Improving Efficiency of IP Alias Resolution based on Offsets between IP Addresses

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

Abstract

In order to get a router-level topology in Internet, IP address alias resolution techniques allow to identify IP addresses that belong to the same router. There are several proposals to make this identification, some based on active measurements and others based on inference studies. The former provides more accuracy and completeness, however efficiency is very low because of the high number of probes needed. These methods probe IP addresses in pairs. With thousands or even more IP addresses to check for aliases, the number of tests gets too high. In order to reduce the number of probes, we propose to select the pairs of IP addresses to test for aliasing using information available a priori. This selection will be based on the offset (numerical distance) between the IP addresses to test. We will show that we can improve efficiency of active alias identification with almost no loss on completeness and without generating probing traffic. The technique is also adaptable to a distributed measurement scenario.

I. Introduction

There are a considerably amount of systems around Internet trying to make a close mapping of its topology. Systems like Skitter [1], Ark [2] , Dimes [3] , Rocketfuel [4] and Scriptroute [5] works by making a large amount of traceroutes from multiple vantage points. The result is a graph composed by nodes (IP addresses) and edges (network links). Some of them make a post-processing to obtain a router level topology using different methods to group the IP addresses owned by the same router. These methods are called *alias resolution* [6]. As a result, a new graph is obtained where nodes are routers, providing a more realistic topology map. Router-level topology maps are important to verify routing algorithms, in calculation and prediction of delay, node localization, traffic engineering, evaluating performance of P2P protocols, path restoration mechanisms, algorithms for building multicast trees and, in general, any study that would need a simulation over a realistic network scenario.

Lots of theoretical studies have worked on which could be the structure of Internet at layer 3. The research community has reached the conclusion that Internet connectivity behaves according to a power-law distribution [7]. However, nowadays this structure can not be demonstrated with a global Internet network map. Alias resolution is an important tool in that ambitious task.

The following metrics can be defined to compare methods for alias resolution [6]: accuracy, completeness and efficiency. Accuracy measures the percentage of discovered or disproven aliases that are correct. When applying alias resolution methods, positive or negative alias can be obtained. In some circumstances and with some methods it is usual to obtain also false positives and false negatives. These will be wrong results that should be minimized. Completeness measures the percentage of aliases discovered with 100% supposed a perfect alias resolution. Although this metric is very important it is difficult to define because a priori we do not know the scenario with perfect alias resolution. This information could be obtained from network operators and Internet providers, but usually they are reluctant to make public their internal network architecture. Finally, efficiency measures the amount of probe traffic used to discover aliases. This traffic is intrusive and it should be limited.

The different proposals for alias resolution can be grouped in two classes: active probing methods and inference methods. 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 control the necessary injected traffic. Besides, this traffic can be confused with scanning or attacks, so they can have problems with packet filtering in firewalls. Not only filtering, some routers could have configured rate-limiting policies to not respond to a high number of requests in a period of time. Inference methods try to deduce alias information by analyzing data from traceroute paths [8] or by getting extra data from DNS [4][6]. 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.

The two main methods in the literature for active probing alias resolution are Mercator and Ally. Mercator [9] was created by CAIDA and its implementation was called iffinder [1]. This method is based on the behavior of routers which return an ICMP error message always from the interface with shortest path to destination. This ICMP error message (port unreachable) is provoked by sending UDP packets to random destination ports on the candidate IP addresses to be aliases. Two IP addresses are alias if the ICMP error messages returned from both have the same source IP address.

The authors thank the partial support of the EU ICT MOMENT Collaborative Project (Grant Agreement No.215225) and Spanish MEC project STRRONG (TEC2007-62192/TCM)

Ally [4] was develop by Rocketfuel project [4]. It is based on the behavior of the IP identifier (IPID) field of the IP header. Typical 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. Ally checks two candidate IP addresses sending three UDP probes with random ports to enforce again ICMP error message whose IP identifiers can be analyzed. Both IP addresses will be alias if the distance between IP identifiers is in between a threshold of 200 sequence numbers [4]. Ally provides the best results for completeness and accuracy in alias resolution [6].

The main problem with active probing for alias resolution is that, for example in Ally method, the number of probes scale with the quadratic of the number of IP address ($O(n^2)$). For large topology maps this huge number of probes is not viable. One possible solution is to relegate active probing methods as a verification of the result provided by inference methods. In this case, active probing is only used to check a possible alias suggested by an inference method. However the completeness of the results are much lower with this procedure [8].

