Open access • Proceedings Article • DOI:10.1145/2504730.2504735

AS relationships, customer cones, and validation — Source link

Matthew Luckie, Bradley Huffaker, Amogh Dhamdhere, Vasileios Giotsas ...+1 more authors

Institutions: University of California, San Diego, University College London

Published on: 23 Oct 2013 - Internet Measurement Conference

Topics: The Internet and Peering

AS Relationships, Customer Cones, and Validation

Matthew Luckie
CAIDA / UC San Diego
mjl@caida.org

Bradley Huffaker
CAIDA / UC San Diego
bradley@caida.org

Amogh Dhamdhere
CAIDA / UC San Diego
amogh@caida.org

Vasileios Giotsas
University College London
V.Giotsas@cs.ucl.ac.uk

kc claffy
CAIDA / UC San Diego
kc@caida.org

ABSTRACT

Business relationships between ASes in the Internet are typically confidential, yet knowledge of them is essential to understand many aspects of Internet structure, performance, dynamics, and evolution. We present a new algorithm to infer these relationships using BGP paths. Unlike previous approaches, our algorithm does not assume the presence (or seek to maximize the number) of valley-free paths, instead relying on three assumptions about the Internet's inter-domain structure: (1) an AS enters into a provider relationship to become globally reachable; and (2) there exists a peering *clique* of ASes at the top of the hierarchy, and (3) there is no cycle of p2c links. We assemble the largest source of validation data for AS-relationship inferences to date, validating 34.6% of our 126,082 c2p and p2p inferences to be 99.6% and 98.7% accurate, respectively. Using these inferred relationships, we evaluate three algorithms for inferring each AS's *customer cone*, defined as the set of ASes an AS can reach using customer links. We demonstrate the utility of our algorithms for studying the rise and fall of large transit providers over the last fifteen years, including recent claims about the flattening of the AS-level topology and the decreasing influence of "tier-1" ASes on the global Internet.

Categories and Subject Descriptors

C.2.5 [**Local and Wide-Area Networks**]: Internet; C.2.1 [**Network Architecture and Design**]: Network topology

Keywords

AS relationships; routing policies; customer cones

1. INTRODUCTION

The Internet consists of thousands of independent interconnected organizations, each driven by their own business model and needs. The interplay of these needs influences, and sometimes determines, topology and traffic patterns, i.e., connectivity between networked organizations and routing across the resulting mesh. Understanding the underlying business relationships between networked organizations provides the strongest foundation for understanding many other aspects of Internet structure, dynamics, and evolution.

Business relationships between ASes, which are typically congruent with their routing relationships, can be broadly classified into two types: customer-to-provider (c2p) and peer-to-peer (p2p). In a c2p relationship, the customer pays the provider for traffic sent between the two ASes. In return, the customer gains access to the ASes the provider can reach, including those which the provider reaches through its own providers. In a p2p relationship, the peering ASes gain access to each others' customers, typically without either AS paying the other. Peering ASes have a financial incentive to engage in a *settlement-free* peering relationship if they would otherwise pay a provider to carry their traffic, and neither AS could convince the other to become a customer. Relationships are typically confidential so must be *inferred* from data that is available publicly. This paper presents a new approach to inferring relationships between ASes using publicly available BGP data.

Measurement and analysis of Internet AS topologies has been an active area of research for over a decade. While yielding insights into the structure and evolution of the topology, this line of research is constrained by systematic measurement and inference challenges [32], many of which our approach proactively addresses. First, the BGP-based collection infrastructure used to obtain AS-level topology data suffers from artifacts induced by misconfigurations, poisoned paths, and route leaks, all of which impede AS-relationship inference. Our algorithm incorporates steps to remove such artifacts. Second, AS topologies constructed from BGP data miss many peering links [6]. We show this lack of visibility does not hinder the accuracy of inferences on the links we do observe. Third, most AS-relationship algorithms rely on "valley-free" AS paths, an embedded assumption about the rationality of routing decisions that is not always valid [32], and which our algorithm does not make. Fourth, import and export filters can be complicated; some operators caveat their c2p links as being region or prefix specific. However, they still describe themselves as customers even if they do not receive full transit. Therefore, we make a c2p inference when any transit is observed between two ASes. We argue that relationship inferences can still be c2p or p2p with the caveat that the c2p relationship may be partial. We develop techniques to mitigate the effects of such *hybrid relationships* when computing an AS's customer cone, described later in

IMC'13, October 23–25, 2013, Barcelona, Spain.
Copyright 2013 ACM 978-1-4503-1953-9/13/10 ...$15.00.
http://dx.doi.org/10.1145/2504730.2504735.

this section. Finally, a single organization may own and operate multiple ASes; we leave the inference of *sibling relationships* as future work because it is difficult to distinguish them from route leaks. We make the following contributions:

We introduce a new algorithm for inferring c2p and p2p links using BGP data. Our algorithm builds on three generally accepted assumptions about industry structure: (1) there is a clique of large transit providers at the top of the hierarchy; (2) most customers enter into a transit agreement to be globally reachable; and (3) cycles of c2p links (e.g., where ASes A, B, and C are inferred to be customers of B, C, and A respectively) should not exist for routing to converge [19]. Our algorithm achieves *near-perfect accuracy* for both p2p and p2c links for the subset of relationships we were able to validate, using the largest set of externally gathered AS relationship validation data collected to date. Since even our three different sources of validation data disagree by 1%, our algorithm reaches the limit of accuracy achievable with available data.

We evaluate our algorithm's accuracy by validating 43,613 (34.6%) of our inferences – the largest validation of AS-relationships performed to date. First, we received nearly 2400 directly reported relationship assertions from operators through the interactive feedback functionality of our public repository of AS relationships [1]. Second, we use policy information encoded in the RIPE WHOIS database to extract more than 6,500 unique relationships; we extract only relationships between ASes where both ASes have policies recorded in the database and both policies are expressed consistently. Third, we extract more than 41,000 relationships using community strings encoded in BGP messages by ASes who publish how these strings map to AS relationships. Our validation data, 97% of which we can share (we cannot share directly reported relationships) will itself contribute to a research area that has long suffered from lack of validation.

We introduce a new algorithm for inferring the *customer cone* of an AS. The customer cone of AS X is defined as the set of ASes that X can reach using p2c links; for AS X the customer cone includes X's customers, as well as X's customers' customers, and so on. The customer cone is not a methodologically clean construct, since some ASes have hybrid relationships; for example, they peer in some regions (or some prefixes), but one AS will purchase from the other for other regions (or other prefixes). We compare the strengths and weaknesses of three distinct approaches for computing customer cones, and explain why we believe one approach is more accurate. We also show how this construct informs our understanding of evolutionary trends such as the flattening of the Internet topology and the financial consolidation of the Internet transit industry.

2. RELATED WORK

Gao [18] was the first to study the inference of AS relationships. Her solution relies on the assumption that BGP paths are hierarchical, or *valley-free*, i.e., each path consists of an uphill segment of zero or more c2p or sibling links, zero or one p2p links at the top of the path, followed by a downhill segment of zero or more p2c or sibling links. The valley-free assumption reflects the typical reality of commercial relationships in the Internet: if an AS were to announce routes learned from a peer or provider to a peer or provider (creating a valley in the path), it would then be offering transit for free. Gao's algorithm thus tries to derive the maximum number of valley-free paths, by selecting the largest-degree AS in a path as the top, and assuming that ASes with similar degrees are likely to be peers (p2p). Gao validated her results using information obtained from a single Tier-1 AS (AT&T). Xia and Gao [36] proposed an improvement to Gao's algorithm that uses a set of ground truth relationships to seed the inference process. Transit relationships are then inferred using the valley-free assumption. Gao's algorithm [18] is used for remaining unresolved links. They validated 2,254 (6.3%) of their inferences using 80% of their validation data and found their algorithm was accurate for 96.1% of p2c links and 89.33% of p2p links.

Subramanian *et al.* [34] formalized Gao's heuristic into the Type of Relationship (ToR) combinatorial optimization problem: given a graph derived from a set of BGP paths, assign the edge type (c2p or p2p, ignoring sibling relationships) to every edge such that the total number of valley-free paths is maximized. They conjectured that the ToR problem is NP-complete and developed a heuristic-based solution (SARK) that ranks each AS based on how close to the graph's core it appears from multiple vantage points. Broadly, as in Gao [18], ASes of similar rank are inferred to have a p2p relationship, and the rest are inferred to have a p2c relationship. Di Battista, Erlebach *et al.* [9] proved the ToR problem formulation was NP-complete in the general case and unable to infer p2p relationships. They reported that it was possible to find a solution provided the AS paths used are valley-free. They developed solutions to infer c2p relationships, leaving p2p and sibling inference as open problems. Neither Subramanian *et al.* or Di Battista, Erlebach *et al.* validated their inferences; rather, they determined the fraction of valley-free paths formed using their inferences.

Dimitropoulos *et al.* [16] created a solution based on solving MAX-2-SAT. They inferred sibling relationships using information encoded in WHOIS databases. Their algorithm attempted to maximize two values: (1) the number of valley-free paths, and (2) the number of c2p inferences where the node degree of the provider is larger than the customer. The algorithm uses a parameter α to weight these two objectives. They validated 3,724 AS relationships (86.2% were c2p, 16.1% p2p, and 1.2% sibling) and found their algorithm correctly inferred 96.5% of c2p links, 82.8% of p2p links, and 90.3% of sibling links. Their validation covered 9.7% of the public AS-level graph and has thus far been the most validated algorithm. However, MAX-2-SAT is NP-hard and their implementation does not complete in a practical length of time for recent AS graphs.

UCLA's Internet Research Laboratory produces AS-level graphs of the Internet annotated with relationships [3]. The method is described in papers by Zhang *et al.* [37] and Oliveira *et al.* [29]. Their algorithm begins with a set of ASes inferred to be in the Tier-1 clique, then infers links seen by these ASes to be p2c; all other links are p2p. Zhang [37] describes a method to infer the clique; Oliveira [29] assumes the Tier-1 ASes are externally available, such as a list published by Wikipedia. There are a growing number of region-specific c2p relationships visible only below the provider AS, causing this approach to assign many p2p relationships that are actually p2c. Gregori *et al.* [23] used a similar approach; for each AS path, their algorithm identifies the relationships possible and infers the actual relationship based on the lifetime of the paths. None of [23, 29, 37] describes validation.

