A Hands-on Look at Active Probing using the IP Prespecified Timestamp Option

Walter de Donato, Pietro Marchetta, and Antonio Pescapé

Department of Computer Engineering and Systems, University of Napoli Federico II
{walter.dedonato,pietro.marchetta,pescape}@unina.it

Abstract. In the last years, network measurements have shown a growing interest in active probing techniques. Recent works propose approaches based on the IP prespecified timestamp option and consider its support to be enough for their purposes. On the other hand, other works found that IP options are usually filtered, poorly implemented, or not widely supported. In this paper, to shed light on this controversial topic, we investigate the responsiveness obtained targeting more than $1.7M$ IPs using several probes (ICMP, UDP, TCP, and SKIP), with and without the IP prespecified timestamp option. Our results show that: (i) the option has a significant impact on the responsiveness to the probes; (ii) a not−negligible amount of targeted addresses return several categories of non RFC−compliant replies; (iii) by considering only the RFC−compliant replies which preserve the option, the probes ranking by responsiveness considerably changes. Finally, we discuss the large−scale applicability of two proposed techniques based on the IP prespecified timestamp option.

Keywords: Internet measurements, Active probing, IP options

1 Introduction

The Internet Protocol version 4 (IPv4), after more than three decades and several minor updates, still represents the core of the Internet and many protocols and services have been built on top of it. IPv4 has provision for optional header fields in order to transport additional information. Particularly, the *Timestamp* (TS) optional header (IP option type 68) is defined along with three variants: (i) each router forwarding the packet, if enough space is available, should add a timestamp; (ii) a $(IP, timestamp)$ couple should be added; (iii) the sender requires a timestamp for up to four "prespecified" IPs [1, 2]. We refer to them as TS_o, TS_i, and TS_p respectively. Since recent works [3–5] reconsidered the utility of TS_p, in this paper we focus our attention on such variant.

Works proposing applications based on TS_p consider its support to be enough for their purposes [3,4]. On the other hand, previous works stated that IP options are usually filtered, poorly implemented, or not widely supported [6, 7].

To the best of our knowledge, both claims have not been properly supported by a large scale analysis comprising a set of destinations statistically significative.

Moreover, previous analysis only considered TCP$_{syn}$ and ICMP$^{echo}_{request}$ probes, thus not considering other possibilities to obtain a reply from a targeted destination.

In this paper, we present a detailed analysis of the TS$_p$ support in Internet obtained by targeting more than $1.7M$ destinations from two vantage points (VPs). For the sake of completeness, we employ four different probes (ICMP, UDP, TCP and SKIP), with and without the TS$_p$ option set. Such analysis allowed us to evaluate the impact of TS$_p$ on the responsiveness to each probe and to investigate the RFC−compliance of different IP stack implementations.

The paper is organized as follows. While in Sec. 2 we discuss the most important related works, in Sec. 3 we briefly describe the background and the adopted methodology. Sec. 4 contains the results of our large−scale measurement campaign. In Sec. 5 we briefly discuss the impact of our findings on some TS$_p$-based applications. Finally, Sec. 6 ends the paper with conclusion remarks.

