

BGP and inter-AS economic relationships^{*}

E. Gregori¹, A. Improta^{2,1}, L. Lenzini², L. Rossi¹, L. Sani³

¹ Institute of Informatics and Telematics, Italian National Research Council
Pisa, Italy

`{enrico.gregori|lorenzo.rossi}@iit.cnr.it`

² Information Engineering Department, University of Pisa, Italy

`{l.lenzini|alessandro.improta}@iet.unipi.it`

³ IMT Lucca, Institute for Advanced Studies, Lucca, Italy

`luca.sani@imtlucca.it`

Abstract. The structure of the Internet is still unknown even though it provides services for most of the world. Its current configuration is the result of complex economic interactions developed in the last 20 years among important carriers and ISPs. Although with only some success, in the last few years some research has tried to shed light on the economic relationships established among ASes. The typical approaches have two phases: in the first, data from BGP monitors is gathered to infer the Internet AS-level topology graph, while in the second phase, algorithms are instantiated on this graph to derive economic tags for all edges between nodes (i.e. ASes). This paper provides significant findings for both steps. Specifically, regarding the second step, a small set of transit-free ASes and the lifespan of any AS paths are the input to an algorithm that we have devised which assigns an economic tag to each AS connection. The quality of inferred tags is expressed by a weight which takes into consideration the lifespan of the connection it refers to, as well as the outcome of the so called two-way validation approach. Regarding the first step the paper reports another valuable contribution targeted at identifying and cleaning the presence of fake connections induced by typos.

Keywords: BGP; Internet; Autonomous System; Tagging algorithm; Inter-AS Economics

1 Introduction and Related Work

The Internet is a collection of Autonomous Systems ASes⁴ (ASes) connected to each other via Border Gateway Protocol (BGP) on the basis of economic

^{*} This paper was partly supported by the MOTIA European Project (JLS/2009/CIPS/AG/C1-016). This publication reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

⁴ An AS is a connected group of one or more IP prefixes run by one or more network operators which has a *single* and *clearly* defined routing policy (RFC 1930)

contracts that regulate the traffic exchanged between them. The real structure of the Internet is still unknown, since there are neither standard methods nor dedicated tools to retrieve the needed information from the Internet itself. Researchers have tried to retrieve this topology using existing tools (e.g. *traceroute*) and by exploiting BGP information obtained from projects that deployed several monitors in significant locations (e.g. on IXPs) across the world. Typically, the Internet is studied as a graph in which nodes are ASes and edges are BGP connections between them, for example see [4,11,7,6]. A particular area of interest is the discovery of economic relationship among ASes.

The undirected graph of the Internet is not sufficient to determine the real importance of each AS since it is not possible to deduce all the possible sequences of ASes that packets can traverse. Contractual agreements could override scientific metrics (e.g. the length of the AS path [5]), thus some of the paths that can be extracted using the undirected graph might not actually be used in the real Internet even though we know each of the connections that make up that path.

Given the heterogeneity of the ASes that form the Internet (e.g. ISPs, CDNs, universities, research networks, factories), AS relationships are fundamental to determine routing policies that will select allowed paths over which inter-AS traffic can flow. The availability of a topology of the Internet that would be aware of the economic relationships among each AS has several practical implications in the real world. For example, a CDN could use this knowledge to select the best places in which to deploy replicas of its server, and a new regional ISP could select the best upstream ASes through which it could connect to the rest of the Internet [13]. AS relationships are also important to yield a better insight into the business choices that led to the creation of the current Internet structure. In the literature, economic relationships between ASes are usually classified into *customer-to-provider* (c2p), *provider-to-customer* (p2c), *peer-to-peer* (p2p) and *sibling-to-sibling* (s2s) [8,4]. In c2p and p2c, an AS (customer) pays another AS (provider) to obtain connectivity to the rest of the Internet. In p2p, a pair of ASes (peers) agree to exchange traffic between their respective customers, typically free of charge. In s2s agreement a pair of ASes (siblings) provide each other with connectivity to the rest of the Internet.