The problematic lack of validation puts AS relationship inference research in a precarious scientific position. Nonetheless, research continues to build on the assumption that meaningful AS relationship inference can be achieved and applied to the study of the Internet, from deployment of security technologies [21], Internet topology mapping [6,8,12] and evolution [13,14], to industry complexity [17] and market concentration [15]. Due to the diversity in inter-domain connectivity, relationship inferences do not (by themselves) consistently predict paths actually taken by packets; such predictive capabilities remain an open area of research [27].

Given its importance to the field of Internet research, we revisited the science of AS relationship inference, with particular attention to validation. Our algorithm does not seek to maximize the number of valley-free (hierarchical) paths, since at least 1% of paths are non-hierarchical (section 3.6). We make inferences using AS path triplets (adjacent pairs of links) which allow us to ignore invalid segments of paths that reduce the accuracy of valley-free path maximization approaches. We also rely on two key assumptions: (1) an AS enters into a c2p relationship to become globally reachable, i.e. their routes are advertised to their provider's providers, and (2) there exists a *clique* of ASes at the top of the hierarchy that obtain full connectivity through a full mesh of p2p relationships. We acquired external validation for 34.6% of the links in our AS graph, which was certainly a challenge but not one as insurmountable as has been suggested (e.g. [9,18,23,33,34]).

3. DATA

In this section we present the sources of data we use: public BGP data, a list of AS allocations to RIRs and organizations, and multiple sources of validation data.

3.1 BGP paths

We derive BGP paths from routing table snapshots collected by the Route Views (RV) project [5] and RIPE's Routing Information Service (RIS) [4]. Each BGP peer is a *vantage point* (VP) as it shows an AS-level view of the Internet from that peer's perspective. For each collector, we download one RIB file per day between the 1st and 5th of every month since January 1998, and extract the AS paths that announce reachability to IPv4 prefixes. When we extract the AS paths, we discard paths that contain AS-sets and compress path padding (i.e. convert an AS path from "A B B C" to "A B C"). We record all AS paths that are seen in any of the five snapshots and use the union to subsequently infer relationships. We use all paths and not just "stable" paths because backup c2p links are more likely to be included if we use all AS paths, and temporary peering disputes may prevent a normally stable path from appearing stable in our five-day window of data.

Figure 1 shows the number of ASes peering with RV or RIS between 1998 and 2013. We also show the number providing full views (routes to at least 95% of ASes). RV is the only source that provides BGP data collected between 1998 and 2000, and while more than two thirds of its peers provided a full view then, it had at most 20 views during these three years. For the last decade, approximately a third of contributing ASes provide a full view. Most (64%) contributing ASes provide routes to fewer than 2.5% of all ASes. The operators at these ASes likely configured the BGP session with the collector as p2p and therefore advertise only customer routes.

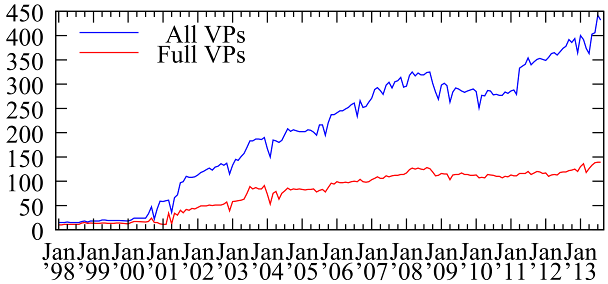


Figure 1: Number of ASes providing BGP data to Route Views and RIS over time. Currently, a third of all contributors provide a full view. The number of ASes providing a full view has not grown since 2008.

3.2 Allocated ASNs

We use IANA's list of AS assignments [2] to identify valid AS numbers assigned to organizations and RIRs. We filter out BGP paths that include unassigned ASes, since these ASes should not be routed on the Internet.

3.3 Validation data directly reported

Our website provides the ability to browse our relationship inferences and submit validation data. We use corrections to two separate inferred datasets, created in January 2010 and January 2011 using a previous relationship inference algorithm. This older algorithm did not produce a cycle of p2c links but assigned many more provider relationships than ASes actually had; 93% of the website feedback consists of p2p relationships, and 62% of that consists of correcting our inference of a c2p relationship to a p2p relationship. In total, we received 129 c2p and 1,350 p2p relationships from 142 ASes through our website. The disparity between our previous inferences and the submissions motivated the new algorithm that we present in this paper.

We separately followed up on unchanged inferences with operators, as well as submissions that seemed erroneous compared to observations in public BGP paths. We received responses from more than 50 network operators. 18 relationships submitted through our website were later acknowledged by the submitting operator to have been inaccurately classified (9 by one operator) or to have changed subsequent to the submission. Additionally, based on email exchanges with operators, we assembled a file containing 974 relationships – 285 c2p and 689 p2p (contained within the "directly reported" circle of figure 2, described further in section 3.6).

3.4 Validation data derived from RPSL

Routing policies are stored by network operators in public databases using the Routing Policy Specification Language (RPSL) [7]. The largest source of routing policies is the RIPE WHOIS database, partly because many European IXPs require operators to register routing policies with RIPE NCC. The routing policy of an AS is stored as part of the *aut-num* record [7]. The aut-num record lists import and export rules for each neighbor AS. An import rule specifies the route announcements that will be accepted from the

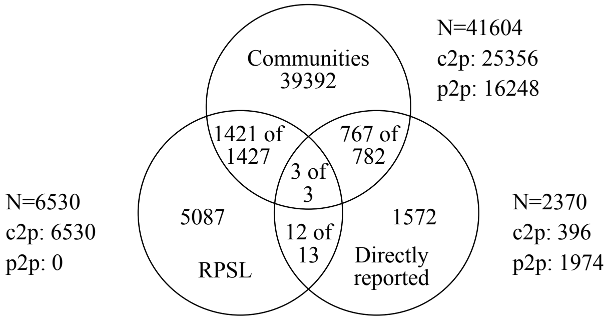


Figure 2: Summary of validation data sets collected and agreement among sets (first number inside intersections is number of overlapping relationships that agree). Overall, 2203 of 2225 relationships agree (99.0%) suggesting a limit on the accuracy of any source of validation data.

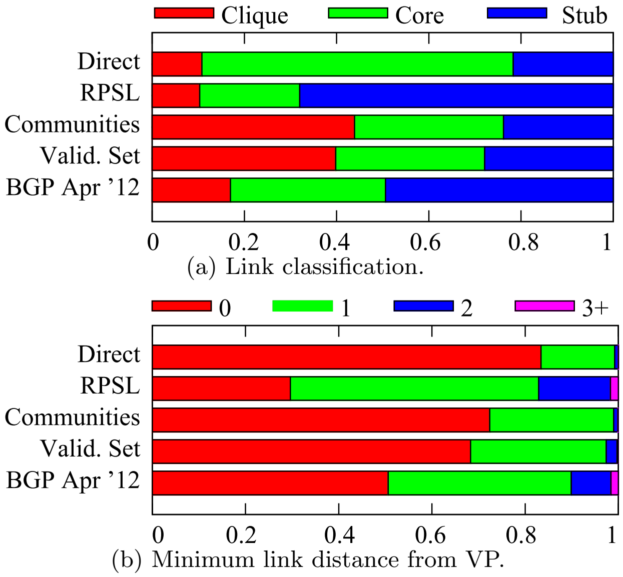
neighbor, while an export rule specifies what routes will be advertised to the neighbor. The special rule ANY is used by an AS to import/export all routes from/to the neighbor, and is indicative of a customer/provider relationship. Using the RIPE WHOIS database from April 2012, we extracted a set of c2p relationships using the following method: if X has a rule that imports ANY from Y, then we infer a c2p relationship if we also observe a rule in Y's aut-num that exports ANY to A. Limiting ourselves to records updated between April 2010 and April 2012 provided us with 6,530 c2p links between ASes with records in RIPE NCC's database.

3.5 Validation data derived from communities

AS relationships can be embedded in BGP community attributes included with each route announcement. Community attributes can be *tagged* to a route when it is received from a neighbor. Community attributes are optional transitive attributes; they can be carried through multiple ASes but could be removed from the route if that is the policy of an AS receiving the route [11]. The tagging AS can annotate the route with attributes of the neighbor and where the route was received, and the attributes can be used to control further announcements of the route. The commonly used convention is for the tagging AS to place its ASN, or that of its neighbor, in the first sixteen bits. The use of the remaining 16 bits is not standardized, and ASes are free to place whatever values they want in them. Many ASes publicly document the meaning of the values on network operations web sites and in IRR databases, making it possible to assemble a dictionary of community attributes and their policy meanings. We used a dictionary of 1286 community values from 224 different ASes assembled from [22] to construct a set of relationships from BGP data for April 2012; in total, there are 41,604 relationships in our set (16,248 p2p and 23,356 c2p).

3.6 Summary of validation data

Figure 2 uses a Venn diagram to show the size and overlap of our validation data sources. Overall, 2203 of 2225 relationships that overlap agree (99.0%), with multiple explanations for the discrepancies. For the directly reported source, some operators reported a few free transit relationships as peering relationships, i.e., they were reported in the traditional economic sense rather than in a routing sense. For the RPSL source, some providers mistakenly imported all routes from their customers and some customers mistakenly exported all routes to their providers. For the BGP communities source, some customers tagged routes exported to a Tier-1 AS as a customer. While there are limits to the accuracy of all our sources of validation, the 99.0% overlap in agreement gives us confidence in using them for validation.



(a) Link classification.

(b) Minimum link distance from VP.

Figure 3: Characteristics of validation data. Relative to BGP data, *clique* links and links directly connected to VPs are over-represented and *stub* links are under-represented.

