Experimenting with Alternative Path Tracing Solutions

Pietro Marchetta, Walter de Donato, Valerio Persico, Antonio Pescapé
University of Napoli Federico II (Italy)
Email: {pietro.marchetta,walter.dedonato,valerio.persico,pescape}@unina.it

***Abstract*—Tracing Internet paths is essential for gathering knowledge about the complex, heterogeneous, highly dynamic, and largely opaque eco-system of networks the Internet is. Currently, only two practical solutions are available: (i) equipping packets with the Record Route IP option to register addresses of the traversed routers; (ii) eliciting ICMP Time Exceeded messages by limiting the Time-to-Live of the injected packets. In this paper, we investigate three alternative path tracing solutions eliciting ICMP Parameter Problem (PP) messages from the network through the injection of malformed packets. After having introduced them, we describe the experimental results of a first campaign aiming at evaluating their ability to collect replies from the traversed routers. Finally, thanks to a large-scale multi-vantage points measurement campaign, we evaluate the ability of the most promising ICMP PP-based solution to discover interfaces and routers not discovered by Paris-Traceroute Multipath Detection Algorithm (MDA). Experimental results (a) confirm the ability of this novel path tracing solution to report interfaces and routers that are not reported by the state of the art tools and also (b) uncover the scenarios in which this new solution appears more helpful.**

I. Introduction

Measuring network paths is important for monitoring, managing, or troubleshooting the Internet. More in general, tracing Internet paths proved to be extremely helpful for increasing our understanding of this highly dynamic and largely opaque ecosystem of networks. However, this operation is strongly complicated by the lack of (i) a standardized approach and (ii) access or control over the global infrastructure determined by the radically distributed ownership of the Internet among its constituent parts (i.e., the Autonomous Systems – ASes).

Currently, only two practical approaches are available for tracing paths: (i) equipping packets with the Record Route IP option to register addresses of the traversed routers; (ii) using the Van Jacobson's Traceroute limiting the Time-to-Live (TTL) of the injected packets to elicit ICMP Time Exceeded messages. Both these solutions suffer from well known limitations: the Record Route option can register no more than nine addresses; Traceroute may provide incomplete and inaccurate information due to a number of reasons such as unresponsive [9] or hidden [4] routers, filtering and rate-limiting, per-packet or uneven load-balancing [1], RFC1812-fully compliant routers [18], [20], or TTL reset [24].

Several optimizations and variants of Traceroute have been proposed over the years (i) to circumvent filtering policies with different types of packets [16]; (ii) to reduce path measurement duration or probing overhead [3], [5], [24]; (iii) to improve the path coverage [27]; (iv) to trace all the paths to the destination in case of load balancing [1]. While current implementations of Traceroute are more robust, accurate and efficient than the original version proposed by Van Jacobson [1], [12], [15], [24], the very basic mechanism – i.e. limiting the TTL of the injected packets – remained essentially unchanged since its first introduction in 1989.

In this work, we explored three novel path tracing solutions based on ICMP Parameter Problem (PP) messages, thus being totally alternative to the TTL-based mechanism. To the best of our knowledge, very few previous works took advantage of ICMP PP messages [22], [25] and ICMP PP was never adopted for tracing Internet paths. We present (Sec. 2) and evaluate (Sec. 3.1) three novel path tracing solutions which inject malformed packets to elicit ICMP PP messages from the traversed routers, rather than relying on TTL expiration. Then (Sec. 3.2), we further analyse – thanks to a large-scale multi-vantage point experimental campaign – the most promising solution. Experimental results show that relying on ICMP PP has the ability to report interfaces and routers that are not reported by the TTL-based approaches, and also uncover the scenarios in which this alternative solution appears more helpful.

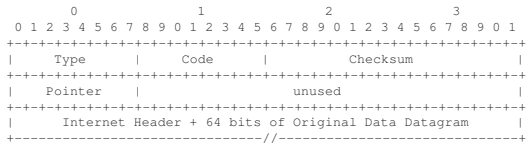
II. Novel path tracing solutions

The novel path tracing solutions explored in this paper take advantage of the Record Route (RR) and Timestamp (TS) IP options to elicit ICMP PP replies. In this section, after recalling the format of these headers and the error conditions determining the generation of ICMP PP messages (Sec. II-A), we detail three novel solutions for tracing Internet paths (Sec. II-B). A prototype written in python implementing these solutions is publicly available to foster other researchers to experiment with them (http://traffic.comics.unina.it/pptr/).

