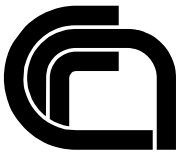


Consiglio Nazionale delle Ricerche

BGP and inter-AS economic relationships

E. Gregori, A. Improta, L. Lenzini, L. Rossi, L. Sani

IIT TR-28/2010

Technical report

Dicembre 2010

Istituto di Informatica e Telematica

BGP and inter-AS economic relationships

E. Gregori[1], A. Improta[2,1], L. Lenzini[2], L. Rossi[1], L. Sani[3]

[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] Student, Faculty of Engineering, University of Pisa, Italy
l.sani@studenti.unipi.it

Abstract. The structure of the Internet is still unknown even if it provides well-known services for a large part of the worldwide population. Its current configuration is the result of complex economic interaction developed in the last 20 years among important carriers and ISPs (i.e. ASes). Although with slight success, in the last few years some research work tried to shed light on the economic relationships established among ASes. Typical approaches employed in the above work proceed along two lines: first, data from BGP monitors spread out all over the world is gathered to infer an Internet AS-level topology graph, and second heuristics taking as input this graph are applied to get economic tags associated to all edges between nodes (i.e. ASes). In this paper we propose an innovative tagging approach leveraging on the lifetime of an AS path to infer the economic relationships on all edges joining the ASes crossed by the path itself, without cutting-off backup links, that bring economic information as well as stable links. The major findings of our approach can be summarized as follows:

- data hygiene before infer the Internet AS-level topology graph
  - study on AS paths loops, human error and their impact on data correctness
- life-time based tagging
  - we do not cut-off bakcup links
  - we evidence those tags are inferred only from a partial viewpoint
  - we evidence the maximum lifetime of the AS path that have contributed to infer the tag of each connection
- classification of candidate Tier-1 AS based on three indexes reflecting the importance of an AS
- explanation and life-time study of non valley-free AS paths

1 Introduction

Internet is a collection of Autonomous Systems (ASes[4]) connected to each other via BGP on the basis of economic contracts that regulate the traffic exchanged between them.

[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 (http://tools.ietf.org/html/rfc1930).

The real structure of the Internet is still unknown, since there are neither standard methods nor available dedicated tools to retrieve the needed information from Internet itself. Researchers during the past years tried to retrieve this topology using existing tools (e.g. *traceroute*) and exploiting BGP information obtained from projects that deployed several monitors in significant locations (e.g. on IXPs) across the world. Typically Internet is studied as a graph in which nodes are ASes and edges are BGP connections among them. To list just a few research works on this topic, see [6], [17], [9], [8]. A peculiar area of interest in this research field is represented by the discovery of economic relationship among ASes. AS relationships are fundamental to determine routing policies that select allowed paths over which inter-AS traffic can flow, since ASes are organizations with different business objective (e.g. ISPs, CDNs, universities, research networks, factories). Moreover, these relationships are important to yield a better insight into the business choices that lead to the creation of the actual Internet structure.

In literature, economic relationship between ASes are usually classified into *customer-provider*, *peer-to-peer* and *sibling-to-sibling* [11, 6]. In the customer-provider agreement, an AS (customer) pays another AS (provider) to obtain connectivity to the rest of the Internet. In the peer-to-peer agreement, a pair of ASes (peers) agree to exchange traffic between their respective customers, typically free of charge. In the sibling-to-sibling agreement a pair of ASes (siblings) provide each other connectivity to the rest of the Internet.

Currently, there are 3 mainly data sources to gather information related to the connectivity between ASes: 1) BGP route collectors; 2) Traceroute measurement infrastructures; 3) Regional Internet Registries (RIRs) databases.

The most reliable source of data are the BGP tables gathered via route collectors managed by projects like RIPE RIS[5] and RouteViews [6] . A route collector is a software router that only collects default free BGP routing information from peering AS border routers. Acting as any other BGP router, those collectors mantain their BGP tables using BGP announcements they receive from neighbors. Notice that data gathered only shows the *best* routes announced by each AS according to its BGP decision process. The only errors that could affect those data arise from typo errors in BGP configurations [6] (i.e. wrong ASN, wrong IP address space) or BGP policy routing error made by ASes administrator during the setup of their BGP sessions. Data provided by these collectors consists of periodic snapshots of their BGP tables and of the BGP announcements they received during time.

Traceroute measurement infrastructures like CAIDA Archipelago[7], DIMES[8] and iPlane [9] make publicly available *traceroute* data they collect from their monitor during time. Traceroute is a network tool that print the sequence of IP

[5] http://www.routeviews.org/
[6] http://www.ripe.net/ris/
[7] http://www.caida.org/projects/ark/
[8] http://www.netdimes.org/new/
[9] http://iplane.cs.washington.edu/

addresses of routers interface that a probe message has traversed in its journey from a source to a destination, exploiting the IP TTL field to elicit an ICMP TIME_EXCEEDED response from each interface along the path. A subsequent mapping from IP address to ASN will reveal a sequence of ASes that can be further used to infer ASes adjacences. However, we do not use traceroute data to obtain AS level connectivity because the current mapping from IP to AS is not completely accurated, (see [3, 15, 12]). Moreover traceroute could produce false IP adjacences[1], and thus false AS connections, and could miss some of them. The major cause of these problems is the potential presence of load balancers along the path traversed by the traceroute probes [21]. This problem has been addressed with Paris traceroute [1], but it has not completely been solved.

RIRs are delegated by the IANA [10] to manage and assign Internet number resources, such as AS numbers (ASNs) and IP addresses. There are 5 RIRs, in charge of 5 different geographic regions[11] AS administrators are required to fill the appropriate RIR database with information related to the ASN and the IP networks inside them. RIRs databases could also contain information about BGP policy routing of an AS, and thus to their adjacences, but it is not mandatory. Moreover there is not any tool to verify if those information are up to-date and consistent with the effective policy declared by ASes. Due to above reasons most of the works in this research field do not rely on information presents in RIRs databases to infer ASes adjacences.

Besides being the most reliable source of information, BGP data allow us to analyze detailed characteristics of each AS path and connection, such as their lifetime. This information will be used to show that a large set of connections is not effectively used to transit traffic and that only for only a small part of the Internet connections can be well-defined an economic tag.

This paper is organized as follows. Firstly, we introduce the reader to the state of the art on tagging algorithms in Sect. 2 and to the main characteristics of BGP in Sect. 3. Then, in Sect. 4 we describe in detail the algorithm proposed and the results obtained. Finally, in Sect. 5 we summarize our conclusions.

2 Related Work

The first work concerning the tagging of the Internet AS-level topology was done in 2000 by Gao [6]. In that work it was proposed to apply an 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 should reflect the economic relationship between them, and that a provider typically has a degree[12] higher than its customers, while two peers typically have a comparable degree size. Based on these concepts, in this work were firstly introduced

[10] http://www.iana.org/)

[11] AfriNIC for Africa, APNIC for Asia and Oceania, ARIN for North America, LACNIC for South and Central America and RIPE for Europe.