The first work on tagging the Internet AS-level topology was [4], which proposed applying a heuristic on public BGP routing information to infer economic relationships between ASes. The heuristic was based on the fact that routes that two ASes exchange must reflect the economic relationship between them, and that a provider typically has a larger node degree⁵ than its customers, while two peers typically have comparable degree. In [4] it was also proved that if *all* ASes respect the export policies imposed by such economic relationships, then the AS path in any BGP routing table must be *valley-free*, i.e. after traversing a p2c or p2p, the AS path cannot traverse a c2p or p2p. Later, [13] formulated the problem of assigning a tag to each connection as an optimization problem, the Type of Relationships (ToR) problem, using the number of *valley-free* paths

⁵ The node degree of a vertex of a graph is the number of edges incident to the vertex. In our case, degree indicates the number of BGP neighbors of an AS.

as an objective function and proposed a heuristic to solve it. The ToR problem has been proven to be NP-complete [1,3], thus several authors have proposed different heuristics and enhancements to resolve it, for example [1,3,2,9].

Other interesting approaches in the tagging issue were developed in [14] and in [11]. The algorithm proposed in [14] started from a partial set of information about the relationships between ASes, inferred using the BGP COMMUNITY attribute and from a set of information gathered through the IRR databases in order to obtain an entire set of AS relationships. However there is not a standard in using the BGP COMMUNITY attribute that could lead to a systematic method to extract useful information, and data available in IRRs have no guarantees regarding completeness and freshness. The algorithm proposed in [11] was based on the fact that BGP monitors at the top of the routing hierarchy, i.e. monitors at Tier-1 ASes, are able to reveal all the downstream p2c over time, assuming that routes follow a no-valley policy. Considering the large number of variables that could affect AS *commercial* agreements and the entropy that is present in the BGP data, we believe that the no-valley approach itself could lead to inaccurate results. In this work, we introduce another kind of approach to this problem. In order to distinguish paths that can lead to correct economic tags from those that merely introduce noise, we believe that the lifetime of an AS path is a fundamental metric that should be considered before tagging. As far as we know, the only work that has considered the lifetime of routes is [11], but the authors only set a threshold to cut off routes that lasted less than two days, potentially cutting off short-lived backup connections.

Our algorithm exploits the AS paths gathered from BGP monitors and also takes into account their lifespan during the inference of the tags, thus preserving backup links. In addition, in order to quantify the reliability of each tag we introduce the concept of *two-way validation* into the tagging algorithm, distinguishing between confirmed and unconfirmed tags.

The rest of the paper is structured as follows. In Sect. 2 we show the presence of false connections in BGP data and their causes. In Sect. 3 we analyze the nature of transient AS paths and we describe in detail the algorithm proposed. In Sect. 4 we analyze the results and we draw conclusions in Sect. 5.

2 BGP Data Gathering and Hygiene

BGP data is widely used in research into the Internet AS-level topology. However, to the best of our knowledge, there are not many works that analyze the correctness of the topological information retrieved. As we mentioned earlier, this kind of data could be affected by errors made during the manual configuration of BGP, which could lead to false AS paths and, thus, to non-existent connections within the AS-level topology. In this section we briefly describe the list of our data sources and how we managed to retrieve the data. We then analyze these data and we will propose a methodology in order to clean up any human mistakes.

The main public projects that are currently available are RIPE RIS and RouteViews. Route collectors (or monitors) deployed by these projects are devices that act like BGP AS border routers, but that do *not* send UPDATE messages, thus not announcing any prefix. Their only purpose is to establish a BGP session with BGP routers under the ownership of other ASes in order to gather routing information. To analyze the BGP table of each monitor we downloaded the snapshot of its first RIB on October 1st, 2010 and all subsequent UPDATE messages up to the end of the month. We did not only download snapshots because we might have missed all those links that are visible only between two snapshots, including some backup links. In addition, UPDATE messages allow us to trace the evolution of each single AS path and each AS connection during October 2010 in terms of its lifespan. To gather the lifespan of each AS path we rely on the BGP-4 specs (RFC 4271). A route is withdrawn from service if: a) the IP prefix that expresses the destination for a previously advertised route is advertised in the WITHDRAWN ROUTES field in the UPDATE message, b) a replacement route with the same prefix can be advertised, or c) the BGP speaker connection can be closed, which implicitly removes all routes the pair of speakers had advertised to each other from service. We consider as the lifespan of an AS path the time span in which it was considered as active by the BGP monitor, i.e. the time interval during which there is at least one active route that includes the considered AS path⁶ in its attributes.