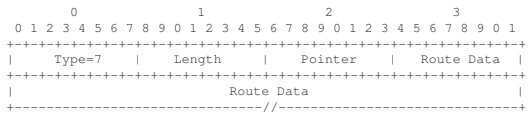
A. ICMP Parameter Problem and IP options

ICMP Parameter Problem - RFC792, RFC1812, RFC1122. According to the standards, when an incoming packet has to be discarded and no other ICMP message covers the detected problem, a router (host) *must* (*should*) send a notification to the source by using an ICMP PP message.

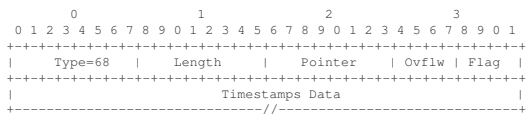
Fig. 1a reports the ICMP PP format: the *type* field is set to 12 while the *code* field can vary among 0 (invalid IP header), 1 (a required option is missing), and 2 (bad length). When code is 0, the pointer field identifies the octet where the error



(a) ICMP Parameter Problem.



(b) IP Record Route option.



(c) IP Timestamp option.

Fig. 1. Format of the headers used within the proposed approach.

occurred: indeed, as usual in case of ICMP error messages, part of the original datagram which caused the error is carried back as payload.

Record Route option - RFC791. The RR option (type 7) provides a way to record the route traversed by a datagram towards its destination. Its format is reported in Fig. 1b. The *length* field counts the option size, while the *pointer* field indicates the first byte of the slot reserved for the next route address and, therefore, its minimum value is 4. The *route data* area is initialized to zero to serve as a container for IPs discovered along the path. When receiving a packet equipped with this option, a network device checks if the pointer does not exceed the option length (i.e., the option is not full), inserts an owned IP address (*usually* the one associated to the outgoing interface [27]), and increments the pointer value accordingly. If the option is full, the packet is normally forwarded without inserting any address. Considering the maximum size of the IP header, the RR option cannot contain more than 9 address slots. For this reason, the RR option represents a valuable yet limited tool for tracing IP paths. The RR option has been used to infer the Internet paths [27], to investigate violations to the destination-based forwarding scheme [6], and to discover the reverse paths [12].

Timestamp option - RFC791. The TS option (type 68) has the format reported in Fig. 1c and it is defined along with three variants according to the *flag* field. When the flag is 0 or 1, if enough space is available, each traversed router is requested to insert in the option data a 32-bit timestamp (TS) or a *(IP, TS)* record respectively. If the value of the flag is 3, the originating host initializes the option data with a set of $(IP, 0)$ records: in this way, the devices from which the timestamp is requested are predetermined. In this paper, we exploit the variant obtained by setting the flag to 0, i.e., we request to any traversed router to insert a timestamp into the option data if enough space is available. The *pointer* field identifies the first byte of the slot reserved for the next timestamp and, thus, its minimum value is 5. When receiving a packet equipped with this option, a network device checks if the pointer does not exceed the option length (i.e., the option is not full), inserts a timestamp, and increments the pointer value accordingly. If the timestamp data area is already full, the packet is forwarded without inserting any timestamp, but the *overflow* value is incremented by one. For this reason, the overflow field counts the number of IP modules that cannot insert timestamps due to lack of space. Since the maximum size of an IP option is 40 bytes, this variant of TS option can contain a maximum of 9 timestamp slots. The TS option has been used to group the addresses owned by the same network device [21], [26], to dissect the RTT [17], to detect third-party addresses and hidden routers in Traceroute traces [18], [23], and to infer reverse paths jointly with the RR option [12].

Soliciting ICMP Parameter problems. The standards explicitly consider the generation of an ICMP PP message in the following conditions:

- RR option: (i) there is *some room but not enough room* to insert a full IP address into the option data; or (ii) the route data area is already full.
- TS option: (i) there is *some room but not enough room* to insert a full timestamp into the option data; or (ii) *the overflow field counts itself overflows*.

