2014

Computing Observed Autonomous System Relationships in the Internet

Lu Ruan
Iowa State University, ruanlu@iastate.edu

Jinu Susan Varghese
Iowa State University

Computing Observed Autonomous System Relationships in the Internet

Lu Ruan
Department of Computer Science
Iowa Stat University
Email: ruanlu@iastate.edu

Jinu Susan Varghese
Department of Computer Science
Iowa Stat University
Email: jsusan@iastate.edu

***Abstract*—Autonomous Systems (ASes) in the Internet use BGP to perform interdomain routing. BGP routing policies are mainly determined by the business relationships between neighboring ASes, which can be classified into three types: provider-to-customer, peer-to-peer, and sibling-to-sibling. ASes usually do not export provider routes and peer routes to providers or peers. It has been proved that if all ASes conform to this common export policy then all AS paths are valley-free. Since AS relationships are not publicly available, several studies have proposed heuristic algorithms for inferring AS relationships using publicly available BGP data. Most of these algorithms rely on the valley-free property of AS paths. However, not all AS paths are valley-free because some ASes do not conform to the common export policy. As a result, inferred AS relationship are inaccurate. Instead of inferring AS relationships, we propose an algorithm for computing observed AS relationships based on transit relationships between ASes that are revealed by BGP data. We analyze the types of mismatches between observed AS relationships and actual AS relationships and show that the mismatches can be used to identify ASes that violate the common export policy.**

I. Introduction

The Internet connects tens of thousands of autonomous systems (ASes) operated by different administrative entities. Routing between ASes is determined by the Border Gateway Protocol (BGP). BGP is a policy-driven interdomain routing protocol that allows each AS to choose its own policy in determining which routes to import from and export to its neighbors. ASes engage in business relationships to exchange data traffic and routing policies of ASes are largely determined by the business relationships between neighboring ASes. AS relationships can be broadly classified into three types: provider-to-customer (p2c), peer-to-peer (p2p), and sibling-to-sibling (s2s). In the p2c relationship, the customer pays the provider for transiting traffic from and to the rest of the Internet. In the p2p relationship, two ASes freely exchange traffic between themselves and their customers, but do not exchange traffic from or to their providers or other peers. In the s2s relationship, two ASes freely transit traffic for each other. A natural outcome of this economic model is that an AS usually does not export its provider routes and peer routes to its providers or peers. Gao [1] proved that if all ASes conform to this common export policy, then all AS paths are *valley-free* (i.e., a p2c edge or p2p edge can only be followed by p2c or s2s edges in an AS path). This implies that a non-valley-free path (or valley path) must contain an AS that violates the common export policy.

Knowing the business relationships between ASes is important for understanding the Internet's structure, inter-domain routing dynamics, and evolution. However, AS relationship data are usually not publicly disclosed. Some studies have proposed heuristic algorithms for inferring AS relationships using publicly available BGP data [1], [2], [3], [4], [5], [6], [7]. Most of these algorithms either assume AS paths are valley-free or aim to maximize the number of valley-free AS paths. However, it has been found that a large number of AS paths in BGP updates and BGP routing tables are not valley-free [8], [9]. Thus, inferred AS relationships are inaccurate and biased towards valley-free paths. Some studies have used inferred AS relationships to identify valley paths and policy violating ASes in these paths [10], [8]. The results can contain both false positives and false negatives due to inaccuracy of inferred AS relationships.

In this paper, we propose a novel approach to obtaining AS relationships using BGP data. Instead of inferring AS relationships based on the valley-free property, we develop an algorithm for computing *observed* AS relationships based on transit relationships between ASes revealed by BGP data. By comparing observed AS relationships with actual AS relationships, we can identify policy violating ASes without identifying valley paths in the first place.

The rest of the paper is organized as follows. Section II provides an overview of the related work. In Section III we define types of AS relationships, common export policies, and the valley-free condition. Section IV presents our algorithm for computing observed AS relationships using BGP data. Section V presents how observed AS relationships can be used to identify ASes that violate the common export policies. Section VI presents our results on observed AS relationships and their application on identifying policy violating ASes. Finally, we conclude our paper and outline our future work in Section VII.

II. Related Work

The AS relationship inference problem was first studied in Gao's seminal work [1]. Gao proved that AS paths are valley-free if all ASes obey the common export policies and developed heuristic algorithms for inferring AS relationships based on the patterns of valley-free paths. Xia and Gao [2] show that Gao's algorithm has low accuracy on p2p relationship inference and propose a new algorithm that significantly improves the p2p accuracy by exploiting partial ground truth AS relationships obtained from BGP community attributes and IRR databases.

Subramanian *et al.* [3] define the Type of Relationship (ToR) problem, which is to assign a type (p2c or p2p, s2s not considered) to each edge in the AS graph derived from a set of BGP paths such that the total number of valley-free paths is maximized. They conjecture that the ToR problem is NP-complete and present a heuristic algorithm for the problem that exploits views from multiple vantage points. The algorithm generates a directed AS graph from each view and assigns a rank to each AS based on its position in the graph. The relationship between two ASes are then inferred by comparing their rankings in different views. Di Battista *et al.* [4] prove that the ToR problem is NP-complete and present heuristics for solving the problem in the general case. They also develop an algorithm for determining the AS relationships in the case in which the problem admits a solution without valley-free violations. Dimitropoulos *et al.* [6] identify the limitations of the ToR formulation that lead to incorrect inferences and present heuristics to more accurately infer AS relationships. They add a second objective to the ToR problem formulation that encourages assignment of p2c relationship along the node degree gradient. Similar to Gao's algorithm, they infer p2p relationships by considering edges adjacent to the top provider of each AS path and assigning p2p relationship if the degree difference of the two ASes is smaller than a threshold. s2s relationships are inferred by searching the IRR databases for ASes belonging to the same organization.