Another alternative could be to improve efficiency of active probing methods by pre-selecting the pairs of IP addresses to check for aliasing. These techniques are called *splitting methods* in [10]. They are based on sending probes only to the set of IP addresses that have more higher probability to be aliases. The selection of which IP addresses have more probability to be aliases are based on TTL [6] and IP identifier [10]. Both will be explained in the following section. These methods have several disadvantages related to the extra probing traffic needed and the difficulty to implement the calculation in a full distributed way.

In this paper we propose a new method to improve efficiency of active probing based on offsets between IP addresses. It has the following advantages. First, it does not introduce extra traffic in the network to provide the pairs of IP addresses with more probability to be aliases. Second, the method is useful when we need to make aliasing tests from different distributed vantage points (very important in large scale topology mapping). Third, configuration parameters for the method are not dependable of the specific network topology under study. The rest of the paper is organized as follows. Section II presents the different existing methods to improve efficiency of active probing methods. Then, our proposal is presented in section III and all methods are evaluated in Internet scenarios using Planetlab and ETOMIC infrastructures. Next section IV presents the use of clustering techniques for improving and generalization of results for our proposal. Finally, conclusions are presented.

II. Existing methods to improve efficiency of active probing for alias resolution

Active probing techniques for alias resolution need to send several packets to two target IP addresses in order to check if they are alias because they belong to the same router. In Mercator only two packets from the same probing source to target IP addresses are needed. If more IP addresses are needed to be verified, an extra probing packet is needed for each IP address. The traffic generated will be proportional to the number of IP addresses ($O(n)$) if the probing station is only one. With Mercator, it is not efficient to distribute the analysis between several probing stations (m), because the traffic generated will increase proportionally ($m * O(n)$). Alternatively, Ally needs three packets for each pair of IP addresses to verify alias and these results can not be reused to compare with a third or more IP address. Therefore the traffic generated will be proportional to the number of possible pairs ($O(n^2)$). The efficiency, specially for Ally, is not good enough.

Some way is needed to improve the efficiency of active probing techniques. The idea is to make some kind of selection looking for the IP addresses which are more likely to be aliases. This would mean a reduction in the number of probing packets improving efficiency but, at the same time, it could provide worse completeness. If not the full search space of IP addresses is used, some aliases will be lost. However, we will try to reduce these losses as much as possible.

One of the proposals in the literature for improving efficiency is based on the TTL (Time-to-Live) field of the IP layer [6]. This study shows that if we get the TTL distance between a pair of IP addresses ($TTL_1 - TTL_2$), it is very probable that this distance was 0 if both IP addresses were alias. The justification is clear: if two IP addresses belong to the same router, they will be at approximately the same distance in number of hops from the source probing station. However, this is not always true because the path to reach each IP address can be very different, crossing different number of hops and then with different distance. To increase completeness, larger TTL distances like 1, 2 or 3 can be considered, but then the number of pairs of IP addresses to check also increases. If we want to improve completeness we will take a larger set of IP addresses and, then, more probing traffic will have to be sent.

This TTL-based method provides fine results but it has related problems. Original traceroutes, from where IP addresses of routers are obtained, are usually made from different vantage points. Therefore TTL information in traceroutes can not be compared between IP addresses because it could have been measured from different source probing stations. So, for each IP address we need first to make a new active probing procedure to get TTL information and this procedure has to be done with all IP addresses. So, more probing traffic is needed at first. Again, in large topology maps we would need to distribute this procedure between different source probe stations, but it is not possible because we would need TTL information for all IP

addresses from all source probe stations, and this would mean increasing the probing traffic proportionally to the number of source probe stations.

Another proposal in the literature for improving efficiency is based on the IPID field of the IP layer [10]. This study shows that for two IP addresses which are alias, the IPIDs in returned packets are very close. Now, the distance between two IPIDs for two packets sent by two different IP addresses must be calculated. With all the offsets between IPIDs we will take those whose IPIDs are closer. Two IPIDs sent from the same router will have higher prov to be close than two IPIDs sent from different routers, so this characteristic is used to reduce the number of pairs to check. The IPID is incremental for all IP packets generated by a router independently of the outgoing interface. This allows to probe the same router from two different source probe stations and then to get the result as if it would be made from the same host. The generation of an IP packet (with the IPID field) in a router is forced sending a UDP packet to a random port as in Ally method. The router will answer with an ICMP error message of port unreachable. However, this test has to be made for each pair of IP addresses in a very short period of time in order to be able to find close IPIDs for packets coming from the same router. Larger periods of time between probing both IP addresses of a router would imply the possibility of more traffic generated by the router to other destinations and therefore it would be more difficult to detect the relation between IPIDs for both IP addresses (the distance between IPIDs would be larger). We need synchronization between the source probe stations, but this makes the implementation harder. Besides, the full process for checking all IP addresses will take longer in order to avoid interferences between different source probe stations. Again, extra probing traffic has to be sent prior to decide which IP addresses have to be checked for aliasing.