Recreating the above conditions at a specific hop along the path causes the probe packet to be discarded and an ICMP PP message to be sent back to the source as notification.

B. ICMP PP-based path tracing solutions

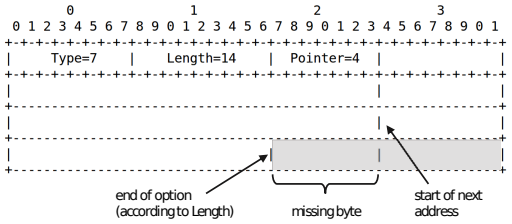
Cut Record Route (CRR). A router forwarding a packet equipped with the RR option, having *some room but not enough room* in the option data for a full IP address, should consider the datagram as damaged, discard it, and eventually send an ICMP PP message to the source. Accordingly, to solicit an ICMP PP message from the i^{th} hop on the path, CRR sets the RR option length (RRLen) such that there is enough space in the option data only for $i-1$ IPs, while only 3 bytes are available for the i^{th} one.

$$RRLen = RRHeaderLen + AddrSize \times (i-1) + BrokenAddr \quad (1)$$

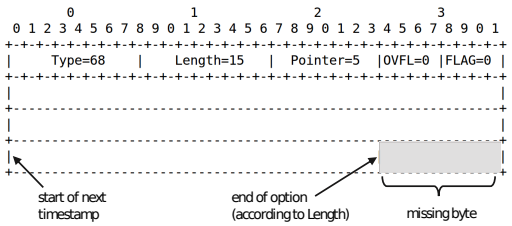
Hence, RRLen is computed as reported in Eq. 1, where: RRHeaderLen is the RR header size (3 bytes); AddrSize is the size of an IPv4 address (4 bytes); BrokenAddr refers to a malformed slot of 3 bytes, thus unable to contain a full address. The pointer field value is initialized to 4, in order to point to the first slot in the RR option data. Thus, the first $i-1$ hops normally manage the option, while only the i^{th} hop detects the malformation, notifying the error to the source. An example of CRR probe to elicit an ICMP PP message form the third hop is reported in Fig. 2a. Note how two padding bytes are introduced to keep the packet consistent with the IP header length field, while just the RR option is malformed.[1] Since the RR option data cannot contain more than 9 slots, the maximum exploring range of CRR is limited to 9 hops.

Cut Timestamp (CTS). Similarly to CRR, CTS elicits an ICMP PP message from the i^{th} hop on the path by exploiting

[1]Padding bytes are treated as End of Options list – RFC791.



(a) CRR probe optional IP header.



(b) CTS probe optional IP header.

Fig. 2. IP options crafted to solicit ICMP PP messages from the 3^{rd} hop: not enough space is available for the third address/timestamp. In grey padding bytes.

a TS option in which enough space is allocated just for $i-1$ timestamps, while only 3 bytes are available for the i^{th} one.

$$TSLen = TSHeaderLen + TSSize \times (i-1) + BrokenTS \quad (2)$$

The TS option length (TSLen) is computed as reported in Eq. 2, where: TSHeaderLen is the size of the TS header (4 bytes); TSSize is the size of a standard timestamp (4 bytes); BrokenTS refers to a malformed slot of 3 bytes, thus unable to contain a full timestamp. An example of CTS probe crafted to solicit an ICMP PP reply from the third hop is reported in Fig. 2b. Note how in this case, just one padding byte is required to properly align the IP header to 32-bits words. Since the TS option data cannot contain more than 9 slots, also the exploring range of CTS is limited to 9 hops.

Overflow in Overflow (OV2). OV2 exploits the 4 bits overflow field of a full-size TS option. Once all the slots in the option data are filled, a packet equipped with the TS option can travel for at most 16 additional hops before being discarded: indeed, a router forwarding a probe with the TS option overflow field at 15 should detect an overflow exception, discard the datagram, and send an ICMP PP message to the source. OV2 creates this condition at the i^{th} router along the path by setting the pointer and overflow fields as reported in Eq. 3.

$$\begin{aligned} 1 \leqslant i \leqslant 16 &\begin{cases} pointer = TSLen + 1 \\ overflow = 16 - i \end{cases} \\ 16 < i \leqslant 25 &\begin{cases} pointer = TSLen - TSSize \times (i-16) + 1 \\ overflow = 0 \end{cases} \end{aligned} \quad (3)$$

If the target is within 16 hops, OV2 relies just on the overflow field: the pointer is set such that the option appears already full, while the overflow value is set to cause the *overflow in overflow* condition after i increments. When the target is x hops far, with $x > 16$, the overflow value is set to zero and $n = x - 16$ slots are left available in the option data: the insertion of n timestamps and 16 increments of the overflow value cause the *overflow in overflow* event at the targeted hop.