Data gathered from BGP monitors need to be cleaned before being used. Several AS paths contain private AS Numbers (ASNs) and the AS_TRANS number 23456 (RFC 4893). In addition, some loops can be found even if the default behavior of BGP is supposed to prevent their formation. We investigated in depth the possible causes of these loops and we found that some of them are caused by typos, as already highlighted in [4] and these may introduce false connections. We found three major causes of loops in BGP data:

a) Human error during AS path prepending. When a BGP router sends an announcement it must prepend its local ASN to the AS path field. BGP allows the manipulation of the AS path length by *prepending* the owned ASN multiple times to influence the routing decision of neighboring ASes. This feature is implemented through rules to identify which announcement must be affected by the manipulation of the AS path and how many times the prepending must be done. Typically these rules are set manually by administrators, thus during this setup errors may be introduced that could in turn generate loops. Table 1 uses real AS paths in order to highlight seven different kinds of human errors.

b) Network Migration. Consider ISP A and B, as represented in Fig.1a. Suppose that A purchases B, then customers of B become customers of A (Fig.1b). The external BGP (eBGP) peering sessions with B’s customers have to be re-configured, requiring significant coordination and planning efforts. The Cisco

⁶ The RFC 4271 defines a route as “the unit of information that pairs a destination with the attribute of a path to that destination”. In this case, the attribute that we are considering is the AS path.

Table 1: Loops caused by human errors

| Error type | (Real) Example |
|--------------------------------|--|
| Lack/excess of a trailer digit | 3561 26821 27474 2747 27474 |
| Lack/excess of a header digit | 286 3549 9731 38077 8077 38077 |
| Lack/excess of a middle digit | 13030 1273 9329 929 9329 |
| Missing space between ASNs | ... 2152 3356 35819 3581935819 35819 35819... |
| Error on a digit | 13030 22212 19024 25782 25785 25782 |
| Error on two digits | 11686 4436 3491 23930 23390 23930 7306 |
| Missing digit cause split ASN | 6939 5603 21441 21 41 21441 |

Local-AS feature allows the *migrated* routers (routers formerly owned by B) to participate in AS A while impersonating AS B towards (previous) customers of AS B. Routers using the Local-AS feature retain the information that the BGP routes have passed the local AS in the AS path. They prepend *B* in inbound eBGP updates and prepend both current ASN *A* and *B* in outbound eBGP updates. In this environment, some loops can be introduced if (previous) customers of B exchange UPDATE messages with each other passing through AS A.

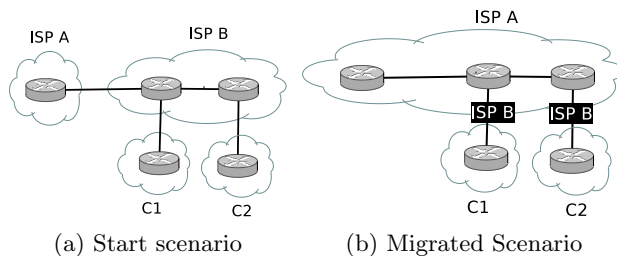


Fig. 1: Network Migration Scenario

c) Split ASes. A split AS is an AS that is divided into two (or more) islands. An AS can be split steadily or could split due to an internal temporary network failure. An example of a steady split AS is AS 48285 (Robtex.com). Consider the following AS path: (44581 48285 16150 5580 48285). AS 16150 and AS 5580 are respectively located in Sweden and the Netherlands. We contacted the Robtex.com administrator⁷ and learnt that AS 48285 is indeed *steadily* split. Thus, to obtain the connectivity between the two islands, the path needs to pass through other ASes (in this case AS 16150 and AS 5580).

Note that AS paths matching case a) can introduce *false* connections, thus it is critical to fix them so as to obtain a reliable topology. On the other hand AS paths in b) and c) will not be fixed because they reflect a real-world situation.

⁷ We would like to thank Robert Olsson, administrator of robtex.com, for the collaboration.

Thus, we clean the topology by fixing typos and by not considering connections involving private ASNs and *AS_TRANS*. The cleaned topology contains 116672 connections and 36437 ASes.

3 From BGP lies to a keen tagging algorithm

The proposed tagging algorithm exploits a list of Tier-1 ASes, denoted by T_{list} , and relies upon the following basic principle: an AS that is not included in T_{list} should be able to reach all the Internet networks, thus there must exist at least one AS path including the considered AS and a Tier-1 AS. As in [11], we use the list gathered from Wikipedia to obtain a list of Tier-1 ASes⁸. The algorithm also assumes that export policies imposed by the p2c, c2p, p2p and s2s relationships described in [4] are respected by ASes. More specifically, it assumes that an AS announces to its customers and siblings all the routes that have been received from its customers, peers and providers, while to its providers and peers it announces only the routes received from its customers.

