Pythia: Yet Another Active Probing Technique for Alias Resolution

Pietro Marchetta, Valerio Persico, Antonio Pescapé
University of Napoli "Federico II", Italy
{pietro.marchetta,valerio.persico,pescape}@unina.it

ABSTRACT

An accurate and exhaustive knowledge of the Internet topology is essential for a deep understanding of such a complex and ever-evolving ecosystem. In this context, a well-known key challenge is represented by *alias resolution*, i.e. the process of grouping under a unique identifier the addresses owned by the same network layer device. While several techniques exist, each solution shows specific limitations such that the alias resolution problem appears far from being definitively solved. In this work, inspired by a previous technique and the lessons learned by experimenting with IP options, we present, evaluate and release Pythia, a novel active probing-based alias resolution technique. Pythia exploits a combination of (i) UDP packet probes and (ii) the IP Prespecified Timestamp option and it is purposely designed to reconstruct a specific category of routers. By using the reliable topological information provided by IGMP probing as a reference, we experimentally evaluate Pythia and compare it to previously proposed techniques according to multiple performance metrics. Experimental results show how Pythia reaches higher performance in terms of applicability and trustworthiness.

Categories and Subject Descriptors

C.2.1 [**Computer-communication networks**]: Network Architecture and Design—*Network topology*

Keywords

IP alias resolution; Internet topology; IP to Router mapping

1. INTRODUCTION

An accurate and complete knowledge of the Internet topology [1] is essential for (i) a deep understanding of such a complex and ever-evolving ecosystem, (ii) for simulating, analyzing, and designing novel networking protocols [2] and (iii) for verifying, correcting, and improving various desirable aspects of the Internet, including its robustness, reliability, efficiency, and security [3].

A very common approach adopted in Internet topology discovery is to launch large scale measurement campaigns based on traceroute [4,5]. This tool sends TTL-limited packets towards a destination, eliciting ICMP Time Exceeded replies from the routers along the path. The source address of the Time Exceeded packets allows to infer an IP-level view of the Internet topology. In order to reconstruct from the IP-level the router-level topology it is necessary to group under a unique identifier the addresses owned by the same network device. This problem is commonly known in literature as *alias resolution* [6].

Unfortunately, while the alias resolution problem has been continuously gathering interest from the research community (as demonstrated by recent works [7, 8]), all the proposed techniques suffer from specific limitations (Sec. 2). As a consequence, the alias resolution problem appears today still far from being definitively solved.

In this work, inspired by a previous technique [9] and the experience gathered by experimenting with IP options [10–12], we present, evaluate and release Pythia, a novel alias resolution technique based on active probing. Pythia exploits a combination of (i) IP Prespecified Timestamp option [13] and (ii) UDP packet probes to reconstruct a specific category of routers, we named *any-interface stamping routers*, representing about one tenth of the devices in the Internet according to previous works [10, 11] (Sec. 3). By using the reliable topological information provided by IGMP probing, we evaluate Pythia and other techniques according to multiple performance metrics to quantify their applicability, accuracy, and trustworthiness (Sec. 4). Experimental results show how Pythia is able to achieve higher performance than the other techniques tested over this category of routers.

2. RELATED WORKS

Several active probing techniques have been proposed over the years to solve the alias resolution problem[1] [14]. In this section, we briefly provide a high-level taxonomy of the previous techniques as well as their main limitations.

Source address. One of the first proposed techniques is known as *common source address* [15]: the addresses A and B are classified as alias if a UDP packet probe sent toward A (B) elicits an ICMP Port Unreachable reply from B (A). A similar approach has been recently proposed in Palmtree [16] that induces the router owning the address A

[1]Active probing is not the only approach. Other techniques infer aliases by analyzing the graph of the topology.

CoNEXT'13, December 9-12, 2013, Santa Barbara, California, USA.

http://dx.doi.org/10.1145/2535372.2535413 .

to generate an ICMP Time Exceeded message. Common source address and Palmtree infer addresses in alias exclusively when they collect replies from addresses different from the targeted ones. As a consequence, these techniques cannot directly tell if two given IP addresses are in alias or not.

