Techniques for better alias resolution in Internet topology discovery

S. García-Jiménez, E. Magaña, D. Morató and M. Izal
Public University of Navarre, Campus de Arrosadía s/n, E-31006 Pamplona, Spain
e-mail: {santiago.garcia, eduardo.magana, daniel.morato, mikel.izal}@unavarra.es *

June 17, 2009

Abstract

One of the challenging problems related with network topology discovery in Internet is the process of IP address alias identification. Topology information is usually obtained from a set of traceroutes that provide IP addresses of routers in the path from a source to a destination. If these traceroutes are repeated between several source/destination pairs we can get a sampling of all IP addresses for crossed routers. In order to generate the topology graph in which each router is a node, it is needed to identify all IP addresses that belong to the same router. In this work we propose improvements over existing methods to obtain alias identification related mainly with the types and options in probing packets.

1 Introduction

In order to analyze important network parameters like delay, congestion, routing or protocols performance, we need to know the network topology. Some examples of networking fields that need this network topology, or even Internet topology, as an input are delay calculation and prediction between end-nodes, geographic localization of nodes, design and testing of routing protocols and traffic engineering, performance evaluation of P2P protocols, recovering mechanisms for path failures or evaluation of algorithms for building multicast trees. In general, any research that needs simulations over a network scenario as similar as possible to reality, will need real network topologies or synthetically generated maps.

In our case we will consider Internet mapping at the router-level. In the reverse-engineer topological features at router-level, most studies use the well-known tool called *traceroute* by Jacobson. This tool has been used by several topology measurement projects with light variants like Skitter [1], Scriptroute [2], Rocketfuel [3] or Dimes [4]. Traceroute, running in a local system, provides the IP addresses of the routers in the path from the local system to the desired destination system. It uses UDP probe packets with TTL field (Time To Live) starting from 1 and increasing one by one for each packet sent. TTL field will be decreased by one in each router in the path to destination, and when it gets 0 value the router will drop the packet and it will generate an ICMP packet with code "time to live exceeded in transit". In the local system we will receive an ICMP packet from each of the routers in the path to destination. These ICMP packets will include the IP address of each of those routers. Repeating this procedure between several systems in certain networks we can get all the IP addresses of the routers in the topology. However a router has usually several interfaces, and each of them has at least one IP address. The IP address of the ICMP packet as specified in RFC 1812 [5] "must be one of the IP addresses associated with the physical interface over which the ICMP message is transmitted" and this interface will be determined by the entry of the routing table for the local system that originated the UDP probe packet. In general, repeating the traceroute between nodes in different networks will provoke ICMP packets from different interfaces for each router. However, due to filtering, some routers do not answer to these probe packets and the IP address of that hop is not shown in traceroute (usually it is marked with an asterisk character). We have checked that this happens in around 7% of Internet hops tested in our final measurements (see section 5).

In order to provide topology information at router-level we need to identify different IP addresses belonging to the same router. These addresses are called *alias* in the literature [6]. The identification

*This work was funded by Integrated Project Evergrow (Contract No 001935) in european FP6/IST/FET Programme, Strep Moment (FP7-STREP-0215225) in european FP7 Programme and Spanish MEC project STRRONG (TEC2007-62192/TCM)

procedure is called router identification or *IP alias resolution* [6]. Alias resolution will allow to reduce the set of expansion trees, coming from the traceroutes, to a network topology graph, with the process of reducing all the nodes that represent the same router to only one node with all its interfaces.

In this paper we present mechanisms to improve the results in alias resolution. They will provide identification with less error in nodes with multiple interfaces. The rest of the paper is organized as follows. Section 2 presents the different existing techniques for alias resolution. Then new proposals in alias resolution are presented in section 3. In next sections all techniques are evaluated in a controlled testbed first and then over the european Internet. Finally, conclusions are exposed.

2 Existing techniques for alias resolution

Existing techniques for alias resolution can be grouped in two classes: active probing methods and inference methods [7]. Active probing techniques are based on sending specific probing packets to the routers and analyzing the replied packets. They are intrusive so it is important to delimit the necessary injected traffic. Besides, this traffic can be confused with scanning or attacks, so many times they can have problems with filtering in routers. Not only filtering, some router have configured rate-limiting policies to not respond to a high number of requests in a period of time.

The other side, inference methods try to deduce alias information by analyzing data from traceroute paths or by getting extra data from DNS for example. These techniques do not need to send probing packets to routers, avoiding all the problems explained before. However, inference methods have limitations in accuracy and completeness. These performance metrics will be discussed in the following section.

2.1 Performance metrics

In order to compare the different methods for alias resolution we have to take into account the following metrics [7]: accuracy, completeness and efficiency. Accuracy measures the percentage of discovered or disproven aliases that are correct. We will consider as *false positives* those aliases identified as positive but that are not really aliases. Equivalently we will consider as *false negatives* those aliases identified as negatives but that are really aliases. We will check inconsistencies between methods as a way to improve the accuracy for each method.