A fundamental characteristic of this algorithm is that it computes the economic relationships by also considering the lifespan of the AS path. The presence of transient AS paths on BGP data has already been highlighted in [10], and we found that several of them, if used, can lead to misleading results in the tagging algorithm. We strongly believe that the tagging decisions made via long lasting paths must not be affected by these transient AS paths that are not used to transit IP traffic, thus our algorithm handles them. This section analyzes this specific class of transient AS paths included in the BGP data. We then introduce a tagging algorithm that also takes into account the presence of such transient paths.

3.1 Transient AS paths

Several AS paths gathered by BGP monitors include AS connections that are not used to transit any traffic. This happens due to erroneous export policies implemented in BGP, as described in [10], where two different classes of BGP misconfigurations are listed that *could* induce false AS paths: *origin* misconfiguration and *export* misconfiguration. In the former, an AS accidentally announces a prefix that it should not be announced, while in *export* misconfiguration an AS sends an announcement to a neighboring AS which violates the commercial agreement between them. These misconfigurations are much more effective when mixed with the BGP convergence times. As stated in [12], in response to path failures some BGP routers may try a number of transient paths before selecting the new best path or declaring that a destination is unreachable, performing the so-called *path exploration*. We analyzed BGP data and found a specific class of

⁸ From http://en.wikipedia.org/wiki/Tier_1_network, the Tier-1 ASes are: AS209 (Qwest), AS701 (Verizon), AS1239 (Sprint), AS1299 (TeliaSonera), AS2914 (NTT), AS3257 (TiNET), AS3356 (Level3), AS3549 (Global Crossing), AS3561 (Savvis), AS7018 (AT&T), AS6453 (Tata)

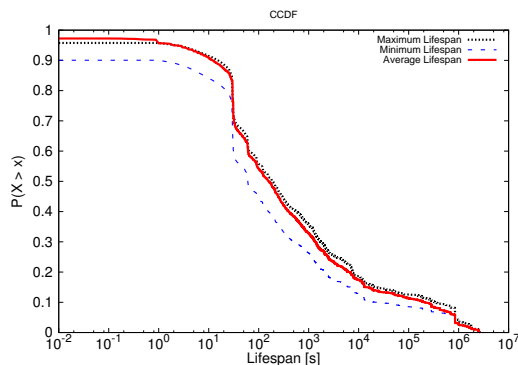


Fig. 2: CCDF of the lifespan of no-valley-free AS paths

transient AS paths that contains two or more ASes in T_{list} separated by one or more intermediate ASes, representing easily recognizable cases of *no-valley-free paths*. We investigated the set of AS paths to find this particular kind of AS path and noted that, on average, 1.3% of the AS paths on each monitor belongs to this class and that the largest number of these paths lasted less than an hour. Figure 2 shows the distribution of the minimum, maximum and average lifetime of each no-valley-free AS path as seen by the *route-views2* monitor of the RouteViews project. As can be seen, about 90% of these paths on *route-views2* lasted 10000 seconds or less. The same behavior was found on all the other monitors.

The peak of lifetimes around 30 seconds in Fig.2 can be explained by the fact that 30 seconds is equal to the suggested value for the *MinRouteAdvertisementIntervalTimer* of BGP (RFC 4271). This timer indicates the minimum amount of time that should elapse between two consecutive announcements regarding the same route. If this default value is used by the vast majority of routers, then several short-lasting no-valley-free paths will be replaced at least every 30 seconds. One plausible explanation for several of these paths is that they are a consequence of the co-effect of the convergence of BGP protocol upon a network failure and the usage of a particular type of outbound policy operated by one of the ASes involved. In more detail, BGP allows the filter of outbound announcements to be set up on a prefix basis. The filter thus prevents an announcement carrying a network prefix that belongs to one of its providers from being propagated to its peers and providers. However this filter has a drawback that can be seen after the end of a BGP connection, either due to a temporary network failure or to the end of an agreement. Consider the scenario in Fig.3. C uses the AS path [D] to reach prefix P, which belongs to its customer D. However, C has stored in its Adj-RIB-In⁹ also the AS paths [A, B, D] and [B,

⁹ Each BGP router maintains a Adj-RIB-In (Adjacent Routing Information Base, Incoming) containing routes received from neighbors.