Shared counter. Since some routers maintain a single counter shared among different interfaces to set the IP-layer identifier (IPID) of the outgoing packets, other techniques perform alias resolution by monitoring the evolution of the IPID value over multiple solicited replies. This approach has been first proposed in *ally* [17] and successively refined in *radargun* [18] and *midar* [19]. Recently, a similar approach has been applied also to IPv6 routers [8]. These techniques work exclusively on devices implementing an IPID counter shared among different interfaces and imply an adequate IPID sampling rate in order to infer the addresses in alias.

Timestamp option. The most related work and source of inspiration for our proposal is the technique introduced by Sherry *et al.* [9], one of the first works demonstrating the potentialities of the IP Prespecified Timestamp option (hereafter simply TS option) for Internet measurements. The TS option allows to prespecify in a single packet up to four IP addresses from which a timestamp is requested. By adopting the notation suggested by [9], hereafter X|ABCD refers to a generic IP packet probe equipped with the TS option, where X is the targeted destination and ABCD is the ordered list of prespecified IPs from which a timestamp is requested. Note that the position of each prespecified address in the ordered list ABCD is essential since it implies that B cannot insert its own timestamp before A, C before B and so on. The basic mechanism proposed in [9] to determine if A and B are in alias or not is to send ICMP Echo Request probes having the format A|ABAB and B|BABA. The technique classifies the addresses as in alias when they provide ICMP Echo Reply messages with four timestamps recorded. The addresses providing only two timestamps are further investigated and declared as in alias only if (i) the provided timestamp values are consistent and (ii) the experimental observations are compliant with topological constraints.

Similarly to the techniques described above, Pythia injects into the network synthetic traffic to infer addresses in alias, however, Pythia has been purposely designed to solve the alias resolution problem for a well-defined category of routers (see Sec. 3). Compared to [9], Pythia exploits (i) the TS option with a different rationale for arranging the addresses in the timestamp requests and (ii) UDP packet probes instead of ICMP Echo Request packets. Thanks to these design choices, Pythia is able to potentially identify all the addresses belonging to the same router, even if only one of these addresses is responsive, unlike all other techniques. This feature appears particularly useful since IP options may expose the traffic to filtering policies [20,21]. More in general, Internet measurements based on IP options represent a new promising research trend as demonstrated by the increasing number of works proposing IP options-based techniques [10, 12, 22–25].

3. PYTHIA

In this section, we describe Pythia, a novel active probing technique for alias resolution. In particular, we first provide a brief overview of the categories of routers managing the TS option. Then, we describe the basic principle exploited by Pythia to infer the alias relation and the algorithm designed to this end. Finally, we discuss both advantages and limitations of our approach.

3.1 TS option and router behaviors

In a previous work [11], thanks to a large-scale measurement campaign targeting more than 1.7M IP addresses, we observed that the routers managing the TS option in the Internet can be classified in the two main categories reported below (interested readers may also refer to a CAIDA online report [26] for more details).

Per-interface stamping routers. When processing the TS option, these routers insert one timestamp when the packet probe passes through the interface associated to a prespecified address. Accordingly, a per-interface stamping router owning the IP address Y provides between 0 and 2 timestamps when it is probed with an ICMP Echo Request Y|YYYY packet[2].

Any-interface stamping routers. These routers insert all the requested timestamps when the prespecified address is associated to *any* owned interface. Differently from per-interface stamping routers, these devices provide exactly 4 timestamps when they own the address Y and are probed with an ICMP Echo Request Y|YYYY packet. Empirical evidences [11] suggest that Juniper routers act in this way.

Our technique is purposely designed to reconstruct any-interface stamping routers that represent about 10.4% of the devices in the Internet according to recent experimental campaigns [10, 11].

3.2 Basic principle

The goal of Pythia is to identify among a set of potential candidates, *all* the addresses owned by the same any-interface stamping router. To this end, the technique exploits UDP packet probes and the TS option as detailed in the following.