2 Related Work

Gunes *et al.* [8] conducted an experimental study of both historical and current responsiveness to probes concluding that the most effective is ICMP, followed by TCP and UDP. They also found a higher responsiveness of network devices to *indirect probes* (i.e. probes launched towards other destinations). Our work has a different goal: while the overall responsiveness is a well investigated topic, we aim at measuring the impact of the TS$_p$ option on the responsiveness to several probes. Fonseca *et al.* [7], using Planetlab, estimated the *transit* filtering of packets crafted with and without TS and *Record Route* (RR) options by using a modified version of traceroute based on ICMP probes. They demonstrated, over a $7.5k$ IPs dataset, that transit filtering is mainly concentrated in a minority of edge ASes. In [6] Medina *et al.* covered the impact of TS and RR options on TCP by analyzing connections towards 500 web servers. Our work extends both analyses to $1.7M$ IPs and to probes other than ICMP and TCP, in order to estimate the overall utility in using TS$_p$ probes, taking into account the effect of transit filtering by using two not−filtered VPs. Sherry *et al.* [3] proposed a novel alias resolution approach based on the TS$_p$ option as well as a measurement study of its support. The latter made use of ICMP$^{echo}_{request}$ probes to target around $267.7k$ destinations. Our work extends such study targeting with several probes more than $1.7M$ destinations in order to globally estimate the impact and the support of the TS$_p$ option as well as the RFC compliance. Our results and hypothesis experimentally justify part of the findings detailed in [3]. Finally, the TS$_p$ option has been recently exploited in the reverse traceroute [4] and to infer router statistics [5]. We evaluate the applicability on large scale scenario of [3] and [4] in the light of the obtained results.

3 Background and Methodology

When using TS$_p$, the originating host composes the option data with a maximum of four $(IP, 0)$ records and sets the pointer field for pointing to the first record.

For instance, a forwarding router should stamp the pointed record only if it contains its own IP address. In such case, the pointer should be incremented to point to the next record. If the router cannot register timestamps due to lack of space, the overflow field should be incremented. The timestamp value should be inserted in a *standard* format, which represents the elapsed time in milliseconds since midnight UT. If such format is not respected the high order bit should be set to one, indicating the use of a *non−standard* value.

In order to estimate its impact on the responsiveness to the probes, a list of addresses is queried with a set of probes crafted with and without TS_p option. The list is extracted from a complete Archipelago [9] *cycle* and filtered to remove non−publicly routable addresses (e.g. 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16, ...). We classify each IP from the list as *Pathending*, if it appears in the Archipelago dataset exclusively as a traceroute destination, and as *Router* otherwise. It is worth to notice that the *Router* set surely contains IP addresses belonging to network devices, while an unknown percentage of *Pathending* IPs consists of end hosts. Each address is then solicited with the following probes: *(a)* $\text{ICMP}^{echo}_{request}$; *(b)* UDP towards a presumably unused port (15616), to collect an $\text{ICMP}^{port}_{unreach}$ message; *(c)* TCP towards an unassigned well-known port (737), to solicit a TCP *reset* reply; *(d)* an IP packet carrying a SKIP message (an obsolete protocol), to solicit an $\text{ICMP}^{proto}_{unreach}$ message. We chose SKIP after a preliminary test demonstrated how unassigned protocol numbers obtain much less answers.

In line with [3], we use the TS_p option according to the (A|BBBB) format with A=B (A represents the destination address and BBBB the ordered list of prespecified IPs). In the following, we refer to the probes with TS_p option respectively as ICMP_p, UDP_p, TCP_p, and SKIP_p. When using ICMP_p and TCP_p probes, the returned option (if present) is extracted from the IP layer of the reply packet, while, regarding UDP_p and SKIP_p, it is extracted from the original probe carried back by the ICMP error packet. A retransmission mechanism allows to deal with potential congestion events and rate limiting policies: before giving up each probe is sent four times with a timeout of two seconds. During a preliminary test, we found that some destinations not always stamp the option. We call such phenomenon *timestamp rate limiting*. In order to deal with it, we apply the retransmission mechanism also when the returned option records are empty.

4 Experimental results

In this section, we present the results obtained with a measurement campaign conducted between the 16^{th} and 20^{th} of June 2011 from two VPs located in Napoli, Italy (NA) and Louvain−la−Neuve, Belgium (LLN)[1]. The collected dataset is freely available online[2]. In a preliminary campaign we also employed 10 Planetlab VPs, which we decided to discard because they do not support the SKIP protocol[3] and their access networks often filter probes with TS_p option.

[1] The authors would like to thank B. Donnet and P. Mérindol for their support.