To sum up, we have two methods for improving efficiency of active probing for alias resolution by reducing the IP addresses to check, but they generate extra probing traffic and even they do not work in a real distributed probing scenario. Both reasons make difficult to apply them to large scale topology maps. In next section, a new method will be proposed that will try to address both limitations.

III. Method based on offsets between IP addresses

A new method to improve efficiency of active probing for alias resolution is proposed in this paper. The idea is to be able to reduce the number of IP addresses to check for aliasing with the following advantages: first, the method will not generate extra traffic, and second, it will be able to be implemented in a distributed way. In order to suggest if two IP addresses have a certain probability for being alias, the method will use the offset between both IP addresses considered as two unsigned integer numbers of 32 bits. Basically the method will use the result of subtracting one IP address from the other ($|IP_1 - IP_2|$) to suggest the relation between them. The offset between two IP addresses will be called *IP offset*.

In fact, the following behavior has been observed in IP addresses: pairs of IP addresses in between certain offsets have more probability to be aliases. This is related with how IP addresses are organized in Internet. An Autonomous System (AS) can be interconnected with other ASs of the same or different Tier (Internet hierarchy level). ASs usually use addressing related with its interconnections. For example, ASs in Tier-2 can interconnect with other Tier-2 ASs using B-class networks, with Tier-1 ASs using A-class networks and with Tier-3 ASs using C-class networks. In this case, the central router of the Tier-2 AS could have interfaces with different addressing schemes depending on the type of AS in the other end. Some of the interfaces in the router will have close addressing (because they interconnect with other ASs of the same tier) and other interfaces could have an addressing belonging to other class, this means, with an important gap in IP offset.

To check this behavior, experimental studies in the real Internet have been made with the distribution of IP addresses and the fact to be aliases. This studies have used Etomic[11][12] and Planetlab[13] measurement infrastructures. Tools and data sets used in this study are openly available in [14].

Etomic provides 18 nodes distributed around Europe (Spain, France, Italy, Hungary, United Kingdom, Belgium, Sweden, etc.) with high-precision network cards and GPS synchronization. The procedure of the experiment has been decomposed on three phases: phase 1 for obtaing IP addresses in the network topology, phase 2 for applying methods to improve efficiency of active probing for alias resolution (reduce the number of pairs of IP addresses to check for alias resolution) and phase 3 for applying the final alias resolution schemes. The objective is to compare the improvement in efficiency for methods applied in phase 2.

In phase 1, traceroutes between those 18 Etomic nodes have been made generating 510 IP addresses for routers in the paths between Etomic nodes. Specifically, paris-traceroute version [15] has been used in order to avoid load balancing in Internet routers. Paris-traceroute sends all probes in a path discovery using the same source and destination port as difference with standard traceroute. As routers in Internet use mainly load balancing by flow instead of by packet, paris-traceroute allows to avoid the problematic related with route flapping. IP addresses of routers in all available routes are obtained by repeating paris-traceroute several times for each source/destination. Paris-traceroute tool has been developed and ported to both Etomic and Planetlab measurement infrastructures.

In phase 2, methods to improve efficiency of active probing for alias resolution are applied. These methods are TTL-based (TTL offset), IPID-based (IPID offset) and IP addresses-based (IP offset, our proposal). The idea is to reduce the number of pairs of IP addresses to test for aliasing. TTL offset and IPID offset methods need extra probing traffic. The TTL offset method

provokes ICMP error packets from the IP addresses under analysis to check their TTL number (one probing packet for each IP address). The IPID offset method provokes ICMP error packets from the IP addresses under analysis to check their IPID numbers (three probing packets for each pair of IP addresses to verify, as in Ally alias resolution method). In the proposed IP offset method, no extra traffic is necessary.

In phase 3, the alias verification has been made applying Mercator, Ally and a modified Ally. Our modification uses different probing packets like ICMP or TCP to receive the ICMP response or error, and the results improve the performance of basic Ally. A custom software has been developed implementing the functionalities for those alias resolution methods. For each pair of IP addresses the output will be positive (alias) or negative (not alias), if all methods agree in the result. If the results are different for each method the output will be labeled as not conclusive. In order to compare the improvement in efficiency applying the methods in phase 2, only positive alias will be considered.