UDP packet probes. UDP packet probes toward a high and presumably unused port allow to avoid ambiguities caused by the devices located along the reverse path. Indeed, these probes elicit ICMP Port Unreachable messages from the destination. Since the ICMP error messages typically return the original packet triggering the error into the payload, it is possible to extract the TS option from the payload of the reply as affected exclusively by the forward path.

Prespecify the destination first. While UDP packet probes allow to avoid ambiguities caused by the reverse path, the routers located along the forward path may still interfere by inserting their own timestamps in the option. To exclude such ambiguities, Pythia prespecifies the destination of the packet probe always as the first address into the TS option: in this way, none of the routers located along the forward path can insert its own timestamp before the targeted router.

The combination of these two mechanisms allows to conclude that any timestamp observed into the TS option returned in the payload of the ICMP Port Unreachable reply has been inserted by the targeted device.

[2]Note that 0 timestamps may be also determined by routers that simply ignore the TS option. From the lack of timestamps is not possible to tell the difference.

3.3 Algorithm

Given an initial set of addresses to alias, Pythia runs in two phases: (i) preliminary test and (ii) alias resolution.

Preliminary test. This phase aims to isolate the addresses owned by any-interface stamping routers and to exclude devices showing anomalous behaviors. To this end, for each address A of the initial set of addresses, Pythia sends two UDP packet probes A|AAAA and A|AZZZ, where Z refers to an address at the University of Napoli, known to be outside the traversed path. The first probe (A|AAAA) allows to split the set of candidate addresses in three main subsets: (a) *unresponsive* addresses, (b) *compliant* addresses − the ones providing 4 timestamps being owned by any-interface stamping routers; (c) *non-compliant* addresses − those providing less than 4 timestamps. As already proposed in [9], the second probe (A|AZZZ) allows to remove from the compliant address set the routers showing anomalous behaviors: since the address Z is surely not located on the traversed path, observing any timestamp associated to Z demonstrates that the targeted router inserts extra timestamps independently from the prespecified addresses. Since this behavior may strongly affect the accuracy of our results, when it is recognized, the corresponding address is considered non-compliant[3]. The sets of unresponsive and compliant addresses represent the input of the following phase.

Alias resolution. In this phase, Pythia performs all the operations required to identify the addresses contained in the initial set owned by the same any-interface stamping router. Let us denote with α and β the ordered lists of addresses respectively compliant and unresponsive. The technique iteratively performs three steps.

1) An address A is popped from α, hereafter we refer to this address also as the *pivot*. During this iteration, the technique tries to infer all the addresses in alias with the pivot. To this end, a new ordered list, named γ, is created by simply concatenating α and β: γ contains all the addresses potentially in alias with A. Let us assume that γ contains the addresses B,C,D,E and so forth.

2) Pythia sends a first UDP packet probe A|ABCD and counts the number of collected timestamps to (i) infer the addresses in alias with the pivot and (ii) determine the next probe to send as reported in Tab. 1. For instance, when the probe A|ABCD elicits an ICMP Port Unreachable message where the returned TS option contains two timestamps (i.e. the ones associated to A and B), this is a clear evidence that A and B are in alias: indeed, the probe is crafted such that only the router owning A is allowed to insert timestamps. Furthermore, the lack of a timestamp associated to C implies that the same router has not recognized this address as an owned one. At the same time, we do not have any clue about D because this address appears just after C in the ordered list of prespecified addresses. Accordingly, when the probe A|ABCD collects two timestamps, we conclude that A is in alias with B but not with C. Since B and C have been already tested, two new addresses are extracted from γ and prespecified in the next probe (A|ADEF). Note that Pythia is able to infer up to 4 addresses in alias within one probe when four timestamps are collected. These UDP packet probes are sent until all the addresses in γ have been tested against the pivot.

3) Once all the addresses contained in γ have been tested against the pivot, Pythia stores the pivot and all the addresses recognized as in alias with it. These addresses are also removed from α and β. As long as a new pivot is available, i.e. α is not empty, the technique performs a new iteration starting from the first step.

A retransmission mechanism is also adopted to deal with possible rate limiting policies employed by the router owning the pivot address. Pythia is publicly available online[4].