[12] The 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.

the classes of economic relationships that are classically used to tag each connection: *provider-customer*, *peer-to-peer* and *sibling-to-sibling*. Moreover, it was also proved that if *all* ASes respect the export policies imposed by the above type of relationships, then the AS path in any BGP routing table must be *valley-free*, i.e. after traversing a provider-customer or peer-to-peer edge, the AS path cannot traverse a customer-provider or peer-to-peer edge.

Later, Subramanian et al. [20] formulated the problem to assign a tag to each connection as an optimization problem, the Type of Relationships (ToR) problem, using the number of *valley-free* paths as objective function: given an untagged graph derived from a set P of AS paths, in which nodes are ASes and edges are connections between them, find a tag-assignment that maximize the number of *valley-free* paths in P using only provider-customer or peer-to-peer tags. They conjectured that such problem is NP-complete and proposed an heuristic to resolve it, using a set of AS paths derived from 10 BGP routing tables available from 10 telnet looking glass servers[13]. They also pointed out that their inference technique does not depend on node degree and can tolerate occasional exceptions to export rules defined in [6].

Based on this work, Di Battista et al. [2] and Erlebach et al. [5] proved that the arisen problem is in fact $NP - complete$, and proposed, indipendently, similar approaches to resolve it, exploting the mapping of the ToR problem into the 2SAT problem[14], but they considered only customer-provider and provider-customer tags. They also discussed that peering relationships cannot be correctly inferred in the ToR problem formulation, as quantified in a later work by Xia et al. [22], which reported that only the 24.63% of the peering connections found in [20] was correct.

Later on, Rimondini et al. [18] compared the results in [20] and [2]. On one hand they observed that the latter outperform the former in terms of number of *valley-free* paths into the resulting tagged graph, on the other hand pointed out that relationships inferred by the latter are further from reality than the former (e.g. well known large ISPs appear as customer of smaller ASes). Also Dimitropolous et al. [4] showed that improved solution to the ToR problem formulated as proposed in [20] will not produce any realistic results. They handled this issues including AS degree into the formulation of the problem.

Another further step was done by Kosub et al. in [13]. In their work, they stated that the resulting tagged graph should not contains cycles (e.g. AS A is customer of AS B that is customer of AS C that is customer of AS A), and showed that adding this constraint to the valley-freeness of AS paths is a feasible task. This theoretical feasibility has been supported by empirical evidence in [10]

[13] A looking glass server is a computer that allow a remote user to view routing information about routers belonging to the organization that own the server.

[14] 2SAT is the problem of determining whether a collection of boolean variables with constraints on pairs of variables can be assigned values satisfying all the constraints. In this context the two possible values are provider-customer or customer-provider links and the constraints regards the valley-freeness of tagged aspaths

by Hummel and Kosub, that defined the Acyclic ToR problem (AToR) proving that it is $APX - complete$[15] and proposing an heuristic to resolve it.

Other interesting approaches in the tagging issue were also developed by Xia and Gao in [22] and by Oliveira et al. in [17]. The algorithm proposed in [22], called PTE (Partialness to Entireness), started from a partial set of information about the relationships between ASes, inferred using BGP COMMUNITY attribute (carried into UPDATES messages) and from a set of information gathered through the IRR databases, in order to obtain an entire set of AS relationships. They filter out from the set of AS paths all those paths that were not *valley-free* using the partial set, and tags the remaining paths using inferences rules that *valley-free* paths should respect. However, nowadays, there is not a standard in using BGP communities that could lead to a systematic method to extract information from them and data available into IRRs has no guarantees on reliability and freshness. The algorithm proposed in [17] is linked to the first Gao approach, and was based on the fact that BGP monitors at the top of the routing hierarchy (i.e. monitors connected with Tier 1 ASes) are able to reveal all the downstream provider-customer connectivity over time, assuming routes follow a no-valley policy. Assuming that the list of Tier1 ASes is already known, they noticed that route collectors deployed by RouteViews and RIPE-RIS covered all the set of Tier1 ASes. Their work exploited this knowledge to tag all the connections viewed by monitors placed at Tier1 ASes with provider-customer or peer-to-peer links.

In this work, we try to introduce another type of approach to this problem. We strongly believe that the underlying problem of all the cited heuristics is that they just assume that the valley-free property is valid for the largest number of available paths, at the point that [20] and its follow-ups try to maximize the number of valid valley-free paths. The valley-free property is only valid in an ideal Internet but the real AS paths are far from being ideal. AS paths gathered by BGP monitors could be false, due to wrong export policies implemented by AS administrators, or due to BGP misconfiguration. An interesting work on BGP misconfiguration was done by Mahajan et al. in [14]. In that work, were listed 2 different classes of BGP misconfigurations 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 the latter an AS sends an announcement to a neighbor AS that violates the commercial agreement between them. The effect of these misconfigurations are much more effective when mixed with the BGP convergence times. As stated in [16], in response to path failures some BGP routers may try a number of transient paths before selecting the new best path or declaring the unreachability to a destination, performing the so-called *path exploration*. In case of BGP misconfigurations, several of these paths could not respect the no-valley rule,

[15] An APX (APproXimable) problem is an NP optimization problem that admits efficient algorithms that can find an answer within some fixed percentage of the optimal answer.

introducing in the set of AS paths some entropy that could affect the results of the algorithm proposed in [6] and [17].

Considering the large amount of variables that could involve AS *commercial* agreements and the entropy that is present at the BGP level, we also think that the no-valley approach itself, as applied in [6], [22] and [17] could lead to inaccurate results. We believe that the lifetime of an AS path is a fundamental index that should be considered *before* the tagging, in order to distinguish paths that can lead to the correct economic relationship tagging from those that are only introducing noise to the algorithm. We will deepen this issue in this work, and we will propose an algorithm that exploits only the raw BGP data gathered from the monitors, independently from the valley-free property.

3 BGP Data Gathering and Hygiene

BGP data is widely used into the research papers about the Internet AS-level topology. However, to the best of our knowledge, there is not any work that analyzes 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, that could introduce false AS paths into the set of AS paths and, thus, non-existent links into the AS-level topology.

In this section we firstly summarize the characteristics of the BGP protocol with the purpose to understand *what* is exactly representing the available data. Then, we will briefly describe the list of our data sources and how we managed to retrieve the data. At the end, we will analyze these data and we will propose a methodology in order to clean it from human mistakes.

3.1 Background on BGP Protocol

Routing information exchanged by ASes fundamentally consist of network prefixes. Each BGP message contains the prefix and some attached BGP attributes. The join of a network prefix and the attached attributes is called *route*. Routes are inserted into UPDATE messages and then sent to BGP neighbors. In practice UPDATE messages are exchanged by BGP AS Border Router (BGP ASBR).

