DRAGO: Detecting, Quantifying and Locating Hidden Routers in Traceroute IP Paths

Pietro Marchetta and Antonio Pescapé
University of Napoli Federico II (Italy)
Email: {pietro.marchetta,pescape}@unina.it

Abstract—**Traceroute is probably the most famous networking tool widely adopted in both industry and research. Despite its long life, however, measurements based on Traceroute are potentially inaccurate, misleading or incomplete due to several unresolved issues. In this paper, we face the limitation represented by *hidden routers* - devices that do not decrement the TTL, being thus totally invisible to Traceroute. We present, evaluate and release DRAGO, a novel active probing technique composed of three main steps. First, a novel Traceroute enhanced by the IP Timestamp option is launched toward a destination. Second, a procedure is applied to quantify the hidden routers contained in the path, if any. Third, a last procedure is performed to identify the exact position in the path of the detected hidden routers. Experimental results demonstrate that the phenomenon is not uncommon: DRAGO detects the presence of hidden routers in at least 6% of the considered Traceroute IP paths and limits the affected area to one fifth of the trace containing these devices.**

I. Introduction

Originally introduced by Van Jacobson in the late eighties, Traceroute [1] is one of the most famous diagnostic tools in the networking field. Ideally, it is able to discover the IP path toward a targeted destination by listing one IP address for each network-layer device traversed along the path. This goal is achieved by injecting into the network packet probes with an increasing value of the Time-to-Live (TTL) field to solicit an ICMP Time Exceeded error message. Today, its practical utility is affirmed both in the industry and research. Traceroute is widely adopted by network operators to investigate network performance problems like, for example, the identification of persistent or transient routing anomalies [2]. On the other hand, researchers make an extensive use of this tool to infer network topological properties [3]–[13], and, more in general, in active monitoring approaches for anomaly detection [14]–[16], performance analysis [17], [18], and geolocation [19], [20]. Unfortunately, despite its long life, it has been profusely demonstrated that Traceroute is not free of limitations [21]–[26]. As a consequence, measurements based on Traceroute may be potentially inaccurate, misleading or incomplete. Among its limitations, while *load balancers* (devices splitting the traffic issued toward the same destination over multiple equal cost paths and thus, causing Traceroute to infer false router-level links and bogus loops [25]) and *anonymous routers* (devices silently discarding the Traceroute packet probes causing the collected traces to be incomplete) have been extensively investigated and partially solved or mitigated [21]–[25], *hidden routers* and issues introduced by their presence are still underestimated.

A hidden router forwards the packets without decrementing the TTL value. As a consequence, these devices are totally invisible to Traceroute. According to [27], "*hidden routers are caused by certain configurations of multi-protocol label switching [28] (MPLS) and result in missing nodes and incorrect link inferences*", potentially having a great impact on the Internet topological properties assessed today. While also middleboxes may act as hidden routers, the magnitude of the phenomenon is today unknown due to the lack of a set of techniques able to recognize the presence of hidden routers.

In this paper, we present and evaluate DRAGO, an active probing technique able to detect, quantify and locate hidden routers in Traceroute IP paths. To achieve this goal, DRAGO performs three main steps: (*i*) a novel Traceroute enhanced by the IP Timestamp (TS) option is launched toward a destination; (*ii*) a procedure is applied to quantify the hidden routers contained in the path, if any; (*iii*) a last procedure is performed to identify the exact position in the path of the detected hidden routers. The key mechanism used to detect hidden routers is based on the comparison between the number of hops that manage the TS option and those decrementing the TTL: there is an evidence of hidden routers on the path every time the number of hops managing the TS option is higher than the ones decrementing the TTL. Experimental results suggest that the phenomenon is not uncommon. DRAGO detects hidden routers in about 6% of the traces of the considered dataset (starting from the information provided by the PREDICT project [29]): DRAGO identifies the exact location for 14% of the detected hidden routers and limits on average the affected area to one fifth of the Traceroute trace containing these devices. The surprisingly high number of affected traces suggests that the phenomenon can not be ignored any more, especially when the objective is to accurately infer the topological properties of Internet.