Table 1: Pythia - Inferences and next probes to send according to the timestamps collected with UDP A|ABCD.

Collected Timestamps	Inference	Next Probe
1	A,B not in alias	A|ACDE
2	A,B in alias; A,C not in alias	A|ADEF
3	A,B,C in alias; A,D not in alias	A|AEFG
4	A,B,C,D in alias	A|AEFG

3.4 Advantages and limitations

To the best of our knowledge, Pythia is the only active probing technique in literature potentially able to identify up to four addresses in alias within a single packet probe whereas, to reach the same result, traditional pairwise techniques would require to test six different pairs of addresses[5]. Besides the linear probing complexity of the preliminary step, Pythia requires a single packet probe to infer if two addresses are in alias or not, whereas other pair-wise techniques such as Sherry *et al.* [9] and Ally [27] require at least two and three probes, respectively. Finally, differently from all the other techniques, Pythia is able to tell if a given address B is in alias or not with the pivot even if B does not reply at all to active probing.

On the other hand, Pythia is not free of limitations. The TS option has a strong impact on the router responsiveness [11,20] reducing the set of addresses that could be used as pivot. Furthermore, once selected, a pivot is targeted with multiple packet probes and this may cause the targeted router to be silent to our probes due to the exceeding of specific rate limiting thresholds. Reordering the probes may strongly help in mitigating this limitation. We left this and other optimizations as future work.

4. EXPERIMENTAL EVALUATION

In this section, we describe (i) the experimental campaign we adopted to evaluate Pythia and to compare it with other techniques; (ii) the set of performance metrics we consider in the evaluation; (iii) the main findings for the evaluation phase.

4.1 Methodology

To experimentally evaluate Pythia, we used the information provided by the MERLIN project [28] as a reference: MERLIN natively provides a router-level view of the network by exploiting IGMP probing [29]. Although affected by several limitations such as filtering [30] and the scope

[3]Other non-RFC compliant behaviors exist [11] and should be taken into account when implementing measurement tools based on IP options.

[4]`http://traffic.comics.unina.it/pythia/`

[5]When transitivity closure is not applied.

Table 2: Performance metrics.

Name	*Acronym*	*Formula*	*Description*
Applicability	APP	$\frac{\|D\|}{\|D\|+\|U\|}$	The fraction of pairs on which the alias resolution technique is able to take a decision
Hit Ratio	HR	$\frac{\|TP\|+\|TN\|}{\|D\|+\|U\|}$	The overall fraction of pairs properly aliased or dealiased by the technique
Mismatch Ratio	MR	$\frac{\|FP\|+\|FN\|}{\|D\|+\|U\|}$	The overall fraction of pairs wrongly aliased or dealiased by the technique
Positive Hit Ratio	PHR	$\frac{\|TP\|}{\|TP\|+\|FN\|+\|UP\|}$	The fraction of pairs in alias properly aliased by the technique
Negative Hit Ratio	NHR	$\frac{\|TN\|}{\|TN\|+\|FP\|+\|UN\|}$	The fraction of pairs not in alias properly de-aliased by the technique
Positive Predictive Value	PPV	$\frac{\|TP\|}{\|TP\|+\|FP\|}$	How much can we trust the technique when two addresses are declared as in alias?
Negative Predictive Value	NPV	$\frac{\|TN\|}{\|TN\|+\|FN\|}$	How much can we trust the technique when two addresses are declared as not in alias?

TP: True Positive	FN: False Negative	UP: Unknown Positive	P= TP ∪ FP	U= UP ∪ UN
FP: False Positive	TN: True Negative	UN: Unknown Negative	N= TN ∪ FN	D= P ∪ N

limited to the multicast enabled part of the network [28], the information provided by IGMP probing is typically considered highly accurate and has been already used as a reference in several previous works [7,9]. During a preliminary experimental campaign based on MERLIN, we collected information about 777 Juniper routers[6] located in 12 distinct ASes of different size (tier-1, transit and stub networks). We tested Pythia on Juniper routers because empirical evidences suggest that these devices act as any-interface stamping routers [10, 11]. In this work we aim at understanding if Pythia performs better than the other techniques on this specific set of routers. In particular, we compared Pythia to Palmtree [7] and *Motu* a publicly available tool developed by CAIDA[8] that implements the technique proposed by Sherry *et al.*