Oliveira *et al.* [5] propose an algorithm for inferring AS relationships using BGP data collected from routers in Tier-1 ASes. The algorithm infers p2c and p2p links based on the no-valley premise. Specifically, if an AS path is revealed by a Tier-1 router, then the first link in the path is either a p2p link or a p2c link and the rest of the links are p2c links. The type of the first link can be determined based on whether the link is visible from another Tier-1 AS.

Luckie *et al.* [7] present an algorithm for inferring p2c and p2p relationships that does not rely on the valley-free property. The algorithm ranks ASes by placing a clique of transit-free ASes at the top, and then sort other ASes by transit degree and node degree. p2c relationships are inferred top-down using this ranking and all remaining links are classified as p2p.

Inferred AS relationships have been used to detect and analyze valley paths in BGP updates [10], [8]. Mahajan *et al.* [10] analyzed BGP updates to identify short-lived AS paths that violate the valley-free condition as probable export misconfigurations and polled the ISP operators involved to verify if it was a misconfiguration and to learn the cause of the misconfiguration. Qiu *et al.* [8] quantitatively characterize BGP updates that violate the valley-free condition over a four-month period. It was found that 1.4% of the BGP updates contain valleys and a substantial percentage of prefixes are affected by valleys. The authors further propose a solution to guard against valley routes by adding state to BGP advertisements that reflects the pattern of the advertised AS path. Giotsas and Zhou [9] argue that using inferred AS relationships cannot provide an objective assessment of valley-free violations because AS relationship inference algorithms assume the universality of valley-free paths, which causes the inferred AS relationships to be biased towards valley-free paths. To address the problem, they extract ground truth AS relationships from routing policies encoded in the BGP community attribute and use this bias-free AS relationship dataset to assess valley-free violations in BGP updates and BGP routing tables. It was found that valley paths are more frequent than reported in [8] and a significant fraction of valley paths are the outcome of complex business relationships and deliberate policies as opposed to a result of BGP misconfiguration.

Inferred AS relationships have also been used to study AS path inference [11], Internet evolution [12], impact of prefix hijacking [13], [14], and in simulation-based evaluation of new BGP route selection schemes [15] and new inter-domain routing protocols [16].

III. AS Relationships, Common Export Policies, and the Valley-Free Condition

AS relationships can be broadly classified into three types: provider-to-customer (p2c), peer-to-peer (p2p), and sibling-to-sibling (s2s). A pair of ASes (X, Y) has p2c relationship if X transits traffic for Y and Y does not transit traffic for X. Here, X is the provider and Y is the customer. X and Y have p2p relationship if all traffic carried on the link between X and Y is originated from X (Y) or its downstream customers and destined to Y (X) or its downstream customers. X and Y have s2s relationship if X transits traffic for Y and Y transits traffic for X. That is, siblings provide mutual transit for each other.

AS relationships can be represented by an AS graph where nodes represent ASes and edges represent the business relationships between ASes. The p2c relationship is represented by two directed edges: a p2c edge from the provider to the customer and a c2p edge from the customer to the provider. The p2p and s2s relationships are represented by undirected edges. If AS pair (X, Y) has p2p (s2s) relationship, then there is an undirected p2p (s2s) edge between X and Y.

The set of routes of an AS is classified into own routes, provider routes, customer routes, and peer routes [1]. Let r be a route of AS X. r is X's own route if all edges in r are s2s edges. r is a customer/provider/peer route if the first non-s2s edge is a p2c/c2p/p2p edge.

ASes usually conform to the following common export policies when exporting their routes to neighbors.

- An AS exports its own routes and its customer routes to its providers and peers. It does not export its provider routes and peer routes to its providers or peers.
- An AS exports all of its routes to its customers and siblings.

An AS path is *valley-free* if it satisfies the condition that a p2c edge or p2p edge is only followed by p2c or s2s edges. Gao [1] proved that if all ASes follow the common export policies, then all AS paths are valley-free. It has been found that some AS paths in BGP updates and BGP routing tables are not valley-free [10], [8], [9], implying that some ASes do not conform to the common export policies.

Identifying valley paths requires the knowledge of AS relationships. Since AS relationships are not publicly disclosed, some studies have used inferred AS relationships to identify valley paths [10], [8]. This may result in false negatives and false positives because inferred AS relationships are not accurate. The difficulty of the problem lies in the fact that on the one hand detecting valley paths (which arise because some ASes violate the common export policies) requires knowledge of AS relationships, on the other hand inferring AS relationships assumes that ASes obey the common export policies in the first place [17].

We argue that it is difficult to accurately infer AS relationships for two reasons. First, some ASes do not conform to the common export policies, either due to BGP misconfigurations or due to complex business relationships. As a result, we cannot rely on the patterns of valley-free paths to accurately infer AS relationships. Second, the BGP data made publicly available by projects like Route Views [18] and RIPE RIS [19] provide only a partial view of the Internet's routing system because the data are collected from a limited number of vantage points [20]. While it is difficult to accurately infer AS relationships, it is possible and useful to compute *observed* AS relationships (i.e., AS relationships *revealed* by the BGP data). In the next two sections, we present an algorithm for computing observed AS relationships and describe how observed AS relationships can be used to identify policy violating ASes.

IV. Computing Observed AS Relationships

In this section, we present an algorithm for computing observed AS relationships using BGP data provided by the Route Views Project [18]. The input to our algorithm is a set of BGP routing table dumps collected by Route Views (RV) routers. Each routing table dump contains BGP routing table entries collected from multiple vantage points (i.e., ASes that peer with an RV router). Each vantage point (VP) provides an AS-level view of the Internet from its perspective. The set of AS paths collected from one VP is referred to as the VP's *view*.

Given a set of BGP routing table dumps, our algorithm first preprocesses the data and then computes the observed AS relationships in two phases as detailed below.