An UPDATE message can be an announcement or a withdrawal. An announcement is an UPDATE message that indicates that the sender AS is able to reach the prefix carried into the message. An attribute that *must* be attached to these kind of messages is the AS_PATH. This field contains the sequence of ASes that the sender of the announcement will traverse to reach the announced prefix, and it is dynamically updated by each ASBR traversed. Another mandatory attribute is the NEXT_HOP, that contains the IP address of the ASBR that has sent the announcement (typically the IP of the interface from which the announcement was sent). A withdrawal is an UPDATE message that indicates that the sender AS is no longer able to reach the prefix carried into the message. For withdrawals the AS_PATH has no meaning, but the NEXT_HOP attribute is mandatory. Due to efficiency reasons an UPDATE could carry more

than one prefix and could be either an announcement (for some prefixes) and a withdrawal (for some other prefixes) at the same time.

Each BGP ASBR mantains a data structure called RIB (Routing Information Base) in which stores all the routes learned from UPDATE messages received from its neighbors. The most relevant fields of this table are [PREFIX, NEXT_HOP and AS_PATH]. When a BGP ASBR receives an announcement, it is able to fill an entry of this type, because the announcement (must) contains all the required information.

Notice that for each prefix a BGP ASBR could have received announcements from several neighbors, i.e. a router could have to choose which route to use to reach a prefix. This choice is made by the *BGP decision process*, that extract the best path to reach each prefix contained into the RIB and maintain them in a routing table, called FIB (Forwarding Information Base). The decision made by the BGP decision process could be affected by the length of the AS_PATH as well as by other BGP attributes attached to each route (e.g. LOCAL_PREF and MED).

When a BGP ASBR receives an announcement of a new prefix or the local BGP process choose a different best path, then it creates an announcement and send it to its BGP neighbors. Due to economic agreements between ASes a BGP ASBR could decide to not announce particular routes to particular neighboring ASes, and when it receives from a neighbor an announcement related to a prefix while into its RIB there is already an entry with the same couple [PREFIX,NEXT_HOP], it replaces the previous entry. Thus, given a neighbor (i.e. a next hop) and given a prefix, there is only one entry into the RIB that matches these fields. Thus, when a BGP ASBR receives a withdrawal from a neighbor is able to delete from its RIB the entry that matches the [PREFIX,NEXT_HOP] fields.

Summarizing, the evolution of the RIB of each BGP ASBR is determined by the updates it receives from its neighbors during time.

3.2 BGP Data Gathering

There are 2 main public projects available at current date: RIPE RIS and RouteViews. Route collectors (or monitors) deployed by these projects are devices that act like BGP ASBRs, but that do *not* send UPDATE messages. In other words route collectors do not announce any prefix and their only purpose is to establish a BGP session with other ASes in order to gather routing information. RIPE-RIS owns AS 12654 and it provides a snapshot of the RIB of each route collector every 8 hours and a collection of all UPDATE messages received in 5 minutes intervals. RouteViews owns AS 6447 and provides a snapshot of the RIB tables every two hours and a collection of UPDATE messages done every 15 minutes[16].

Table 1 reports information about the location of each monitor and the number of the ASes that established a BGP session with the considered monitor. It is interesting to notice that 17 monitors over 20 are located on IXPs. For these

[16] In this work, we only considered the monitors that provides data in MRT format

Table 1: Monitors location

RouteViews	Location	n.of Neighbors	% over tot. IXP member
route-views2	University of Oregon, USA	33	–
route-views4	University of Oregon, USA	10	–
route-views.kixp	KIXP, Nairobi, Kenya	10	–
route-views.eqix	Equinix, Ashburn, VA	16	7%
route-views.isc	ISC (PAIX),Palo Alto CA, USA	14	8%
route-views.linx	LINX, London, GB	22	3%
route-views.wide	DIXIE (NSPIXP),Tokyo, J	6	100%
RIS	Location	n.of Neighbors	% over tot. IXP member
rrc00	RIPE NCC, Amsterdam, NL	14	–
rrc01	LINX, London, GB	75	12.5%
rrc03	AMS-IX, Amsterdam, NL	93	16%
rrc04	CIXP, Geneva, CH	13	29%
rrc05	VIX, Vienna, A	48	43%
rrc06	JPIX, Otemachi, J	4	6%
rrc10	MIX, Milan, IT	17	21%
rrc11	NYIIX, New York, USA	31	17%
rrc12	DE-CIX, Frankfurt, D	50	9%
rrc13	MSK-IX, Moscow, RUS	19	14%
rrc14	PAIX,Palo Alto, USA	17	10%
rrc15	PTTMetro-SP, Sao Paulo, BR	9	25%
rrc16	NOTA, Miami, USA	6	5%

monitors the table also reports the percentage of the total number of IXP members[17] that is connected with the monitor. To analyze the BGP table of each monitor, we downloaded the snapshot of its RIB of May 1st, 2010 and all the subsequent UPDATE messages up to the end of the month. Such data allow us to trace the evolution of the RIB of each monitor during the month of May. We did not limited to download only snapshots because we could have missed all those links that are visible only for few seconds between the snapshots. Moreover, downloading UPDATE messages allow us to trace the evolution of each single AS path and each AS connection during the month in terms of its lifetime.

As already highlighted, the most important information that we use to infer AS adjacences is the AS_PATH attribute carried into announcements.

3.3 BGP Data Hygiene

Data gathered from BGP monitors need to be cleaned before the usage. In particular, AS paths can contain private AS numbers[18] and the AS_TRANS number 23456[19].

Moreover, it is also well-known that the default behavior of BGP is to prevent the formation of loops. However [6] pointed out that this not completely true, since some loops can be found analyzing AS paths. Thus, we have deeply investigated the possible causes of these loops, since it is important to understand if

[17] The total number of members has been retrieved using PeeringDB database(http://www.peeringdb.com)

[18] http://www.iana.org/assignments/as-numbers/as-numbers.xml

[19] http://tools.ietf.org/html//rfc4893

a loop is caused by an human error, and thus could introduce false connections, or not.

Deepening the analysis of such loops, we found 3 major causes:

a) Human error during AS paths prepending
b) Network migration
c) Steadily or occasionally splitted ASes

a) Human error during AS paths prepending. When a BGP router sends an announcement to a neighbor it must prepend its local AS number to the AS path field before sending out the UPDATE message. BGP allows a router to manipulate the length of the AS path field by *prepending* its AS number multiple times. This feature is useful to influence the routing decision of neighbor ASes. Routers' IOSes allow to define 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 it is during this setup that are possibly introduced errors that could generate loops. Table 2 illustrates six differents kinds of human errors with a real AS path example. Notice that these paths bring *false* connections, thus it is critical to fix them to obtain a reliable topology.