Completeness measures the percentage of aliases discovered with the total supposed a perfect resolution. Although this metric is very important it is difficult to define. For example, if we consider aliases by IP, 100% of IP identified in pairs would not be a completenes of 100%, because some IP addresses belonging to different interfaces of the same router would not be identified for the same router. We can identify several results in the process of alias indetification depending on the method: positive, negative, not conclusive (unknown) and error. The results of an alias identification will be *positive* if the method associates both IP addresses as belonging to the same router. The result will be *negative* if the method certifies that both IP addresses can not belong to the same router. If this negative alias is obtained for all possible pairs of a certain IP address, we can assure that there is no other IP address that belongs to the same router in the discovered topology. In same conditions, the methods will not be able to confirm one thing nor the other, so the result will be called *not conclusive*. Finally, methods based in probing requests and responses will be subjected to *error* conditions as packet filtering or specific behaviors in routers. For example, timestamp based methods will not be applicable to those routers that do not incorporate that option.

The results of each method will have to be one of the four presented: positive, negative, not conclusive and error. The percentage of all of them will determine the completeness of the method: we will prefer methods with high percentage of positive or negatives. The existence of errors or not conclusive results will indicate the limitation of each method. Depending on how these metrics are aggregated we can produce results by IP address or by pair of IP addresses. In the first case, all results for all the pairs formed by one IP address are aggregated. In the second case, each pair is considered independently. Considerations over one or the other will be presented later.

Other metrics related with completeness are number of nodes and degree. Number of nodes in the topology can give an idea of complexity reduction in the graph. At first, each IP address is represented with a node in the graph. A node is connected to other nodes that were successor or predecessor of that IP address for some traceroute. If we check that two IP addresses are from the same router, they are aggregated in the same node. Therefore, a measurement of effectiveness will be related with the reduction in the number of nodes in the topology. Degree refers to the number of edges that we found in each node of the graph, this means the number of IP addresses identified as belonging to the same router

Finally, efficiency measures the amount of probe traffic used to discover aliases. Mainly, active probing methods are characterized by introducing extra traffic in the network, and this intrusive traffic should be delimited.

Inconsistencies will have to be considered when comparing different methods for alias identification. We will have an inconsistency when, for a certain pair of IP addresses, one method gives a positive alias and another method gives a negative alias. We will have to reduce inconsistencies as much as possible.

2.2 Active probing methods

In this section existing methods based on active probing are described.

2.2.1 Based on source IP address (Mercator)

The method based on active probing is described in [6], implemented by CAIDA [8] in the tool *iffinder* and used in several works [9][10]. This method consists on sending UDP datagrams from the same host to all the IP addresses that could belong to the same router. The destination UDP port is chosen randomly and if there is no application listening on that port, the router sends an ICMP packet with type "destination port unreachable". As explained before, these ICMP messages are generated from the interface with path to destination (the probing source host). So if all probed IP addresses belongs to the same router, all the ICMP messages will have the same source IP address and we will have a positive alias resolution.

The name of "Mercator" appears in [11] where some improvements are made to this method. The probing is repeated several times to account for unstable routing tables and source routing is used to inject packets into specific network destinations.

However, nowadays it is usual to filter out those ICMP packets in the routers. In our study (section 5) we have found that 92.55% of probing packets do not receive the ICMP notification because of filtering in the target router. Therefore, the effectiveness of this method is very limited. Besides, we have detected that the ICMP notifications can be sent from different interfaces depending on the input interface of the probing packet (violating RFC 1812 [5]). This means that there are routers that reply with ICMP messages to the same probing host from different IP addresses, all of these addresses belonging to the same router. Therefore this method would not be able to identify two IP addresses that are for sure not belonging to the same router (a negative alias).

2.2.2 Based on IP identifier counter (Ally)

An alternative method based on active probing was proposed in [3]. It uses the same type of probes based on UDP packets to provoke ICMP notifications but in this case the method uses the IP identification field (IPID) in the IP header to check aliases. This IP identifier is originally used in the procedures of fragmentation and reassembly. This field has the same value for all fragments belonging to an original IP datagram before fragmentation, so it is used to reassemble the original packet in destination. Typical TCP/IP implementations of IP identifier use a counter which is incremented by one for each packet created in the host, independently of destination, protocol or service. Therefore, several IP packets received from the same host and near in time will have close values in the IP identifier field. The differences in the counter will be caused by other IP traffic generated in between by that host to other destination.