First we focus our study in IP offset method for phase 2. For Etomic scenario, Figure 1 shows the histogram for the offsets between IP addresses (IP offset), considering the full set of pairs tested (a), considering only pairs that are aliases (with methods in phase 3) (b), and the survival function for previous figures (c).

Figure 1.a shows that IP offset for the full set of IP addresses is distributed along almost all the IP offset space. This means that IP addresses in each pair can be in a very different subnet. However in Figure 1.b IP offset for the IP addresses that are alias is concentrated in two clear points: around 0 and around 2.15e+09 (half the number of possible IP addresses, 2^{32}). This distribution is more clearly presented in the survival function shown in Figure 1.c where the two steps present in the 'Aliases' curve correspond with the previous points of interest. The survival function for all IP address space is distributed along the IP offset axis while the survival function for aliases has two clear steps.

The intuitive explanation for the specific distribution of IP offset is related with the adjacency between Autonomous Systems (AS). IP addresses of routers in the same AS have an IP offset around 0, and IP addresses of border routers between ASs have an IP offset around 2.15e+09. This explanation will be objective of future work.

The behavior observed in Figure 1.c can be used to estimate the range of IP offsets that can be alias with higher probability. These IP offsets will be concentrated in the steps of the figure. This behavior could be thought as specific of the Etomic paths considered. However we have observed this behavior in between several nodes of Internet. In concrete, the same analysis has been repeated for 18 Planetlab nodes located around world: USA, Germany, United Kingdom, Spain, France, Korea, Taiwan, etc. Traceroutes have been made between all these nodes. In this case we have obtained 369 IP addresses for routers in the paths between Planetlab nodes. The result for the distribution of IP offsets is presented in Figure 2. The survival function for all IP addresses is very similar to the one from the previous scenario. The survival function for alias pairs is a bit different: the two steps commented before can be observed around the same IP offsets, but in this case the second step is less significant.

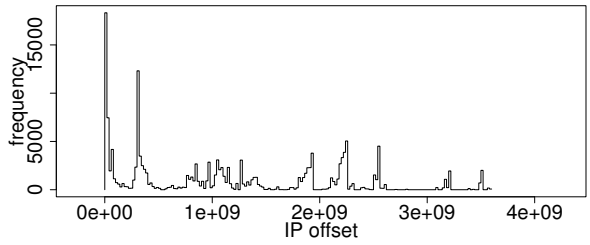
The method based on IP offset needs some mechanisms to select the ranges from figures 1.c or 2 where the probability of being alias is maximized. As a first approximation, two zones which the steps in the survival function can be distinguished visually. The range of IP offset can be specified for both zones. Then the percentage of alias and not alias pairs that are present in both zones can be analyzed in order to obtain the efficiency of the proposed method.

The range of IP offset around 0 includes the subnetworks /30 and /31 that belongs to point-to-point links where both IP addresses are for sure in different routers. This would correspond to IP offset value equals to 1. The problem is that the mask for each IP address is not known and then it is not possible to know wich IP addresses belong to a point-to-point link. An approximation could be to ignore IP offset with value equals to 1, but this is an oversimplification. For example, in our Etomic scenario there are 4 aliases with IP offset equals to 1. Therefore, this IP offset value can not be discarded as candidate to be indicator of aliases.

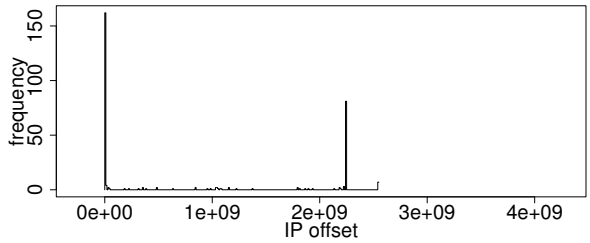
Using the IP addresses for the Etomic scenario, the efficiency is calculated comparing the percentage of IP addresses-pairs tested with the percentage of alias resolved positively. For good efficiency we need a high percentage of alias resolved positively for a low percentage of pairs tested. Figure 3 shows the efficiency for IP offset method compared with the methods in the state of the art for phase 2: TTL and IPID based. In IP offset method, 10 points are obtained in Figure 3 by increasing the size of the zones around the steps in the previous survival function in Figure 1.c. In TTL method, the different points are obtained by increasing the TTL threshold one by one, starting with a zero TTL offset. In IPID method, different points are obtained again using different thresholds for IPID offset.