A. Preprocessing Data

We first extract IPv4 AS paths from the BGP routing table dumps. We then sanitize the AS paths as follows. First, if an AS path contains an AS set at the end of the path, we remove the AS set if it contains more than one AS numbers (ASNs). Otherwise, we keep the single ASN in the set as the last ASN in the path. Second, we discard AS paths that contain invalid ASNs. An ASN is invalid if it is unallocated or reserved. Allocations of ASNs is published by IANA [21] and we use it to identify invalid ASNs. Third, we discard AS paths that contain loops. An AS path contains a loop if an ASN appears more than once in the path and the appearances are not adjacent to each other. For example, AS path <A B C B D> contains a loop and will be discarded. Last, we remove the duplicate ASNs that arise from AS prepending. For example, AS path <A B B B C> becomes <A B C> after duplicate ASNs are removed.

B. Phase One: Processing AS Paths Containing a Tier-1 AS

Our algorithm computes observed AS relationships in two phases. In phase one, we process the set of AS paths that contain a Tier-1 AS. This set is denoted by S_1.

To compute observed AS relationships, we maintain two counters, $cnt(X,Y)$ and $cnt(Y,X)$, for each pair of ASes X and Y that are adjacent in an AS path. Both counters are initialized to 0. If we observe X provide transit for Y in an AS path, then we increase $cnt(X,Y)$ by 1. If we observe Y provide transit for X in an AS path, then we increase $cnt(Y,X)$ by 1. Basically, $cnt(X,Y)$ counts the number of times we observe X provide transit for Y and $cnt(Y,X)$ counts the number of times we observe Y provide transit for X.

In order to determine the transit relationships between adjacent pairs of ASes in an AS path, we need to know the top provider of the AS path (i.e., the AS that is located at the highest level of the routing hierarchy). If two adjacent ASes X and Y appear before the top provider, then Y provides transit for X. If they appear after the top provider, then X provides transit for Y.

Every AS path in S_1 contains a Tier-1 AS, so the first Tier-1 AS in the path is the top provider. We process every path p in S_1 to update the counters as follows. Let $p =< A_1\ A_2\ \cdots\ A_n >$ and let A_j ($1 \leq j \leq n$) be the first Tier-1 AS in p. We know that A_i provides transit for A_{i-1} for $2 \leq i \leq j$ and A_i provides transit for A_{i+1} for $j \leq i \leq n-1$. Thus, we increase $cnt(A_i, A_{i-1})$ by 1 for $2 \leq i \leq j$ and we increase $cnt(A_i, A_{i+1})$ by 1 for $j \leq i \leq n-1$.

1) Determining Observed AS Relationships: We say an edge between X and Y is visible from a VP A if A's view contains an AS path in which X and Y are adjacent. The common export policies described in Section III imply the following: 1) A p2c edge from X to Y is visible by X's upstream providers; 2) An s2s edge between X and Y is visible by their upstream providers; 3) A p2p edge between X and Y is not visible by their upstream providers. These further imply that if an edge is visible from a Tier-1 AS A, this edge can be a p2c edge, an s2s edge, or a p2p edge adjacent to A, but it cannot be a p2p edge between two non-Tier-1 ASes. This leads to the following rules for determining observed AS relationships based on the values of the two counters.

- If $cnt(X,Y) > 0$ and $cnt(Y,X) = 0$, then X and Y have p2c relationship, i.e., X is a provider of Y.

- If $cnt(X, Y) > 0$ and $cnt(Y, X) > 0$ and X or Y (or both) is a Tier-1 AS, then X and Y have p2p relationship.
- If $cnt(X, Y) > 0$ and $cnt(Y, X) > 0$ and both X and Y are non-Tier-1 ASes, then X and Y have s2s relationship.

The set of observed AS relationships computed in this phase is denoted by R_1. These AS relationships are computed based on transit relationships *observed* from AS paths that contain a Tier-1 AS. Given R_1, we can classify every AS path in our dataset as *determined* or *undetermined*. An AS path is *determined* if the relationship for every adjacent pair of ASes in the path is known (i.e., the relationship exists in R_1). An AS path is *undetermined* if it contains at least one pair of ASes whose relationship is unknown.

C. Phase Two: Processing Undetermined AS Paths

Let S_2 denote the set of undetermined AS paths. In phase two, we process the paths in S_2 to resolve all unknown AS relationships. Note that an undetermined AS path does not contain any Tier-1 AS because all AS paths that contain a Tier-1 AS are determined.

1) Finding the Top Provider of an AS path: The key to resolving unknown AS relationships in undetermined AS paths is to find the top provider in these paths. Once the top provider of an AS path is determined, it is straightforward to determine the transit relationships between adjacent pairs of ASes in the path. Gao's algorithm [1] relies on AS degree to determine the top provider. Specifically, the first largest degree AS in an AS path is considered to be the top provider. We propose a more reliable way of finding the top provider by considering a new AS attribute named *distance*. The *distance* attribute of an AS indicates its minimum distance from a Tier-1 AS and can be calculated as follows. We construct an undirected graph G in which the nodes represent the ASes that appear in the AS paths in our dataset. We add an edge between two nodes in G if the two nodes are adjacent in an AS path. All Tier-1 nodes have distance 0. To find the distance of a non-Tier-1 node v, we compute the shortest path from v to any Tier-1 node, and the length of the path (in hops) is the distance of v.

In a p2c relationship, typically the provider has larger degree and smaller distance than the customer. However, violations of the degree and distance rules exist. To determine the frequency of such violations, we examine the ground truth AS relationships collected in [7]. We find that 1.08% of the p2c relationships violate the degree rule (i.e., the provider has smaller degree than the customer) and 0.82% violate the distance rule (i.e., the distance of the provider is bigger than the distance of the customer). Since distance violation is less frequent than degree violation, we believe using the distance attribute to identify the top provider is more reliable than using the degree attribute.