An OV2 probe crafted to solicit an ICMP PP reply from the third hop contains a full-length TS option of 40 bytes, where the pointer is set to 41 (the option is full) and the overflow field is set to 13 causing the overflow in overflow exception at the third hop. Note that in this case, no padding bytes are required. Since the overflow field allows up to 16 increments and the TS option data can contain up to 9 slots, the OV2 maximum exploring range is 25 hops.

Rebuilding the path from the collected replies. Since IP options are not universally supported [7], in order to assign the source address of the ICMP PP messages to a specific hop along the path, we take advantage of the TTL-based distance covered by each probe along its travel: such distance is computed as the difference between the TTL value initially set into the probe packet and the one carried back in the payload of the ICMP PP message.

III. Experimental results

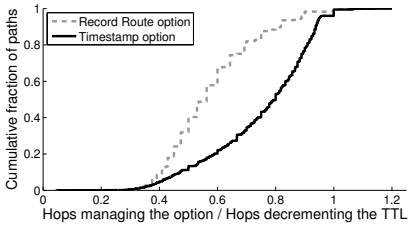
In this section, after comparing the novel path tracing solutions through a preliminary experimental measurement campaign (Sec. III-A), we further explore the most promising ICMP PP-based solution together with Paris-Traceroute MDA, the state-of-the-art implementation for the TTL-based path-tracing approach (Sec. III-B).

All the results reported in this section are related to the interfaces discovered exclusively along the path and not at the targeted device. Since different path tracing solutions may report different interfaces of the same router, we employed state-of-the-art *alias resolution* techniques to group together the addresses owned by the same router: we combined IGMP probing [19], Iffinder [13], Pythia [21] and Midar [14] from 40 Planetlab nodes to obtain rich and accurate IP aliasing information. Multiple techniques and vantage points (VPs) increase the coverage and the confidence of our results. All the experimental campaigns were performed in September 2014.

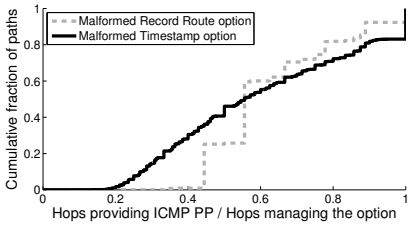
A. Comparing ICMP PP-based solutions

Methodology. To explore the effectiveness of the ICMP PP-based path tracing solutions, we traced paths with CRR, CTS, and OV2 towards $20K$ destinations in distinct /24 subnets from our laboratory in Napoli. We selected the destinations among those addresses steadily responsive both to ping (according to the PREDICT project [10]), and to UDP probes equipped with non malformed RR and TS options.

Results. The main findings from this experimental campaign are: (i) a large fraction of devices per path managed the IP options, thus being compliant with the explored solutions; (ii) although non RFC-compliant implementations exist, the ICMP PP-based solutions were all actually able to elicit replies from the majority of devices processing the option; (iii) OV2 outperformed CRR and CTS in terms of discovered interfaces and routers.

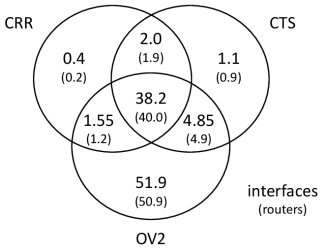


(a) Hops managing the option.



(b) Hops providing ICMP PP.

Fig. 3. Considering as a reference the number of devices per path decrementing the TTL, the TS (RR) option was managed by 75% (57%) of the devices. Malforming the TS (RR) option triggered ICMP PP replies from 61% (62.5%) of these devices.



(a) Intersections (%).