We assembled our validation data set by combining our four data sources in the following order: (1) directly reported using the website, (2) RPSL, (3) BGP communities, and (4) directly reported in an email exchange. Where a subsequent source classified a link differently, we replaced the classification; we trust relationships acquired through email exchanges more than relationships submitted via the website. Our validation data set consists of 48,276 relationships: 30,770 c2p and 17,506 p2p.

To estimate possible biases in our validation data set, we compared its characteristics with those of the April 2012 BGP dataset in terms of the link types and the minimum distance from a VP at which those links were observed. The closer the link is to a vantage point, the more likely we are to see paths that cross it. We classify links as *clique* (one endpoint is in the clique), *core* (both endpoints are not in the clique and are not stubs), and *stub* (one endpoint is a stub and the other endpoint is not in the clique). Figure 3(a) shows *clique* links are over-represented in our validation data set as compared to BGP data, while *stub* links are under-represented. This disparity is due to the validation data from BGP communities, which mostly comes from large ASes. Figure 3(b) shows links directly connected to a VP (distance 0) are over-represented in our validation data relative to April 2012 BGP data, likely due to the communities dataset, many of which involve ASes that provide VPs.

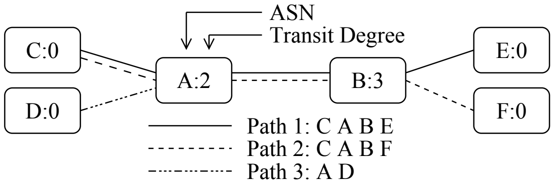


Figure 4: Computing the transit degree of ASes using paths. While the node degrees of ASes A and B are both three, A's transit degree is two because A it is not observed to announce D's prefixes to any neighbors. Nodes with a transit degree of zero (C, D, E, F) are stub ASes.

4. RELATIONSHIP INFERENCE

Our algorithm uses two metrics of AS connectivity: the *node degree* is the number of neighbors an AS has; and the *transit degree* is the number of unique neighbors that appear on either side of an AS in adjacent links. Figure 4 illustrates the transit degree metric; ASes with a transit degree of zero are stub ASes. We use transit degree to initially sort ASes into the order in which we infer their relationships, breaking ties using node degree and then AS number. ASes inferred to be in the clique are always placed at the top of this rank order. Sorting by transit degree reduces ordering errors caused by stub networks with a large peering visibility, i.e., stubs that provide a VP or peer with many VPs.

4.1 Assumptions

We make three assumptions based on discussions with operators and generally understood industry structure.

Clique: multiple large transit providers form a peering mesh so that customers (and indirect customers) of a transit provider can obtain global connectivity without multiple transit provider relationships.

A provider will announce customer routes to its providers. All ASes, except for those in the clique, require a transit provider in order to obtain global connectivity. We assume that when X becomes a customer of Y, that Y announces paths to X to its providers, or to its peers if Y is a clique AS. Exceptions to this rule include backup and region-specific transit relationships.

The AS topology can be represented in a directed acyclic graph. Gao *et al.* argue there should be no cycle of p2c links to enable routing convergence [19].

4.2 Overview

Algorithm 1 shows each high-level step in our AS relationship inference algorithm. First, we sanitize the input data by removing paths with artifacts, i.e., loops, reserved ASes, and IXPs (step 1). We use the resulting AS paths to compute the node and transit degrees of each AS, and produce an initial rank order (step 2). We then infer the clique of ASes at the top of the hierarchy (step 3). After filtering out poisoned paths (step 4), we apply heuristics to identify c2p links (steps 5-10). Finally, we classify all remaining unclassified links as p2p. Our algorithm contains many steps, a consequence of trying to capture the complexity of the real-world Internet and mitigate limitations of public BGP data [32]. The output from this process is a list of p2c and p2p relationships with no p2c cycles, by construction.

Algorithm 1 AS relationship inference algorithm.

```
Require: AS paths, Allocated ASNs, IXP ASes
 1: Discard or sanitize paths with artifacts (§4.3)
 2: Sort ASes in decreasing order of computed transit degree, then node degree (§4)
 3: Infer clique at top of AS topology (§4.4)
 4: Discard poisoned paths (§4.3)
 5: Infer c2p rels. top-down using above ranking (§4.5)
 6: Infer c2p rels. from VPs inferred not to be announcing provider routes (§4.5)
 7: Infer c2p rels. for ASes where the provider has a smaller transit degree than the customer (§4.5)
 8: Infer customers for ASes with no providers (§4.5)
 9: Infer c2p rels. between stubs and clique ASes (§4.5)
10: Infer c2p rels. where adjacent links have no relationship inferred (§4.5)
11: Infer remaining links represent p2p rels. (§4.5)
```

4.3 Filtering and Sanitizing AS Paths

We first sanitize the BGP paths used as input to our algorithm, especially to mitigate the effects of BGP path poisoning, where an AS inserts other ASes into a path to prevent its selection. A poisoned path implies a link (and thus relationship) between two ASes, where in reality neither may exist. We infer poisoning, or the potential for a poisoned path, in AS paths (1) with loops, or (2) where clique ASes are separated. We filter out paths with AS loops, i.e., where an ASN appears more than once and is separated by at least one other ASN. Such paths are an indication of poisoning, where an AS X prevents a path from being selected by a non-adjacent upstream AS Y by announcing the path "X Y X" to provider Z, so if the route is subsequently received by *Y* it will be discarded when the BGP process examines the path for loops [24]. For BGP paths recorded in April 2012, 0.11% match this rule. After we have inferred the clique in step 3, we also discard AS paths where any two ASes in the clique are separated by an AS that is not in the clique. This condition indicates poisoning, since a clique AS is by definition a transit-free network. For BGP paths recorded in April 2012, 0.03% match this rule.

We also filter out paths containing unassigned ASes; in BGP paths from April 2012, we observed 238 unassigned ASes in 0.10% of unique paths. 222 of these ASes are reserved for private use and should not be observed in paths received by route collectors. In particular AS23456, reserved to enable BGP compatibility between ASes that can process 32-bit ASNs and those that cannot [35], is prevalent in earlier public BGP data and can hinder relationship inferences.

We also remove ASes used to operate IXP route servers because the relationships are between the participants at the exchange. Unfortunately we know of no public database of IXP route server ASes, nor do we know of any algorithm that reliably identifies an IXP route server. To identify IXP route server ASes, we manually searched for IXP ASes using a combination of routing and WHOIS data: if the name of the AS in WHOIS contains RS or IX, we include the AS in our list if the AS announces less address space than a /23. This process yielded a list of 25 ASes known to operate route servers; we remove these ASes from paths so that the IX participants are adjacent in the BGP path.

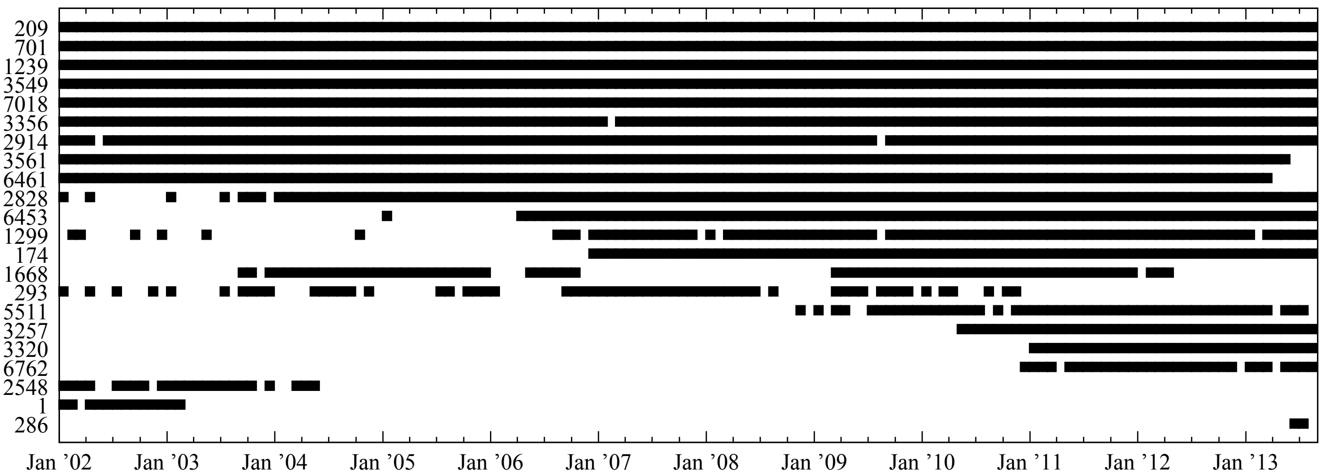


Figure 5: ASes inferred to be in the clique over time. We plot the 22 of the 26 ASes inferred to be in the clique at any time after January 2002. The clique's small size and consistent membership lend confidence in our inference methodology. AS names change over time so we do not label them. As an artifact of the AS3356/AS3549 merger process in 2013, clique member AS6461 was inferred not to be in the clique.

Finally, we discarded all paths from 167.142.3.6 for May and June 2003, and from 198.32.132.97 between March and November 2012; the former reported paths with ASes removed from the middle of the path, and the latter reported paths inferred from traceroute.

4.4 Inferring Clique

We attempt to infer the ASes present at the top of the hierarchy. Since Tier-1 status is a financial circumstance, reflecting lack of settlement payments, we focus on identifying transit-free rather than Tier-1 ASes. First, we use the Bron/Kerbosch algorithm [10] to find the maximal clique C_1 from the AS-links involving the largest ten ASes by transit degree.[1] Second, we test every other AS in order by transit degree to complete the clique. AS Z is added to C_1 if it has links with every other AS in C_1 and it does not appear to receive transit from another member of C_1; i.e. no AS path should have three consecutive clique ASes. Because AS path poisoning may induce three consecutive clique ASes in a false BGP path "X Y Z", we add AS Z to C_1 provided there are no more than five ASes downstream from "X Y Z". A nominal value of five ASes will still capture clique ASes even in paths poisoned multiple times, yet is unlikely to wrongly place a large transit customer in the clique since a clique AS is likely to announce (and we are likely to observe [28]) more than five customers of large transit customers. If an AS would be admitted to C_1 except for a single missing link, we add that AS to C_2. Finally, because an AS might be in C_1 but not in the clique, we re-use the Bron/Kerbosch algorithm to find the largest clique (by transit degree sum) from the AS-links involving ASes in C_1 and C_2. The product of this step is a clique of transit-free ASes.