Given an AS path $p \in S_2$, we find the AS in p that has the smallest distance and let it be the top provider of p. In case multiple ASes have the same smallest distance, we break the tie as follows. We divide the ASes with the smallest distance into groups of consecutive ASes and let the largest degree AS in the last group be the top provider of p. For example, consider an AS path <A, B, C, D, E, F, G> in which C, E, and F have the smallest distance. These 3 ASes fall into two groups, the first group contains C and the second group contains E and F. E and F belong to the last group, so the one with the larger degree is the top provider. In case of a tie in degree, the first largest degree AS in the last group is chosen as the top provider. That is, E is the top provider if E and F have the same degree.

The reason for choosing the top provider from the last group of smallest distance ASes is the following. Consider an AS path for a specific prefix. The BGP updates for the prefix are originated by the last AS in the AS path and propagated from right to left among the ASes in the path. In the above example, BGP updates for a prefix in G are originated by G and then propagated to F, E, D, and so on. Thus, the largest degree node in the last group of smallest distance nodes should be the top provider for traffic destined for the prefix in G. Suppose E is the top provider. The path segment <C, D, E> shows that the top provider E propagates an update down the routing hierarchy to D, which then propagates the update up the routing hierarchy to C (note that D has larger distance than C and E). This indicates that a valley exists in the path. While both our algorithm and Gao's algorithm [1] rely on the top provider to determine transit relationships between adjacent pairs of ASes in an AS path, our algorithm is different from Gao's algorithm in two ways. First, we choose the top provider based on the distance attribute (with tie breaking using the degree attribute) while Gao relies on the degree attribute to pick the top provider. Second, Gao assumes AS paths are valley-free. As a result, at most one of the edges adjacent to the top provider in an AS path can be inferred as p2p. In contrast, we do not assume valley-free paths. As a result, both edges adjacent to the top provider in an AS path may be observed as p2p.

2) Resolving Unknown AS Relationships: To resolve all the unknown relationships, we sequentially process all AS paths in S_2 as follows. Let $p =< A_1\ A_2\ \cdots\ A_n >$ be an AS path in S_2 and let A_j $(1 \le j \le n)$ be the top provider of p determined using the above method. We increase $cnt(A_i, A_{i-1})$ by 1 for $2 \le i \le j$ if the relationship of A_i and A_{i-1} is unknown. That is, A_i provides transit for A_{i-1} if they appear before the top provider. We increase $cnt(A_i, A_{i+1})$ by 1 for $j \le i \le n-1$ if the relationship of A_i and A_{i+1} is unknown. That is, A_i provides transit for A_{i+1} if they appear after the top provider.

After we process all paths in S_2, we go through two steps to resolve unknown AS relationships. In the first step, we use the following rules to assign AS relationships to AS pairs based on the values of the two counters.

- If $cnt(X, Y) > 0$ and $cnt(Y, X) = 0$, then X and Y have p2c relationship, i.e., X is a provider of Y.
- If $cnt(X, Y) > 0$ and $cnt(Y, X) > 0$, then X and Y have p2p relationship.

If $cnt(X, Y) > 0$ and $cnt(Y, X) > 0$, then X and Y can have either p2p or s2s relationship. If the link between X and Y is not visible by any upstream provider, then X and Y have p2p relationship, otherwise they have s2s relationship. In the first step, we temporally assign p2p relationship if $cnt(X, Y) > 0$ and $cnt(Y, X) > 0$. In the second step, we change some p2p relationships to s2s as follows. We examine each path in S_2. If the path contains a p2p link and the p2p link is not adjacent to the top provider of the path, then we change the link type to s2s because the link is visible by an upstream provider (i.e., the top provider). For example, consider the AS path <A B C D E F> where C is the top provider. If D and E is assigned p2p relationship in the first step, then we change their relationship to s2s in the second step. At the end of the second step, we have resolved all unknown AS relationships, and the set of resolved AS relationships is denoted by R_2.

In summary, our algorithm computes a set of observed AS relationships $R = R_1 \cup R_2$ in two phases. In phase one, we compute the set R_1 by processing AS paths that contain a Tier-1 AS. The top provider of each of these paths is a Tier-1 AS. In the second phase, we compute the set R_2 by processing undetermined AS paths according to R_1. For these AS paths, we pick the largest degree node in the last group of consecutive smallest distance nodes as the top provider. The set $R = R_1 \cup R_2$ contains the observed AS relationships for all AS pairs that appear in the AS paths in our dataset.

V. Identifying Policy Violating Nodes Using Observed AS Relationships

An AS that does not conform to the common export policies is referred to as a policy violating node (PVN). In this section we present two ways of identifying PVNs using observed AS relationships.

A. Comparing Observed AS Relationships with Actual AS Relationships

We can identify PVNs by comparing observed AS relationships with the ground truth AS relationships. The largest source of ground truth AS relationships to date was assembled by Luckie *et al.* [7]. Their dataset contains around 47,000 p2p and p2c relationships. For a pair of ASes (X, Y), the observed AS relationship could be different from the ground truth AS relationship and the mismatch can be classified into four types.

- **Type 1**: **The observed relationship is p2c but the actual relationship is p2p**. The observed relationship of (X, Y) is p2c means that we observe X provide transit for Y but we do not observe Y provide transit for X. The mismatch is due to *incomplete data*. That is, Route Views routers peer with a limited number of VPs and the views from these VPs do not allow us to observe p2p relationship for (X, Y). This type of mismatch can be reduced if the number of VPs increase. In particular, if a new VP reveals an AS path in which Y provides transit for X, then we would be able to observe the p2p relationship between X and Y.
- **Type 2**: **The observed relationship is s2s but the actual relationship is p2p**. The fact that the observed relationship is s2s indicates that the p2p link between X and Y is visible from an upstream provider. This means that X or Y (or both) has violated the common export policies by exporting its peer route to a provider.
- **Type 3**: **The observed relationship is p2p or s2s but the actual relationship is p2c (i.e., X is provider of Y)**. This means that the actual customer Y has violated the common export policies by exporting a route learned from provider X to a peer or provider, causing us to observe Y provide transit for X in addition to observe X provide transit for Y. As a result, the observed relationship for (X, Y) is p2p or s2s.
- **Type 4**: **The observed relationship is c2p (i.e., X is customer of Y) but the actual relationships is p2c (i.e., X is provider of Y)**. This means that the actual customer Y has has violated the common export policies by exporting a route learned from provider X to a peer or provider, causing us to observe Y provide transit for X. This is similar to the Type 3 mismatch; the only difference is that here we do not observe X provide transit for Y, so the observed relationship of (X, Y) is c2p.