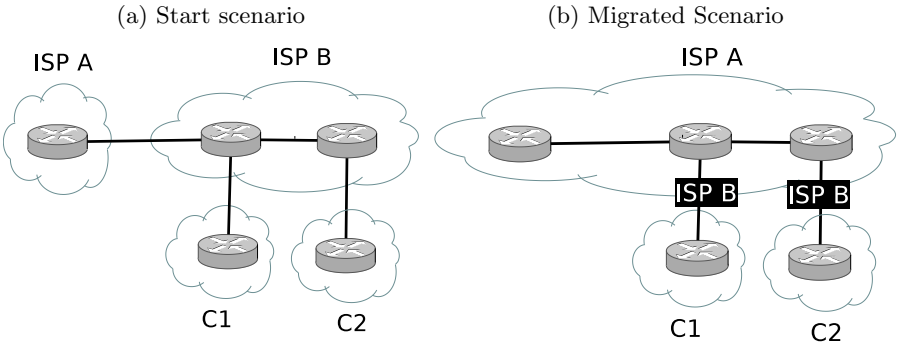
Table 2: Loops caused by human errors

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	... 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 AS Number	6939 5603 **21441 21 41 21441**

b) Network Migration. Consider 2 different ISPs: A and B. Now suppose that A purchases B, then customers of B become customers of A. Thus the external BGP (eBGP) peering sessions with the customers of B have to be reconfigured, requiring significant coordination and planning efforts. The Cisco Local-AS feature allow the *migrated* routers (former B routers) to partecipate 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 the local-AS (B) in inbound eBGP updates and prepend both actual AS number (A) and local-AS (B) in outbound eBGP updates. In this environment, some loops can be introduced if (previous) customers of B exchange UPDATE messages with the rest of the world passing

through AS A.

Fig. 1: Network Migration Scenario



c) Steadily or occasionally splitted ASes. A splitted AS is an AS that is divided in two (or more) islands. An AS can be splitted steadily or could be splitted due to an inside temporary network failure. An example of AS involved in the first case is owned by Robtex (AS 48285). Consider the following AS path: (44581 48285 16150 5580 48285). AS 16150 and AS 5580 are respectively located in the Netherlands and in Sweden. Contacting Robtex administrator we learnt that AS 48285 is *steadily* splitted. Thus, to obtain the connectivity between the two islands, it needs to pass through other ASes (in the considered case AS 1650 and AS 5580).

The cleaned topology contains 116672 connection and 36437 ASes.

4 BGP and Economics

The research of the real Internet AS-level topology and the deduction of the economic relationships developed between couples of ASes are challenging topics that researchers face since the mid 90s. These two topics are strictly related one to each other. The inferring of the AS economic relationships is fundamental, since the undirected graph of the Internet is not sufficient to determine the real importance of each AS [8], because it is not possible to deduce from the undirected graph all the feasible sequences of ASes that packets can traverse. Indeed contractual agreements could override scientific metrics (e.g. the length of the AS path, as discussed in [7]), thus some undirected paths could be not included in the set of the really used paths.

The availability of such a detailed topology has several practical implications, as already pointed out in [20]. For example, a CDN can use this knowledge to

select the best places in which deploy replicas of its server or a new regional ISP can select the best upstream ASes to which connect to the rest of the Internet.

As already said above, there has been several efforts to find a reliable algorithm to discover the economic relationships incurring among a couple of ASes. However, all these algorithms introduce a common error in assuming that the valley-free is a property that every path must respect.

In order to recover from the above drawback, we propose a new tagging algorithm that, given a set of AS paths (including the no-valley free paths which are a common source of errors in the other available algorithms), it infers the economic relationships among ASes crossed by those paths.

In this section we firstly introduce the basilar tools needed to infer AS relationships, then we show the no-valley-free AS paths and we investigate on their nature. As last step, we propose an enhanced algorithm to deal with such AS paths and show our results.

4.1 Tagging Algorithm

In this section we first describe the basic principles on which the tagging algorithm is based and second we focus on the steps through which it proceeds. The proposed tagging algorithm exploits the list of Tier-1 ASes, denoted by $Tlist$, and a set of AS paths which are in fact taken as inputs.

We consider as Tier-1 ASes all those ASes that can reach every other network on the Internet without purchasing IP transit from another AS. This means that T_{list} contains only ASes that 1) are directly connected to each other (i.e. they form a clique), 2) are able to reach all the Internet networks and 3) do not have any provider on top of them.

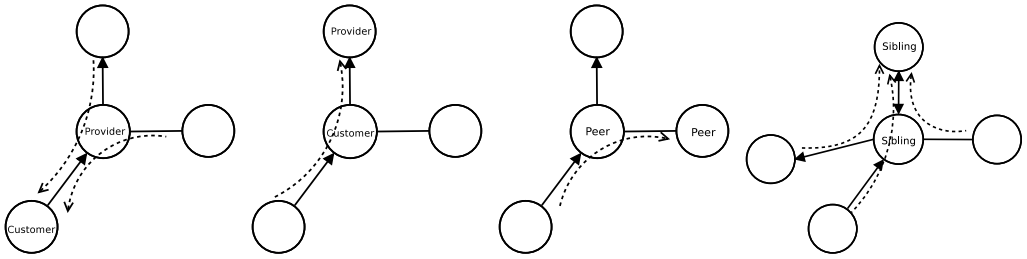
The new algorithm relies upon the following basic principle: if an AS not included in $Tlist$ can reach all the Internet networks, there will be at least one AS path which include it and at least one Tier-1.

Moreover, the algorithm assumes that export policies imposed by the provider-customer, peering-to-peering and sibling-to-sibling relationships described in [6] are respected by ASes. More specifically, an AS announces to its customers and siblings all the routes received from its customers, peers and providers, while to its providers and peers it announces only the routes received from its customers.

In figure 2 are schematically represented the export policies described above.

To find the relationship established among two neighboring ASes, A and B, it firstly collects all the AS paths that contain the pair [A, B][20] in the set P_1 and all the AS paths that contain the pair [B, A] in the set P_2. P_1 contains all those AS paths in which A is using B to reach at least a destination and P_2 contains all those AS paths in which B is using A to reach at least another destination. Notice that one of these sets could be empty, or because one AS does not use the other to reach any destination, or because the set of AS paths taken in input could be incomplete.

[20] We use square brackets to indicate a directed connection, while we use round brackets to indicate an undirected connection



(a) customer-provider (b) provider-customer (c) peer-to-peer (d) sibling-to-sibling

→ = customer-provider — = peer-to-peer ↔ = sibling-to-sibling

Fig. 2: Export policies

Then, it checks the information carried by AS paths into the set P_1, as reported in Fig.3. Suppose that one of such AS paths is like (S, ... , A, B, T_1, ...,D), where S is the source AS, D is the destination AS while T_1 belongs to T_{list}. In this case the algorithm infers that B is a provider of A, because B is announcing to A routes retrieved from T_1. In other words, B is *providing* connectivity to a portion of Internet to A, then the algorithm tags B to be a provider of A.