Figure 5 shows ASes inferred to be in the clique since January 2002. Nine ASes have been in the clique nearly every month, and ASes that are inferred to be in the clique are almost continuously present. The consistency of the inferred clique and our discussions with operators give us confidence in our clique inference methodology. However, peering disputes and mergers of ASes can disrupt our inference of the clique. ASes may form alliances to prevent de-peering incidents from partitioning their customers from the Internet. If such a disconnection incident triggers activation of a backup transit relationship, a peer will disappear from the clique and instead be inferred as a customer of the allied peer. The process of merging ASes can can also result in peers being inferred as customers. For example, in 2013 Level3 (AS3356) gradually shut down BGP sessions established with Global Crossing (AS3549), shifting sessions to AS3356. In order to maintain global connectivity during this merger process, Level3 advertised customers connected to AS3549 to peers that were only connected to AS3356. As a result, ASes in the clique appeared to be customers of AS3356, when in reality they were peers. Specifically, in figure 5, AS6461 was not inferred to be a member of the clique because it had shifted all peering ports with Level3 to AS3356.

4.5 Inferring Providers, Customers, and Peers

The remainder of the algorithm infers p2c and p2p relationships for all links in the graph. Step 3 infers p2p relationships for the full mesh of links between clique ASes. The rest of this section uses figure 6 as reference.

AS path triplets: We make inferences using only AS path triplets (adjacent pairs of links). Triplets provide the constraints necessary to infer c2p relationships while allowing us to ignore non-hierarchical segments of paths, and are more computationally efficient than paths. For example, in figure 6 we break path 1 into two triplets: "1239 3356 9002" and "3356 9002 6846".

Notation: We use the notation in table 1 to describe relationships between ASes. A p2c relationship between X and Y is presented as "X > Y". The notation reflects providers as typically greater (in degree or size or tier of a traditional hierarchical path) than their customers. A triplet with no inferred relationships is presented as "X ? Y ? Z".

[1] Starting with ten ASes reveals most clique ASes and is small enough to prevent the incorrect inference of a clique below the top of the hierarchy. If there are multiple cliques, we select the clique with the largest transit degree sum.

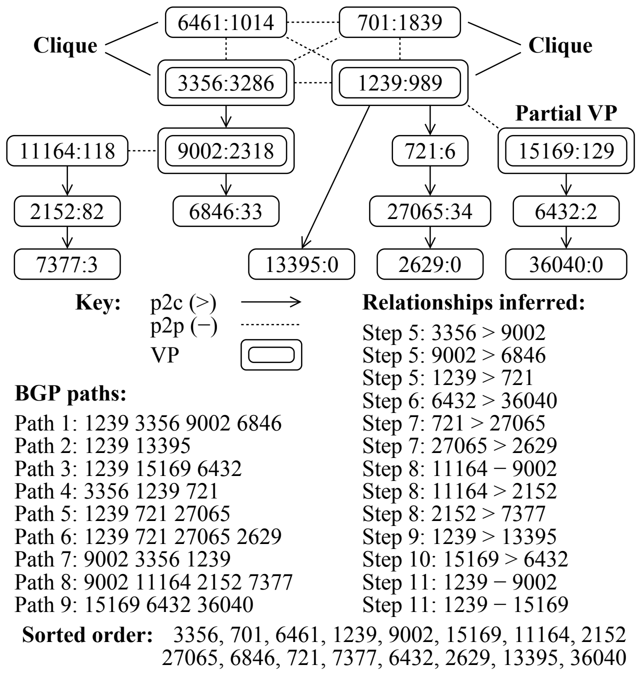


Figure 6: Inferring providers, customers, and peers. Each AS is labeled "AS Number:Transit Degree". VPs are in double squares, and (by definition) on the left side of all raw BGP paths. We use this set of paths to illustrate the inferences made at each step of our algorithm. Relationships listed use notation in Table 1.

Notation	Description
X < Y	X is a customer of Y
X − Y	X is a peer of Y
X ? Y	No inferred relationship

Table 1: Notation used to describe relationships.

Sorting of ASes: We sort ASes in the order we will estimate their c2p relationships. ASes in the clique are placed at the top, followed by all other ASes sorted by transit degree, then by node degree, and finally by AS number to break ties. We sort by transit degree rather than node degree to avoid mistaking high-degree nodes (but not high-transit degree, e.g., content providers) for transit providers. Figure 6 shows the sorted order of the ASes in that graph.

Preventing cycles of p2c links: When we infer a c2p relationship we record the customer AS in the provider's customer cone, and add the AS to the cones of all upstream providers. Any ASes in the customer's cone not in the cones of its upstream providers are also added to the cones of those providers. We do not infer a c2p relationship if the provider is already in the AS's cone, to prevent a cycle of p2c links.

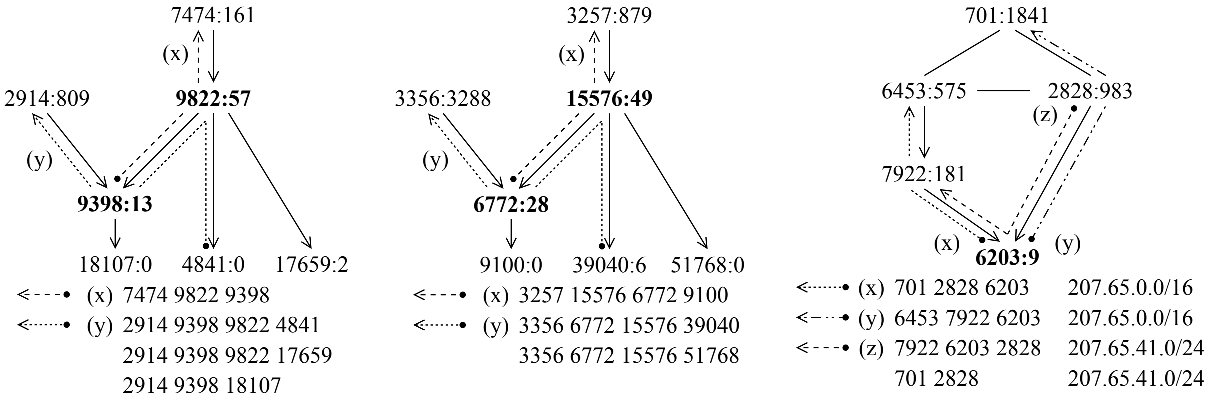
Step 5: Infer c2p relationships top-down using ranking from Step 2. This step infers 90% of all the c2p relationships we infer, and is the simplest of all our steps. We visit ASes top-down, skipping clique ASes since they have no provider relationships. When we visit AS Z, we infer Y > Z if we observe "X − Y ? Z" or "X > Y ? Z". To have observed "X − Y", X and Y must be members of the clique (step 3). To have inferred "X > Y" by now, one must have visited Y in a previous iteration of this step. No cycle of c2p links can be formed because c2p relationships are assigned along the *degree gradient*, i.e. no c2p relationship is inferred between two ASes where the provider has a smaller transit degree, a necessary condition to create a cycle. For example, in figure 6, we first consider (after the four clique ASes) c2p relationships for 9002 (3356), then 15169 (none), etc.

The order of the ASes in the triplet is important, at this step and for most of the remaining steps. To minimize false c2p inferences due to misconfigurations in one direction of a p2p relationship (an AS leaks provider or peer routes to peers), we infer a c2p relationship when we observe the provider or peer closer than the customer to at least one VP in at least one triplet. This heuristic builds on the intuition that an AS enters a provider relationship to become globally reachable, i.e., at least one VP should observe the provider announcing the customer's routes. For example, when we infer 3356 > 9002 in figure 6, we use triplet "1239 3356 9002" from path 1 and not triplet "9002 3356 1239" from path 7 because 3356 appears before 9002 in path 1.

Step 6: Infer c2p relationships from VPs inferred to be announcing no provider routes. We assume that "partial VPs" providing routes to fewer than 2.5% of all ASes either (1) export only customer routes, i.e., the VP has configured the session with the collector as p2p; or (2) have configured the session as p2c, but have a default route to their provider, and export customer and peer routes to the collector. Given a path "X ? Y ? Z" where X is a partial VP and Z is a stub, the link XY can either be p2c or p2p, requiring Y > Z. In figure 6, we use path 9 and the inference of 15169 as a partial VP to infer 6432 > 36040.

Step 7: Infer c2p relationships for ASes where the customer has a larger transit degree. Given a triplet "W > X ? Y" where (1) Y has a larger transit degree than X, and (2) at least one path ended with "W X Y" (i.e. Y originates a prefix to X), then we assign X > Y. We assume c2p relationships against the degree gradient are rare, although they can arise from path poisoning. Condition (2) mitigates the risk of using poisoned paths because poisoned segments of a path do not announce address space. In figure 6 we infer 721 (X) > 27065 (Y) because path 5 shows 27065 announcing a prefix to 721. In the absence of path 5, we would infer that 2629 poisoned path 6 with 27065 and we would not assign a c2p relationship. When we assign X > Y, we also assign Y > Z where we observe triplets "X > Y ? Z"; in figure 6 we use path 6 to infer 27065 > 2629.