[2] `http://www.grid.unina.it/Traffic/Data/TSp_16-20_June_2011.tar.gz`.

[3] Planetlab nodes currently support TCP, UDP, ICMP, GRE and PPTP protocols [10].

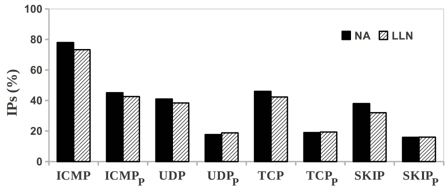


Fig. 1: Responsiveness to the probes per vantage point

After removing non−publicly routable addresses (1.4%), 1, 776, 095 destinations were extracted from the Archipelago's cycle started on the 13^{th} of June 2011. The obtained IPs resulted to be equally divided into *Pathending* (49.99%) and *Router* (50.01%).

All the results from the two VPs are very similar: for instance, as reported in Fig.1, the responsiveness to each probe is consistent between them. Therefore, given such consistency and for space constraints, in the following we discuss the results of the VP located in Napoli.

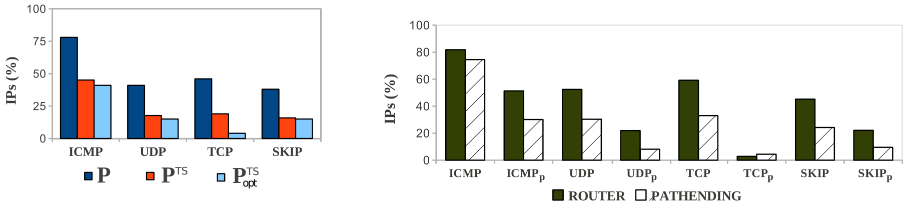
About 19% of the destinations were unresponsive to our probes, while a small portion (2.25%) returned non RFC−compliant replies (we call them *anomalies*). Hence, disregarding the anomalies, we first quantify the support of TS_p and its impact on the responsiveness to the probes. Then, we deeply investigate and characterize the isolated non RFC−compliant behaviors.

4.1 Support analysis

Responsiveness In Fig.2(a) the amount of destinations responsive to probes without option (P) is compared with the amount of them replying when TS_p is enabled by preserving the option (P^{TS}_{opt}) or regardless of this (P^{TS}).

In line with [8], the most effective probe without option is ICMP (78.1%) followed by TCP (46.1%), UDP (41.4%) and SKIP (34.7%). The insertion of TS_p heavily impacts the responsiveness to each probe (−33% ICMP, −24% UDP, −28% TCP, −19% SKIP), but preserves the ranking order. However, applications relying on TS_p generally require the reply to preserve the option and the ranking significantly changes when considering only such replies: ICMP_p (40.7%), SKIP_p (15.8%), UDP_p (15%) and TCP_p (3.6%). It is worth to notice how most replies to TCP_p probes were received without option, while this effect is marginal for the other probes. Moreover, as shown in Fig.2(b), *Router* IPs resulted more responsive than *Pathending* ones for all the probes, with the only exception of TCP_p. In the rest of the paper, all the replies not preserving the TS_p option will not be taken into account.

Tab. 1(a) and 1(b) show the relation among different probes with respect to the responsiveness. Each element (i, j) represents the percentage of destinations



(a) impact of the TS_p option

(b) breakdown on *Router* and *Pathending* IPs

Fig. 2: Responsiveness to the probes

Table 1: Responsiveness relation among different probes

(a) without TS_p option (%)

	ICMP	UDP	TCP	SKIP
ICMP	**78.1**	40.6	44.9	32.6
UDP	40.6	**41.4**	37.6	30.1
TCP	44.9	37.6	**46.1**	28.9
SKIP	32.6	30.1	28.9	**34.7**

(b) with TS_p option (%)