Otherwise, if the AS path is like
(S, ... T1, $_1$, A, B, ..., D) it infers that A is a provider of B, because their relationship cannot be neither peer-to-peer nor customer-provider. This because if A and B have a peer-to-peer relationship, this would mean that A is acting as transit AS between one of its providers or peers (T_1)[21] and another peer (B), violating the export rules imposed by the peer-to-peer agreement. The same rationale can be applied to show that A and B cannot have a customer-provider relationship.

As last case, if the AS path does not contain any of the T_{list} ASes, the algorithm infers that neither B nor A seems to use the other party as provider. This means that A and B potentially have a peer-to-peer agreement. The relationship is termed *potential peering* because the set P_2 could still contain an AS path including [B, A] and a T_1 AS

Notice that examining the set P_1 the algorithm could find different tags for the same connection. For example it could find that [A, B] is both a provider-customer and a customer-provider relationship. Thus, rules are needed to merge different tags assigned to the same connection. These rules are summarized in Table 3a and can be justified by the export policies mentioned above. If A and B have a provider-customer relationship, this means that A can reach the customers of B and B can reach the customers, the peers and the providers of A; on the other hand if A and B have a sibling-to-sibling relationship, this

[21] By definition a Tier-1 does not have any provider.

```
1  function tag_connection(AS A, AS B) {
       put all AS paths containing [A, B] into P_1
3      foreach AS path p_k in P_1 {
           if there is an AS T ∈ T_list right to B
5              if B ∉ T_list
                   Tag[A,B] = customer−provider
7              else
                   Tag[A,B] = potential−peering
9          elsif there is an AS T ∈ T_list left to A
               if A ∉ T_list
11                 Tag[A,B] = provider−customer
               else
13                 Tag[A,B] = potential−peering
           else
15             Tag[A,B] = potential−peering

17         if not exists Current_Tag[A,B]
               Current_Tag[A,B] = Tag[A,B]
19         elsif Current_Tag[A,B] is different from Tag[A,B]
               if Current_Tag[A,B] = provider−customer and Tag[A,B] = customer−provider
21                 Current_Tag[A,B] = sibling−to−sibling
               elsif Current_Tag[A,B] = potential−peering and Tag[A,B] = customer−provider
23                 Current_Tag[A,B] = customer−provider
               elsif Current_Tag[A,B] = potential−peering and Tag[A,B] = provider−customer
25                 Current_Tag[A,B] = provider−customer

27     }
   }
```

Fig. 3: Tagging of the [A,B] connection

means that A can also reach the providers and the peers of B, in addition to its customers. Thus, a sibling-to-sibling relationship *includes* the information carried by a provider-customer relationship. This rationale is valid also for the other cases. Thus, upon have assigned a tag to the connection examining each path, the algorithm checks if there is already a tag for the same connection (this is always true except the first time), and eventually update the tag according to the above rules, as reported in Fig.3 from line 17.

As further step, the algorithm checks the information carried by the AS paths introduced by the set P_2 as done for the set P_1, and infer a tag for the [B, A] connection. Another merge procedure is needed after this step, because the tags inferred for [A, B] and [B, A] connections must be turned into *one* tag for the (A, B) pair. This procedure follows the same rules listed in Table 3a, but handling the fact that a provider-customer for the [A, B] connection is a customer-provider for the [B, A] connection. Moreover, at this point we are able to identify the real peer-to-peer relationships when are found the potential-peering tag for the connection [A, B] and [B, A]. These rules are summarized into Table 3b. All those pairs (A, B) that experience only a single potential-peering tag [A, B], missing the inverted [B, A], cannot be turned into peer-to-peer relationship directly since, accordingly to Table 3b, the missing inverted tag could turn the relationship also in provider-customer, sibling-to-sibling or customer-provider. The only case left regards the relationships involving the ASes in T_list. For example consider an AS path like (S ... A, B, T_1, ..., D). If another AS in T_list, appears next to T_1,

Table 3: Merging Rules

(a) Direct Merge

[A, B]	**p2c**	**pp**	**c2p**	**s2s**
p2c	p2c	p2c	s2s	s2s
pp	p2c	pp	c2p	s2s
c2p	s2s	c2p	c2p	s2s
s2s	s2s	s2s	s2s	s2s

(b) Inverse Merge

[B, A]	**p2c**	**pp**	**p2p**	**c2p**	**s2s**
p2c	s2s	p2c	s2s	p2c	s2s
pp	c2p	p2p	s2s	p2c	s2s
p2p	p2c	p2p	p2p	c2p	s2s
c2p	c2p	c2p	c2p	s2s	s2s
s2s	s2s	s2s	s2s	s2s	s2s

Legend: p2c = provider-customer, pp = potential-peering, c2p = customer-to-provider, s2s = sibling-to-sibling

Table 4: Two list of candidate Tier-1 ASes

(a) *Wikipedia* list

ASN	**Name**
209	Qwest
701	Verizon
1239	Sprint
1299	TeliaSonera
2914	NTT
3257	TiNET
3356	Level3
3549	Global Crossing
3561	Savvis
7018	AT&T
6453	Tata

(b) *Jellyfish* list

ASN	**Name**	**Degree**
174	Cogent	3037
3356	Level3	2949
7018	AT&T	2458
701	Verizon	2122
3549	Global Crossing	1528
209	Qwest	1472
1239	Sprint	1280
2828	XO	968
6461	AboveNet	865
3257	TiNET	855
2914	NTT	725

ASN	**Name**	**Degree**
1299	TeliaSonera	625
6453	Tata	581
3320	Detusche Telekom	571
6762	Telecom Italia Sparkle	221
3561	Savvis	435
5511	France Telecom	157
1668	ATDN	67

the algorithm marks that T_1 as a provider of B. This because we assume that the relationships among two ASes included in T_list are all peer-to-peer connections and thus B cannot be a peer of T_1. Otherwise, it marks B as a potential peer of T_1.

4.2 Tier-1 List Gathering

As mentioned above, one of the inputs of the algorithm is the list of Tier-1 ASes. On the Web there are several rumors about which ASes should be considered as part of the Tier-1 set, but there is not any reliable algorithm to delineate the real list of Tier-1 ASes. As far as we know, the only work on economic relationship among ASes that uses a list of Tier-1 is [17]. This work relied on a list of 11 ASes (see Table 4a) available on wikipedia[22].

Another way to gather a list of Tier-1 ASes, called *core*, has been introduced in [19], even if that list is not used to infer any economic relationship. This work exploited the fact that the degree value of a node could be considered as an index of its importance in the graph and that the core nodes have to form a

[22] http://en.wikipedia.org/wiki/Tier_1_network

full mesh network among them to have full connectivity to the Internet. This algorithm applied to available BGP data, infers 18 candidate Tier1-ASes, which are reported in Table 4b along with their degree.