Step 8: Infer customers for provider-less ASes. We visit provider-less ASes top-down, skipping clique members because their customers were inferred in step 5. This step is necessary because steps 5 and 7 require an AS to have a provider in order to infer customers. Examples of provider-less ASes are some regional and research networks, e.g., TransitRail. For each provider-less AS X, we visit each of its neighbors W top-down. When we observe triplet "W X Y", we infer W − X because we never observed W announcing X to providers or peers in previous steps; therefore, X > Y. We remove the condition in step 5 that the peer AS must be closest to the VP, because provider-less ASes are mostly observed by downstream customers providing a public BGP view. In figure 6, 11164 (X) is a provider-less AS; we use path 8 to infer 9002 (W) − 11164 (X) and 11164 (X) > 2152 (Y). When we assign X > Y, we also assign Y > Z where we



(a) Siblings (9398 and 9822) (b) Mutual Transit (6772 and 15576) (c) Leaking or Poisoning (6203)

Figure 7: p2c-c2p valleys caused by unconventional routing policies between (a) siblings, (b) mutual transit, and (c) leaking/poisoning. Each AS is labeled with its transit degree, which influences the order of p2c inferences. An AS-to-organization map would resolve (a) but not (b) because the ASes in (b) are independent ASes. Leaking as in (c) results in paths with spurious p2c-c2p valleys when an AS leaks a route from a provider to another provider. Note that in (c) AS6203 could instead be poisoning to prevent AS2828 from selecting the more specific route (traffic engineering); we believe (c) to be an example of a prefix-list leak because AS2828 is the only origin AS observed for that prefix. All examples are present in April 2012 BGP data.

observe triplets "X > Y ? Z"; in figure 6 we also use path 8 to infer 2152 (Y) > 7377 (Z).

Step 9: Infer that stub ASes are customers of clique ASes. If there is a link between a stub and a clique AS, we classify it as c2p. This step is necessary because step 5 requires a route between a stub and a clique AS to be observed by another clique AS before the stub is inferred to be a customer. Stub networks are extremely unlikely to meet the peering requirements of clique members, and are most likely customers. In figure 6, path 2 reveals a link between 1239 and 13395, but there is no triplet with that link in the set, perhaps because it is a backup transit relationship.

Step 10: Resolve triplets with adjacent unclassified links. We again traverse ASes top down to try to resolve one link as p2c from triplets with adjacent unclassified links. We do this step to avoid inferring adjacent p2p links in step 11, since adjacent p2p links imply anomalous behavior, e.g., free transit or route leakage. We loosen the requirement in step 5 that the first half of the triplet must be resolved. When we visit Y, we search for *unresolved triplets* of the form "X ? Y ? Z", and attempt to infer Y > Z. For each unresolved triplet "X ? Y ? Z", we look for another triplet "X ? Y < P" for some other P. If we find one, we infer X < Y (and Y will be inferred as a peer of Z in step 11). Otherwise we search for a triplet "Q ? Y ? X", which implies Y > X, and therefore we would resolve both sides of the original unresolved triplet to X < Y and Y > Z. Since we are only confident of resolving one side of the original triplet (embedding an assumption that most p2c links have already been resolved at earlier steps), we make no inferences in this case. Otherwise, we infer Y > Z in this step, and X − Y in step 11.

Step 11: Infer p2p links: We assign p2p relationships for all links that have no inferred relationships.

4.6 Complex Relationships

Sibling Relationships and Mutual Transit: Our algorithm does not infer sibling relationships, where the same organization owns multiple ASes, which can therefore have unconventional export policies involving each sibling's peers, providers, and customers. Similarly, we do not infer mutual transit relationships, where two independent organizations provide transit for each other in a reciprocal arrangement. Both of these arrangements can lead to paths (and triplets) that violate the valley-free property, and in particular produce p2c-c2p valleys in paths. Gao's algorithm [18] inferred that two ASes involved in a non-hierarchical path segment were siblings, which maximizes the number of valley-free paths. Dimitropoulos *et al.* used WHOIS database dumps to infer siblings from ASes with similar organization names, because policy diversity among siblings makes it difficult to infer siblings from BGP data [16]. Our algorithm does not attempt to resolve these unconventional routing policies because it is difficult to accurately classify them; as a result, our algorithm produces p2c-c2p valleys in paths.

Figure 7 provides three examples of non-hierarchical path segments caused by siblings (figure 7(a)), mutual transit (figure 7(b)), and route leaks or path poisoning (figure 7(c)). In figure 7(a), ASes 9398 and 9822 are ASes owned by the same organization, Amcom Telecommunications, which implements complex export policies with these ASes. Specifically, we observe customers of 9822 are exported to 9398's peers and providers, and routes originated by 9398 are exported to 9822's providers. These policies induce a p2c-c2p valley in a path, because we infer 9398 is a customer of both 9822 (path x) and 2914, and observe 9398 to announce customers of its inferred provider 9822 to its other inferred provider (path y). In figure 7(b), independently operated ASes 6772 and 15576 implement mutual transit, owing to complementary traffic profiles: AS6772 is an access provider with mostly inbound traffic, while AS15576 is a content provider, with mostly outbound traffic. A WHOIS-derived database of sibling relationships does not help infer mutual transit arrangements. Finally, route leaks and path poisoning can also result in a p2c-c2p valley in a path. Figure 7(c) provides an example of a route leak from 6203,

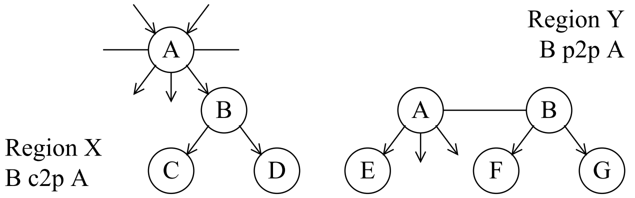


Figure 8: Two ASes may have hybrid relationships. In this example B is a customer of A in region X and B is a peer of A in region Y. However, our algorithm infers a single relationship between B and A: c2p. If A routes rationally, it will only advertise paths to F and G from B to customers.

where it provides transit to a /24 prefix announced by one of its providers (AS2828) to another provider (AS7922). The prefix AS2828 announces is a more specific prefix of a /16 prefix that AS6203 announces, and has likely leaked because of a prefix list configured in AS6203. An alternative explanation is that AS6203 is poisoning that path so that AS2828 cannot select the more specific prefix. We believe a route leak is the more plausible explanation because AS2828 is the only AS to originate the prefix. Without the use of prefixes, it is not possible to distinguish this valley from mutual transit or siblings. We do not currently use prefixes in our inference algorithm; detecting and validating route leaks and path poisoning remains an open problem.

We used sibling inferences derived from WHOIS database dumps [25] to evaluate how the sibling links in our topology were classified. We did not use these sibling inferences in our algorithm because (1) we did not have supporting WHOIS databases going back to 1998, (2) in validation the sibling inferences contained a significant number of false-positives (where two ASes were falsely inferred to belong to an organization), and (3) they do not help distinguish mutual transit between independent ASes from other anomalies such as path poisoning and route leaks. In total, there were 4537 links observed between inferred siblings for April 2012; we inferred 4238 (93%) of these to be p2c. Because we inferred most of the siblings to have p2c relationships (i.e. a transit hierarchy) we also used the sibling inferences to examine if the ordering of ASes in paths supported the classification. Of the 312 organizations for which we observe at least two siblings in a path, 275 (88%) had a strict ordering; i.e. AS x was always observed in a path before sibling AS y. For example, we observed 21 Comcast sibling ASes in BGP paths with at least two siblings present; all of these ASes were connected beneath AS7922. It is possible that Comcast's siblings exported routes received from their peers (if any) to other siblings. However, we did not see any peer links from beneath AS7922 (perhaps due to limited visibility), and it would make more engineering sense to share peer routes among siblings by connecting the peer to AS7922.

Partial Transit, Hybrid Relationships, and Traffic Engineering: Our algorithm infers either a p2p or c2p relationship for all links. A partial transit relationship typically restricts the propagation (and visibility) of routes beyond a provider to the provider's peers and customers. Similarly, complex import and export policies can produce hybrid relationships [27]. Figure 8 depicts an AS pair with a c2p relationship in one region, and a p2p relationship elsewhere. Because publicly available BGP data only includes the best paths announced from each AS, there is no way to distinguish hybrid relationships from traffic engineering practices such as load balancing across providers. Similarly, ASes A and B may enter into a p2p relationship, but B may not advertise all customers or prefixes to A, requiring A to reach those via a provider. We consider hybrid relationships and traffic engineering when computing the customer cone of each AS, which we describe in section 5. Modeling routing policies remains an open problem [27].

The presence of hybrid relationships can cause our algorithm to incorrectly infer a c2p link as a p2p link. In figure 8, if we observe no providers of A announcing routes to E via A then our algorithm infers the c2p link between A and E as p2p. This inference occurs because step 10 collapses triplets with no relationships inferred for either link; i.e., it does not adjust the triplet "E ? A > B". A separate step is necessary to distinguish partial transit relationships visible from a VP below the provider from peering relationships.

Paid Peering: An assumption behind c2p and p2p relationships is that the customer pays the provider and p2p relationships are settlement-free, as historically p2p relationships were viewed as mutually beneficial. Modern business relationships in the Internet are more complicated; ASes may enter into a paid-peering arrangement where an AS pays settlements for access to customer routes only. Multiple network operators confirmed several aspects of paid-peering: (1) approximately half of the ASes in the clique (figure 5) were paid-peers of at least one other AS in the clique as of June 2012; (2) paid-peering occurs between ASes at lower-levels of the AS topology; and (3) routes from paying and settlement-free peers have the same route preference. This last condition prevents us from distinguishing paid-peering from settlement-free peering using BGP data alone.

Backup Transit: A backup transit relationship occurs when a customer's export policies prevent their routes from being exported outside a provider's customer networks. The export policies used while the provider is in backup configuration are identical to peering; the difference between backup transit and paid peering is due to export filters instead of a contractual agreement. Our algorithm infers most backup transit relationships as peering.

4.7 Validation

In this section we evaluate the positive predictive value (PPV) and true positive rate (TPR, or *recall*) of our heuristics against our validation dataset (section 3.6). Our AS relationships dataset consists of 126,082 links; we validated 43,613 (34.6%) of these. Table 2 shows the PPV of inferences made at each step of our algorithm. Most relationship inferences are made in steps 5 (56.4%) and 11 (37.7%), and both of these stages have excellent PPV (99.8% and 98.7%).