Comparing completeness in Figure 3, IP offset method provides the best results with low percentage of pairs tested: with 10% of pairs tested for aliases, around 85% of aliases are obtained. As the percentage of pairs tested for aliasing is increased, the results are close to those provided by the TTL method. The behaviour of IPID method is special because the working zone is concentrated around 30% of pairs tested, getting the best results compared with other methods.

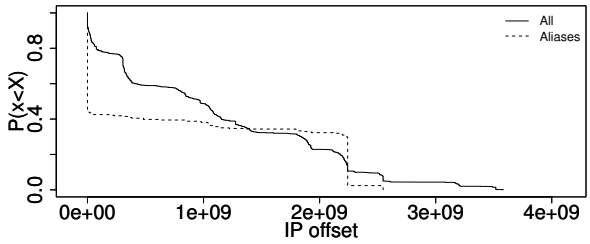
Although IP offset method provides results for efficiency near to other methods in the state of the art or even better, the big advantages of the proposal are related with its following two main characteristics. First, it must be noted that IP offset method is applied only over the values of IP addresses directly, without injecting any extra probing traffic in the network as needed by TTL and IPID based methods. This means avoiding interferences over real network traffic and, what is most important, the results can be obtained very fast even for very large topologies. Second, the alias resolution procedure can be made in a



(a)



(b)



(c)

Fig. 1. Histogram of IP offsets for all IP address space (a), for pairs that are aliases (b), and survival function for both (c)

truely distributed way from different vantage points because the method does not depend on the localization of the probing node neither synchronization between probing nodes. In TTL method, we have explained that in order to compare TTL values to different IP addresses the measurements have to be made from the same vantage point as it requires to compare TTLs values in pairs considering all possible pairs. In IPID offset method, in order to avoid interference between probes coming from different probing sources some kind of synchronization would be needed, increasing complexity and the time needed to complete the method.

These two advantages are summarized in Figure 4 where the amount of probing traffic necessary for each method in phase 2 is presented compared with the number of source probing stations (grade of distribution). Figure 4 present the results for Etomic scenario. IP offset method does not need any probing traffic, the line is located in 0 bytes transmitted. TTL offset needs to send one probe packet for each target IP address from each source probing station. Its traffic will be proportional to the number of nodes (n), the number of source probing stations (m) and the size of probe packets ($s = 64bytes$), total bytes necessary would be $m * n * s$. IPID offset needs even more traffic because the probing must be done in pairs and two probe packets are needed for each pair. In this case, the generated traffic does not depend on the number of source probing stations because pairs to check can be distributed between source probing stations. So the traffic generated in IPID offset method would be $((n * (n-1))/2) * 2 * s)$. The distribution of tasks for phase 2 between several source probing stations is not efficient for TTL method as shown in Figure 4. Besides, more probing traffic will imply an increase in the time necessary to complete the task. Therefore, IP offset method provides the best results considering the amount of traffic necessary (none) and the effect of distribution in several source probing stations.

The next step will be to avoid manual intervention to define the interesting ranges of IP offset. Previously we have used a visual method to define these ranges. In the following section we provide a systematic method based on clustering algorithms:

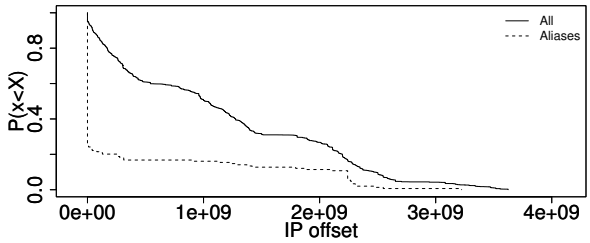


Fig. 2. Survival function for IP offsets in all pairs of IP addresses and in pairs that are aliases

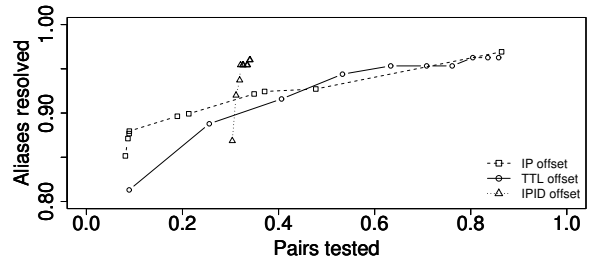


Fig. 3. Comparison of methods for improving efficiency of active probing techniques in alias resolution

the Expectation Maximization (EM) and the K-means algorithms.