The Ally tool sends two probe packets almost back-to-back to two IP addresses (potential aliases), receiving two ICMP packets with type "destination port unreachable" and IP identifier x and y. One second later, a third probe packet is sent to the IP address that sent first the previous ICMP. Then a third ICMP with IP identifier z is received. The two IP addresses will be alias if $x < y < z$ with $|z - x| < 200$. If there were not IP traffic generated by the router in between: $x + 2 = y + 1 = z$. It must be noted that IP traffic generated by a router is related mainly with management tasks (routing protocols, SNMP, ping, traceroute, etc.) and does not take into account the forwarded packets by the router. The threshold of 200 sequence numbers in one second is chosen taking this into account.

However, there are implementations of routers that do not follow this characteristic in the increase of IP identifier. In our study we have discovered other behaviors like using a reset policy for IP identifier or even using a random value (sometimes based on timestamp and sometimes is a real random value). Again this method is affected by filtering of ICMP messages in routers. If we have to verify all possible pairs of IP addresses in a topology, it can be very costly. The search space can be reduced using the hierarchy embedded in DNS names and considering only IP addresses with nearby TTLs. Finally, the method can yield false positives when the IP identifiers of different routers happen to appear synchronized, but we will check that this probability is very low.

2.3 Inference methods

2.3.1 Based on graph analysis

The data collected by traceroutes can be used to construct a set of expansion trees (directed graphs) using the IP addresses as nodes and the pairs of IP addresses as edges. We will have an expansion tree for each source of the traceroute. The labor here is to join information for several expansion trees with different sources and destinations in order to get a final graph with routers as nodes and links between routers as edges. Several heuristics for that process have been proposed in the state of the art:

- Two addresses that immediately preceed a common successor are aliases [7]. It assumes that links are point-to-point and that ICMP "time to live exceeded in transit" of traceroute uses the input interface of the router as the source address. Both assumptions are not realistic in Internet so the heuristic may fail for a large percentage of cases.
- Different IP addresses that appear in the same traceroute trace can not be aliases [7]. It assumes that routing loops are not present. Recognizing routing loops in a trace can be not so simple.
- Analytical Alias Resolver (AAR) [12] searches for potential path symmetry between two end points, locating point-to-point subnets (/30 or /31 networks) with IP addresses in each direction. If a match is observed, aliases can be found from the proper alignment of the path traces (one in each direction).
- Analytical and Probe-based Alias Resolver (APAR) [13] is an improvement on the previous proposal. It extends the analysis to multi-access links (up to /22 networks) introducing the possibility of false positives in the alias resolution. This case requires to have collected at least half of the addresses in the multi-access network. To reduce the number of false positives, a ping query is sent from a host to each IP address in the network so candidate aliases should be at similar TTL distances (the TTL field of the IP header in the ICMP reply packet is analyzed this time).

2.3.2 Based on DNS

In [3][7] the alias resolution scheme is based on drawing inferences from systematic naming conventions in DNS names. Usually ISPs follow a convention in naming interfaces of routers. For example the names SO0-0-0.EB-Pamplona0.red.rediris.es and FEO-1-2.EB-Pamplona0.red.rediris.es refers to two interfaces in a router (EB-Pamplona0.red.rediris.es). Hierarchy embedded in DNS names and lexigraphically adjacency in name convention for router interfaces make it possible. This convention is different for each ISP so it requires a reverse-engineering process that is manual, limiting the usability of the method.

Another application of DNS information is testing aliases generated with other methods [12]. Similarities in DNS names will identify alias pairs.

3 New techniques for alias resolution

One of the problems in Ally method [3] is the filtering of ICMP packet with type "destination port unreachable". In our measurements we get only around 8% responses. However, we can modify the probing packets in order to get more replies of different kinds from the router, all of them providing the IP identifier field of the IP header. The idea is the same, polling IP addresses in pairs, checking that the replied IP identifiers follow a monotically increasing sequence.

3.1 Based on IP identifiers

We can improve the results of Ally method if we use other types of traffic. We can modify the probing traffic, the probing interval and the inter-packet time compared with Ally. In Ally two probing packets (UDP) were sent to two target addresses, one after the other ($10\mu s$ interpacket delay) and after receiving the first reply from one of the destinations, a third packet is sent after one second to the IP address that replied sooner. The threshold to consider the increasing sequence of IP identifiers is 200 between first and third packets.

In our proposal *IPID(UDP)*, we send 3 UDP packets equally spaced by 200 msec, sending first and third probing packet to one IP address and second probing packet to the other IP address. In this case the threshold will be 40 units of IPIDs (between first and second packet, and between second and third packet) to consider that the IP identification of received packets are in a monotically increasing sequence. This is the same threshold than Ally adapted to the new time scale.