The paper is organized as follows. Sec. II provides some background on the TS option; Sec. III presents the details of DRAGO; Sec. IV-A describes how the technique works in a sample scenario; Sec. IV-B reports the results of the evaluation phase; Finally, Sec. V compares the proposed solution with related works while Sec. VI ends the paper with concluding remarks.

II. Background

Recently, we have seen a growing interest on Internet measurements based on the TS option [30]. This option has been used to identify the addresses owned by the same network device [31], to develop a reverse Traceroute [32], to estimate the network delay [33] and to detect third-party addresses in Traceroute IP paths [26]. A deep study on the level of support for this option is reported in [34]. While we foresee more and more applications, in this work we use the TS option to count the IP modules managing the IP option on the path: DRAGO exploits the TS option to identify incongruities between the number of IP modules decrementing the TTL and those managing the option.

The TS option includes a 4 bytes header and it is defined along with three variants according to the *flag* field. In this work, we exploit the basic variant obtained by setting the flag field to 0. With this variant, each IP module, forwarding a packet equipped with the TS option, is requested to insert one timestamp in the option data (if enough space is available) or to increment by one the *overflow* field (when the option data is full). Basically, the overflow counts the number of hops that could not insert a timestamp due to lack of space. Since the maximum size of an IP option is 40 bytes and each timestamp requires 4 bytes, the TS option can contain no more than 9 timestamps. In addition, the overflow field consists of 4 bits and no more than 15 hops can increment its value before the overflow is reset. This implies that a packet probe equipped with the TS option allows to count up to 24 hops managing the option (9 hops inserting the timestamp and 15 incrementing the overflow).

III. Detecting Hidden Routers with Drago

In this section, we describe how DRAGO detects and locates hidden routers along an IP paths. As depicted in Fig. 1, DRAGO works as follows: (*1.*) a novel Traceroute enhanced by the TS option is launched toward the destination; (*2.*) a procedure to detect and approximately locate the presence of hidden routers in the traces is applied to the Traceroute trace; (*3.*) a last procedure is applied to reduce, as much as possible, the uncertainty on the position of the detected hidden routers starting from the output of the previous step.

Step 1: Traceroute enhanced with the TS option. This step aims at counting the number of devices managing the TS option and those decrementing the TTL on the path toward a destination. To reach this goal a novel Traceroute is used: the injected TTL limited Traceroute probes are also equipped with the TS option. In this way, all the hops along the path are requested to insert a timestamp in the option's data or to increment the overflow. This enhanced Traceroute collects ICMP$-$TE messages from the hops. From each collected ICMP$-$TE reply, our Traceroute extracts the source address and also the number of devices managing the TS option up to the replying router. The latter information is computed by inspecting the TS option brought back in the payload of the ICMP$-$TE error message[1].

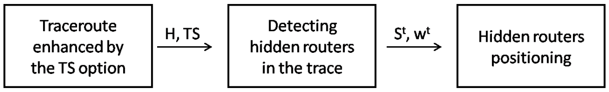

Traceroute enhanced by the TS option
H, TS
Detecting hidden routers in the trace
S^t, w^t
Hidden routers positioning

Figure 1: The three steps performed by DRAGO.

Hereafter, we adopt the following notation (see Tab. I as a reference for the notation introduced across the paper): h_i is the i-th hop discovered by our Traceroute along the path toward the destination; H is a Traceroute trace made by n hops $h_1 h_2 ... h_n$; TS_i is the number of timestamps plus the overflow increments contained in the TS option brought back in the payload of the ICMP$-$TE reply provided by h_i. Basically, TS_i represents the number of IP modules which actively managed the TS option up to the TTL-decrement performed in h_i. As a consequence, TS_n represents the overall number of IP modules managing the TS option on the path. TS is clearly monotonically non-decreasing with i.

The trace H and the associated TS represent the input of the second step.