Table 3 compares the PPV of our inferences and those made by four other popular inference algorithms for April 2012 BGP paths. Our algorithm correctly infers 99.6% of c2p relationships and 98.7% of p2p relationships. We asked the authors of SARK [34], CSP [27], and ND-ToR [33] for the source code for their algorithm, or for a set of relationships inferred from April 2012 BGP paths; we did not receive either. Gao's PPV for p2p relationships is the highest of the algorithms tested because it makes the fewest number of p2p inferences of all the algorithms, inferring many more c2p relationships than exist in the graph. The algorithm that per-

Step	Description	Validation (PPV)	Fraction
3	clique at top of AS topology	136 p2p @ 100%	153 (0.12%)
5	**c2p relationships top-down**	26664 c2p @ 99.8%	**71160 (56.4%)**
6	c2p relationships from VPs announcing no provider routes	116 c2p @ 99.1%	532 (0.42%)
7	c2p relationships to smaller degree providers	205 c2p @ 96.1%	2420 (1.92%)
8	relationships for ASes with no providers	120 c2p @ 93.3%	842 (0.67%)
		152 p2p @ 96.7%	333 (0.26%)
9	c2p relationships for stub-clique	422 c2p @ 95.0%	651 (0.52%)
10	collapse adjacent links with no relationships	524 c2p @ 94.7%	2474 (1.96%)
11	**p2p relationships for all other links**	15274 p2p @ 98.7%	**47517 (37.7%)**
		43613 @ 99.3%	126082 (100%)

Table 2: Validation of inferences (PPV) and number/fraction of inferences made at each step.

Algorithm		c2p			p2p	
	PPV	TPR	Errs	PPV	TPR	Errs
	(%)	(%)	(1/)	(%)	(%)	(1/)
CAIDA	99.6	99.3	250	98.7	99.3	77
UCLA	99.0	94.7	100	91.7	98.8	12
Xia+Gao	91.3	98.6	11	96.6	81.1	29
Isolario	90.3	98.0	10	96.0	82.4	25
Gao	82.9	99.8	5.8	99.5	62.5	200

Table 3: Our AS relationship algorithm accurately classifies both c2p and p2p relationships, with high precision (PPV) and recall (TPR).

forms closest to ours is UCLA's; our improvements result in six times fewer false peering inferences. We assembled additional historical validation datasets by extracting relationships from archives of the RIPE WHOIS database (RPSL, section 3.4) and public BGP repositories (BGP communities, section 3.5) at six month intervals between February 2006 and April 2012. The validation performance of Gao, UCLA, and CAIDA algorithms are quantitatively similar as shown in table 3.

We investigated the types of errors that these four algorithms produce, focusing on the two cases with significant occurrence: where we correctly infer c2p (p2p), but another algorithm mistakenly infers p2p (c2p). We note that when the ground truth is p2p, Gao often infers the link as c2p, usually with the customer having a smaller degree than the provider. On the other hand, UCLA and Isolario produce errors where a p2p link is inferred to be c2p, often with the customer having a larger degree than the provider. The UCLA algorithm often infers c2p links to be p2p because it uses the visibility of a link from tier-1 VPs to draw inferences, and defaults to a p2p inference for links it cannot see (see section 2). We agree with the intuition behind this visibility heuristic and use a variant of it in our algorithm, but we use additional heuristics to accommodate for phenomena that inhibit visibility through tier-1 VPs, e.g., traffic engineering, selective announcements.

We compared our inferences with 82 partial transit relationships that were flagged by a community string. Our algorithm correctly inferred 69 (84%) of them as p2c; 66 p2c inferences where made in step 10. In comparison, UCLA's dataset identified only 13 (16%) of the partial transit relationships as p2c. We also compared our inferences against a small set of 27 backup p2c relationships, of which only 2 were correctly identified as p2c. Validation data for partial and backup transit relationships is scarce because of their rarity and their limited visibility.

It is well-known that the public view misses a large number of peering links [6]. While our algorithm can only make inferences for links we observe, an important question is whether its accuracy is affected by a lack of (or increasing) visibility. We performed the following experiment 10 times. We start with paths selected from a random set of 25% of VPs, and successively add VPs to obtain topologies seen from 50%, 75% and all VPs. We measure the PPV of our inferences on each topology subset. We found that the PPV of c2p inferences was consistently between 99.4% and 99.7% on all topology subsets. We found that the PPV of p2p links varied between 94.6% and 97.7% with 25% of VPs, and 97.2% and 98.4% with 50% of VPs, indicating that our algorithm performs better when it has more data (VPs) available. Consequently, if our visibility of the AS topology increases in the future (e.g., due to new VPs at IXPs), the accuracy of our algorithm at inferring the newly visible links should not be affected.

We agree that collecting validation data is difficult and time-consuming [9, 16, 18, 34] because the relationships are considered confidential and often covered by a non-disclosure agreements, but we gathered validation data from multiple sources that represent more than a third of the publicly available graph. The previous most validated work was Dimitropoulos *et al.* [16], which validated 9.7% of inferences.

5. CUSTOMER CONES

In this section, we use our AS relationship inferences to construct the *customer cone* of each AS. The customer cone is defined as the ASes that a given AS can reach using a customer (p2c) link, as well as customers of those customers (indirect customers). An AS is likely to select a path advertised by a customer (if available) over paths advertised by peers and providers because the AS is paid for forwarding the traffic. The most profitable traffic for an AS is traffic forwarded between customers, as the AS is paid by both.

The customer cone is a metric of influence, but not necessarily of market power. Market power requires the ability to restrict the mobility of customers; in general, an AS can enter into a provider relationship with whoever offers a suitable service. For large transit providers, particularly those in the clique where a full p2p mesh is required for global connectivity, the customer cone defines the set of ASes whose service might be disrupted if the AS were to have operational difficulty. We compare three algorithms to infer an AS's customer cone, and reason why one construction is the most realistic. We discuss the effect of topology flattening on the utility of the customer cone metric, and use our inferences to show how the Internet has flattened from an inter-domain routing perspective.

5.1 Algorithms to compute the customer cone

Due to ambiguities inherent in BGP data analysis, there are multiple methods to infer the customer cone of a given AS. We compare three methods: recursively inferred, BGP observed, and provider/peer observed. All three methods infer the set of ASes that can be reached from a given AS following only p2c links, and the three methods infer the same customer cone for nearly all but the largest ASes.

Recursive: the customer cone of an AS A is computed by recursively visiting each AS reachable from A by p2c links. For example, if B is a customer of A, and C is a customer of B, then A's customer cone includes B and C. Some prior work has defined and used the recursive customer cone (e.g. [15,16]), but this definition unrealistically assumes that a provider will receive all of its customers' routes, and thus be able to reach them following a customer link. This definition can thus distort the size of a customer cone.

BGP observed: given a set of relationships and corresponding BGP paths, C is included in A's customer cone if we observe a BGP path where C is reached following a sequence of p2c links from A. This method addresses two problems of the recursive method. First, A may provide transit for some prefixes belonging to B, but not for B's customers; the BGP observed method will not recursively include customers of B in A's cone that are never announced to A. Second, the error induced by hybrid relationships is reduced because an AS should not announce prefixes received from the peer segment of the hybrid relationship to providers; in figure 8, A's providers will not include E and F in their customer cone unless they receive routes to those ASes from another customer, though A's cone will include those ASes. The main limitations of the BGP observed cone method are: (1) the customer cones of ASes with hybrid relationships will still include customers of peers, and (2) the customer cones of ASes that provide a VP are more likely to be complete and therefore larger as an artifact of the collection apparatus.

Provider/Peer observed: given a set of relationships and corresponding BGP paths, we compute the customer cone of A using routes observed from providers and peers of A. This method addresses the two limitations of the BGP observed method: because A will not announce paths received from the peering portion of a hybrid relationship with AS B to providers and peers, we will not include customers of B observed from the peering portion in the customer cone of AS A. Similarly, because the customer cone of A is computed based on what neighbors of A announce, the presence of a VP at A will no longer inflate A's customer cone relative to ASes that do not provide a VP. The limitation of the provider/peer observed method is that we are only able to view best paths, rather than all paths, so we may underestimate the customer cones of some ASes.

ASN	VP	PP Obs.	BGP Obs.	Recursive
3356	⋆	46.8 (1)	59.1 (1)	78.0 (1)
3549	⋆	45.2 (2)	54.2 (2)	72.3 (2)
3257	⋆	32.6 (3)	33.8 (5)	59.3 (5)
174		31.1 (4)	39.9 (4)	65.1 (3)
1299	⋆	29.3 (5)	40.0 (3)	64.6 (4)
2914	⋆	24.6 (6)	29.8 (6)	57.4 (6)
6453	⋆	18.9 (7)	28.1 (7)	55.8 (7)
6762	⋆	16.9 (8)	18.5 (9)	44.5 (11)
1239	⋆	15.2 (9)	21.0 (8)	51.0 (8)
3491		13.8 (10)	13.9 (12)	32.1 (13)
701	⋆	12.0 (11)	18.2 (10)	47.4 (9)
2828		11.3 (12)	11.4 (13)	45.7 (10)
7018	⋆	10.2 (13)	15.3 (11)	43.7 (12)
1273		8.4 (14)	8.4 (14)	26.7 (14)
6939	⋆	8.1 (15)	8.3 (15)	18.6 (15)

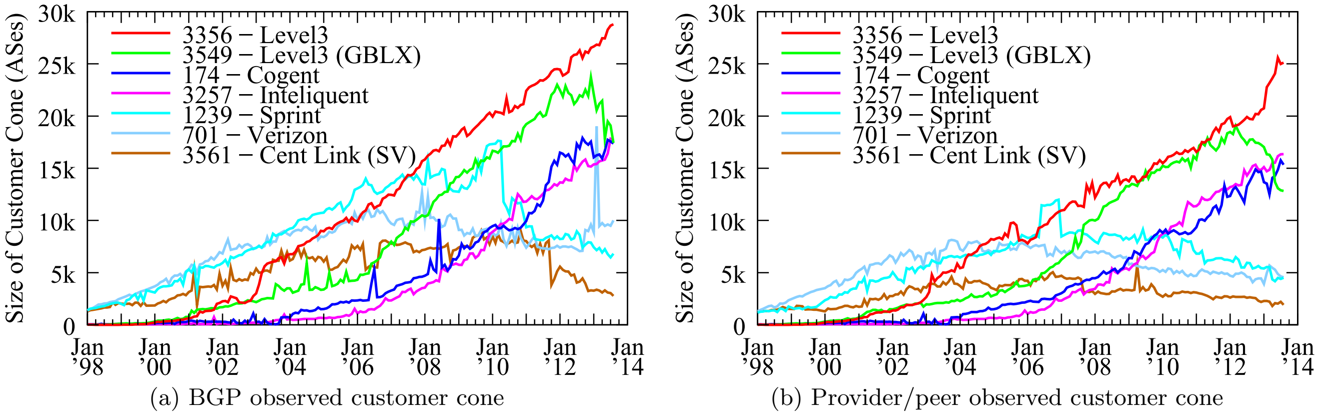
Table 4: For April 2012, the fifteen largest ASes by provider/peer observed customer cone size, their customer cone sizes inferred with recursive and BGP observed algorithms, and their rank by customer cone size. The size of each AS varies significantly, but their ranks are similar.