	ICMP_p	UDP_p	TCP_p	SKIP_p
ICMP_p	**40.7**	13.2	3.5	13.5
UDP_p	13.2	**15.0**	3.2	11.6
TCP_p	3.5	3.2	**3.6**	2.6
SKIP_p	13.5	11.6	2.6	**15.8**

responsive to both the probes on the i^{th} row and j^{th} column. The main diagonal, therefore, points out the amount of destinations which responded to a specific probe. Without option, ICMP probes showed a significant marginal utility compared to the others. Anyway, even the other probes showed some marginal utility compared to ICMP: UDP (0.8%), TCP (1.2%) and SKIP (2.1%). When TS_p is enabled the scenario is similar: while only part of the IPs which replied to ICMP_p also provided replies to the other probes, UDP_p and SKIP_p collected answers respectively from $31k$ and $41k$ addresses which were unresponsive to ICMP_p.

Table 2: Timestamp rate limiting phenomenon

	D_0	$\mathrm{D}_0 \cap \mathrm{D}_1$	$\mathrm{D}_0 \cap \mathrm{D}_2$	$\mathrm{D}_0 \cap \mathrm{D}_3$	$\mathrm{D}_0 \cap \mathrm{D}_4$
ICMP_p	98,024	2,443	299	0	0
UDP_p	54,649	643	0	0	0
TCP_p	420	0	0	0	0
SKIP_p	56,213	646	0	0	0

Option Management Henceforth we use the following notation: the term D_j represents the set of destinations which respond to the generic TS_p probe by stamping the prespecified address j times, while the D_j^{probe} notation refers to a specific probe. For instance, D_1^{icmp} is the set of destinations which, solicited with ICMP_p probes, returned replies containing only one stamped record.

Regarding the *timestamp rate limiting* (see Sec. 3), Tab.2 reports for each probe the number of destinations classified both as D_0^{probe} and D_{1-4}^{probe}. This behavior mostly involved *Router* IPs probed with ICMP_p. To handle such phenomenon in the next analyses, we reassigned the involved destinations using the following criterion: an address belonging to both D_0^{probe} and D_j^{probe} is removed from D_0^{probe} to be exclusively part of D_j^{probe}. This process leads to the results reported in Tab.3(a), where the number of stamps per probe is pointed out as percentage of the responsive destinations.

Tab.3(a) suggests the rule followed by most devices to manage TS_p: *the option is stamped once every time the probe passes through the interface associated to the currently pointed prespecified address.* Since UDP_p and SKIP_p probes, unlike ICMP_p, return the option as affected by the forward path only, the similarity among $\mathrm{D}_1^{icmp} \bigcup \mathrm{D}_2^{icmp}$, D_1^{udp} and D_1^{skip} supports such hypotesis. Tab.3(b), in which

Table 3: Deep analysis of the returned TS_p options

(a) breakdown of the replies on the probes(%)

	TOT	D_0	D_1	D_2	D_3	D_4
ICMP_p	$723k$	13.2	$\underline{26.4}$	$\underline{54.9}$	∼0	5.5
UDP_p	$267k$	20.2	$\underline{74.5}$	0.1	0	5.1
TCP_p	$620k$	0.7	∼0	99.3	∼0	∼0
SKIP_p	$281k$	19.8	$\underline{80.1}$	0.1	0	∼0

(b) intersection between D_i^{icmp} and D_j^{udp}

		j=0	j=1	j=2	j=3	j=4
	TOT	$54k$	$198.9k$	246	-	$13.7k$
i=0	$95.3k$	$\mathbf{27.9k}$	306	-	-	-
i=1	$190.8k$	$12.1k$	$\mathbf{32.8k}$	112	-	-
i=2	$397.5k$	519	$\underline{147.8k}$	**92**	-	-
i=3	168	6	2	19	**-**	-
i=4	$39.6k$	2	2	5	-	$\mathbf{13.2k}$

the (i, j) element represents the size of $\mathrm{D}_i^{icmp} \bigcap \mathrm{D}_j^{udp}$, deeper investigates such scenario: the big intersection between D_2^{icmp} and D_1^{udp} ($147.8k$) confirms again our hypothesis. Hence, if a (D|DDDD) probe enters and leaves the destination node through the same interface D, the option is stamped twice, otherwise just once. As we will discuss in Sec. 5, such behavior may reduce the applicability of the technique proposed in [3].