Step 2: Detecting and quantifying hidden routers. This step aims at detecting the presence of hidden routers in the trace H pointing out also the portion of the trace in which those devices lie. To explain how the step works, we introduce a new variable called $INCR$:

$$INCR_i = \begin{cases} TS_1 & \text{if i = 1} \\ TS_i - TS_{i-1} & \text{otherwise} \end{cases} \qquad (1)$$

$INCR$ reports the hop-by-hop number of IP modules managing the TS option. For example, $INCR_i = z$ implies that there are exactly z IP modules managing the TS option in the transition $h_{i-1}\ h_i$. Since there are $n+1$ hops in the path (considering also the source machine), there is an evidence of hidden routers in the path every time $TS_n > n+1$. Indeed, in this case, there are more devices managing the TS option than the ones decrementing the TTL. The condition reported above can be applied also to any portion of the trace: this is the basic mechanism used in DRAGO.

First, all the longest subsequences $S^1, ...S^p$ of consecutive non-zero elements in $INCR$ are extracted. Each subsequence S^t contains s^t elements and it is related to a specific portion of the trace made by $s^t + 1$ hops. The subsequence S^t contains hidden routers every time the following condition is verified:

$$\sum_{i=1}^{s^t} S_i^t > s^t + 1 \qquad (2)$$

Basically, there are hidden routers in a subsequence when the number of involved hops is lower than the IP modules managing the TS option in the associated portion of the trace. In particular, in the subsequence S^t there are exactly w^t hidden routers:

$$w^t = \max\left(\quad 0 \quad ; \quad \sum_{i=1}^{s^t} S_i^t - (s^t + 1) \quad \right) \qquad (3)$$

[1] Usually, the original probe (TS option included) triggering the ICMP error is brought back to the source host inside the payload of the ICMP message.

Accordingly, the overall number of hidden routers W contained in the trace is:

$$W = \sum_{t=1}^{p} w^t \tag{4}$$

The set of sequences S^t containing hidden routers represents the input of the next step.

Step 3: Hidden routers positioning. Up to now, all that we know is that w^t hidden routers are located somewhere in the subsequence S^t. Especially when the final goal is to accurately map the network topology, such level of accuracy is not enough. Hence, the goal of this step is to reduce, as much as possible, the uncertainty about the position of the w^t hidden routers detected in the subsequence S^t.

A first possibility is to analyze the elements of the subsequence S^t one-by-one. Note that, while $S^t_i > 2$ definitively uncovers hidden routers (and also their exact position in the trace), $S^t_i = 2$ is hard to interpret: it may suggest the presence of a hidden router but this is not always the case. Indeed, both the TTL-decrement and the TS option management are performed at the IP layer of the TCP/IP stack. In distinct implementations, the two operations may be performed in different order and such circumstance has an impact on $INCR$ and the extracted S^t. For example, Fig. 2 shows the $(i\text{-}1)\text{-}th$ and $i\text{-}th$ hops discovered toward the Traceroute destination as well as the order in which the TTL and the TS option are managed: the first hop manages the TTL before the TS option while the opposite happens in the second hop. In this scenario, $INCR_i$ is 2 but there are no hidden routers in this portion of the trace. Analyzing one-by-one the elements in S^t may not uncover all the hidden routers contained in the trace: indeed, each node may manage at most once the TS option and its contribution should count no more than once. Hence, analyzing entire portions of S^t may reveal additional hidden routers.

Hereafter, we use $S^t_{i,j}$ to refer to the elements S^t_i,...,S^t_j in the subsequence S^t. In each portion $S^t_{i,j}$, there are exactly $w^t_{i,j}$ hidden routers:

$$w^t_{i,j} = \max\left(\quad 0 \quad ; \quad \sum_{k=i}^{j} S^t_k - (j - i + 2) \quad \right) \tag{5}$$

Table I: Notation used in the paper.

Notation	Description
H	$h_1\ h_2\ ..\ h_n$, vector where h_i is the i-th hop discovered by Traceroute along the path.
TS	$TS_1\ TS_2\ ..\ TS_n$, vector where TS_i is the number of IP modules managing the TS option up to h_i.
$INCR$	$INCR_1\ INCR_2\ ..\ INCR_n$, vector where $INCR_i$ is the number of IP modules managing the TS option in the transition $h_{i-1}\ h_i$.
S^t	t-th subsequence of consecutive non-zero elements extracted from $INCR$. It contains s^t elements.
w^t	Hidden routers contained in S^t.
$S^t_{i,j}$	S^t_i, .. , S^t_j vector of elements in the subsequence S^t .
$w^t_{i,j}$	Hidden routers contained in $S^t_{i,j}$.
W	Total number of hidden routers contained in the trace.