From the routers of the reference dataset we extracted 6,503 addresses and applied the following methodology to deal with the quadratic probing and computational complexity of the employed techniques. Each tested technique was evaluated on 100 different chunks. To generate a chunk, we performed three steps: (1) we first randomly selected 10 routers of the reference dataset and extracted all their addresses; (2) from this set of addresses we randomly selected up to 50 IPs; finally (3) we generated all the possible combinations of two addresses starting from the IPs sampled during the previous step. Techniques requiring in input a list of addresses, such as Pythia and Palmtree, were fed with the lists obtained during the second step, while those requiring in input IP pairs (Motu) were fed with the lists obtained at the third step. This two-step sampling process allowed to (i) strongly reduce the time required to obtain the experimental results and (ii) preserve in each chunk a significant number of addresses actually in alias. Finally, since a well-known problem for active probing technique is the dependence of the obtained results on the used vantage point, we tested Pythia and the other techniques from 12 PlanetLab nodes [32].

[6]The *IGMP* probing provides also some indications about the brand of the router. Interested readers may refer to [31] for more details.

[7]http://itom.utdallas.edu/tools.html

[8]http://www.caida.org/tools/measurement/motu/

4.2 Performance Metrics

Properly evaluating and comparing alias resolution techniques is not straightforward. In this work, we adopt multiple performance metrics as explained in the following.

Two given addresses can be classified by a generic alias resolution technique as (i) in alias, (ii) not in alias or (iii) unknown − i.e. they are not-classifiable for some reasons − independently on how the technique works. Accordingly, to compare different techniques tested over the same initial set of addresses, one possibility is to consider all the pairs extracted from this set.

By inspecting the results generated by a specific technique, the set of pairs can be split in three disjoint sets: $P-$ pairs classified as in alias; $N-$ pairs classified as not in alias; $U-$ not-classifiable pairs. By taking into account the reference dataset, these three sets can be respectively split in True Positive (TP) and Negative (TN), False Positive (FN) and Negative (FN), Unkown Positive (UP) and Negative (UN). It follows that $P = TP \cup FP$, $N = TN \cup FN$ and $U = UP \cup UN$. Furthermore, we refer to the set containing all the pairs classified by the technique as the Decision set $D = P \cup N$.

We used these sets to evaluate the tested techniques according to the performance metrics reported in Tab. 2. These metrics allow to estimate the level of applicability (*applicability*), accuracy (*hit and mismatch ratio*) and trustworthiness (*positive and negative predictive value*) for each technique.

4.3 Experimental results

In this section, we present the results obtained with a measurement campaign conducted between the 1^{th} and 14^{th} of May 2013 from 12 PlanetLab nodes. For each vantage point, we created a unique file containing all the pairs probed in the chunks and the corresponding outcomes of the alias resolution technique tested. Since Palmtree cannot directly infer if two given IP addresses are in alias or not, we considered transitivity closure on its results. The final dataset is publicly available online[4]. Fig. 1 reports the distributions of the performance metrics over the vantage points (1a-g) and aggregated statistics (1b).

Applicability. Pythia is able to classify many more pairs than the other tested techniques (Fig. 1a): on average, Pythia, Motu and Palmtree classified one pair for every 2.6, 11.6, 28.6 pairs. Thus, Pythia was 4.5 and 11 times more ap-

plicable than Motu and Palmtree, respectively. This result can be explained by considering that, unlike the other techniques, Pythia is able to classify a pair even if only one of the two addresses replies.

Hit and Mismatch Ratio. Pythia showed a higher hit ratio but also a higher mismatch ratio when compared to the other techniques (Fig. 1c and 1d). However, we registered an absolute gain in hit ratio that is much more significant than the loss we observed in terms of mismatch ratio: by comparing Pythia to Motu (Palmtree), the hit ratio grew on average from 8.5% (3.4%) to 37.8% whereas the mismatch ratio from 0.1% (< 0.1) to 1%.