The above classification of mismatches shows that there are two causes for mismatches between observed AS relationships and actual AS relationships: 1) There is insufficient data for us to observe the actual AS relationship; 2) Some AS has violated the common export policies. Type 1 mismatch is due to incomplete data and the other three types of mismatch are due to policy violation by some AS. In a type 2 mismatch, one or both peers are PVNs and in a type 3 or type 4 mismatch, the actual customer is the PVN.

1) Discussion: A common approach to identifying PVNs is to find valleys in AS paths [10], [8], [9]. An AS path that violates the valley-free condition contains one or more of the following four types of valleys.

1) **p2c-c2p**: a p2c edge is followed by a c2p edge.
2) **p2p-c2p**: a p2p edge is followed by a c2p edge.
3) **p2c-p2p**: a p2c edge is followed by a p2p edge.
4) **p2p-p2p**: a p2p edge is followed by a p2p edge.

We use X>Y to denote X is a provider of Y, X<Y to denote X is a customer of Y, and X−Y to denote X and Y are peers. Then the above four types of valleys can be written as X>V<Y, X−V<Y, X>V−Y, and X−V−Y, where X, V, and Y are the ASes involved in the valley. In all four cases, the middle AS V is a PVN. Specifically, in X>V<Y, V exports a route learned from provider Y to provider X. In X−V<Y, V exports a route learned from provider Y to peer X. In X>V−Y, V exports a route learned from peer Y to provider X. In X−V−Y, V exports a route learned from peer Y to peer X.

In order to identify PVNs by finding valleys in AS paths, we need knowledge of AS relationships. Since AS relationships are not publicly disclosed, prior studies use either inferred AS relationships [10], [8] or partial ground truth AS relationships [9]. Both approaches have their limitations. Identifying valleys using inferred relationships can produce false positives and false negatives due to inaccuracy of inferred AS relationships. For example, if an AS path contains the segment A>O−B according to inferred AS relationships, then O is identified as a PVN. If the actual AS relationship between O and B is O>B, then O is not a PVN. Here, a node identified as a PVN is actually a policy conforming AS (i.e., false positive). Consider another example, suppose an AS path contains the segment A>O>B according to inferred AS relationships, so O is not identified as a PVN. If the actual AS relationship between O and B is O−B, then O is actually a PVN. Here, a node not identified as a PVN is actually a PVN (i.e., false negative). These examples show that using inferred AS relationships cannot reliably identify PVNs.

To address the inaccuracy of inferred AS relationships, Giotsas and Zhou [9] extract ground truth AS relationships from routing policies encoded in BGP community attribute. They are able to extract AS relationships of more than 30% of the AS links and analyze the valley-free violations using this partial truth data. This approach has two limitations. First, it cannot identify all valleys due to the the lack of complete truth data. Second, sometimes the PVN cannnot be identified even though a valley is found in an AS path. For example, suppose an AS path contains the segment A>B?C<D. Here B?C means the actual AS relationship between B and C is unknown. This segment contains a valley, however, we cannot identify the PVN without knowing the actual AS relationship between B and C. Specifically, C is the PVN if B>C and B is the PVN if B<C.

Using observed AS relationships, we can identify PVNs without identifying valleys in AS paths. By comparing observed AS relationships with the actual AS relationships, we can identify the PVNs from type 2, type 3, and type 4 mismatches. Since we do not have complete ground truth AS relationship data, we are not able to identify all PVNs. However, the observed AS relationships can be made available to the network operators to enable them to identify policy violations. Suppose an network operator is provided with the observed AS relationships between its AS and its neighboring ASes, the operator can compare the observed AS relationship with the actual AS relationship for each neighboring AS. If the operator's AS is found to be a PVN in a type 2, type 3, or type 4 mismatch, then the operator should determine if the violation of the common export policies is intentional (i.e., due to special economic models) or unintentional (i.e., due to BGP misconfigurations). If the policy violation is intentional, then no corrective action is needed. Otherwise, the operator should identify BGP misconfigurations and correct them.

B. Finding P2P-P2P Valleys

While comparing observed AS relationships with actual AS relationships can identify PVNs that export a peer route to a provider (type 2 mismatch) and PVNs that export a provider route to a provider or peer (type 3 and type 4 mismatches), it cannot identify PVNs that export a peer route to a peer. To identify such PVNs, we need to find X−V−Y valleys in AS paths using the observed AS relationships and identify V as a PVN. Note that V may not be an actual PVN because the actual AS relationship between X and V and between V and Y may not be p2p. If the operator of V is given a set of X−V−Y valleys that it is involved in, it can then check if it has p2p relationship with both X and Y. If yes, V is indeed a PVN and the operator should identify and correct BGP misconfiguration if the X−V−Y valley is not intentional. For example, an AS V can play the role of a mediate AS that allows two ASes X and Y to peer with each other through it. In this case the X−V−Y valley is intentional and no corrective action is needed.

In summary, PVNs that export a peer route to a provider and PVNs that export a provider route to a provider or peer can be identified by comparing observed AS relationships with the actual AS relationships. PVNs that export a peer route to another peer can be identified by finding p2p-p2p valleys using observed AS relationships. Thus, observed AS relationships provide useful information for network operators to identify policy violations and subsequently identify and correct BGP misconfigurations if the policy violation is unintentional. We plan to create a website that allows a user to enter an ASN and obtain the observed AS relationships between the AS and its neighbors as well as a list of p2p-p2p valleys in which the AS is a PVN. This information allows network operators to identify policy violations by their ASes. The website will also allow network operators to report identified policy violations and their causes (i.e., deliberate policy for complex AS relationships or BGP misconfiguration). The collected data on policy violations and their causes will allow us to gain a better understanding of complex AS relationships and routing policies associated with them.