IV. Using clustering for IP offset

The steps in the survival function presented in figure 1.c concentrate the ranges of offsets between IP addresses that have to be considered in our analysis. Alias resolution methods are applied only to these ranges in order to obtain true aliases with more probability. Although these ranges could be selected visually, they could be optimized for each scenario, so an automatic procedure would be preferred. Two clustering algorithms have been considered for this task. The following study presents the results depending on the training procedure and the grade of generalization of results for different scenarios.

Data from Etomic and Planetlab scenarios are aggregated in order to obtain a complex scenario for the analysis. This will be named the complete scenario. This complete scenario would allow to check if generic results can be obtained that could be applied to different scenarios. At the same time, two clustering algorithms are considered: EM [16] and K-means [17]. The training data for both clustering algorithms can be diverse and will be explained next. Results are shown in Figure 5. First, EM clustering algorithm is able to provide the optimal number of clusters for the training data. If we use the true aliases (those offsets of IP addresses that provide true aliases) as training data, EM provides 3 clusters. This is called 'EM optimal aliases' in figure 5. For those three clusters, two of them correspond to the ranges of true aliases, and the third cluster can be discarded. K-means clustering algorithm needs the number of cluster to be specified at advance. Using the same number of clusters (3) for K-means, the results are not good, too far from the visual approximations so they are discarded.

Alias resolution methods provide true and false aliases. False aliases (those pairs of IP addresses that are not alias for sure) can also be considered as a training set for EM clustering algorithm. In this case, EM generates 15 clusters. Three of them correspond to true aliases, each with different percentage of true aliases. Aggregating one or more of these three clusters we can select the percentage of pairs tested and the related percentage of alias resolved. This results are named 'EM optimal not aliases' in figure 5.

This number of 15 clusters has been fixed for EM clustering algorithm and true aliases as training set. This gives more flexibility, providing several points in figure 5 with the legend 'EM 15 cluster aliases'. This would allow to priorize percentage of alias resolved or percentage of pairs tested.

Finally, this number of 15 clusters has been fixed for K-means clustering algorithm, using true aliases as training set (named 'KM 15 cluster aliases' in figure 5) and false aliases as training set (named 'KM 15 cluster not aliases' in figure 5).

As observed in figure 5, EM provides better results with low percentage of pairs tested. The importance of the training set is more related with the number of points available in the figure and therefore flexibility in the point of work to use. K-means clustering algorithm is much faster than EM clustering algorithm, giving good results with high percentage of pairs tested.

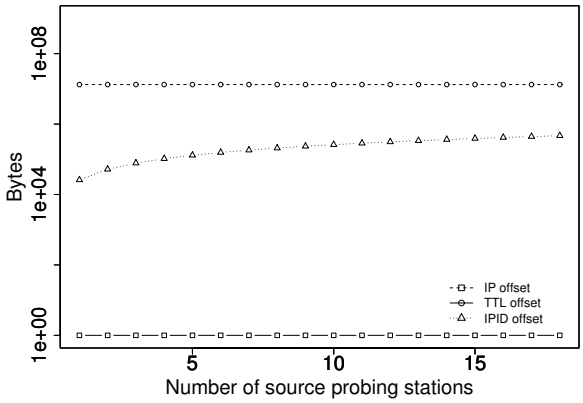


Fig. 4. Comparison of probing traffic level in methods for improving efficiency of active probing techniques in a distributed probing scenario

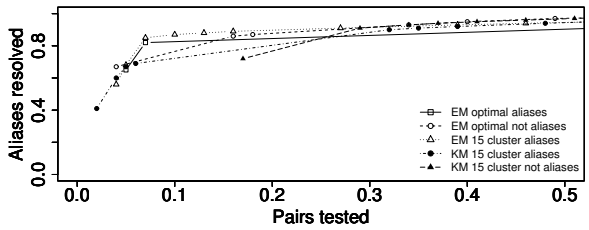


Fig. 5. Comparison of clustering algorithms trained with the complete scenario and different training sets

It must be noted that this is the complete (worst) scenario where we are combining IP addresses from Etomic and Planetlab scenarios. Following, we are going to consider both scenarios individually, analyzing the effect of using the training set from one scenario and applying the results in the other scenario. The four combinations plus the complete scenario results are presented in figures 6-10 for the clustering algorithms and different training sets revised before. Specifically, for each clustering algorithm (EM and K-means) the number of clusters is changed using alias and not alias inputs for training. Besides, different combinations of data from Etomic and Planetlab scenarios are used, combining the data sets as training and as testing. For example, data sets with pairs of IP addresses that are alias in Etomic are used as training in clustering algorithms and then the resulting clusters are used to classify possible alias for Etomic and Planetlab pairs of IP addresses. The *All* data set refers to the aggregation of data from Etomic and Planetlab data sets.