ICMP "echo request" can be used to obtain a replied ICMP "echo reply" from an IP address with a copy of the IP data included in the original packet. The replied packet will have an IPID in the IP header that can be used to check the sequence of IP identifiers as before. We will call this method *IPID(ECHO)*. However, there are implementations that use the IPID of the replied packet as a copy of the ICMP identifier in the ICMP header of the request packet. Normally the ICMP identifier in the ICMP echo request is numbered sequentially starting in 0, and the replied ICMP identifier copies the same value. We have observed that the replying router sometimes copies the ICMP identifier also in the IP identifier. Our proposal can support this behavior. If both addresses reply with the IPID sequence 0,1,2,... then we can confirm nothing and the alias will be unknown. If only one address replies with the IPID sequence 0,1,2,... then the alias is false. In any other case we will have to apply the default IPID identification method looking for the increasing sequence.

Another possibility is to use ICMP "timestamp request" that will provoke an ICMP "timestamp reply" from the target IP address. Those packets exchange three timestamps, but we will keep using the IPID contained in the replied packet. Again the default IPID identification method will be used. We will call this method *IPID(TSTAMP)*.

Finally, we can use TCP packets to force replying packets from the target IP address. If we send a TCP SYN packet (connection establishment) to a destination port where no application is listening, we should receive a TCP RESET packet from the target IP address. Again, we can use the IPID in this TCP RESET packet to apply the default IPID identification method. This will be the *IPID(TCP)* method.

3.2 Based on timestamping

A new set of probing methods based on timestamping are proposed. The idea is to get some reference of the clock in the router with certain IP address. If we get timestamps from two IP addresses we can check if they follow a monotically increasing sequence in a way similar to the IPID identification method. We have to get alternative timestamps for each IP address and all of them must follow a sequence to be considered as aliases.

TSTAMP(ICMP) uses the ICMP "timestamp request" that will provoke a ICMP "timestamp reply" from the target IP address. The replied packet will contain three timestamps as follows. The first timestamp is the originate timestamp, which contains the last time the sender touched the packet. The receive timestamp is the time that the echoing host first touched the packet and the transmit timestamp is the last timestamp set just previous to sending the replied packet. We consider the transmit timestamp of replied packets in order to check the monotically increasing sequence of timestamps.

Another variation in order to get router timestamps is *TSTAMP(TCP)*. This is based in sending a TCP SYN packet (connection establishment) to a destination port where no application is listening but this time with the timestamp option activated in the TCP header. This will cause to receive a TCP RESET from the target IP address with the TCP timestamp field. This timestamp is usually the uptime of the router, this means the time a router has been up and running. Its resolution is at least one second [14]. So this timestamp can be used to differentiate between different routers because hopefully they will have different uptime. We will use this TCP timestamp field to check the monotically increasing sequence of timestamps when we poll alternatively two IP addresses that candidate to be aliases. The TCP timestamp support is optional and it could be not implemented in the router. If this option is implemented in the target router, when a connection request carries the TCP timestamp option the target router is enforced to answer with this option activated too [14].

However, these techniques have specific problems. They can not give positive aliases because the timestamp is not enough to differentiate IP addresses from different routers. Router timestamps that follow an internal clock can be NTP (Network Time Protocol) synchronized with others, very usual in network deployments. Router timestamps related with the time measured since the router was booted, can be again synchronized with routers that, for example, suffer the same power outage. So the incidences of false positives can be very high. For this reason, these methods will not produce positives in alias identification but they will be able to provide negatives. For example, if two IP addresses produce timestamps with big separation, this would indicate that both IP addresses belong to routers with different booting time and therefore they belong to different routers.

3.3 Probability of false positives with Ally

In order to decrease the incidence of false positives in identifications with Ally method we can increase the number of probing packets. In this work we propose to probe alternatively both IP addresses under test with several packets and then create the discrete series formed by IPIDs of each received packet. In case that both addresses belong to the same router and it uses incremental IPIDs then the monotically

Table 1: Percentage of routers with certain behavior in the generation of IP identifiers

Probes	*Incremental*	*Random*	*RandomT*	*Reset*	*Filtered*
UDP(Ally)	22.83	0.14	0	4.66	72.34
ICMP ECHO	18.53	0.37	0	50.13	30.95
ICMP TSTAMP	15.52	4.48	13.84	0.08	66.06
TCP	24.45	4.13	0	4.47	66.92

increasing sequence will be identified and therefore both addresses will be aliases (figure 1 (left)). Figure 1 (right) shows an example of false positive in Ally where the first three packets follow a monotically increasing sequence (positive alias for Ally). However, considering packet 5 we can observe that both IP addresses follow different sequences. This could be discovered with only two more packets than standard Ally. We will analyze how increasing the number of probing packets will reduce the probability of false positives.

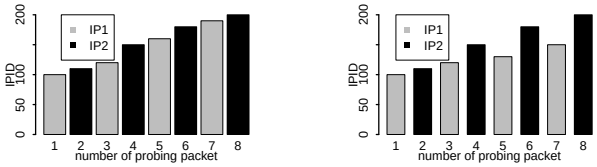

Figure 1: IP identifier sequence for positive (left) and false positive (right) aliases in Ally method

In order to get an estimation of false positives (accuracy) in Ally method, we should differentiate the following behaviors in the generation of IP identifiers:

- Incremental: the default behavior supposed in Ally, all IP packets generated by a host will increase the IP identifier in one unit.