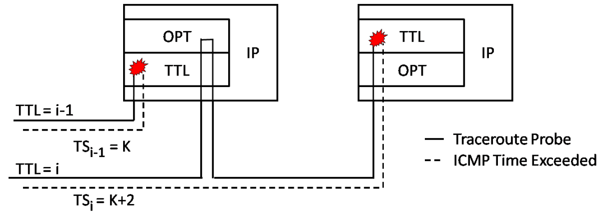


Figure 2: Different implementations of the TCP/IP stack and their impact on TS and $INCR$.

To accurately locate the hidden routers, the technique should count the number of hidden routers contained in all the possible portions of the subsequence S^t. To explore only a subset of all the possibilities, the technique makes use of a binary tree. Each node in the tree is related to a specific portion $S^t_{i,j}$ and it is labelled with the corresponding $w^t_{i,j}$. The root node in the tree is related to the entire subsequence S^t_{1,s^t} and it is labelled with the total number of hidden routers contained in S^t (w^t).

At the beginning, the tree contains only the root node. The technique generates the two child nodes of a generic node $S^t_{i,j}$ only if $w^t_{i,j} > 0$. When this occurs, the technique *explodes* the node $S^t_{i,j}$ by generating the two child nodes $S^t_{i,j-1}$ and $S^t_{i+1,j}$: a child node is associated to the sequence of its parent shortened of either the first or the last element. The ratio is that a portion (a node in the tree) must be further investigated (exploded) only if there are still evidences of hidden routers. At the end of this process, the tree contains several levels (in the worst case, s^t levels). All the nodes at the same level are related to distinct portions of the subsequence with the same size: the higher is the level the lower is the size of the portions associated to the nodes. The paradigm adopted to build the binary tree is *depth-first*. The exploration of a branch ends when one of the following conditions is verified:

- The node to explode is associated to a portion made by a unique element $S^t_{i,i}$: the $w^t_{i,i}$ hidden routers are exactly located.
- Both the child nodes $S^t_{i,j-1}$ and $S^t_{i+1,j}$ of the last exploded node $S^t_{i,j}$ do not contain hidden routers, i.e. $w^t_{i,j-1} = 0$ and $w^t_{i+1,j} = 0$: hidden routers are visible at the parent node $S^t_{i,j}$ ($w^t_{i,j} > 0$) but disappear in the child nodes. In this case, we conclude that $w^t_{i,j}$ hidden routers are contained in the portion $S^t_{i,j}$ of the trace but it is not possible to locate such devices with a higher accuracy.

The last phenomenon may also affect a subset of the hidden routers: this happens when the parent node in the tree contains more hidden routers then the ones visible in the two child nodes. For those hidden routers, the technique is not able to further shrink the affected portion of trace.

The source code of DRAGO is freely available at [35].

Limitations. At this time, DRAGO is not able to distinguish if an $INCR_i = 1$ is caused by one of the hops traced by Traceroute or it is caused by a hidden router. From this point of view, DRAGO estimates a lower bound of hidden routers affecting the path. This version of DRAGO does not manage subsequences containing anonymous routers since the corresponding *TS* value is undetermined. In addition, the technique may suffer from false positives due to the so called *lazy* routers [27]: these devices do not decrement the TTL when the packet is equipped with an IP option, thus violating the standard RFC791. We left the design of a more sophisticated solution able to deal with anonymous and lazy routers for future works.