In figure 6 with only three clusters in EM clustering algorithm with optimal alias training, there are only two points corresponding to the two clusters with true aliases for each scenario. In this case, the Etomic scenario with Etomic training set provides better results. However, this difference is almost insignificant in the rest of the figures. The reason is that IP offset survival functions were very similar in Etomic 1 and Planetlab 2 scenarios. This independence of the training set is very important because it will allow us to define ranges for offsets between IP addresses that will be applicable within any topology discovering process. This means that offset ranges (clusters) generated with Etomic scenario can be applied to Planetlab scenario without any modification, and eventually to any network scenario (this is an ongoing research). The generality of the results is another advantage of the proposed IP offset method.

The EM clustering trained with true aliases and 15 clusters (Figure 8) provides the best results: around 90% of aliases resolved for around 10% of pairs tested. In this case, the application on the Etomic scenario with Etomic training set provides slightly better results than the other combinations (in the order of 2% of improvement for alias resolved). Using training with not aliases in EM clustering (shown in Figure 7) provides again 15 clusters but results are worse than with previous training.

K-means algorithm presents worse results that EM even using the same number of clusters. Figure 9 shows the K-means training with not alias input and Figure 10) shows the K-means training with alias input. K-means for both cases needs a larger percentage of pairs tested to obtain the same number of alias resolved than EM. K-means algorithm is faster than EM but in this case speed is not important because the clusters can be precomputed and applied to very different topology scenarios as shown before for Etomic and Planetlab.

An interesting conclusion from Figures 6-10 is that results almost do not depend on the training set used in the clustering.

This means, for example, that clusters obtained with Etomic training set are applicable to Planetlab scenario without large deviation in the results. Clusters are independent of the scenario or the specific set of IP addresses to analyze. This will allow to have pre-calculated clusters to classify certain IP offsets related to possible alias. Then, the procedure to apply the IP offset method will be simpler with pre-calculated clusters.

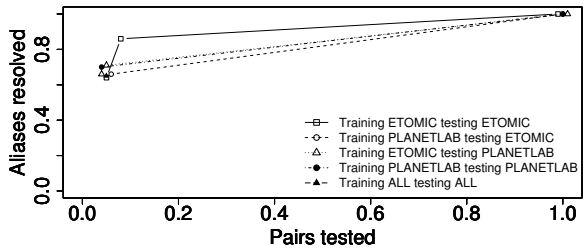


Fig. 6. EM alias optimal clustering using different trainings sets

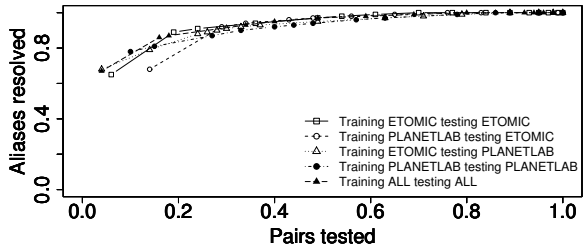


Fig. 7. EM not alias optimal clustering using different trainings sets

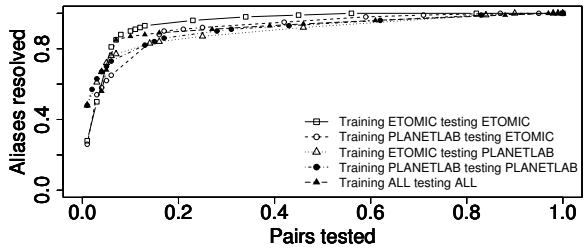


Fig. 8. EM alias (15 clusters) clustering using different trainings sets

V. Conclusions

The reduction of cost in alias resolution is very important because of the high number of probing traffic and time needed in this task. This paper proposed to check for aliasing only on a subset of pairs created based on the IP offsets between them. The results have shown the advantages compared with other proposals. It does no need to inject probing traffic (that sometimes could be even considered as malicious traffic), avoiding interfering with normal network traffic and speeding the full process by several orders of magnitude. Besides, the proposal can be used in a distributed alias resolution system because of its independence on the source probing node. This would allow to speed up even more the alias resolution procedure. Finally, the results are generalizable, providing similar results independently of the training set used.