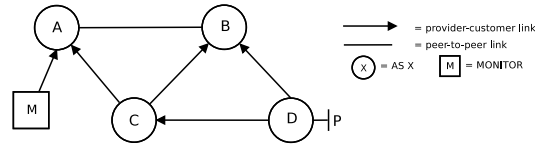


Fig. 3: Scenario

D], received from its providers A and B respectively. Now suppose that due to a network failure in P, D sends a withdrawal to B and C. C's BGP decision process will remove from its RIB the AS path [D] to reach network P and will search for another way to reach it before declaring to its neighbors that network P has been withdrawn. Since it has not yet received any withdrawal message from B concerning P, the direct consequence is that C will select¹⁰ [B, D]. If C performs an outbound filtering implemented as described above, it will then announce to A the route [B,D] to reach P, even if it is clearly in contrast with the p2c agreement signed with B. This is because network P appears in the list of networks that can be advertised to all the providers. Furthermore, since an AS typically prefers a route toward a customer over a route toward a peer or a provider, A will select the AS path [C, B, D] to reach P and it will announce it to the monitor M, thus causing the creation of a no-valley-free path. In practice, M sees that C is transiting traffic between its providers A and B for a short time (see Fig. 2) since the network P will be withdrawn from the Internet at the end of the convergence of BGP protocol. Note that this is an issue for the tagging algorithm, but not for the AS-level topology discovery tools. The connections among ASes that appear during these transients all exist, even though the traffic does not actually pass via the given AS path.

3.2 Tagging algorithm

The main piece of information that can be exploited to infer AS relationships is the set of AS paths that we can gather from the BGP data stored by monitors and their lifespan. After the data hygiene phase, performed as described in Sect. 2, our algorithm assigns an economic tag to each connection of the topology, which also indicates the level of reliability for each of them. The algorithm is organized in three main steps:

a) inference of all the possible economic relationships for each AS connection. In this step, the algorithm analyzes every single AS path separately from the others and assigns a raw economic tag to each direct¹¹ connection found in each AS path. It does this by exploiting the presence of ASes in T_{list} in order to highlight

¹⁰ For simplicity's sake, we consider the length of AS paths as the only relevant decision factor in BGP decision process

¹¹ We denote as *direct connection* a connection in which the direction *is* relevant i.e. the direct connection [A,B] is different from the direct connection [B,A].


```

1 foreach AS path
2   foreach direct connection [A,B]
3     if ( $T_1 \in T_{list}$  follows [A,B] in the AS Path)
4       Tag[A,B] = c2p
5     if ( $T_1 \in T_{list}$  precedes [A,B] in the AS Path)
6       Tag[A,B] = p2c
7     if (does not exist any  $T_1 \in T_{list}$  )
8       Tag[A,B] = p2p

```

Fig. 4: Step a) of the algorithm

```

1 foreach direct connection [A,B]
2   foreach Tag[A,B] from the longest-lasting to the shortest-lasting
3     if (exists direct_Tag[A,B])
4       if (lifespan [direct_Tag[A,B]] and lifespan [Tag[A,B]] are comparable)
5         direct_Tag[A,B] = direct_merge(direct_Tag[A,B], Tag[A,B])
6     else
7       direct_Tag[A,B] = Tag[A,B]

```

Fig. 5: Step b) of the algorithm

which AS is transiting traffic for another AS. For each direct [A,B] connection inside the path considered the algorithm proceeds as reported in Fig.4. Consider an AS path such as ($\dots A B \dots T_1 \dots$). The algorithm infers that B is a provider of A (line 2), because B is announcing to A routes retrieved from T_1 . In other words, B is *providing* A with connectivity to a portion of the Internet.

Consider now an AS path such as ($\dots T_1 \dots A B \dots$). In this case the algorithm infers that A is a provider of B (line 4), because their relationship cannot be both p2p and c2p. This is because if A and B established a p2p relationship, then A would transit traffic between one of its providers or peers (T_1)¹² and another peer (B), thus violating the export rules imposed by the p2p agreement. The same argument can be applied to show that A and B cannot have a c2p relationship. If the AS path considered does not contain any of the T_{list} ASes following or preceding [A,B], the algorithm infers that A and B have a p2p relationship because neither A nor B uses the other as a provider (line 6). Note that the aim of the algorithm is not to infer the relationships among ASes included in T_{list} , since they are assumed to be p2p. This step of the algorithm also maintains for each tagged connection the maximum lifetime of the AS paths used to infer it.

b) inference of a single economic relationship for each direct AS connection. In this step, the algorithm uses the results of step a) to infer for each *direct* AS connection [A,B] a unique economic tag among all the tags collected in step a)