IV. DRAGO AT WORK

A. A working example

In this section, we provide an example of how DRAGO works. Fig. 3 shows a sample trace collected during the first step: the novel Traceroute discovered 10 hops toward the destination. These devices manage the TTL field and thus replied to Traceroute with ICMP−TE error messages. By inspecting the TS option brought back in the payload of these ICMP−TE messages, our Traceroute stored in TS the number of IP modules managing the option as registered hop-by-hop. In this example, TS=[1 1 3 3 6 7 8 10 12 12]. Starting from TS, the second step computes $INCR$ as described in Eq. 1. In this example, $INCR$=[1 0 2 0 3 1 1 2 2 0]. Then, p subsequences S^t of non-zero consecutive elements contained in $INCR$ are extracted. In this case, p=3 with S^1=[1], S^2=[2], S^3=[3 1 1 2 2]. Then, the Eq. 2 is applied on the extracted subsequences to detect hidden routers: only S^3 reveals hidden routers. By applying the Eq.3, we can count the number of contained hidden routers: $w^3 = 3$. Those 3 hidden routers may lie in any position of the trace associated to S^3 (h_4 h_5 h_6 h_7 h_8 h_9). To reduce such uncertainty, the third step builds the binary tree reported in Fig. 4. At the beginning, the tree is made just by the root node. This node is associated to the entire subsequence $S^3_{1,5}$ containing $w^3 = w^3_{1,5} = 3$ hidden routers. Since $w^3_{1,5} > 0$, the node $S^3_{1,5}$ is exploded in $S^3_{1,4}$ and $S^3_{2,5}$. The technique implements a deep-first exploration. Hence, the next considered node is $S^3_{1,4}$ and $w^3_{1,4} = 2$ is computed by applying the Eq. 5. In turn, the node $S^3_{1,4}$ is exploded in $S^3_{1,3}$ ($w^3_{1,3} = 1$) and $S^3_{2,4}$ ($w^3_{2,4} = 0$). Then, $S^3_{1,3}$ is exploded in $S^3_{1,2}$ ($w^3_{1,2} = 1$) and $S^3_{2,3}$ ($w^3_{2,3} = 0$). The node $S^3_{1,2}$ is further exploded in $S^3_{1,1}$ ($w^3_{1,1} = 1$) and $S^3_{2,2}$ ($w^3_{2,2} = 0$). The exploration of this branch is terminated: we reached the leaf of the binary tree and we found the exact position ($S^3_{1,1}$) of a hidden router (between h_4 and h_5). In addition, note that we count 2 hidden routers in $S^3_{1,4}$ but the child nodes provided details only about one router. We forcedly conclude that another hidden router is located somewhere in $S^3_{1,4}$ but we could not better identify its position. According to the deep-first paradigm, the technique analyzes $S^3_{2,3}$ and then $S^3_{2,4}$: both nodes are not exploded since they do not contain hidden routers. Then, node $S^3_{2,5}$ is exploded and the exploration continues as before until the $S^3_{4,5}$ is exploded in $S^3_{4,4}$ and $S^3_{5,5}$. While $w^3_{4,5} = 1$, either $w^3_{4,4} = 0$ either $w^3_{5,5} = 0$: a hidden router visible in $S^3_{4,5}$ disappeared in the lower level of the tree. We can conclude that a last hidden router lies somewhere in $S^3_{4,5}$ and the technique stopped.

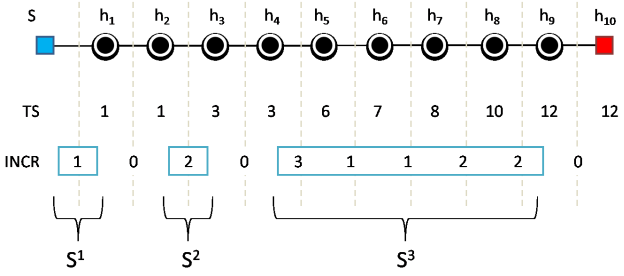


Figure 3: H, INCR and S^t in a sample scenario.

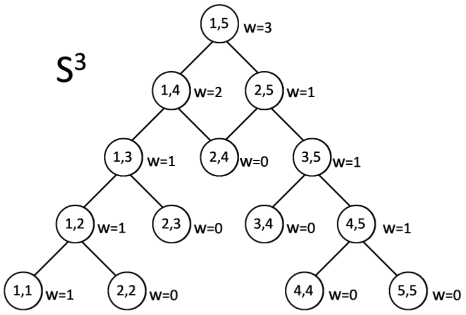


Figure 4: The binary tree for the subsequence S^3 of Fig. 3.