Observed AS relationships can also help Internet researchers to understand the extent to which the publicly available BGP data can reveal the actual relationships between ASes.

VI. RESULTS

We use BGP routing table dumps collected by all Route Views routers on May 1, 2014 at 02:00 to compute observed AS relationships. Our dataset contains around 7.58 million unique AS paths collected from 632 VPs. We manually identify 16 Tier-1 ASes according to Wikipedia [22]; 12 of these Tier-1 ASes are VPs. A total of 136,559 observed AS relationships are computed from our dataset.

A. Observed AS Relationships in Phase One

The set R_1 computed in phase one contains 91,210 observed AS relationships. These include 90,417 (99.13%) p2c relationships, 197 (0.22%) p2p relationships, and 596 (0.65%) s2s relationships. All p2p links observed in phase one are adjacent to Tier-1 ASes.

We compare R_1 with the ground truth AS relationships collected by Luckie *et al.* [7]. Their dataset consists of two sets of ground truth AS relationships. The first set, denoted by T_{comm}, contains over 41,000 p2c and p2p relationships derived from BGP community attributes in BGP route announcements. The second set, denoted by T_{RPSL}, contains over 6,500 p2c relationships derived from routing policies stored in public databases using the Routing Policy Specification Language

(RPSL). T_{comm} and T_{RPSL} have 1,430 overlapping relationships, out of which 6 do not agree. We remove these 6 relationships from T_{comm} and T_{RPSL} and compute the union of the two sets. The union, denoted by T_{all}, contains 46,698 AS relationships, including 16,243 p2p relationships and 30,455 p2c relationships.

R_1 and T_{all} have 20,873 overlapping relationships, out of which 19,960 (95.63%) relationships match. The 913 mismatches can be classified into four types, as shown in Table I. We see in the table that the most frequent type of mismatch is type 1, which occurs 774 (84.78%) times. Thus, most of the mismathes are due to incomplete data, which causes p2p relationships to be observed as p2c relationships. Such mismatches can be recuced by increasing the number of VPs. There are 16 type 2 mismatches, which are caused by one or both peers violating the common export policies. There are 101 type 3 mismatches and 22 type 4 mismatches, both of which are caused by the actual customer violating the common export policies.

TABLE I
MISMATCHES BETWEEN OBSERVED AS RELATIONSHIPS IN PHASE ONE AND GROUND TRUTH

Type of Mismatch	obs=p2c truth=p2p	obs=s2s truth=p2p	obs=s2s/p2p truth=p2c	obs=c2p truth=p2c
#Occurrences	774	16	101	22
Percentage	84.78%	1.75%	11.06%	2.41%

To understand how actual p2p relationships are observed from the BGP data, we examine the overlapping p2p relationships between T_{all} and R_1. We find 903 overlapping p2p relationships, which can be classified into three types: match (i.e., observed relationship is p2p), observed as p2c, and observed as s2s. The number of occurrences and the percentage of each type are shown in Table II. We see that 85.71% of the overlapping p2p relationships are observed as p2c. This means that a large fraction of the p2p links adjacent to Tier-1 ASes are observed as p2c links due to the limited number of VPs.

TABLE II
OVERLAPPING P2P RELATIONSHIPS BETWEEN T_{all} AND R_1

Match	Observed as p2c	Observed as s2s
113 (12.51%)	774 (85.71%)	16 (1.77%)

To understand how actual p2c relationships are observed from the BGP data, we examine the overlapping p2c relationships between T_{all} and R_1. We find 19,970 overlapping p2c relationships, which can be classified into three types: match, observed as p2p or s2s, and observed as c2p. The number of occurrences and the percentage of each type are shown in Table III. We see that 99.38% of the overlapping p2c relationships match and the rest are mismatches caused by policy violations of the customers in the p2c relationships. The results show that policy violations cause a very small percentage of the p2c links to be observed as other types.

TABLE III
OVERLAPPING P2C RELATIONSHIPS BETWEEN T_{all} AND R_1

Match	Observed as p2p/s2s	Observed as c2p
19,847 (99.38%)	101 (0.51%)	22 (0.11%)

B. Observed AS Relationships in Phase Two

The set R_2 computed in phase two contains 45,349 observed AS relationships, out of which there are 41,462 (91.43%) p2c relationships, 2,264 (4.99%) p2p relationships, and 1,623 (3.58%) s2s relationships. The p2p links observed in this phase are between non-Tier-1 ASes. We observe many more p2p links in phase two than in phase one (2,264 vs. 197), indicating that more peering occurs at lower level of the routing hierarchy.

R_2 and T_{all} have 6,521 overlapping relationships, out of which 5,516 relationships do not match. The number of occurrences and the percentage of the four types of mismatches are shown in Table IV. We see that the most frequent type of mismatch is type 1, which occurs 5,123 (92.88%) times. Thus, most of the mismatches are due to incomplete data, which causes p2p relationships to be observed as p2c. This is consistent with the findings in [23], [24], [5] that the public view misses a large number of p2p links, especially p2p links between lower tier ASes. In order to observe a p2p relationship between X and Y, we need a VP in X or X's downstream customer and a VP in Y or Y's downstream customer. Since there are only a limited number of VPs and most of the VPs are located at higher levels of the routing hierarchy [20], we are only able to observe a small fraction of the p2p links between non-Tier-1 ASes. The second most frequent type of mismatch is type 2, which occurs 364 times. Here, a p2p relationship is observed as an s2s relationship because one or both peers export the peer route to a provider, causing the p2p link to be visible by an upstream provider. There are 13 type 3 mismatches and 16 type 4 mismatches, both of which are caused by the customer violating the common export policies.