¹² By definition a Tier-1 AS does not have any provider.

```

1 foreach direct [A,B] connection
2   get direct_Tag[A,B]
3   if ( exists direct_Tag[B,A])
4     if ( lifespan [direct_Tag[B,A]] and lifespan [direct_Tag[A,B]] are comparable)
5       confirmed_Tag[A,B] = inverse_merge(direct_Tag[A,B], direct_Tag[B,A])
6     else
7       unconfirmed_Tag[A,B] = direct_Tag[A,B]
8   else
9     unconfirmed_Tag[A,B] = direct_Tag[A,B]

```

Fig. 6: Step c) Final tagging and two-way validation

for the same direct connection. The algorithm first orders the tags inferred for the direct connection $[A,B]$ in descending order of lifespan. It then analyzes each tag from the longest-lasting to the shortest-lasting, as illustrated in Fig.5 (line 2). The lifespan of each tag plays a major role, since the algorithm allows the current examined tag ($\text{Tag}[A,B]$) to affect the *current* resulting tag for the same direct connection ($\text{direct_Tag}[A,B]$) iff its lifespan does not differ by more than N_{MAG} order of magnitude from the longest-lasting tag found for the same direct connection, i.e. iff the two lifespans are *comparable* (line 4). Note that the algorithm examines each tag in descending order of lifespan, thus $\text{direct_Tag}[A,B]$ contains the longest-lasting tag for the direct connection $[A,B]$. With this solution, the algorithm simply ignores the transient paths to infer the economic relationship for those connections that are found both in transient and stable paths, while it still analyzes connections that are found only as short-living since it assumes that they are backup connections. The merging rules used in this step (*direct_merge* at line 5) are reported in Table 2a and can be justified by the export policies described at the beginning of this section. If A and B have a p2c relationship, i.e. B and A have a c2p relationship, this means that A can reach only B's customers while B can reach A's customers, A's peers and A's providers. On the other hand, if A and B have a s2s relationship, A can reach B's providers B's peers and B's customers, while if A and B have a p2p relationship, they only reach their respective customers. Thus, the merge of a s2s tag with either p2c or p2p tags results in a s2s tag and the merge of a p2c (c2p) tag with a p2p tag results in p2c (c2p) tag.

c) Final tagging and two-way validation. In the final step, the algorithm uses the results of step b) to infer the economic relationship for each indirect connection¹³, as illustrated in Fig.6. For each *direct* $[A,B]$ connection the algorithm first gets its resulting tag ($\text{direct_Tag}[A,B]$, line 2) and checks if there is a resulting tag for the direct connection $[B,A]$ ($\text{direct_Tag}[B,A]$, line 3). If both exist then it checks whether their lifespans are comparable (line 4) and if necessary merges

¹³ We denote as *indirect connection* a connection in which the direction *is not* relevant, i.e. the indirect connection $[A,B]$ is equal to the indirect connection $[B,A]$.

Table 2: Rules to merge tags (resulting tags are referred to the connection (A,B))

| (a) Direct Merge | | | | | (b) Inverse Merge | | | | |
|------------------|------------|------------|------------|------------|-------------------|------------|------------|------------|------------|
| [A, B] | [A, B] | | | | [A, B] | [B, A] | | | |
| | p2c | p2p | c2p | s2s | | p2c | p2p | c2p | s2s |
| p2c | p2c | p2c | s2s | s2s | p2c | s2s | p2c | p2c | s2s |
| c2p | s2s | c2p | c2p | s2s | c2p | c2p | c2p | s2s | s2s |
| p2p | p2c | p2p | c2p | s2s | p2p | c2p | p2p | p2c | s2s |
| s2s | s2s | s2s | s2s | s2s | s2s | s2s | s2s | s2s | s2s |

the two tags (*inverse_merge*, line 5). The merging rules, listed in Table 2b, are very similar to those introduced in step b), but they also take into account the fact that a p2c for the [A,B] connection is a c2p for the [B,A] connection. In this step it is also possible to find which tag has a *two-way validation*, i.e. which tag is inferred from both the tags of the direct connections [A,B] and [B,A]. Then, a label is assigned for each tag which indicates if it is two-way validated or not, distinguishing between *confirmed* (line 5) and *unconfirmed* tags (line 7 and 9).

4 Results

This section presents the results of the tagging algorithm obtained using $N_{MAG} = 1, 2, \infty$ ¹⁴. The results are shown in Table 3. Note that the number of *confirmed* tags is ranging from 4.5% to 6.4% for raising N_{MAG} . However, if the value of N_{MAG} is increased there is more probability that transient paths will affect the final tag decision, thus lowering the reliability of the algorithm itself. This is confirmed by Fig.7 that shows the CCDF of the minimum lifespan of AS paths that participate to the tag of each connection. As can be seen from this figure, if the lifespan is not considered then a large portion of transient AS paths are contributing to the tag of each connection, possibly leading to a wrong result.