- Random: is typical to find router implementations where the IP identifier is chosen randomly (the 16 bits field allows numbers in the range 0-65535). Some peculiarities can be found in this case, for example that the IPID can change randomly each second, and keep the same value for all packets within that second. This special behavior will be called RandomT.

- Reset: another behavior that we find in some routers is to generate IPIDs following an increasing sequence starting in 0 per router interface. Every new packet would provide an increment of IPID in one unit.

We have checked previous behaviors in the generation of IP identifiers for the real scenario (section 5), getting the percentages shown in table 1 for each kind of probing packet: UDP, ICMP ECHO, ICMP TIMESTAMP and TCP. It must be noted the high percentage of probing packets that are filtered out and that did not produce any result. The desired incremental behavior in IP identifiers is better in TCP probes. Later we will present the improvements in alias resolution with the modification of probing packets.

In order to analyze the probability of false positives in Ally method, we will need to consider the cases in which belonging the two IP addresses (IP_1,IP_2) to two different routers (R_1,R_2) they are identified as alias. In this case we receive three IP identifiers in the ICMPs x (from R_1), y (from R_2) and z (from R_1 or R_2). Here we do not consider the problems related with packet filtering and rate-limiting that will affect the characteristic of completeness. Depending on the behavior of both routers we can distinguish the following cases:

- Random-Random: IP identifier is random for both routers. In this case, we have to calculate the probability of $x < y < z$ with $|z - x| \leq 200$, keeping the Ally threshold. The first x can be any, it is not important, but if we consider x at offset 0, if $y = 1$ then $z \in \{2, 3, 4, ..., 199\}$. If $y = 2$ then $z \in \{3, 4, 5, ..., 199\}$, and so on up to $(y = 198, z = 199)$. Probability of this case would be: $P_{R,R} = \sum_{i=1}^{198} \frac{i}{65536^2} = 4.58\ 10^{-6}$

- Random-Incremental: IP identifier is random for the first router and incremental for the second. Now R_2 is incremental but as we only have one of its IP identifiers (y), its initial value is random too. For this reason, the probability is the same as Random-Random: $P_{R,I} = P_{R,R} = 4.58\ 10^{-6}$

- Incremental-Random: IP identifier is incremental for the first router and random for the second. In this case, we have to calculate the probability of $x < y < z$ with $\{x, z\}$ incremental. We need to determine the distance between x and z, and then calculate the probability of $y \in (x, z)$. This distance depends on each router and time of the day, because packet generation rate can change with both variables. Because of the diversity of this behavior, figure 2 shows the distribution of distances for IP identifiers between two consecutive packets received from all routers in our scenario. The mean value is 9 in a time interval of 0.4 seconds, so if we have 1 second between x and z as Ally, we will need to suppose a distance of 22. The random IP identifier y will need to stay in between those 22 numbers. The probability results: $P_{I,R} = \frac{22}{65535} = 2.7\ 10^{-4}$

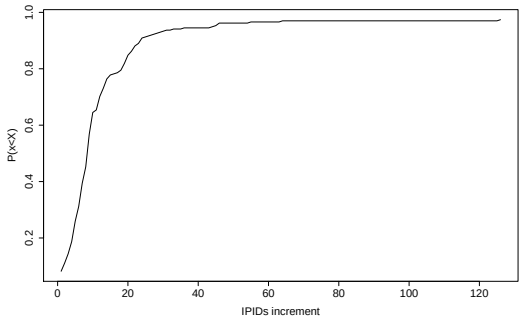

Figure 2: Distribution of distances for IP identifiers between two consecutive packets for all routers

- Incremental-Incremental: IP identifier is incremental for both routers. The probability will be the same than the previous case because for y there is only one realization and then it can be considered random. $P_{I,I} = P_{I,R} = 2.7\ 10^{-4}$
- Random-Reset: This case is the same than Random-Random. We have a static number (the 0 made by the reset policy) and we can have two random values who can make an incremental sequence with it $x < 0 < z$. If $x = (-1\ mod\ 65536)$ then $z \in \{2, 3, 4, ..., 199\}$. If $x = (-2\ mod\ 65536)$ then $z \in \{2, 3, 4, ..., 198\}$. Probability of this case would be again: $P_{R,0} = \sum_{i=1}^{198} \frac{i}{65536^2} = 4.58\ 10^{-6}$
- Incremental-Reset: The probability will be the same than in the Incremental-Random case. We have an increment of 22 IPIDs between first and last packet and 0 must be in between of both. The probability results: $P_{I,0} = \frac{22}{65535} = 2.7\ 10^{-4}$