Path tracing Solution	Interfaces #	Interfaces %	Routers #	Routers %
CRR	7,671	42.15	7,252	43.36
CTS	8,399	46.15	7,980	47.71
OV2	17,560	96.47	16,229	97.02
Union	**18,200**	**100**	**16,727**	**100**

(b) Discovered interfaces and routers.

Fig. 4. Interfaces and routers discovered by the ICMP PP-based solutions.

We calculated the fraction of devices managing the option (i.e. following the RFC) over those decrementing the TTL along the forward path by targeting each destination with UDP probes equipped with non-malformed RR and TS option. From the payload of the elicited ICMP Port Unreachable reply, we extracted the UDP probe as arrived at destination, option included. Then, we computed the number of devices decrementing the TTL as the difference between the TTL value we originally set in the UDP probe and its final value at the destination. Finally, we computed the number of devices managing the option as the number of IP addresses (timestamps and overflow increments) contained in the returned RR (TS) option. On average, 75% (57%) of the devices decrementing the TTL on each path proved to manage the TS (RR) option (Fig. 3a). Furthermore, some paths involve more devices managing the option than those decrementing

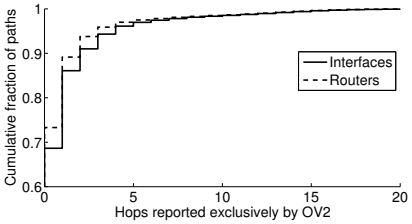


Fig. 5. Hops per path reported only by OV2 – OV2 reported thousands of interfaces and routers not listed by MDA highlighting standard-compliant devices managing the TS option. For about 31% of the paths, OV2 discovered at least one additional interface.

the TTL, suggesting the presence of middleboxes or hidden routers on the path [23].

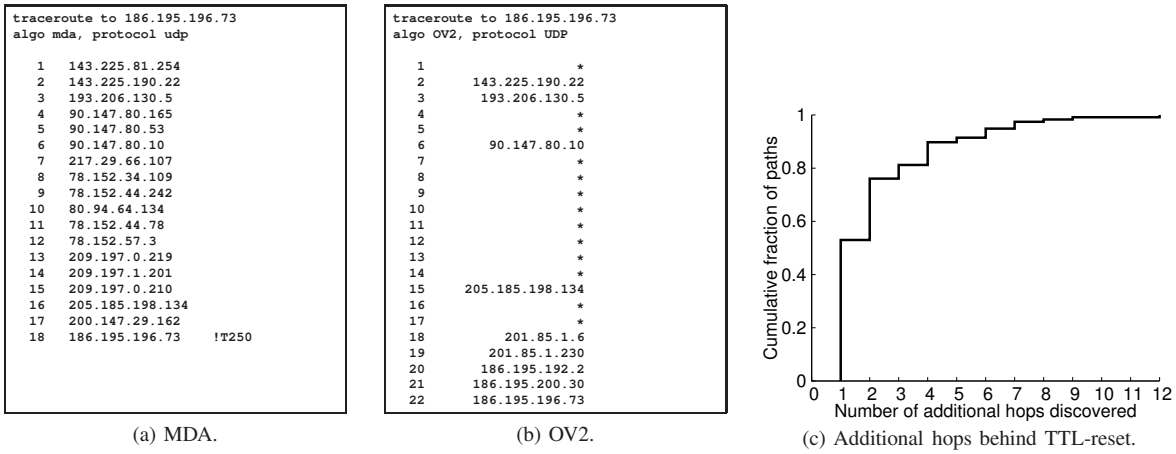
Not all the devices managing the IP options provided ICMP PP replies (Fig. 3b). On each path, malformed TS (RR) options solicited ICMP PP replies from 61% (62.5%) of the devices managing the option on average. Root causes of lack of replies include: (a) routers not generating the ICMP PP reply but silently discarding the issued probes; (b) ICMP PP replies filtered along the reverse paths; (c) devices exposing non RFC-compliant behaviors, i.e., manipulating the IP option but not properly recognizing the probe malformations. In particular, some devices incorrectly interpreted the *some room but not enough room* condition as a full option, thus normally forwarding the probes along the path, while other devices reset the overflow field when it reaches its maximum value. In our campus network, we observed both the abnormal behaviors exposed by CISCO 6500 series routers.