The efficiency is similar or even better than the one for the methods in the literature. This proposal achieves to obtain 90% of aliases resolved postively with only 10% of pairs tested. As the percentage of pairs tested is increased, the alias resolution converges to perfect alias identification. Between the proposed clustering algorithms, EM clustering algorithm with 15 clusters

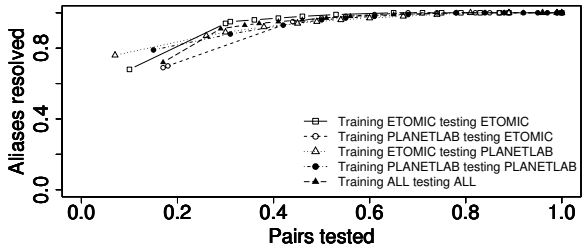


Fig. 9. KM not alias (15 clusters) clustering using different trainings sets

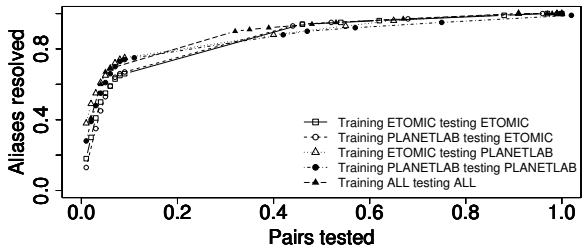


Fig. 10. KM alias (15 clusters) clustering using different trainings sets

and true aliases as training set provides the best results in all scenarios. This clustering can be made one time with a scenario, and then the clusters can be reused with other scenarios with almost similar performance.

As future work, the intuitive explanation of IP offset distribution and its relation with aliasing must be addressed. Second, the application of IP offset method in IPv6 can be analysed. Both methods in state of the art, TTL-offset and IPID-offset, can be applied for IPv6. However, it seems that IP offset method will need a modification.

REFERENCES

[1] Bradley Huffaker, Daniel Plummer, David Moore, and Kc Claffy, "Topology discovery by active probing," in *Proc. the Symposium on Applications and the Internet (SAINT)*, January 2002.

[2] CAIDA, "ARK, Archipelago Measurement Infrastructure," http://www.caida.org/projects/ark/, 2002.

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

[4] Neil Spring, Ratul Mahajan, and David Wetherall, "Measuring ISP Topologies with Rocketfuel," *IEEE/ACM Transactions on Networking*, vol. 12, no. 1, pp. 2–16, February 2004.

[5] Neil Spring, David Wetherall, and Tom Anderson, "Scriptroute: A public internet measurement facility," in *4th USENIX Symposium on Internet Technologies and Systems*, 2002.

[6] Neil Spring, Mira Dontcheva, Maya Rodrig, and David Wetherall, "How to resolve IP aliases," UW CSE Technical Report 04-05-04, Department of Computer Science and Engineering, University of Washington, Seattle, 2004.

[7] Sevcan Bilir, Kamil Sarac, and Turgay Korkmaz, "Intersection characteristics of end-to-end internet paths and trees," in *13TH IEEE International Conference on Network Protocols*, November 2005, pp. 378–390.

[8] Mehmet Gunes and Kamil Sarac, "Analitycal IP alias resolution," ICC '06. IEEE International Conference, 2006.

[9] Jean-Jacques Pansiot and Dominique Grad, "On routes and multicast trees in the internet," ACM SIGCOMM Computer Communication Review, 1998.

[10] Hal Burch, "Measuring an IP network in situ," Carnegie Mellon University, PhD thesis, ISBN 0-542-01549-8, 2005.

[11] Daniel Morato, Eduardo Magaña, and Mikel Izal et al., "The European Traffic Observatory InfraestruCture (ETOMIC): A testbed for universal active and passive measurements," in *Proc. TRIDENTCOM 2005*, 2005, pp. 283–289.

[12] Public University of Navarre and Collegium Budapest, "ETOMIC: European traffic observatory measurement infrastructure," http://www.etomic.org.

[13] Princeton University, "PLANETLAB: An open platform for developing, deploying, and accessing planetary-scale services," http://www.planet-lab.org.

[14] Santiago Garcia-Jimenez et al., "Tools and data sets in alias resolution research," http://www.tlm.unavarra.es/∼santi/research.

[15] Brice Augustin, Xavier Cuvellier, Benjamin Orgogozo, Fabien Viget, Matthieu Latapy Timur Friedman, Clemence Magnien, and Renata Teixeira, "Avoiding traceroute anomalies with paris traceroute," in *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, October 2006, pp. 153–158.

[16] D. B. Rubin A. P. Dempster, N. M. Laird, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society, Series B*, vol. 39, no. 1, pp. 1–38, 1977.

[17] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*. 1967, vol. 1, pp. 281–297, University of California Press.