We also investigated the small amount of destinations not respecting the previous rule. Analyzing D_3^{icmp}, we often observed records containing timestamps according to the $t_1 t_1 t_2$ pattern, with t_2 slightly higher than t_1. On the other side, regarding D_2^{udp} we found $t_1 t_1$ patterns, which suggests that the option is stamped twice when entering the node, but only once when leaving it. We deepened the analysis of D_4^{icmp} and D_4^{udp} by using IGMP probes with the MERLIN [11] platform. We only received replies from Juniper routers[4], while doing the same on D_1^{icmp} gave no replies. Moreover, we never observed Cisco routers stamping the option more than twice. Hence, we foresee novel fingerprinting and alias resolution techniques relying on how TS_p is managed.

4.2 RFC compliance analysis

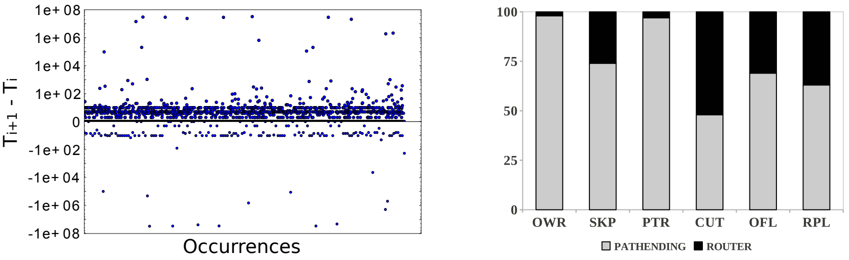
Timestamp Format According to the RFC 791, a standard timestamp should always be lower than $86.4 * 10^5$ $(24h * 3600s * 1000)$, while a non−standard value should belong to the range $[2^{31}, 2^{32}]$. Hence, the range $]86.4 * 10^5, 2^{31}[$ consists of non RFC−compliant values. Among the $660k$ destinations stamping at least once, we found timestamp values according to the following distribution: 87.6% standard, 11.3% non−standard, 1.15% non RFC−compliant. We also found 449 destinations stamping different probes using different formats and 9 of them doing it inside the same answer.

Focusing our attention on the standard values, we analyzed the difference between contiguous not null timestamps from the same reply. Fig.3(a) shows such values for ICMP_p replies, where we identify three cases: *(i)* small positive and *(ii)* negative differences, *(iii)* both positive and negative huge differences. According to the rule described in Sec.4.1, the first case represents an estimation of the reply−generation delay on the destination node. Although limited by the milliseconds resolution, such estimation may represent a valid alternative to classic techniques based on round−trip time. The second case corresponds to transient anomalies which quickly disappeared. The third case represents a persistent behavior we observed on just 38 destinations, which seem to stamp the option by using two different clocks. Since such replies contain four timestamps following the $t_1 t_2 t_2 t_3$ pattern, where often $t_1 \approx t_3$, we speculate the presence of a middlebox along the path which is responsible of inserting t_1 and t_3.

Anomalies Disregarding timestamp values, 40013 targeted destinations provided non RFC−compliant replies, which lead us to the following taxonomy:

- **OWR**: some prespecified IP addresses are overwritten;

[4] DVMRP [12] codes $3.x$ are commonly associated to Juniper, while $12.x$ to Cisco.