Reset-Random and Reset-Incremental are not distinguishable from Incremental-Random and Incremental-Incremental respectively. Taking into account the probability of each of the behaviors from table 1 (UDP row for Ally), the total probability of false positives is:

$$\begin{aligned} P &= P_{R,R} * 0.000025 + P_{R,I} * 0.00413100 + \\ &\quad + P_{I,R} * 0.004131 + P_{I,I} * 0.68260644 + \\ &\quad + P_{R,0} * 0.000843 + P_{I,0} * 0.13929732 = 2.23\ 10^{-4} \end{aligned} \tag{1}$$

This probability is not too high, getting around two false positives for 10,000 iterations of the method. However, in real network topologies is usual to work with miles of IP addresses and then millions of pairs, making this probability significative.

In order to reduce the probability of false positives, one possibility is to increase the number of probing packets. Using the distribution of IP identifier distances (figure 2) and the probability of each behavior in IP identifier generation (table 1), a simulation has been made in order to study the effect of increasing the number of packets in the probability of false positives. Initial IP identifiers for each router are generated with a uniform random distribution. The results are shown in figure 3 from 3 to 60 packets. It can be observed that the bigger improvement is for packets up to 10. Although this would imply to generate more traffic, the improvements in the reduction of false positives could make it worthwhile.

4 Evaluation in a controlled testbed

One of the big difficulties in the deployment of alias identification and analysis of topologies in general is that usually it is impossible to verify the correctness of the results. It is not possible to access to specific

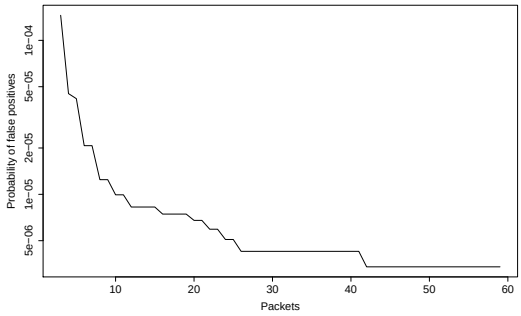

Figure 3: Probability of false positives with the number of packets

information about the network topology in the global Internet composed by thousands of Autonomous Systems. Operators are reluctant to make public information about their network resources, and then it is not possible to confirm the validity of alias identification methods. For this reason, we have first tested our proposal in a controlled testbed.

Our testbed is composed by seven Cisco routers and one PC with Linux operating system working as a router. There are different technologies for the links: serial, Ethernet, POS (Packet over SONET) over STM-1, DOCSIS and ATM. A host is used as probing station from which all alias identification methods are launched. In the testbed there are 27 IP addresses and we can check aliases between 351 pairs of IP addresses. In a real scenario the IP addresses of router interfaces are discovered using *traceroutes* between probing stations distributed along all different networks [8]. This discovering process is not relevant for our study about checking the accuracy of different alias discovering methods. Therefore, we will suppose all IP addresses of the scenario as known.

The results for each method from the literature (Mercator, Ally) and our proposals are analyzed. As explained before, depending on how these metrics are aggregated we can produce results by IP address or by pair of IP addresses. In this case the results will be aggregated by IP address, obtaining for each method the percentage of positive, negative, not conclusive and error. All of them for a certain method will sum up 100%. The positive metric presents the percentage of IP addresses that have been paired with at least another IP address. This means that at least one pair have given a positive alias. The negative metric presents the percentage of IP addresses that have given negative alias for all possible pairing. Note that in our testbed it is not possible because all routers have at least two interfaces. The not conclusive metric presents the cases when all pairs with certain IP address have not given a positive neither all negatives, without errors. Finally, error metric covers all other possibilities, whenever there are at least an alias with error. In this case, presenting the results by IP, 100% for positive does not mean a perfect alias resolution because a router can have three or more interfaces. With 100% positive we will only be able to confirm that all IP addresses have been paired with another one. This means that we could get more than 100% positive if all interfaces will be grouped with all the other interfaces for the same router.

For the testbed we have not found inconsistencies between the results from different methods. As we have the real topology information, we have checked that all aliases in the topology are discovered and that there are not false positives in this resolution.

Ally and IPID(TCP) pair all IP addresses of the topology with at least one other (100% in positive metric). Mercator only gets 83.3%, IPID(UDP) 37.5%, IPID(ECHO) 16.6% and IPID(TIME) 8.4%. In the majority of the routers in the testbed the TCP timestamp option is not available, so we obtain only errors with the TSTAMP(TCP) method. In the other timestamp based method, TSTAMP(ICMP), although there are replied packets the results are not conclusive because a good increasing sequence in timestamps can be due to clock synchronization between different routers. However, as in this controlled testbed we know that router clocks are not synchronized, this method would indeed recognize 100% of aliases.