In our opinion both lists present some drawbacks. Wikipedia is a web-based encyclopedia that anyone can edit, so the list can be the result of multiple manipulations. On the other hand the web page considered is concerning a very specific and technical topic, 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 list obtained using the algorithm proposed in [19] could contain the real Tier-1 only if all the connections among nodes composing the core are settlement-free peer-to-peer connections. The problem is that some ASes in the list could have established peer-to-peer or provider-customer relationships with the other ASes and still be present in the list. This can be confirmed by the different characteristics that ASes included in the list have.

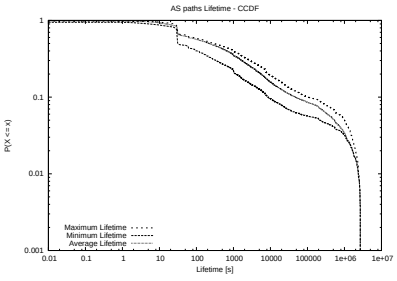
To highlight these differences, we have analyzed some metrics that should reflect the real importance of an AS on the Internet. In our opinion Tier-1 ASes should have a) an higher *degree*, b) an higher *centrality* and c) a larger *scope cone* than non-Tier-1 ASes. The role of degree to highlight the importance of an AS has already been highlighted in several works, among which [6], and to the best of our knowledge it is the only metric that has been used to this purpose in past research works. However, degree by itself cannot capture how much an AS is important *for* other ASes. This concept is better captured by the centrality and the scope cone. The centrality of an AS indicates how much this AS is exploited by other ASes to transit traffic. It is computed as the number of AS paths in which the considered AS transits traffic, i.e those AS paths in which it appears between two ASes, divided by the number of the total number of AS paths. As will be pointed out in Sect.4.3 must be taken into account the presence of short-lasting paths, that are not used to transit traffic. Thus, we have defined the *weighted* centrality of an AS as its centrality multiplied by the average lifetime of the AS paths in which the considered AS *seems* to transit traffic. The third metric is the scope cone of an AS, defined as how many different ASes it can reach. An AS with a large scope cone is an AS preferred over one with smaller scope cone, since the former is used, directly or indirectly, by an higher number of ASes than the latter. Notice that these metrics make sense because AS paths gathered by a monitor are the results of the BGP decision process of the neighbors of the monitors, thus can be considered as the *best* AS paths for the neighbor policies.

The resulting values for degree, scope cone and centrality are reported respectively in Table 4b, 5 and 6 in descending order, and they show that some ASes in the list should be considered more important than others.

Additional hints of the incorrectness of the provided lists is given by the presence of some peculiar AS path patterns that include 3 consecutive ASes of the T_{list}. We spotted x% of these patterns in the wikipedia list and y% in the jellyfish list. These patterns should be seen only if the Tier-1 ASes are not all interconnected via settlement-free peering relationships, since in this case one

Fig. 4: CCDF of AS paths containing 3 consecutive ASes of T_{list}

(a) Jellyfish



(b) Wikipedia

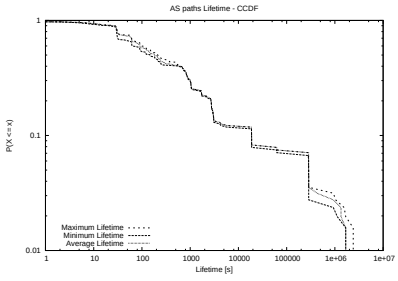


Table 5: Scope Cone of candidate Tier-1 ASes

AS	Scope Cone	ASN	Scope Cone
Level3	25809	Sprint	13726
Global Crossing	23727	AT&T	12464
TiNET	20366	Savvis	12174
NTT	19683	Verizon	11993
Cogent	19670	Detusche Telekom	10506
Telia	19607	XO	8569
AboveNet	18294	ATDN	8042
Tata	14906	Qwest	6616
Telecom Italia Sparkle	13867	France Telecom	3000

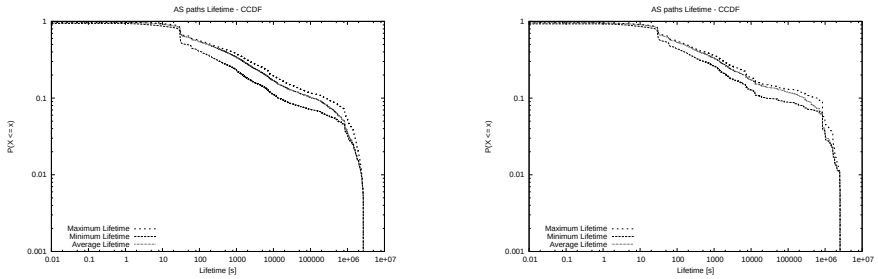
of them is transiting traffic for others. Moreover, some of these patterns are lasting too much to represent a simple transient path created through a router misconfiguration, as can be seen in Fig. 4a and 4b related to AS paths viewed by the monitor *route-views2*.

4.3 No-valley-free AS Paths and Enhanced Algorithm

As already highlighted, the ASes included in the T_{list} do not need to buy IP transit, thus none of the AS paths should contain two T_1 ASes separated by a third AS, i.e. no-valley-free paths. We have investigated the set of AS paths to find this particular pattern and spotted that an average per-monitor of 3% of them match using the Jellyfish list and Y% of them match using the Wikipedia list. Analyzing BGP data, we have found that the largest number of the no-valley-free AS paths lasted shortly during the month. For example, Fig.5a and Fig.5b show 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 Route-Views project, when considering as Tier-1 respectively the ASes into the Jellyfish list and Wikipedia list. It can be seen that in both cases about the 90% of the anomalous AS paths lasted 10000 seconds or less.

Table 6: Weighted Beetwenness of candidate Tier-1 ASes

AS	Weighted Betweenness	ASN	Weighted Betweenness
Level3	142208.11	Verizon	40476.27
Global Crossing	125335.11	Qwest	26661.98
Cogent	74169.38	Telecom Italia Sparkle	22603.99
TiNET	53415.22	AboveNet	22286.35
NTT	49889.07	XO	17608.11
Tata	49065.09	Detusche Telekom	16221.2
Telia	43438.02	Savvis	10715.61
AT&T	42088.77	France Telecom	4801.10
Sprint	41852.71	ATDN	176.13



(a) *Jellyfish* list (b) *Wikipedia* list

Fig. 5: Lifetime CCDF of anomalous AS paths

The same behavior has been found on all the other monitors considered. 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 detail, BGP allows to set up the filter of outbound announcements using the prefixes of the routes that can be advertised. This way, for example, the filter will prevent that an announcement carrying a network prefix that belongs to one of its provider is propagated to its peers and providers. However such a filter contains a drawback that could rise after the death of a BGP connection, both due to a temporary network failure or due to the end of an agreement. Consider the scenario represented in Fig.6. C uses the AS path [D] to reach the prefix P, that belong to its customer D. However, C has also stored in its Adj-RIB-In[23] also the AS paths [A, B, D] and [B, D], received from its providers A and B respectively.