The results also highlight the presence of a large percentage of unconfirmed p2p relationships, which decrease for increasing values of N. This is caused mainly by lack of information, but also by the fact that the reverse connection that could confirm the tag is short lasting, and thus not considered by the algorithm. A particular scenario created due to lack of information is depicted in Fig.8. The figure shows that, due to the BGP decision processes, the monitors cannot gather all the possible AS paths. Thus the results of our algorithm can be misleading.

Consider S to be the source AS, T an AS included in T_{list} and M the monitor. Consider also that A, B, C and T are the transit providers of S, that in this case acts as a stub AS, i.e. it does not transit any IP traffic for any other AS, but only for end users. Supposing that T decides that the best path to S is via A, the AS paths that are gathered by M in the steady state of this scenario will be

¹⁴ $N_{MAG} = \infty$ represents the case in which the lifespan is not considered i.e. every AS path contribute to the tagging algorithm outcome.

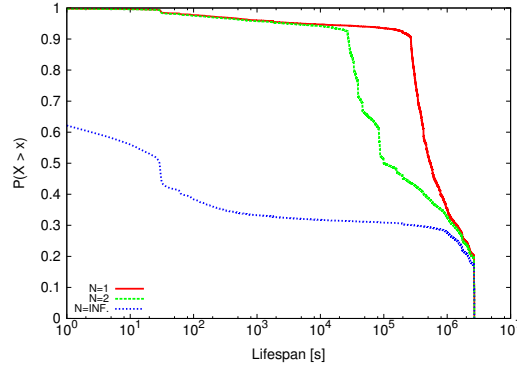


Fig. 7: CCDF of min lifespan of AS paths used

(T, A), (T, B), (T, C), (T, A, S), (B, S) and (C, S). Applying our algorithm to this scenario leads to a single unconfirmed p2c relationship for [A, S], while the couples [B, S] and [C, S] will be interpreted as unconfirmed p2p. This could be solved by connecting the BGP collector to the stub AS. There would thus be a higher probability to gather AS paths involving T, i.e. (S, A, T), (S, B, T) and (S, C, T), which would then transform the relationships [S, B] and [S, C] into an unconfirmed p2c.

The same scenario could be applied to small and medium ISPs, since it is the effect of BGP decision processes, and this phenomenon also introduces a large number of unconfirmed p2p which, in the real world, are p2c relationships. Some of these connections can be spotted by analyzing the stub ASes. A typical stub AS has several p2c relationships to transit ASes due to multihoming and it does not develop any p2p relationships. Particular real world examples of this class of ASes are ASes owned by banks, car manufacturers, universities and local ISPs. To find these kinds of ASes, we analyzed all the AS paths available and we considered as stub ASes all the ASes that only compare as last hop in the AS paths, i.e. those ASes that do not transit any IP traffic for other ASes. Due to their nature these ASes are likely to be customers in the relationships that involve them, thus there is a higher probability that unconfirmed p2p relationships involving them could be turned into p2c relationships. As reported in Table 3, the large number of unconfirmed p2p relationships involving stub ASes supports this rationale. The probability of being a real p2c relationship is very high for unconfirmed p2c relationships involving stubs. Note that if the connections involving stubs and unconfirmed p2c are considered as confirmed p2c, the percentage of confirmed tags would raise to near 60% for all the N_{MAG} considered.

Tags inferred by the algorithm depends on the list of Tier-1 considered, thus we investigated its correctness. Wikipedia is a web-based encyclopedia that any-

Table 3: Tag results

| | Tag type | Total | Unconfirmed | |
|------------------|----------|-------|---------------|---------------------|
| | | | Total (%) | Involving Stubs (%) |
| $N_{MAG}=1$ | p2c | 71841 | 69707 (97.0%) | 53089 (73.8%) |
| | p2p | 43398 | 41507 (95.6%) | 12990 (29.9%) |
| | s2s | 1378 | - | - |
| $N_{MAG}=2$ | p2c | 72394 | 70034 (96.7%) | 53331 (73.7%) |
| | p2p | 42556 | 40716 (95.7%) | 12748 (30.0%) |
| | s2s | 1667 | - | - |
| $N_{MAG}=\infty$ | p2c | 74949 | 71642 (95.9%) | 54180 (14.8%) |
| | p2p | 39078 | 37235 (95.3%) | 11898 (30.5%) |
| | s2s | 2590 | - | - |