TABLE IV
MISMATCHES BETWEEN OBSERVED AS RELATIONSHIPS IN PHASE TWO AND GROUND TRUTH

Type of Mismatch	obs=p2c truth=p2p	obs=s2s truth=p2p	obs=s2s/p2p truth=p2c	obs=c2p truth=p2c
#Occurrences	5,123	364	13	16
Percentage	92.88%	6.60%	0.24%	0.29%

T_{all} and R_2 have 6,050 overlapping p2p relationships. The three types of these p2p relationships are shown in Table V. We see that 84.68% of the overlapping p2p relationships are observed as p2c, indicating that a large fraction of the p2p links between non-Tier-1 ASes are observed as p2c links due to the limited number of VPs.

T_{all} and R_2 have 471 overlapping p2c relationships. The three types of these p2c relationships are shown in Table VI. We see that 93.84% of the overlapping p2c relationships match and the rest are mismatches caused by policy violations of the customers in the p2c relationships. The results show that

TABLE V
OVERLAPPING P2P RELATIONSHIPS BETWEEN T_{all} AND R_2

Match	Observed as p2c	Observed as s2s
563 (9.31%)	5123 (84.68%)	364 (6.02%)

policy violations cause a small percentage of the p2c links to be observed as other types.

TABLE VI
OVERLAPPING P2C RELATIONSHIPS BETWEEN T_{all} AND R_2

Match	Observed as p2p/s2s	Observed as c2p
442 (93.84%)	13 (2.76%)	16 (3.40%)

In phase two, we use distance-based approach to find the top provider in an AS path. That is, we choose the largest degree node in the last group of consecutive smallest distance nodes to be the top provider. An alternative approach is degree-based [1], which selects the first largest degree node in an AS path as the top provider. We compared the two approaches and found that the distance-based approach is more reliable. Specifically, with degree-based approach, the resulting R_2 has 474 matches with T_{all}. On the other hand, the distance-based approach results in 1005 matches, which is over two times of the matches produced by the degree-based approach.

C. Policy Violating Nodes

1) Identifying PVNs by Comparing Observed AS Relationships with the Truth: Given the set $R = R_1 \cup R_2$ of observed AS relationships, we can identify PVNs by comparing R with T_{all} to find type 2, type 3, and type 4 mismatches. For a type 2 mismatch, one or both ASes are PVNs. For a type 3 or type 4 mismatch, the actual customer is the PVN. We obtain 380 type 2 mismatches by comparing R and T_{all}, from which we identify 154 candidate PVNs. In a type 2 mismatch, one or both peers violate the common export policies by exporting a peer route to a provider, so we identify both peers as candidate PVNs. By comparing R and T_{all}, we find 114 type 3 mismatches and 38 type 4 mismatches, from which we identify 142 PVNs. These PVNs violate the common export policies by exporting a provider route to a provider or peer.

Note that our set of identified PVNs is a proper subset of all the PVNs because we do not have complete ground truth AS relationship data. Specifically, only 20% of our observed AS relationships can be found in T_{all}.

2) Identifying PVNs by Finding P2P-P2P Valleys: As noted in Section V-B, PVNs that export a peer route to a peer cannot be identified by comparing observed AS relationships with the truth because the policy violation does not result in a mismatch. In order to find such PVNs, we need to find p2p-p2p valleys in AS paths and identify the node adjacent to both p2p links as the PVN.

Using R_1, we find 69,978 p2p-p2p valleys and 18 PVNs in the set S_1 of AS paths we process in phase one. To understand how mismatches between observed AS relationships and actual AS relationships affect the results, we replace the observed AS relationships with the actual AS relationships when there is a mismatch and the resulting AS relationships are referred to as the *corrected* AS relationships. Using corrected AS relationships, we find 107,587 p2p-p2p valleys and 49 PVNs in S_1. The results are shown in Table VII. We see that using observed AS relationships fails to identify a large number of p2p-p2p valleys and PVNs as compared to using corrected AS relationships. This is because 790 p2p links are observed as p2c or s2s links so that p2p-p2p valleys involving these p2p links cannot be identified. We find that all 18 PVNs identified using observed AS relationships belong to the set of 49 PVNs identified using corrected AS relationships. That is, using observed AS relationships produces zero false positives and 31 false negatives. Another finding is that all 16 Tier-1 ASes are PVNs. This means that it is common for Tier-1 ASes to export a peer route to another peer.

Using the set of observed AS relationships $R_1 \cup R_2$, we find no p2p-p2p valleys in the set S_2 of AS paths we process in phase two. This is because 5,487 p2p links are observed as p2c or s2s in phase two. To identify p2p valleys, we modify our algorithm in phase two so that we do not assign s2s relationships. That is, if $cnt(X, Y) > 0$ and $cnt(Y, X) > 0$, then we assign p2p relationship to (X, Y). With this modification, we identify 10,116 p2p-p2p valleys and 103 PVNs. On the other hand, using corrected AS relationships results in 42,748 identified valleys and 134 identified PVNs, as shown in Table VII. The set of observed PVNs and the set of corrected PVNs have 100 common PVNs. This means that using observed AS relationships to identify PVNs results in 3 false positives and 34 false negatives.

TABLE VII
P2P-P2P VALLEYS AND PVNS IDENTIFIED BY OBSERVED AS RELATIONSHIPS AND CORRECTED AS RELATIONSHIPS

	#p2p-p2p valleys	#PVNs
S_1 observed	69,978	18
S_1 corrected	107,587	49
S_2 observed	10,116	103
S_2 corrected	42,748	134

D. Summary

We make the following conclusions from our results. First, a large fraction of p2p links are observed as p2c links due to incomplete data from a limited number of VPs. As a result, inferring AS relationships from BGP data can overestimate the number of p2c links and underestimate the number of p2p links. Second, most of the mismatches between observed AS relationships and actual AS relationships are caused by incomplete data. The other mismatches are caused by policy violations and can be used to identify ASes that export a peer route to a provider and ASes that export a provider route to a provider or peer. Third, observed AS relationships can be used to identify ASes that export a peer route to a peer by finding p2p-p2p valleys in AS paths. The identified PVNs are highly likely to be correct (i.e., false positive rate is low). However, it can fail to identify some PVNs because many p2p links are observed as p2c links.