Now suppose that due to a network failure in P, D sends to B and C a withdrawal announcement. C's BGP decision process will remove from its RIB the

[23] For each neighbor, BGP maintains a Adj-RIB-In (Adjacent Routing Information Base, Incoming) containing the NLRI received from the neighbors

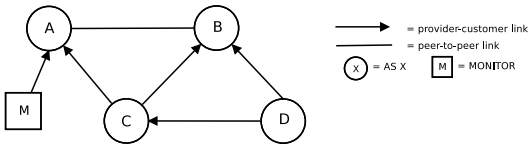


Fig. 6: Scenario

AS path [D] to reach network P and will search for another way to reach it before declaring network P as withdrawn to its neighbors. Since it has not received any withdrawal message from B concerning P yet, the direct consequence is that C will select[24] [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 provider-customer agreement signed with B. This because network P appears in the list of networks that can be advertised to all the providers. Furthermore, considering that 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, causing the birth of a no-valley-free path. In practice, M see that C is transiting traffic between its providers A and B for a 7 short time (see Fig.5) since the network P will be withdrawn from the Internet at the end of the convergence of BGP protocol.

It is interesting to notice the peak of lifetimes around 30 seconds in Fig.5, that could be explained by the fact that 30 seconds is equal to the suggested value for the *MinRouteAdvertisementIntervalTimer* of BGP[25]. This timer indicates the minimum amount of time that should elapse between two consecutive announcements regarding the 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. It can also be seen that about the 90% of the no-valley-free paths have a lifetime shorter than 100 seconds, meaning that the results of the tagging algorithm introduced in Sect.4.1 are obviously biased by the presence of these transient paths. For example, if the tagging algorithm tag as provider-customer both the pair [A, B] found inside a path lasted one month and the pair [B, A] found inside a transient path, it will be inferred an non-existent sibling-to-sibling relationship among the pair (A,B) instead of the correct provider-customer relationship. Clearly, this consideration is valid also when the tagging algorithm infer two (or more) different tags for the same pair [A, B].

It must be noticed that this is an issue for the tagging algorithm, but not for the AS-level topology. The connections among ASes that appears during these

[24] For easiness, we consider as only relevant decision factor in BGP process the length of AS path

[25] cfr. http://tools.ietf.org/html//rfc4271

```
foreach path p_k in the set of AS paths {
   Partial_Tag = tag_connection(A, B, p_k)

   if not exists Tag[A,B]
      if exists Tag[B,A]
         if (conn-life[A,B] == 0 and path-life[p_k] == 0)
         or (path-life[p_k] != 0 and (conn-life[B,A], path-life[p_k]) are comparable)
            if Tag[B,A] == pp and Partial_Tag == pp
               Tag[A,B] = p2p;
               conn-life[A,B] = conn-life[B,A]
               delete Tag[B,A]
               delete conn-life[B,A]
            elsif Tag[B,A] == p2c and Partial_Tag == p2c
               Tag[A,B] = sibling-to-sibling;
               conn-life[A,B] = conn-life[B,A]
               delete Tag[B,A]
               delete conn-life[B,A]
            elsif Partial_Tag[A,B] overrides Tag[B,A]
               Tag[A,B] = Partial_Tag;
               conn-life[A,B] = conn-life[B,A]
               delete Tag[B,A]
               delete conn-life[B,A]
      else //neither Tag[A,B] nor Tag[B,A] exists
         Tag[A,B] = Partial_Tag
         conn-life[A,B] = path-life[p_k]
   else // exists Tag[A,B] (but not Tag[B,A]
      if (conn-life[A,B] == 0 and path-life[p_k] == 0)
      or (path-life[p_k] != 0 and (conn-life[A,B], path-life[p_k]) are comparable)
         if Partial_Tag overrides Tag[B,A]
            Tag[A,B] = Partial_Tag[A,B]

}
```

Fig. 7: Tagging of the [A,B] connection

transients are all existent, even if the traffic is not effectively passing via the given AS path (false link definition in [14]).

We strongly believe that the tagging decisions made via long lasting paths should not be affected by transient paths that are not used to transit traffic. Thus, we have modified the previous described tagging algorithm to take into account such problem. The enhanced version of the algorithm is reported into Fig.7. In addition to the previous version, this algorithm takes as input the set of AS paths ordered by *descending* values of lifetime. Thus, the enhanced algorithm firstly infer the relationships for the (A,B) connection from long-lasting paths, then pass to analyze shorter-lasting paths. If a path with a shorter lifetime induces the algorithm to override a tag for the connection, then it compares its lifetime with the lifetime of the path that lead to infer the previous tag. If the two lifetimes are not comparable, then it does not allow the shorter path to influence the previous tagging decision. We consider as comparable the lifetimes of two paths if they differ no more than N orders of magnitude.

Notice that this algorithm do not cut-off backup connections since it does not impose any time-threshold. Connections that are found in transient paths as well as in stable paths are simply ignoring the transient paths to infer the economic relationship, while connections that are found only as short-living will

be assumed to be backup connections and will be tagged as any other stable connection. As can be seen in Fig.8, the vast majority of tags have been inferred from AS paths that were visible almost for a month. Table 7 reports the results of the enhanced tagging algorithm obtained using as order of magnitude N = 1, 2, 3, 4. The results highlight the presence of a large number of potential-peering connections that decrease for crescent values of N. This is caused mainly by lack of information, but it could also be caused by the fact that the reverse connection that could upgrade, for example, the potential peering in an effective peering is short lasting, and then not considered by the algorithm. Notice that if more monitors were used several of these potential peering relationships could upgrade to a peer-to-peer, provider-customer or sibling-to-sibling relationship.

Fig. 8: Lifetime CCDF of tagged connection

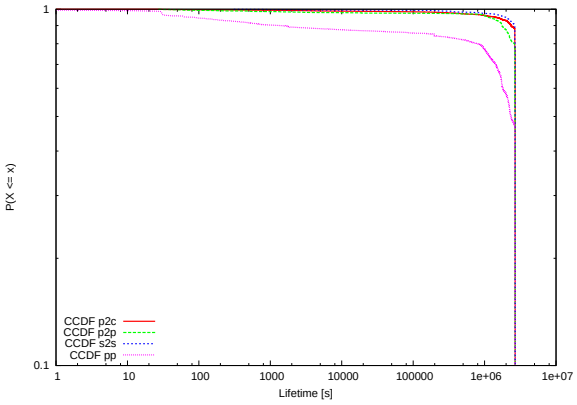


Table 7 reports the results of the enhanced tagging algorithm obtained using respectively the jellyfish and wikipedia list and using as order of magnitude N = 1, 2, 3, 4. The results highlight the presence of a large number of potential-peering connections that decrease for crescent values of N. This can be justified by the fact that not all potential peerings are caused by the completely lack of information, but also by the fact that the reverse connection that could turn, for example, a potential peering in an effective peering was short-lasting. Notice that if more monitors were used several of these potential peering relationships could upgrade to a peer-to-peer, provider-customer or sibling-to-sibling relationship.