This testbed presents a very optimistic scenario where almost all the routers come from the same manufacturer. Besides, it has very relaxed security, with no packet filtering above the default configurations. In this scenario, a traditional method as Mercator (port unreachable based) resolves 83.3% of alias and the other method Ally reaches 100% of alias identification (one alias for all IP addresses in the testbed). Therefore, in a scenario without packet filtering in nodes, methods from the state of the art can achieve complete resolution of aliases. However, in real scenarios through Internet we will see how it affects, and how these traditional methods are clearly not enough.

Table 2: Results of IP address alias identification in a real scenario (in % per pairs)

Method	*Positive*	*Negative*	*Not conclusive*	*Error*	*Nodes acumulated*	*Links acumulated*	*Total identified*
Mercator	.02	0	9.35	90.63	545	710	0.02
Ally	.03	7.35	0	92.62	520	692	7.40
IPID(UDP)	.06	7.77	0	92.17	506	685	11.79
IPID(ECHO)	.21	54.81	19.12	25.86	440	588	62.03
IPID(TCP)	.01	3.27	.31	96.41	434	580	63.08
IPID(TIME)	.06	12.52	7.91	79.51	434	580	63.17
TSTAMP(TCP)	0	0	0	100	434	580	63.17
TSTAMP(TIME)	0	0	7.22	92.78	434	580	63.17
.....	.....	.....	.....	.....	.....	.....	.....
Total	0.30	62.87	11.58	25.25	434	580	63.17

5 Evaluation over the Internet

Once the methodology has been verified in a controlled testbed where we have been able to check inconsistencies and false positives/negatives, the next step is to extend the study to a real scenario in Internet. We have used the ETOMIC monitoring platform [15] that allows to launch experiments in 18 nodes distributed around Europe. ETOMIC nodes are connected to networks in Universities, Research Centers and enterprises providing real information about connectivity to Internet in different points around Europe, as shown in 4. We have used these ETOMIC nodes to discover the topology of the european Internet that provides connectivity between them, and then be able to identify alias of different routers. We made a set of traceroutes between each pair of ETOMIC nodes in December 2007 obtaining 427 IP addresses.

Figure 4: Evaluation over the Internet using ETOMIC infrastructure

Table 2 shows the results in alias identification obtained for this european network. This time results are presented in pairs of IP addresses: 100% positive would indicate that all pairs of IP addresses are alias which is not realistic. Therefore, the percentages of negative alias are bigger considering results per pairs. Again we have not detected any inconsistency in the identification between different methods. Mercator and Ally are the classical methods present in the literature and all the other are our proposals. IPID(UDP) is our modification of Ally with the same number of 3 probing packets but changing the timing (200 msec inter-packet time). The rest of the methods were explained before. The total identified column represents the percentage of pairs of IP addresses that were identified as positive or negative alias. Besides this value for each row is the accumulated with previous rows, providing a final result that is combination of different methods.

Classic methods as Mercator and Ally have very low rate of success in the discovering process. For Mercator, 0.02% of positives with no negatives, this means a high percentage of errors (90.62%) caused by packet filtering in routers. Another cause of this high percentage of errors is the different behaviors of routers that send ICMP error not from the interface closest to the destination as recommended.

Ally produces better results, 0.03% of positives and 7.35% of negatives. These improvements for the same probing packets as Mercator (UDP packets to arbitrary destination port forcing an ICMP error reply), are caused by those cases in which Mercartor is not able to recognize the alias as explained before.

The results for the new methods are presented now. IPID(UDP) gets better results than Ally,

although they are very similar. The difference in timing is the reason of the success. Rate limiting of ICMP packets in routers produce worst results in Ally as it sends two probe packets almost back-to-back.

IPID(ECHO) produces better results getting 0.21% of positives and 54.81% of negatives in pairs. This method offers new alias identification not discovered by previous methods. This type of ICMP packets (ECHO/REPLY) suffers less filtering in routers, providing better results in alias identification situations. Also this method produces unkown identification: routers are replying with ICMP packets but because of the diversity of behaviors in IPIDs the identification is not possible.

Again, IPID(TIME) gives good results compared with classic methods, but it does not produce new positive results compared with IPID(ECHO). IPID(TIME) produces some negative aliases not discovered by previous methods.

Timestamp based methods do not add new results. TSTAMP(TCP) does not give any result because no router in our measurements is responding with the timestamp option activated in the TCP header. This behavior is vendor specific and in more extensive experiments we could get improvements. Finally, TSTAMP(ICMP), although is getting replies for a big percentage of probes, the provided data does not produce conclusive results.

Comparing results in table 2, a global result of 0,30% of pairs have been aliased. This means an improvement of more 0,25% identification with previous methods in the literature. Besides, 62.87% of pairs are confirmed not to be aliases. Although there were replies from routers, some methods are not always able to give a positive or negative identification. These are represented in the 'not conclusive' column.