(a) difference of timestamps in contiguous TS_p records

(b) sources of the anomalies

Fig. 3: Non RFC−compliant behaviors

- **SKP**: the destination stamps the option by skipping one or more records (e.g. the second IP is stamped, but not the first one);
- **PTR**: the pointer field is inconsistent with respect to the number of timestamps;
- **CUT**: the original packet carried by the ICMP error message is truncated before the end of the option;
- **OFL**: the overflow field counts several *extra−stamps*, but the number of timestamps is less than four;
- **RPL**: the option data is replaced with part of the original packet header.

It is worth to notice that the CUT anomaly is different from a missing option. Indeed, the IHL field of the IP header carried back by the ICMP error message is 15 in the first case, pointing out the presence of the option, and 5 in the latter case. Regarding the PTR anomaly, the pointer value should belong to the set $\{5, 13, 21, 29, 37\}$, but we also found non−standard values: *253* and *9*.

Fig.3(b) shows that most anomalies, with the exception of CUT, were generated by destinations belonging to the *Pathending* set. Although such set may also contain IPs assigned to routers, the phenomenon seems reasonably related to end hosts having buggy TCP/IP stack implementations. As shown in Tab.4(a), which underlines the relationships among different anomalies, a destination can generate replies affected by more than one of them. The (i, j) element represents the amount of destinations (as percentage of 40013) affected by both the anomalies on the i^{th} row and the j^{th} column. OWR (73.3%) and PTR (74.8%) are the most common anomalies and appear simultaneously in most cases. As expected, since in the RPL anomaly part of the original packet is copied over the option data without modifying the pointer, such anomaly implies OWR and PTR ones. Moreover, all the IPs affected by the SKP anomaly are also source of the PTR anomaly. Finally, part of the addresses providing CUT replies to UDP_p and $SKIP_p$ probes also generated different anomalies when answering to $ICMP_p$ and TCP_p probes. Such behavior is more evident by looking at Tab. 4(b), which

Table 4: Detailed analysis of anomalies

(a) relation among anomalies (%)

	OWR	SKP	PTR	CUT	OFL	RPL
OWR	**73.32**	0.03	63.60	0.02	0.02	0.96
SKP	0.03	**0.08**	0.08	-	0.04	-
PTR	63.60	0.08	**74.84**	0.02	0.04	0.96
CUT	0.02	-	0.02	**15.47**	-	0.01
OFL	0.02	0.04	0.04	-	**0.06**	-
RPL	0.96	-	0.96	0.01	-	**0.96**

(b) breakdown of anomalies on the probes

	TOT	$ICMP_p$	UDP_p	TCP_p	$SKIP_p$
OWR	29.3k	24.8k	293	3.8k	3.7k
SKP	32	28	4	2	3
PTR	29.9k	28.5k	725	3	4.5k
CUT	6.2k	–	5.6k	–	3.5k
OFL	26	26	–	6	–
RPL	383	–	249	1	287

shows how a specific anomaly relates to the different probes: while CUT and RPL only affect UDP$_p$ and SKIP$_p$, all the other anomalies mainly affect ICMP$_p$, which results to be the most affected probe.

Deepening OWR anomaly Regarding the OWR anomaly, we found prespecified IPs overwritten in different ways, which we discuss below.

The 85% of IPs generating OWR anomalies returned replies in which only the first IP address is overwritten. We further divide them in two cases: (a) 99.7% not stamping any address, (b) 0.3% stamping at least the first IP. The case a mostly involves *Pathending* destinations which failed to properly stamp the current option record by writing the timestamp in the location reserved to the address. Such hypothesis is confirmed by several findings: the returned option has always the pointer set to 13, meaning stamped once; by swapping the first prespecified IP with one not on the path towards the destination, the anomaly disappears: the first prespecified IP is not overwritten and the option is not stamped at all, as confirmed by the pointer value. The case b reveals the presence of network devices confusing TS$_p$ with the TS$_i$ option variant on the path to the destination, since the first TS$_p$ record is filled with both the IP address and the timestamp of such device and the pointer is properly incremented. To better understand such behavior we targeted the same destinations by using TTL limited TS$_p$ probes, in order to reach only the indicted device. As expected, the anomaly appeared a few hops before reaching the destination.