5.2 Evaluation

Table 4 displays the customer cone sizes of the 15 largest ASes as a percentage of all ASes in the graph using the three methods, as well as their rank order. The rank order is largely independent of the method used to compute the customer cone; for example, the same seven ASes are the largest seven ASes computed with all algorithms. But the table also shows that the recursive cone is significantly larger than the BGP observed cone – for nine of the fifteen ASes shown in Table 4, the recursively defined customer cone is at least twice the size. We found significant incongruity between the customer cones constructed for ASes for which there is also a VP; for example, AS3356 only reaches 60-76% of the ASes in its recursively-defined customer cone over a p2c link. This incongruity makes the recursive method less realistic than the two alternatives we describe.

The BGP observed cone is often larger than the provider/peer observed customer cone for large ASes. There are three exceptions in table 4: ASes 1273, 2828 and 3491, none of which provide a VP. AS174's BGP observed cone is larger than its provider/peer observed cone despite not providing a VP, because one of its customers does. The provider/peer observed method avoids over-inflating ASes that provide a VP relative to ASes that do not, as an AS relies on peers and providers selecting their routes and those routes being observed by a VP to reveal the AS's entire customer cone.

Figure 9 shows the customer cone sizes of ASes that were in the top three (by customer cone size) at any point over the past eleven years, computed using BGP observed and provider/peer observed algorithms. BGP observed cones (figure 9(a)) have spikes that coincide with views of some peering routes received from neighbors with whom the AS has a hybrid relationship. In particular, AS1239's customer cone is consistently larger between October 2009 and May 2010 because a customer provided a view of the routes advertised by AS1239's peer. The provider/peer observed cones (figure 9(b)) have fewer such spikes because these peer routes were not advertised to other peers. A notable exception is AS1239's customer cone between June and December 2006, which corresponds incorrect inference of backup provider links as peer links due to an adjacent hybrid relationship (see section 4.6). In figure 8, if our algorithm incorrectly infers the c2p link between E and A as p2p, it will also infer that F and G are in A's customer cone. The provider/peer



(a) BGP observed customer cone

(b) Provider/peer observed customer cone

Figure 9: The size of the customer cones using BGP observed and provider/peer observed algorithms for the seven ASes that were among the three largest ASes between Jan. 1998 and Aug. 2013. The three largest ASes in Jan. 1998 (701, 1239, and 3561) are no longer in the top three. The BGP observed cone has several spikes, including a six-month spike for AS1239 between Oct. 2009 and May 2010 that is not present in the provider/peer observed cone. We believe the provider/peer observed customer cone is the more realistic customer cone method.

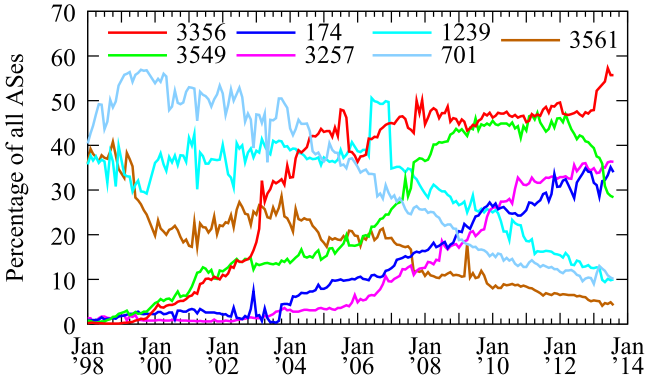


Figure 10: Relative size of provider/peer observed cone over time. 701 acquired part of 3561 in 1999 and moved customers across.

observed cone seems to be the most robust methodology available to infer AS customer cones provided a customer link is not mistakenly inferred as a peer link.

5.3 Customer cone over time

Figure 9(b) plots the seven ASes that ranked in the top three ASes by provider/peer observed customer cone size at any point from January 1998. We can observe several interesting trends with just these seven ASes. First, the three ASes ranked in the top three for January 1998 (ASes 701, 1239, and 3561) are no longer in the top three. In absolute terms, the customer cone of 701 decreased in size between January 2002 and October 2012. The customer cone of 3356 reflects two other interesting events: (1) in early 2003, AS1 (Genuity/BBN) merged with 3356 to create the third largest network at the time, and (2) in late 2010, 3549 (the second largest AS by customer cone) was purchased by Level3 (the largest AS by customer cone). 3549's customer cone has since shrunk as new customers connect to 3356 and some of 3549's customers moved across.

Figure 10 plots the customer cone sizes for the same seven ASes, but as a fraction of the topology size. We see: (1) ASes 701, 1239, and 3561 all had the same customer cone size in January 1998, (2) some customers of 3561 (MCI) shifted into 701 (Worldcom) due to the MCI-Worldcom merger in 1998, (3) 1239 held a third of the ASes in its customer cone for ten years until 2008, and (4) while 3356 had the largest customer cone in 2012, its relative cone size, i.e., as a fraction of the entire AS topology, was slightly smaller than AS701's was in January 2000. This last fact reflects massive growth in the Internet's AS topology since 2000, in addition to the consolidation undertaken by both ASes, yielding the largest customer cones of the two respective decades.

Since most companies providing Internet transit are by now also in other lines of business and do not report financial information specific to their transit business, we cannot correlate BGP consolidation with financial performance. But we know that of the three ASes whose relative customer cone sizes have plummeted in the last decade (701, 1239, 3561), two of them (Verizon and Sprint) have moved into more profitable cellular service.

Renesys produces market intelligence data using the customer cone notion [31]. They declined to share their data or method with us to enable a comparison because it is core to one of their commercial products. Comparing our rank order in table 4 with their "Bakers Dozen" from 2012 [30] shows minor differences in rank order.

5.4 Topology Flattening

The introduction of CDNs and richer peering has resulted in a flattening of the Internet topology [20, 26] where ASes avoid sending traffic via transit providers. An intriguing question is how valid is the customer cone in a flattened Internet topology? How many paths still travel to the top of a given cone to reach destinations?

We develop a new metric to track this potential shift in routing behavior as flattening occurs. While public BGP data contains a small fraction of all peering links [6] we can study shifts in routing behavior from the paths of individual

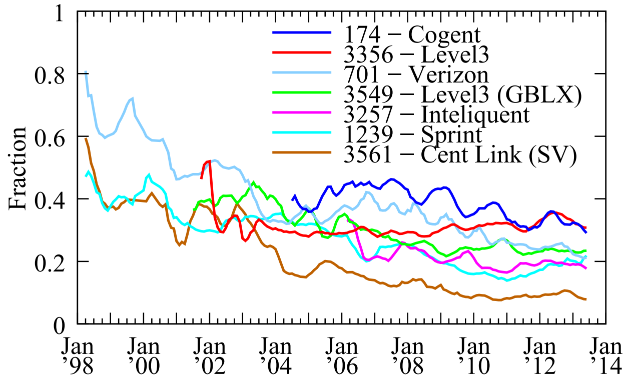


Figure 11: Fraction of ASes in X's customer cone that were reached via X from an AS in X's customer cone over time. Most ASes show a decline in the fraction of cone-internal paths.

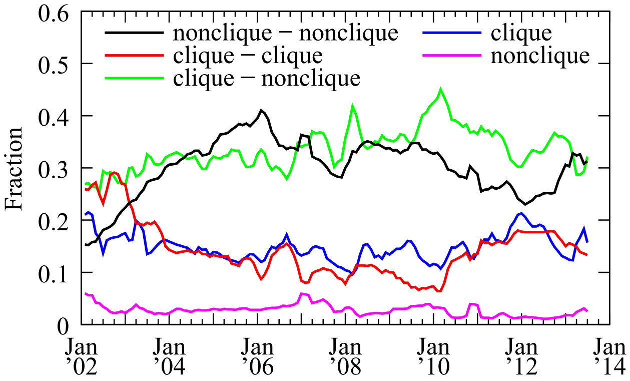


Figure 12: Characteristics of nodes or links at top of each path. The fraction of paths crossing a non-clique peering link peaked in January 2006.

VPs because they reveal the peering links they use. For each VP that provides a full view to RV or RIS and is also in X's customer cone, we compute the fraction of *cone-internal* paths, i.e., fraction of paths from that VP that transit X (the cone's top provider) when reaching another AS in X's cone which is not also in the customer cone of the VP. Figure 11 shows the five-month moving average of this fraction for the seven ASes once they have at least 1000 ASes in their cone. Five of these networks show a steady decline in the fraction of cone-internal paths.