one can edit, so the list can be the result of multiple manipulations. On the other hand, the web page containing the list of Tier-1 ASes concerns a very specific and technical topic and it is hard to believe that a common user without any particular skill in this subject could build up a detailed list like that. The presence of a low number of AS paths that include three consecutive ASes of the T_{list} would seem to indicate that it is incorrect, because these patterns would only be able to be seen if the Tier-1 ASes were not all interconnected via settlement-free peering relationships, since in this case one of them is transiting traffic for others. For example, route-views2 spot 239 different AS paths showing this pattern. Since, on average, on each route collector 20% of the AS paths that match this pattern last more than one hour, they are very unlikely to represent only simple transient paths created through router misconfigurations. In our future work we plan to study in detail the ASes that make up the core of the Internet.

5 Conclusion

We have exploited BGP data to discover what economic relationships are established between couples of ASes using an algorithm that works directly on BGP raw data. We found that BGP data can contain several anomalous paths that are clearly in contrast with the valley-free rule introduced in [4]. We traced their dynamics over a month and discovered that most of them lasted very few seconds and that they are the result of the combination of a particular common BGP

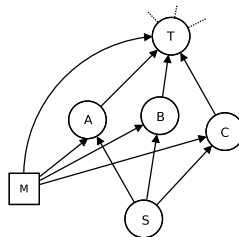


Fig. 8: BGP monitor placement pitfalls

misconfiguration and the BGP convergence delay. Our algorithm is keen of these events and is able to handle them using the lifespan of each AS Path. To infer the relationship, the algorithm relies on a priori knowledge of a list of Tier-1 ASes in order to understand whether an AS is transiting traffic for another AS. Using the concept of *two-way validation*, this algorithm is also able to assign a level of reliability to each inferred tagged connection from the Internet topology. It does this in order to compensate for the drawbacks introduced by the incompleteness of the BGP data in our possession. Our results indicate that current BGP datasets can select the two-way validated economic relationship only in 4.5 - 6.4% of the connections depending on the N_{MAG} considered. Further work will study methodologies to obtain a much higher coverage.

References

1. Battista, G.D., Patrignani, M., Pizzonia, M.: Computing the types of the relationships between autonomous systems. IEEE INFOCOM (2003)
2. Dimitropoulos, X., Krioukov, D., Huffaker, B., kc claffy, Riley, G.: Inferring AS relationships: dead end or lively beginning? 4th Workshop on Efficient and Experimental Algorithms (WEA) (2005)
3. Erlebach, T., Hall, A., Schank, T.: Classifying customer-provider relationships in the internet. TIK-Report (145) (Jul 2002)
4. Gao, L.: On inferring autonomous system relationships in the internet. IEEE/ACM TRANSACTIONS ON NETWORKING 9(6), 733–745 (Dec 2001)
5. Gao, L., Wang, F.: The extent of as path inflation by routing policies (2002)
6. Gregori, E., Improta, A., Lenzini, L., Orsini, C.: The impact of ixps on the as-level topology structure of the internet. Computer Communications (2010)
7. He, Y., Sigano, G., Faloutsos, M., Krishnamurthy, S.: Lord of the links: A framework for discovering missing links in the internet topology. IEEE/ACM TRANSACTIONS ON NETWORKING 17(2) (Apr 2009)
8. Huston, G.: Interconnection, peering, and settlements. INET'99 Abstracts Book (1999)
9. Kosub, S., Maaß, M.G., Taubig, H.: Acyclic type-of-relationship problems on the internet. CAAN06, LNCS #4235, pp. 98–111. Springer (2006)
10. Mahajan, R., Wetherall, D., Anderson, T.: Understanding bgp misconfiguration. In: In Proc. ACM SIGCOMM. pp. 3–16 (2002)
11. Oliveira, R., Pei, D., Willinger, W., Zhang, B., Zhang, L.: Quantifying the completeness of the observed internet as-level structure. UCLA Technical Report, TR 080026 (Sep 2008)
12. Oliveira, R., Zhang, B., Izhak-ratzin, R.: Quantifying path exploration in the internet (2006). Internet Measurement Conference (IMC) (2006)
13. Subramanian, L., Agarwal, S., Rexford, J., Katz, R.H.: Characterizing the internet hierarchy from multiple vantage points (2002)
14. Xia, J., Gao, L.: On the evaluation of as relationship inferences. IEEE GLOBE-COM (2004)