At the beginning of the process, all that we knew was that 3 hidden routers exist somewhere in the portion of the trace associated to S^3 (h_4 h_5 h_6 h_7 h_8 h_9). By applying the third-step of our technique, we concluded that (*a.*) a first hidden router is *exactly* located in S^3_1 (i.e. between h_4 and h_5); (*b.*) a second hidden router is located somewhere in $S^3_{1,4}$ (i.e in the portion of the trace h_4 h_5 h_6 h_7 h_8); (*c.*) the last hidden router is located somewhere in $S^3_{4,5}$ (i.e h_7 h_8 h_9).

This result is achieved by inspecting 14 out 15 portions of the of the subsequence S^3.

B. Experimental Results

In this section, we report the main findings of the evaluation.

To evaluate DRAGO, we selected $25K$ destinations in distinct ASes among the addresses showing stable responsiveness to *ping* according to the PREDICT project [29]. These addresses have been selected by using the IP-to-AS mapping service provided by Cymru [36]. We have launched DRAGO toward these destinations from our laboratory at the University of Napoli. To deal with load balancers, the novel Traceroute launched during the first step has been instructed to generate probes as part of the same flow by replicating the internal mechanism adopted in Paris Traceroute[2] [25]. After having

[2] Another fixed option before the TS option allows to deal with load balancers that simply assume UDP port numbers placed just after the IP header. A preliminary campaign exploiting this more sophisticated approach has qualitatively confirmed the results reported in this Section.

removed filtered traces and those affected by loops, the final dataset consists of $22K$ traces containing more than $45K$ addresses.

From the traces of the dataset, we have extracted $49,956$ unique transitions $h_{i-1}h_i$ not involving anonymous routers and the corresponding $INCR_i$ value. Besides few exceptions, all the transitions showed a stable number of intermediate hops managing the TS option, i.e. every time the transition $h_{i-1}h_i$ appears in a trace, the corresponding $INCR_i$ value is always the same. Tab. II reports the number of transitions showing the same $INCR$ value and the traces in which those transitions appear. By adopting a conservative approach, an initial set of hidden routers is already visible by analyzing single transitions showing $INCR$ values higher than 2: 100 transitions uncovered alone the presence of consecutive hidden routers invisible to Traceroute (4 in the worst case, i.e. when the corresponding $INCR$ value is 6).

At the same time, the evaluation of entire subsequences S^t extracted from $INCR$ can potentially uncover additional hidden routers. Tab. III shows the number of subsequences and traces containing a specific number of hidden routers. Considering the entire dataset, $29,756$ subsequences determined by distinct patterns of hops have been analyzed: $1,348$ (4.5% of the total) have at least one hidden router. From the trace point of view, about 6% of all the Traceroute traces in the dataset contains at least one hidden router. Taking into account how the phenomenon has been largely ignored, such a value appears surprisingly high and suggests that hidden routers are not uncommon and may heavily affect the assessed results achieved by classic topology discovery techniques based on Traceroute.

After the second DRAGO step, the *position range* (the number of different positions potentially hosting a hidden router) for each hidden router coincides with the size of the subsequence: the higher is the range, the higher is the uncertainty on the position of the hidden router. DRAGO performs the third step to reduce such uncertainty. Fig. 5(a) shows the position range of each detected hidden router after the second and third step. Clearly, when the position range is 1, the hidden router is exactly located. The black portion in the figure is the gain achieved in accuracy: while on average, the position range of a hidden router decreases from 5.3 to 3.3 (-37%), the hidden routers exactly located grows from 7% to 14%. Fig. 5(b) shows the size of the position range as a fraction of the Traceroute trace. From the second to the third step, this fraction decreases on average from 0.32 to 0.19, i.e the final area identified by DRAGO as affected by hidden routers represents on average less than $\frac{1}{5}$ of the Traceroute trace containing these devices. These results are achieved efficiently thanks to the binary tree: the positioning of the detected hidden routers did not require the inspection of all the possible portions in each subsequence. Fig. 5(c) reports the distribution of the fraction of explored portions for the subsequences containing at least 2 elements: on average, only 57% of all the possible portions are explored.