To explore this decline in the use of these top (clique) providers to transit traffic, we evaluate the topological characteristics of top links of each path. Figure 12 plots the fraction of nodes or links in each topological category since 2002, for a set of seven VPs that consistently provided a public view.[2] Between 2002 and 2006 the fraction of paths where the top link was a peering link between two non-clique ASes rose from 15% to 40% of paths, while the fraction of paths where the top link was a peering link between two clique ASes fell from 30% to 10% of paths in 2009. Since 2006, the fraction of observed paths crossing a peering link between a clique AS and a lower-tier AS has increased, while the fraction of paths using a non-clique peering link has dropped. Considering all paths using clique ASes to reach destinations (clique-nonclique links, clique-clique links, or clique nodes at the top of paths), over 80% of paths used some clique AS back in 2002, bottoming out in 2006 (just above 60%), followed by a slow rise back to 77% today. This trend reversal is perhaps a result of Tier-1 ASes adjusting peering strategies as a reaction to ASes peering below the clique (so-called *donut peering*) to recover some transit revenue.

Our AS relationship inferences also shed light on the continually increasing richness of peering in the Internet. As the number of full VPs has increased an order of magnitude since 2000 (from 12 to 125 in October 2012), the number of p2p links observable from these VPs jumped by two orders of magnitude (from about 1K to 52K), and increased as a fraction of the entire graph from 10% (in 2000) to 38%. (even after the number of full VPs stabilized in 2008). This increase in peering (flattening) was not observed by individual VPs, most (75%) of which experienced almost no change in the *fraction* of links inferred as p2p. Instead, the increase in relative presence of p2p links in the graph is due to individual VPs seeing more unique p2p links.

6. SUMMARY AND FUTURE WORK

We have presented, and validated to an unprecedented level, a new algorithm for inferring AS relationships using publicly available BGP data. Our algorithm tolerates prevalent phenomena that previous algorithms did not handle. We validated 34.6% of our relationship inferences, finding our c2p and p2p inferences to be 99.6% and 98.7% accurate, respectively. Since even different sources of our validation data disagree by 1%, our algorithm reaches the limit of accuracy achievable with available data. We have published 97% of the validation data set, the relationship inference code, and inferred relationships publicly at `http://www.caida.org/publications/papers/2013/asrank/`

Analysis of the Internet at the AS granularity is inherently challenged by measurement and inference in a dynamic complex network. A known concern is that public views of the AS topology capture only a fraction of the p2p ecosystem, since so few ASes share their full view of the Internet with BGP data repositories. Another challenge is the variety of complex peering relationships that exist, which may be possible to infer by adapting our algorithm or incorporating additional data from either the control or data plane. Meanwhile, we have derived techniques to mitigate the effects of such relationships on application of relationship inferences.

Our inferences shed new light on the flattening Internet topology, revealing a decline in the fraction of observed paths traversing top-level (clique) ASes from 2002 (over 80%) bottoming out in 2006 (just above 60%), followed by a slow rise back to 77% today, perhaps as these clique ASes adjust their peering strategies to try to recover some transit revenue. Our customer cone metric reveals other interesting aspects of Internet evolution: the largest customer cone in the Internet has rarely had more than half of the routed ASes in it. In 2000, AS701 had a customer cone that contained half of the active ASes, and today AS3356's cone has half of the active ASes. An area of future research would be to explore the predictive capabilities of customer-cone-related metrics on Internet evolution and dynamics, such as characteristics that correlate with an impending switch from a c2p to a p2p relationship.

[2] ASes 513, 1103, 1221, 2497, 3333, 4608, and 4777.

Acknowledgments

We thank the numerous operators who contacted us with validation data and candidly discussed aspects of their network's operations. We thank Robert Beverly, David Clark, Xenofontas Dimitropoulos, Dimitri Krioukov, and Olaf Maennel for their feedback on an earlier version of this paper. We also thank the anonymous reviewers and Ethan Katz-Bassett for their feedback. The work was supported by the National Science Foundation under grants CNS-0958547 and CNS-1017064, and by the U.S. Department of Homeland Security (DHS) Science and Technology (S&T) Directorate Cyber Security Division (DHS S&T/CSD) BAA 11-02 and SPAWAR Systems Center Pacific via contract number N66001-12-C-0130. This material represents the position of the author(s) and not necessarily that of NSF or DHS.

7. REFERENCES

[1] AS Rank Website. `http://as-rank.caida.org/`.

[2] Autonomous System (AS) Numbers. `http://www.iana.org/assignments/as-numbers/as-numbers.xml`.

[3] Internet Topology Collection. `http://irl.cs.ucla.edu/topology/`.

[4] RIPE (RIS). `http://www.ripe.net/ris/`.

[5] U. Oregon Route Views Project. `http://www.routeviews.org/`.

[6] B. Ager, N. Chatzis, A. Feldmann, N. Sarrar, S. Uhlig, and W. Willinger. Anatomy of a Large European IXP. In *ACM SIGCOMM*, 2012.

[7] C. Alaettinoglu, C. Villamizar, E. Gerich, D. Kessens, D. Meyer, T. Bates, D. Karrenberg, and M. Terpstra. Routing policy specification language (RPSL), 1999.

[8] B. Augustin, B. Krishnamurthy, and W. Willinger. IXPs: Mapped? In *ACM SIGCOMM IMC*, 2009.

[9] G. D. Battista, T. Erlebach, A. Hall, M. Patrignani, M. Pizzonia, and T. Schank. Computing the Types of the Relationships Between Autonomous Systems. *IEEE/ACM Transactions on Networking*, 2007.

[10] C. Bron and J. Kerbosch. Algorithm 457: Finding All Cliques of an Undirected Graph. *CACM*, 1973.

[11] R. Chandra, P. Traina, and T. Li. BGP Communities Attribute, 1996.

[12] K. Chen, D. R. Choffnes, R. Potharaju, Y. Chen, F. E. Bustamante, D. Pei, and Y. Zhao. Where the Sidewalk Ends: Extending the Internet AS Graph Using Traceroutes from P2P Users. In *ACM CoNEXT*, 2009.

[13] A. Dhamdhere and C. Dovrolis. The Internet is Flat: Modeling the Transition from a Transit Hierarchy to a Peering Mesh. In *ACM CoNEXT*, Dec 2010.

[14] A. Dhamdhere and C. Dovrolis. Twelve Years in the Evolution of the Internet Ecosystem. *IEEE/ACM Transactions on Networking*, 2011.

[15] A. D'Ignazio and E. Giovannetti. Antitrust Analysis for the Internet Upstream Market: a Border Gateway Protocol Approach. *Journal Of Competition, Law and Economics*, 2006.

[16] X. Dimitropoulos, D. Krioukov, M. Fomenkov, B. Huffaker, Y. Hyun, and kc claffy. AS Relationships: Inference and Validation. *CCR*, 2007.

[17] P. Faratin, D. Clark, S. Bauer, W. Lehr, P. Gilmore, and A. Berger. The Growing Complexity of Internet Interconnection. *Communications and Strategies*, 2008.

[18] L. Gao. On Inferring Autonomous System Relationships in the Internet. *IEEE/ACM Transactions on Networking*, 2001.

[19] L. Gao and J. Rexford. Stable Internet Routing Without Global Coordination. *IEEE/ACM Transactions on Networking*, 2001.

[20] P. Gill, M. Arlitt, Z. Li, and A. Mahanti. The flattening Internet topology: Natural evolution, unsightly barnacles or contrived collapse? In *PAM*, 2008.

[21] P. Gill, M. Schapira, and S. Goldberg. Let the Market Drive Deployment: A Strategy for Transitioning to BGP Security. In *ACM SIGCOMM*, 2011.

[22] V. Giotsas and S. Zhou. Valley-free Violation in Internet Routing - Analysis Based on BGP Community Data. In *IEEE ICC*, 2012.

[23] E. Gregori, A. Improta, L. Lenzini, L. Rossi, and L. Sani. BGP and Inter-AS Economic Relationships. In *IFIP Networking*, 2011.

[24] G. Huston. Exploring Autonomous System Numbers. *The Internet Protocol Journal*, 2006.

[25] K. Keys and B. Huffaker. Mapping autonomous systems to organizations: CAIDA's inference methodology.

[26] C. Labovitz, S. Iekel-Johnson, D. McPherson, J. Oberheide, and F. Jahanian. Internet inter-domain traffic. In *ACM SIGCOMM*, 2010.

[27] W. Mühlbauer, S. Uhlig, B. Fu, M. Meulle, and O. Maennel. In Search for an Appropriate Granularity to Model Routing Policies. In *ACM SIGCOMM*, 2007.

[28] R. Oliveira, D. Pei, W. Willinger, B. Zhang, and L. Zhang. In search of the elusive ground truth: the Internet's AS-level connectivity structure. In *ACM SIGMETRICS*, June 2008.

[29] R. Oliveria, D. Pei, W. Willinger, B. Zhang, and L. Zhang. Quantifying the Completeness of the Observed Internet AS-level Structure. Technical Report TR-080026-2008, UCLA CS Dept., 2008.

[30] Renesys. A baker's dozen, 2012 edition. `http://www.renesys.com/blog/2013/01/a-bakers-dozen-2012-edition.shtml`.

[31] Renesys. Rankings, damned rankings, and statistics. `http://www.renesys.com/tech/presentations/pdf/menog2.pdf`.

[32] M. Roughan, W. Willinger, O. Maennel, D. Perouli, and R. Bush. 10 lessons from 10 years of measuring and modeling the Internet's autonomous systems. *JSAC*, 2011.

[33] Y. Shavitt, E. Shir, and U. Weinsberg. Near-Deterministic Inference of AS Relationships. In *ConTel*, 2009.

[34] L. Subramanian, S. Agarwal, J. Rexford, and R. H. Katz. Characterizing the Internet Hierarchy from Multiple Vantage Points. *IEEE INFOCOM*, 2002.

[35] Q. Vohra and E. Chen. BGP Support for Four-octet AS Number Space, 2007.

[36] J. Xia and L. Gao. On the evaluation of AS relationship inferences. In *IEEE Globecom*, 2004.

[37] B. Zhang, R. Liu, D. Massey, and L. Zhang. Collecting the Internet AS-level Topology. *CCR*, 2005.