In table 2 other metrics have been included. All addresses discovered by initial traceroutes were 579. This means an initial topology composed by 579 nodes, each of them representing one IP address. With the alias identification process we can aggregate several IP addresses in the same node, identifying several interfaces of a router. With classic methods we reduce the nodes up to 545. With our proposal we have reduced the number of nodes up to 434, this means that we have been able to allocate more IP addresses to a reduced number of routers. The reduction in the number of links in the topology is similar.

The possible pairs of IP addresses that we had to check for aliasing are 167,331. That would be very costly in time and traffic introduced in the network. However, we can reduce this number. First, we can apply the transitive property: if IP_1 is an alias for IP_2, all aliases for IP_1 are aliases for any of IP_2's aliases. Secondly, we can test only each IP discovered in one direction of the traceroute ($SX \rightarrow SY$) with the corresponding one in the opposite direction ($SY \rightarrow SX$). As the path could be not symmetric, we check each IP address in one direction with several IP addresses discovered in the opposite direction.

Depending on the asymmetry of paths, taking the adjacents in one hop could be not enough. We study the case of taking all possible aliases for each address with all addresses in the opposite path of the traceroute. Reducing the potential aliases to check also reduces the identification success from 63.17% to 52.20%. Depending on our constraints we could prioritize identification or complexity reduction. By this mechanism we reduce the potential aliases to check to 1% of total pairs. With this reduction, if we focus only in true aliases we obtain 15% of aliases using only around 3% of probes.

6 Conclusions

In this paper we have described the problematic related with identification of different interfaces, each interface with different IP address, that belong to the same router. This procedure is called alias identification and it is vital in the process of discovering network topologies. We have analyzed the methods proposed in the literature for alias resolution, and we have detected that one of the big problems is the filtering of replies to probing packets in routers. We have proposed modifications using different kinds of probing packets, trying in all of them to get more replies from routers for alias resolution. Improvements have been also proposed in the processing of the replies and in the timing between probe packets. Our proposed modifications can improve alias identification in 55.77%.

Future improvements can be related with the reduction of traffic generated and time needed to complete those methods. Inference methods can help in this task, using active probing methods as a complement to improve the results offered by the first ones.

References

[1] D. McRobb, K. Claffy, and T. Monk, "Skitter: CAIDA's macroscopic Internet topology discovery and tracking tool," Available from http://www.caida.org/tools/measurement/skitter/, 1999.

[2] N. Spring, D. Wetherall, and T. Anderson, "Scriptroute: A public internet measurement facility," in *4th USENIX Symposium on Internet Technologies and Systems*, 2002.

[3] N. Spring, R. Mahajan, and D. Wetherall, "Measuring ISP topologies with Rocketfuel," in *Proc. ACM SIGCOMM*, 2002.

[4] Y. Shavitt and E. Shir, "DIMES: Let the internet measure itself," *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 5, pp. 71–74, October 2005.

[5] F. Baker, "Requirements for IP version 4 routers," RFC 1812, June 1995.

[6] J.J. Pansiot and D. Grad, "On routes and multicast trees in the Internet," *ACM SIGCOMM Computer Communication Review*, vol. 28, no. 1, pp. 41–50, January 1998.

[7] N. Spring, M. Dontcheva, M. Rodrig, and D. Wetherall, "How to resolve IP aliases," Tech. Report 04-05-04, Washington Univ. Computer Science, 2004.

[8] B. Huffaker, D. Plummer, D. Moore, and K. Claffy, "Topology discovery by active probing," in *Proc. the Symposium on Applications and the Internet (SAINT)*, January 2002.

[9] A. Broido and K.C. Claffy, "Internet topology: Connectivity of ip graph," in *Proc. SPIE International Symposium on Convergence of IT and Communication*, August 2001.

[10] J. Leguay, M. Latapy, T. Friedman, and K. Salamatian, "Describing and simulating internet routes," in *Proc. IFIP Networking*, May 2005.

[11] R. Govindan and H. Tangmunarunkit, "Heuristics for internet map discovery," in *Proc. IEEE INFOCOM*, 2000.

[12] M. Gunes and K. Sarac, "Analytical IP alias resolution," in *IEEE ICC 2006*, June 2006.

[13] M. Gunes and K. Sarac, "Resolving IP aliases in building traceroute-based internet maps," Technical report, University of Texas at Dallas, 2006.

[14] V. Jacobson, R. Braden, and D. Borman, "TCP extensions for high performance," RFC 1323, May 1992.

[15] D. Morato, E. Magaña, M. Izal, J. Aracil, F. Naranjo, F. Astiz, U. Alonso, I. Csabai, P. Haga, G. Somin, J. Seger, and G. Vattay, "The European Traffic Observatory InfraestruCture (ETOMIC): A testbed for universal active and passive measurements," in *Proc. TRIDENTCOM 2005*, 2005, pp. 283–289.