Another 13% of IPs reset part of the prespecified addresses when replying to TCP$_p$ probes. In order to identify the sources of such anomalies, we targeted such destinations using again the MERLIN platform. All the IGMP replies returned the following DVMRP codes: 37.90 and 21.95. Hence, we tried to detect a possible association between such codes and a specific brand/OS by targeting the same destinations with the *nmap* tool [13][5]. We found a highly probable association with Microsoft Windows versions: code 37.90 should correspond to version 2003, while 21.95 to version 2000.

The remaining 2% of destinations mixed the previously described behaviors.

Deepening RPL anomaly While RPL replies normally return already known information, for a specific destination we observed a peculiar behavior which may cause security concerns: probed several times with UDP$_p$, the option data appeared replaced each time in a different way. We identified such replacements as packet headers presumably stored in a dynamic buffer at the destination. In this way, we were able to collect remote MAC and IP addresses, mostly coming from ARP requests. Unfortunately, common OS fingerprinting techniques were not able to discover more information about such device.

5 Applicability of TS$_p$−based techniques

The results reported in Sec.4 allow a general discussion about the recently introduced techniques based on TS$_p$.

[5] Since nmap OS fingerprinting consists in an aggressive probing process, we limited its use only to specific cases involving a reduced amount of IPs.

Reverse traceroute [4], when the RR option is unable to discover the next hop, takes advantage of TS_p in two different ways. In the first case, a candidate IP R − extracted from pre-collected topology information − is prespecified in ICMP_p probes from S using the (D|DR) format, where D is the last discovered hop on the reverse path. In the second case, in order to avoid transit filtering, a spoofed ICMP_p probe is sent, using the (D|R) format, from a selected VP to D acting as S. In both cases, if S receives a reply in which R is stamped, such address is part of the path from D to S. Based on our results, 40.7% of destinations answered to ICMP_p preserving the option, but only 86.8% of them stamped the option. Thus, such approach works with about 35% of IPs from our dataset. Moreover, if R itself belongs to D_0^{icmp} (i.e. 13.2% of IPs), the spoofing approach is not effective.

The alias resolution technique proposed in [3] relies on TS_p as described in the following: for each pair (A, B) of candidate IPs, two ICMP_p probes having (A|ABAB) and (B|BABA) format are sent respectively towards A and B. If both probes obtain replies stamped four times, A and B are alias. According to the rule defined in Sec.4.1, a D_2^{icmp} router stamps twice the (A|ABAB) probe only if the packet enters the node from interface A and exits from interface B and the same happens for the (B|BABA) probe by inverting the crossing order. This explains why in [3] they often obtain replies stamped twice for the first probe and without stamps for the second, which they partially recover exploiting topological constraints. However, they state to obtain much more success in identifying alias pairs for D_4^{icmp} addresses than for D_2^{icmp} ones. Our results confirm that the aliasing technique works well with D_4^{icmp} destinations (2.2%), and demonstrate that D_2^{icmp} IPs (22.3%) are not compliant with the technique, while D_1^{icmp} destinations (10.7%) support it[6]. Hence, from a single VP, the aliasing approach works on 12.9% of cases. Despite the relatively lower amount of collected replies, UDP_p and SKIP_p may represent a valid alternative to implement a similar technique, since they are not affected by the reverse path.