Finally, OV2 proved to be the most effective solution among the ICMP PP-based ones (Fig. 4a), reporting most of the interfaces and routers collected by CRR and CTS. OV2 alone reported about 52% of the total discovered interfaces and routers, mainly due to the larger exploring range. Note that, although limited in number, we also observed routers providing ICMP PP messages depending on (a) the adopted IP option – CRR reports interfaces and routers invisible to CTS and OV2 – and (b) the type of malformation – although based on the same option, CTS discovered routers not reported by OV2. We left as future work pinpointing why some devices replied to CRR and CTS but not to OV2.

B. OV2 performance analysis

We investigated whether OV2, the most promising ICMP PP-based solution, is able to discover interfaces or routers not reported by Paris-Traceroute MDA, the widely recognized state-of-the-art implementation for TTL-based path tracing solution. Experimental results confirmed the ability of this alternative solution to report additional information on the traversed paths.

Methodology. We reimplemented OV2 on top of Paris-Traceroute code to let it work properly also from Planetlab nodes. We employed 40 Planetlab nodes at different sites world-wide: each node has been instructed to first issue packets equipped with RR and TS options towards our University network, to assess if this particular type of traffic was filtered



(a) MDA. (b) OV2. (c) Additional hops behind TTL-reset.

Fig. 6. MDA probes reached the destination with a residual TTL value of 250 suggesting the reset of the TTL value to 255 along path that defused MDA probes (6a). By not relying on the TTL, OV2 was able to identify these hidden hops (6b, hops 18-21). Figure 6c shows the distribution of the hops discovered by OV2 in the portion of the path totally hidden to TTL-based path tracing mechanisms due to TTL reset.

by the node's access network, where typically filtering occurs [7]. We found that the access networks of three Planetlab nodes steadily filter packets equipped with IP options. Each of the remaining 37 nodes randomly selected $10K$ destinations from a set of addresses highly responsive to ping according to PREDICT [10]: each path was traced with both OV2 and MDA (95% confidence [1]) one after the other. Note that MDA traces all the possible paths to the destination in case of load balancers while our current implementation of OV2 only traces one of these paths behaving as the classic Traceroute. Accordingly, OV2 is likely to report fewer devices in each path by design. With this approach, we aim at comparing OV2 with the TTL-based tracing solution configured to achieve its full potential.

Results. The experimental campaign showed that (i) almost 60% of the routers in the core network support the TS option. Furthermore, OV2 reported interfaces and routers not listed by MDA. This utility resulted higher (ii) in those paths traversing devices which reset the TTL field and (iii) towards specific ASes. We provide more details in the following.

Jointly, MDA and OV2 reported along the traced paths about $296K$ interfaces for a total number of $231K$ routers: about 57% of these devices proved to support the TS option by replying when probed with OV2. This value is a lower bound of the real support of this option since many devices managing the TS option do not provide the ICMP PP replies (Sec. III-A). To the best of our knowledge, this is the first quantification of uniquely identified routers supporting the TS option. Previous works mainly focused on whether this type of traffic suffers from filtering [7] or on the performance penalties determined by IP options [8].

Globally, OV2 discovered $3,442$ interfaces and $2,476$ routers not listed by MDA. This happened even if OV2 is not configured to trace all the paths to the destination or to inject a number of probe packets exponentially growing with the number of load balancers encountered along the path. Results were confirmed by repeated measurements. From a per-path point of view, OV2 listed at least one interface (router) not reported by MDA in 31.3% (26.7%) of the scanned paths (Fig. 5). About 31.2% of the interfaces reported only by OV2 are additional interfaces of the same routers already discovered by MDA, while the remaining ones belong to devices not tracked by MDA.