VII. Conclusion and Future Work

In this paper, we present an algorithm for computing observed AS relationships using publicly available BGP data. We argue that it is difficult to accurately infer AS relationships using BGP data due to two reasons. First, some ASes do not conform to the common export policies. As a result, not all AS paths are valley-free. Second, publicly available BGP data provide only a partial view of the Internet's routing system because the data are collected from a limited number of VPs. Instead of inferring AS relationships, we compute observed AS relationships based on transit relationships between ASes that are revealed by BGP data. We show that the observed AS relationships can be used to identify policy violating nodes by finding mismatches with the actual AS relationships and by identifying p2p-p2p valleys in AS paths.

In our future work, we plan to create a website that allows users to obtain the observed AS relationships and the p2p-p2p valleys for a given ASN. This information allows network operators to identify policy violations of their ASes. The website will also allow network operators to report identified policy violations and their causes, based on which we will study complex AS relationships and their associated export policies.

References

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

[2] Jianhong Xia and Lixin Gao, "On the evaluation of AS relationship inferences," in *IEEE Global Telecommunications Conference*, 2004, vol. 3, pp. 1373–1377.

[3] Lakshminarayanan Subramanian, Sharad Agarwal, Jennifer Rexford, and Randy H Katz, "Characterizing the Internet hierarchy from multiple vantage points," in *IEEE INFOCOM*, 2002, vol. 2, pp. 618–627.

[4] Giuseppe Di Battista, Maurizio Patrignani, and Maurizio Pizzonia, "Computing the types of the relationships between autonomous systems," in *IEEE INFOCOM*, 2003, vol. 1, pp. 156–165.

[5] R. Oliveira, Dan Pei, W. Willinger, Beichuan Zhang, and Lixia Zhang, "The (in)completeness of the observed Internet AS-level structure," *IEEE/ACM Transactions on Networking*, vol. 18, no. 1, pp. 109–122, 2010.

[6] Xenofontas Dimitropoulos, Dmitri Krioukov, Marina Fomenkov, Bradley Huffaker, Young Hyun, kc. claffy, and George Riley, "AS relationships: inference and validation," *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 1, pp. 29–40, 2007.

[7] Matthew Luckie, Bradley Huffaker, Amogh Dhamdhere, Vasileios Giotsas, and kc claffy, "AS relationships, customer cones, and validation," in *The Internet Measurement Conference*, 2013, pp. 243–256.

[8] Sophie Y Qiu, Patrick Drew McDaniel, and Fabian Monrose, "Toward valley-free inter-domain routing," in *IEEE ICC*, 2007, pp. 2009–2016.

[9] Vasileios Giotsas and Shi Zhou, "Valley-free violation in Internet routing - analysis based on BGP community data," in *IEEE ICC*, 2012, pp. 1193–1197.

[10] Ratul Mahajan, David Wetherall, and Tom Anderson, "Understanding BGP misconfiguration," in *ACM SIGCOMM Computer Communication Review*, 2002, vol. 32, pp. 3–16.

[11] Z Morley Mao, Lili Qiu, Jia Wang, and Yin Zhang, "On AS-level path inference," in *ACM SIGMETRICS Performance Evaluation Review*. ACM, 2005, vol. 33, pp. 339–349.

[12] Amogh Dhamdhere and Constantine Dovrolis, "Twelve years in the evolution of the Internet ecosystem," *IEEE/ACM Transactions on Networking (ToN)*, vol. 19, no. 5, pp. 1420–1433, 2011.

[13] Mohit Lad, Ricardo Oliveira, Beichuan Zhang, and Lixia Zhang, "Understanding resiliency of Internet topology against prefix hijack attacks," in *37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, 2007, pp. 368–377.

[14] Hitesh Ballani, Paul Francis, and Xinyang Zhang, "A study of prefix hijacking and interception in the Internet," *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 4, pp. 265–276, 2007.

[15] Philip Brighten Godfrey, Matthew Caesar, Ian Haken, Yaron Singer, Scott Shenker, and Ion Stoica, "Stabilizing route selection in BGP," *IEEE/ACM Transactions on Networking*, 2014, to appear.

[16] Lakshminarayanan Subramanian, Matthew Caesar, Cheng Tien Ee, Mark Handley, Morley Mao, Scott Shenker, and Ion Stoica, "HLP: a next generation inter-domain routing protocol," in *ACM SIGCOMM*, 2005, pp. 13–24.

[17] Nick Feamster, Hari Balakrishnan, and Jennifer Rexford, "Some foundational problems in interdomain routing," in *Proceedings of Third Workshop on Hot Topics in Networks (HotNets-III)*, 2004, pp. 41–46.

[18] *University of Oregon Route Views Project*, http://www.routeviews.org.

[19] *RIPE's Routing Information Service*, http://www.ripe.net/data-tools/stats/ris/.

[20] Matthew Roughan, Walter Willinger, Olaf Maennel, Debbie Perouli, and Randy Bush, "10 lessons from 10 years of measuring and modeling the Internet's autonomous systems," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 9, pp. 1810–1821, 2011.

[21] *Autonomous System (AS) Numbers*, http://www.iana.org/assignments/as-numbers/as-numbers.xhtml.

[22] *Tier-1 network*, http://en.wikipedia.org/wiki/Tier_1_network.

[23] Hyunseok Chang, Ramesh Govindan, Sugih Jamin, Scott J Shenker, and Walter Willinger, "Towards capturing representative AS-level internet topologies," *Computer Networks*, vol. 44, no. 6, pp. 737–755, 2004.

[24] Rami Cohen and Danny Raz, "The Internet dark matter-on the missing links in the AS connectivity map.," in *IEEE INFOCOM*, 2006.