Positive and Negative Hit Ratio. Considering that the vast majority of the pairs in the dataset consists of addresses not in alias (about 80% of all the pairs), one could imagine that the higher hit ratio of Pythia is determined exclusively by pairs correctly identified as not in alias: this intuition is only partially true. Indeed, both the positive and negative hit ratio for Pythia resulted higher than the other techniques (Fig. 1e and 1f), although the gain was much more significant over the pairs actually not in alias. Experimental results showed that Pythia, Motu and Palmtree were able to correctly identify respectively 57.3%, 47.3% and 19.8% of the pairs actually in alias.

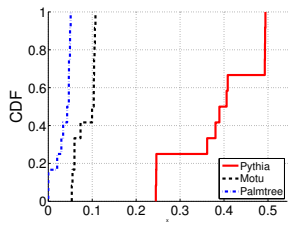
Positive and Negative Predictive Value. Compared to the other techniques, Pythia showed a similar positive predictive value and a much higher negative predictive value. Accordingly, when Pythia declares a pair of addresses as in alias, its level of trustworthiness is comparable to the other techniques. At the same time, when it declares a pair of IPs as not in alias, its level of trustworthiness is three times higher than Motu[9].

To deepen the comparative evaluation of Motu and Pythia, we also performed a per-pair analysis. To this end, we first aggregated the data collected by all the vantage points. In this process, we did not observe conflicts among the decisions taken by the same technique from different vantage points.

Tab. 3 reports the breakdown of the pairs on the decisions taken by Motu and Pythia. Thanks to the information provided by the reference dataset, we split the decision set of each technique in correct and wrong decisions. No decision refers to the pairs declared as not-classifiable by the technique. Globally, correct, wrong and no decisions account for 10.6%, 0.2% and 89.2% for Motu and 48.5%, 1.0% and 50.5% for Pythia, respectively. The subset of pairs classified by both the techniques represent 10.8% of the total in the dataset: when the techniques judged the same pair, they always took the same (mostly correct) decision with very few exceptions. On the other hand, both the techniques were not able to classify more than a half of the total pairs (50.5%). Interestingly, while Motu is not able to provide any additional information about the pairs not classified by Pythia, the latter showed a significant marginal utility when compared to Motu. Indeed, Pythia was able to classify 46.7% of all the pairs not classified by Motu, taking the correct decision in 97.9% of these cases.

This result suggests that for the subset of pairs classifiable by both the techniques, Motu and Pythia are essentially equivalent. However, Pythia was able to take correct decisions on a wide set of pairs not classifiable by Motu.

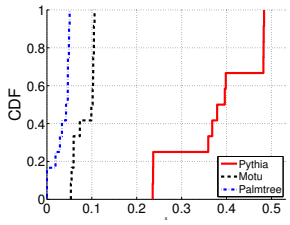
[9]Palmtree does not perform dealiasing.



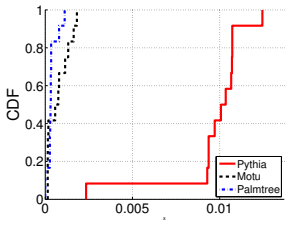
(a) Applicability.

Metric	Pythia	Motu	Palmtree
APP	38.8 (9)	8.6 (2)	3.5 (2)
HR	37.8 (9)	8.5 (2)	3.4 (2)
MR	1.0 (0)	0.1 (0)	0.0 (0)
PHR	57.3 (6)	47.3 (12)	19.2 (10)
NHR	33.6 (10)	0.1 (0)	0.0 (0)
PPV	99.6 (0)	99.8 (0)	98.60 (2)
NPV	96.5 (1)	33.3 (33)	-

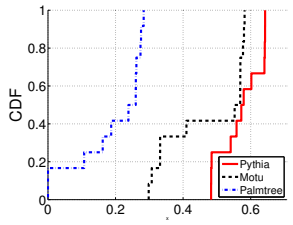
(b) Stats in%: Mean(St Dev).