By focusing on those paths where OV2 reported many additional interfaces and routers, we isolated a specific scenario where this alternative tracing solution appeared particularly interesting, i.e., in case of TTL reset [24]. There is TTL reset when a traversed device along the path resets the TTL to a high value, thus defusing the Traceroute probes that directly reach the targeted destination. TTL reset causes (a) the portion of the path between the device resetting the TTL and the targeted destination to be totally hidden to TTL-based path tracing mechanism, no matter how sophisticated they are; (b) the number of hops to the destination to be underestimated. In addition, (c) using multiple vantage points does not help when the TTL reset occurs in the proximity of the destination and in case of limited route diversity. Root causes of TTL reset include MPLS tunnel misconfigurations [11] and middleboxes purposely configured to limit path tracing into corporate networks [24]. In these cases, adopting a tracing solution not relying on the TTL field may provide information on the hidden portion of the path. One instance is showed in Fig. 6 reporting the MDA and OV2 trace toward the same destination. MDA packets arrived at the destination with a residual TTL value of 250 (note the flag !T250 in Fig. 6a): a TTL reset to 255 caused the last hops before the destination to be invisible to MDA. By not relying on the TTL, OV2 was able to trace this portion of the path invisible for MDA (Fig. 6b). In our experimental campaign, we observed evidences of TTL reset in $1,383$ paths, in 12% of these paths, OV2 discovered on average (at most) 2.2 (12) additional hops behind the device resetting the TTL (Fig. 6c).

Also, we broke the measurement results down by AS. As shown in Fig.7a, OV2 revealed at least one additional IP for 960 ASes, most of which were stub networks. In the best case (AS26615), OV2 discovered over 200 additional IPs.

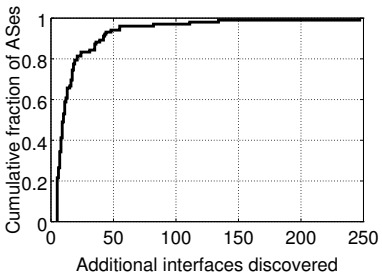
Guided by this result, we targeted $28,715$ destinations active according to the PREDICT dataset and located in different prefixes announced by AS26615 with both MDA and OV2. Considering only the interfaces belonging to this AS, we obtained the results reported in Fig.7b: while MDA discovered about the same IPs from all the VPs, OV2 revealed different IPs when working from different VPs. Up to 72 additional IPs were revealed from a single VP, detecting in total 180 IPs more than MDA. About (a) 35% of the interfaces reported only by OV2 appeared a few hops before the destination ([2]), (b) 60% appeared several hops after the last hop listed by MDA, and (c) only 5% appeared at the same hop count as the destination reported by MDA.

IV. Conclusion

In this paper, we experimentally explored three novel path tracing solutions we publicly released, named CRR, CTS, and OV2. These solutions inject into the network packets equipped with malformed IP options in order to elicit ICMP Parameter Problem messages from the traversed routers. We experimentally observed that all these solutions were actually able to elicit replies from the network, with OV2 outperforming CRR and CTS. With a large-scale multi-vantage point experimental campaign, we observed that OV2 is able to report interfaces and routers not discovered by the classic TTL-based path tracing solutions, thus complemeting them. Experimental analyses highlighted how this complementarity is higher for specific stub networks and in those paths traversing devices that reset the TTL field.

ACKNOWLEDGEMENTS

This work is partially funded by the MIUR projects: PLATINO ($PON01\_01007$), SMART HEALTH ($PON04a2\_C$), S2-MOVE ($PON04a3\_00058$).



(a) IPs discovered only by OV2 per AS.

VP location	MDA	MDA∪OV2	Gain
China	69	141	104.3%
TX, USA	69	119	72.5%
Brasil	70	100	42.9%
NC, USA	71	98	38.0%
Japan	70	75	7.1%
Union	**74**	**254**	**243.2%**

(b) OV2 utility towards AS26615 (interfaces).

Fig. 7. Compared to MDA, OV2 revealed at least one additional IP for 960 ASes (7a). When targeting from 5 VPs the top AS for additionally discovered IPs (AS26615), using OV2 in addition to MDA lead to a gain of 243% in terms of discovered IPs (7b).

References

[1] B. Augustin, T. Friedman, and R. Teixeira. Measuring multipath routing in the Internet. *IEEE/ACM Trans. on Networking*, 19(3):830–840, June 2011.

[2] CISCO. TTL Expiry Attack Identification and Mitigation. http://www.cisco.com/web/about/security/intelligence/ttl-expiry.html.

[3] Í. Cunha, R. Teixeira, D. Veitch, and C. Diot. Predicting and tracking internet path changes. In *ACM SIGCOMM 2011*, pages 122–133, New York, NY, USA, 2011. ACM.