5 Conclusions

In this paper we have exploited BGP data provided by RouteViews and RIPE-RIS projects to discover which economic relationships are established among

Table 7: Tag results

(a)

	Jellyfish	***Wikipedia***
N = 1		
p2c	72722	71568
p2p	1710	1831
s2s	1379	1280
pp	40653	41898
N = 2		
p2c	73380	72348
p2p	1684	1802
s2s	1684	1583
pp	39716	40844

(b)

	Jellyfish	***Wikipedia***
N = 3		
p2c	74024	73096
p2p	1635	1754
s2s	2009	1908
p2p	38796	39819
N = 4		
p2c	74645	73922
p2p	1651	1778
s2s	2315	2197
pp	37853	38680

couples of ASes, proposing our own algorithm that works directly on BGP raw data. In detail, the proposed algorithm relies upon the a priori knowledge of a list of Tier-1 ASes to understand if an AS is transiting traffic for another AS. Since the *exact* list of these ASes is unknown, we used two lists available on the net. We also investigated the accuracy of these lists exploiting three metrics that reflect the importance of an AS, i.e. degree, centrality and scope-cone, and we found that both lists are not completely correct, since they contain ASes that seem to have lesser importance than others. Applying the algorithm with both lists, we found that in BGP data can be spotted several anomalous paths that are clearly in contrast with the valley-free rule introduced in [**Gao**]. To analyze them we traced their dynamics during the month, uncovering that about the 90% of them lasted very few seconds and are the results of the combination of a particular common BGP misconfiguration and the BGP convergence delay. Once highlighted this issue, we proposed an enhanced version of our tagging algorithm that handle also the lifespan of each AS path that contributes to the tagging process.

In addition, during the process of building up the algorithm, we found that several AS paths contains loops, even if these kind of situations are normally prevented by BGP. We analyzed the causes of such loops and we found that nearly the 10% of them are caused by human errors introduced during AS path prepending. The total number of ASes involved in these errors is very small, but they are still introducing inaccuracy to the topology.

References

[1] B. Augustin, X. Cuvellier, B. Orgogozo, F. Viger, T. Friedman, M. Latapy, C. Magnien, and R. Teixeira. “Avoiding traceroute anomalies with Paris traceroute”. In: *IMC* (2006).

[2] G. Di Battista, M. Patrignani, and M. Pizzonia. “Computing the Types of the Relationships between Autonomous Systems”. In: *IEEE INFOCOM* (2003).

[3] H. Chang, S. Jamin, and W. Willinger. “Inferring AS-level Internet topology from router-level path traces”. In: (2001).

[4] X. Dimitropoulos, D. Krioukov, D. Krioukov, kc claffy, and G. Riley. “Inferring AS Relationships: Dead End or Lively Beginning?” In: *4th Workshop on Efficient and Experimental Algorithms (WEA)* (2005).

[5] T. Erlebach, A. Hall, and T. Schank. “Classifying Customer-Provider Relationships in the Internet”. In: *TIK-Report* 145 (July 2002).

[6] L. Gao. “On Inferring Autonomous System Relationships in the Internet”. In: *IEEE/ACM TRANSACTIONS ON NETWORKING* 9.6 (Dec. 2001), pp. 733–745.

[7] L. Gao and F. Wang. “The extent of AS path inflation by routing policies”. In: (2002).

[8] E. Gregori, A. Improta, L. Lenzini, and C. Orsini. “The impact of IXPs on the AS-level topology structure of the Internet”. In: *Computer Communications* (2010).

[9] Y. He, G. Sigano, M. Faloutsos, and S. Krishnamurthy. “Lord of the Links: A Framework for Discovering Missing Links in the Internet Topology”. In: *IEEE/ACM TRANSACTIONS ON NETWORKING* 17.2 (Apr. 2009).

[10] B. Hummel and S. Kosub. “Acyclic Type-of-Relationship Problems on the Internet: An Experimental Analysis”. In: *Proceedings of the 7th ACM SIGCOMM Internet Measurement Conference (IMC’2007), ACM Press* (2007), pages 221–226.

[11] G. Huston. “Interconnection, Peering, and Settlements”. In: *INET’99 Abstracts Book* (1999).

[12] Y. Hyun, A. Broido, and kc claffy. “Traceroute and BGP AS path incongruities”. In: *www.caida.org/outreach/papers/2003/ASP/* (2003).

[13] S. Kosub, M. G. Maaß, and H. Taubig. “Acyclic type-of-relationship problems on the Internet”. In: *CAAN06, LNCS #4235, pp. 98–111. Springer* (2006).

[14] R. Mahajan, D. Wetherall, and T. Anderson. “Understanding BGP Misconfiguration”. In: *Computer COmmunication Review* (2002).

[15] Z. M. Mao, J. Rexford, J. Wang, and R. H. Katz. “Towards an accurate AS-level traceroute tool”. In: ().

[16] R. Oliveira, B. Zhang, and R. Izhak-ratzin. “Quantifying Path Exploration in the Internet (2006)”. In: *Internet Measurement Conference (IMC)* (2006).

[17] R. Oliveira, D. Pei, W. Willinger, B. Zhang, and L. Zhang. “Quantifying the Completeness of the Observed Internet AS-level Structure”. In: *UCLA Technical Report, TR 080026* (Sept. 2008).

[18] M. Rimondini. “Statistics and comparisons about two solutions for computing the types of relationships between Autonomous Sys-

tems". In: *http://www.dia. uniroma3.it/ compunet/files/ToR-solutions-comparison.pdf* (2002).

[19] G. Siganos, S. L. Tauro, and M. Faloutsos. "Jellyfish: A Conceptual Model for the AS Internet Topology". In: *Journal of Communications and Networks* (2006).

[20] L. Subramanian, S. Agarwal, J. Rexford, and R. H. Katz. "Characterizing the Internet hierarchy from multiple vantage points". In: (2002).

[21] F. Viger, B. Augustin, X. Cuvellier, C. Magnien, M. Latapy, T. Friedman, and R. Teixeira. "Detection, understanding, and prevention of traceroute measurement artifacts". In: *Computer Networks: The International Journal of Computer and Telecommunications Networking* 52.5 (2008).

[22] J. Xia and L. Gao. "On the Evaluation of AS Relationship Inferences". In: *IEEE GLOBECOM* (2004).