6 Conclusion

Targeting more than 1.7M destinations with a set of probes crafted with and without the TS_p option, we draw the following conclusions: (i) the TS_p option has an important impact on the responsiveness to the probes (−33% ICMP, −24% UDP, −28% TCP, −19% SKIP); (ii) by considering just the replies preserving the option, as required by most applications, the probes ranking by responsiveness considerably changes (ICMP 40.7%, SKIP 15.8%, UDP 15%, TCP 3.6%); (iii) a limited amount of destinations not always stamp (*timestamp rate limiting*); (iv) the option is commonly stamped once every time the packet passes through the interface associated to the currently pointed prespecified IP; (v) around 2.25% of destinations showed non RFC−compliant behaviors classifiable in six non−disjoint categories, while about $7.6k$ IPs made use of timestamp values not allowed by the RFC. In the light of our findings, we evaluated the large−scale applicability of recent proposals based on the TS_p option, demonstrating that,

[6] Such percentage may significantly increase by using multiple VPs.

from a single VP, the alias resolution technique [3] is effective just on 12.9% of destinations, while the reverse traceroute [4] can potentially work on 35% of IPs when the TS_p option is required.

In the future, we plan to (i) further investigate the TS_p option support per Autonomous System by exploiting more unfiltered VPs from the BISmark platform [14] and to propose novel measurement techniques based on it; (ii) exploit the TS_p option in active probing approaches for the monitoring of Internet Outages [15].

References

1. Su, Z.S.: Rfc 781: A specification of the internet protocol (ip) timestamp option (May 1981)
2. Postel, J.: Internet Protocol. RFC 791 (Standard) (September 1981)
3. Sherry, J., Katz-Bassett, E., Pimenova, M., Madhyastha, H.V., Anderson, T., Krishnamurthy, A.: Resolving ip aliases with prespecified timestamps. In Proc. of IMC '10, New York, NY, USA, ACM (2010) 172–178
4. Katz-Bassett, E., Madhyastha, H.V., Adhikari, V.K., Scott, C., Sherry, J., van Wesep, P., Anderson, T.E., Krishnamurthy, A.: Reverse traceroute. In Proc. of NSDI'10, USENIX (2010) 219–234
5. Ferguson, A.D., Fonseca, R.: Inferring router statistics with ip timestamps. In Proc. of CoNEXT'10 Student Workshop, New York, NY, USA, ACM (2010)
6. Medina, A., Allman, M., Floyd, S.: Measuring the evolution of transport protocols in the internet. SIGCOMM Comput. Commun. Rev. **35** (April 2005) 37–52
7. Fonseca, R., Porter, G.M., Katz, R.H., Shenker, S., Stoica, I.: Ip options are not an option. Technical report (2005)
8. Gunes, M.H., Sarac, K.: Analyzing router responsiveness to active measurement probes. In Proc. of PAM '09, Berlin, Heidelberg, Springer-Verlag (2009) 23–32
9. Claffy, K., Hyun, Y., Keys, K., Fomenkov, M., Krioukov, D.: Internet mapping: From art to science. In Proc. of CATCH'09, Washington, DC, USA, IEEE Computer Society (2009) 205–211
10. Huang, M.: VNET: PlanetLab Virtualized Network Access. Technical Report PDN–05–029, PlanetLab Consortium (June 2005)
11. Marchetta, P., Mérindol, P., Donnet, B., Pescapé, A., Pansiot, J.J.: Topology discovery at the router level: a new hybrid tool targeting ISP networks. IEEE JSAC, Special Issue on Measurement of Internet Topologies **29**(6) (October 2011)
12. Pusateri, T.: Distance vector multicast routing protocol version 3 (DVMRP). Internet Draft (Work in Progress) draft-ietf-idmr-dvmrp-v3-11, Internet Engineering Task Force (October 2003)
13. Lyon, G.F.: Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning. Insecure, USA (2009)
14. Sundaresan, S., de Donato, W., Feamster, N., Teixeira, R., Crawford, S., Pescapé, A.: Broadband Internet Performance: A View From the Gateway. In Proc. of SIGCOMM'11, ACM (2011)
15. Dainotti, A., Squarcella, C., Aben, E., Claffy, K.C., Chiesa, M., Russo, M., Pescapé, A.: Analysis of Country-wide Internet Outages Caused by Censorship. In Proc. of IMC'11, Berlin, Germany, ACM (November 2011)