[4] G. Detal, B. Hesmans, O. Bonaventure, Y. Vanaubel, and B. Donnet. Revealing middlebox interference with tracebox. In *ACM IMC*, pages 1–8, 2013.

[5] B. Donnet, P. Raoult, T. Friedman, and M. Crovella. Efficient algorithms for large-scale topology discovery. In *ACM SIGMETRICS PER*, volume 33, pages 327–338, 2005.

[6] T. Flach, E. Katz-Bassett, and R. Govindan. Quantifying violations of destination-based forwarding on the Internet. In *ACM SIGCOMM IMC*, pages 265–272, 2012.

[7] R. Fonseca et al. IP Options are not an option. Technical Report UCB/EECS-2005-24, University of California, Berkeley, 2005.

[8] P. Fransson and A. Jonsson. End-to-end measurements on performance penalties of ipv4 options. In *IEEE GLOBECOM'04*, volume 3, pages 1441–1447, 2004.

[9] M. H. Gunes and K. Saraç. Resolving anonymous routers in Internet topology measurement studies. In *INFOCOM*, pages 1076–1084, 2008.

[10] IP Address Hitlist. PREDICT ID USC-LANDER internet_address_hitlist_it57w-20131127. 2011-07-26 to 2013-12-30. http://www.isi.edu/ant/lander.

[11] Juniper Networks. Technical documentation: Disabling Normal TTL Decrementing. https://stage.juniper.net/techpubs/en_US/junos/topics/usage-guidelines/mpls-disabling-normal-ttl-decrementing.html.

[12] E. Katz-Bassett, H. V. Madhyastha, V. K. Adhikari, C. Scott, J. Sherry, P. Van Wesep, T. E. Anderson, and A. Krishnamurthy. Reverse traceroute. In *USENIX NSDI*, volume 10, pages 219–234, 2010.

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

[14] K. Keys, Y. Hyun, M. Luckie, and K. Claffy. Internet-scale IPv4 Alias Resolution with MIDAR. *IEEE/ACM Trans. on Networking*, 21(2):383–399, 2013.

[15] M. Luckie. Scamper: A scalable and extensible packet prober for active measurement of the Internet. In *ACM SIGCOMM IMC*, 2010.

[16] M. Luckie, Y. Hyun, and B. Huffaker. Traceroute probe method and forward IP path inference. In *ACM SIGCOMM IMC*, 2008.

[17] P. Marchetta, A. Botta, E. Katz-Bassett, and A. Pescapé. Dissecting Round Trip Time on the Slow Path Using a One-Packet Approach. In *PAM*, 2014.

[18] P. Marchetta, W. de Donato, and A. Pescapé. Detecting Third-Party Addresses in Traceroute Traces with IP Timestamp Option. In *PAM*, 2013.

[19] 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. *IEEE JSAC*, 29(9):1776–1787, 2011.

[20] P. Marchetta, V. Persico, E. Katz-Bassett, and A. Pescapé. Don't Trust Traceroute (Completely). In *ACM CoNEXT Student Workshop*, 2013.

[21] P. Marchetta, V. Persico, and A. Pescapè. Pythia: Yet another active probing technique for alias resolution. In *ACM CoNEXT*, 2013.

[22] P. Marchetta, V. Persico, and A. Pescapé. The greenhouse effect attack. In *Communications and Network Security (CNS), 2014 IEEE Conference on*, pages 516–517. IEEE, 2014.

[23] P. Marchetta and A. Pescapè. DRAGO: Detecting, Quantifying and Locating Hidden Routers in Traceroute IP Paths. In *IEEE GIS*, 2013.

[24] T. Moors. Streamlining traceroute by estimating path lengths. In *IEEE IPOM*, 2004.

[25] S. Qian, Y. Wang, and K. Xu. Utilizing destination options header to resolve ipv6 alias resolution. In *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, pages 1–6. IEEE, 2010.

[26] 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*, pages 172–178, 2010.

[27] R. Sherwood, A. Bender, and N. Spring. Discarte: A disjunctive internet cartographer. In *ACM SIGCOMM*, pages 303–314, 2008.