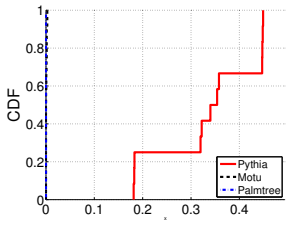
(c) Hit Ratio.



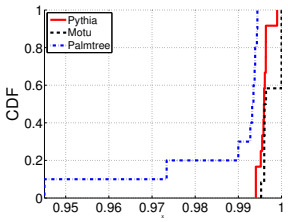
(d) Mismatch Ratio.



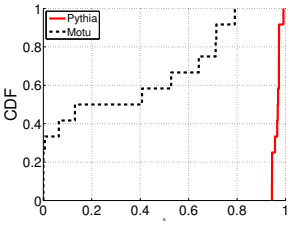
(e) Positive Hit Ratio.



(f) Negative Hit Ratio.



(g) Positive Predictive Value.



(h) Negative Predictive Value.

Figure 1: Performance metric distributions over the vantage points.

5. CONCLUSION

In this work, we presented and experimentally evaluated Pythia, a novel active probing technique for alias resolution. Pythia exploits a combination of IP Prespecified Timestamp option and UDP packet probes to reconstruct a specific category of routers. By using as a reference the highly reliable topological information provided by IGMP probing, we experimentally compared Pythia to Palmtree and a previous technique based on the Timestamp option, according to multiple performance metrics. Experimental results demonstrated that Pythia is applicable on a wider set of addresses mainly because it is potentially able to investigate the alias relation even if not all the considered addresses are responsive. Finally, when taking a decision about two addresses, the level of trustworthiness of Pythia is similar to the other tested techniques when judging addresses actually in alias and much higher for the ones actually not in alias. As future

Table 3: Pythia versus Motu: pair breakdown on classification (%).

PYTHIA \ MOTU	Correct Decision	Wrong Decision	No Decision	
Correct Decision	10.6	0	37.9	→ **48.5**
Wrong Decision	0.001	0.2	0.8	→ **1.0**
No Decision	0.007	0	50.5	→ **50.5**
	↓	↓	↓	
	10.6	**0.2**	**89.2**	

work, we plan to improve Pythia, provide a wider evaluation and deepen the impact of alias resolution on route stability: preliminary results suggest that traceroute may cause the inference of ghost routing changes [33].

6. ACKNOWLEDGMENTS

The work of the authors is partially funded by the PLATINO (PON01_01007) and S^2−MOVE (PON04a3_00058) projects financed by MIUR.

7. REFERENCES

[1] B. Donnet and T. Friedman. Internet topology discovery: a survey. *IEEE Communications Surveys and Tutorials*, 9(4), 2007.

[2] S. Floyd and V. Paxson. Difficulties in simulating the Internet. *IEEE/ACM Trans. Networking*, 9(4):392–403, August 2001.

[3] M. H. Gunes and K. Sarac. Analytical IP alias resolution. In *IEEE ICC*, 2006.

[4] V. Jacobson et al. Traceroute. `ftp://ftp.ee.lbl.gov/traceroute.tar.gz`.

[5] A. Botta, W. De Donato, A. Pescapé, and G. Ventre. Discovering topologies at router level: Part II. In *GLOBECOM'07*, pages 2696–2701. IEEE, 2007.

[6] K. Keys. Internet-scale IP alias resolution techniques. *ACM SIGCOMM CCR*, 40(1):50–55, 2010.

[7] L. Spinelli, M. Crovella, and B. Eriksson. Aliascluster: A lightweight approach to interface disambiguation. In *IEEE International Global Internet Symposium*, 2013.

[8] R. Beverly, W. Brinkmeyer, M. Luckie, and J. P. Rohrer. IPv6 Alias Resolution via Induced Fragmentation. In *PAM*, 2013.

[9] J. Sherry, E. Katz-Bassett, M. Pimenova, H.V. Madhyastha, T. Anderson, and A. Krishnamurthy. Resolving IP aliases with prespecified timestamps. In *ACM SIGCOMM IMC*, 2010.

[10] P. Marchetta, W. de Donato, and A. Pescapé. Detecting third-party addresses in traceroute traces with IP timestamp option. In *PAM*, 2013.

[11] W. de Donato, P. Marchetta, and A. Pescapé. A Hands-on Look at Active Probing using the IP Prespecified Timestamp Option. In *PAM*, 2012.

[12] P. Marchetta and A. Pescape. DRAGO: Detecting, quantifying and locating hidden routers in traceroute IP paths. In *INFOCOM, 2013 Proceedings IEEE*, pages 3237–3242, 2013.

[13] Zaw-Sing Su. Rfc 781: A specification of the internet protocol (IP) timestamp option, May 1981.

[14] K. Keys. Internet-Scale IP Alias Resolution Techniques. *ACM SIGCOMM CCR*, 2010.

[15] J-J. Pansiot and D. Grad. On routes and multicast trees in the Internet. *ACM SIGCOMM CCR*, 1998.

[16] M. Tozal and K. Sarac. Palmtree: An IP alias resolution algorithm with linear probing complexity. *Computer Communications*, 2010.

[17] N. Spring, R. Mahajan, and D. Wetherall. Measuring ISP topologies with Rocketfuel. *ACM SIGCOMM CCR*, 32(4), 2002.

[18] A. Bender, R. Sherwood, and N. Spring. Fixing Ally's growing pains with velocity modeling. In *ACM SIGCOMM IMC*, 2008.

[19] K. Keys, Y. Hyun, M. Luckie, and K. Claffy. Internet-scale IPv4 Alias Resolution with MIDAR. *IEEE/ACM Trans. Networking*, 2013.

[20] R. Fonseca, G. Porter, R. Katz, S. Shenker, and I. Stoica. IP options are not an option. *Tech. Rep. Univ. of California*, 2005.

[21] P. Fransson and A. Jonsson. End-to-end measurements on performance penalties of ipv4 options. In *IEEE GLOBECOM*, 2004.

[22] R. Sherwood, A. Bender, and N. Spring. Discarte: a disjunctive internet cartographer. *ACM SIGCOMM CCR*, 38(4), 2008.

[23] E. Katz-Bassett, H.V. Madhyastha, V.K. Adhikari, C. Scott, J. Sherry, P. Van Wesep, T. Anderson, and A. Krishnamurthy. Reverse traceroute. In *USENIX NSDI*, 2010.

[24] A.D. Ferguson and R. Fonseca. Inferring router statistics with IP timestamps. In *ACM CoNEXT Student Workshop*, 2010.

[25] T. Flach, E. Katz-Bassett, and R. Govindan. Quantifying violations of destination-based forwarding on the Internet. In *ACM SIGCOMM IMC*, 2012.

[26] Alistair K. Tsps investigations, caida report. `http://www.caida.org/~alistair/projects/tsps-investigations.html`.

[27] Neil Spring, Ratul Mahajan, David Wetherall, and Thomas Anderson. Measuring ISP topologies with Rocketfuel. *IEEE/ACM Trans. Netw.*, 12(1):2–16, February 2004.

[28] P. Marchetta, P. Mérindol, B. Donnet, A. Pescapè, and J-J. Pansiot. Topology discovery at the router level: A new hybrid tool targeting ISP networks. *JSAC*, 29(9):1776–1787, 2011.

[29] Pusateri. Distance vector multicast routing protocol version 3 (dvmrp). *IETF, Internet Draft (Work in Progress) draft-ietf-idmr-dvmrp-v3-11*, 2003.

[30] P. Marchetta, P. Mérindol, B. Donnet, A. Pescapé, and J-J. Pansiot. Quantifying and mitigating IGMP filtering in topology discovery. In *IEEE GLOBECOM*, 2012.

[31] P. Mérindol, B. Donnet, J.-J. Pansiot, M. Luckie, and Y. Hyun. MERLIN: MEasure the Router Level of the INternet. In *NGI*, 2011.

[32] Andy Bavier et al. Operating system support for planetary-scale network services. In *NSDI*, 2004.

[33] P. Marchetta, V. Persico, E. Katz-Bassett, and A. Pescapé. Don't trust traceroute (completely). In *ACM CoNEXT Student workshop*, 2013.