Table II: INCR values.

INCR	Unique Transitions	Involved Traces
0	13,458	21,885
1	31,705	21,930
2	5,323	21,757
3	56	248
4	18	21
5	21	21
6	5	5

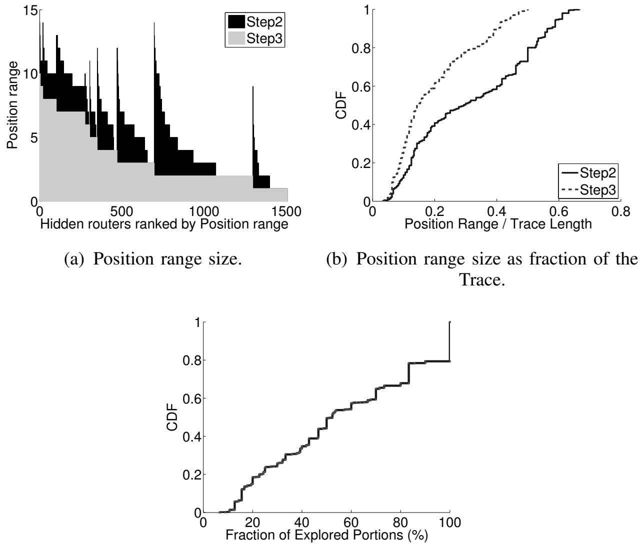
Table III: Hidden routers.

Hidden Routers	Unique Subsequences	Involved Traces
0	28,408	20,603
1	1,222	1,211
2	98	91
3	23	22
4	5	5

Finally, for the subset of hidden routers exactly located we have computed the hop distance from the Traceroute source (Fig. 6(a)) and destination (Fig. 6(b)): 70% of these devices are just one hop far from the destination. Some middleboxes do not decrement the TTL by default (like the Cisco firewall Adaptive Security Appliance [37]) or refresh the TTL of the incoming packets [38], then a portion of the detected hidden routers could be middleboxes located in the proximity of the destination.

V. Related Work

To the best of our knowledge, very few researchers have addressed the hidden routers problem. Sherwood *et al.* [27], [39] proposed a solution based on a novel Traceroute enhanced by the Record Route (RR) option to identify load balancers, anonymous routers, addresses owned by the same device and, possibly, hidden routers. The injected probes collect along the path additional IP addresses in the RR option. They used the disjunctive logic programming to merge the addresses stored in the RR option and the traceroute data uncovering 329 hidden routers (0.3% of all the discovered devices). DRAGO exploits a different IP option, the TS option, which provides an almost three times larger exploring range compared to the RR option (24 hops against 9 hops). In addition, the disjunctive logic programming is a computationally complex

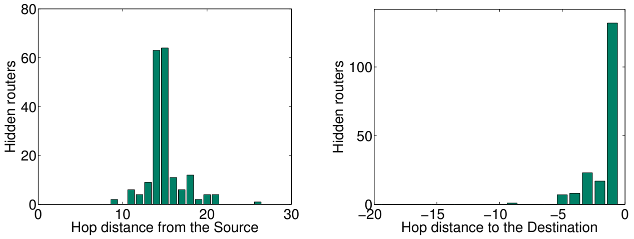


(a) Position range size.

(b) Position range size as fraction of the Trace.

(c) Explored portions in each subsequence.

Figure 5: Step 3 - Hidden routers positioning.



(a) Relative to the source. (b) Relative to the destination.

Figure 6: Positions of hidden routers exactly located.

solution [27] while our technique is much lighter. On the other hand, DRAGO is not able to identify any address of the hidden router and the same hidden router could be acknowledged multiple times. Note that the two techniques may be also profitably merged: hidden routers could be first recognized and located with DRAGO and then, the solution proposed in [27], [39] could be applied to identify addresses of the detected hidden routers.

VI. Conclusion

In this work we presented and evaluated DRAGO, an active probing technique able to detect, quantify and locate hidden routers in Traceroute IP paths. The experimental campaign demonstrates how these devices are not so rare: we inferred the presence of hidden routers in 6% of Traceroute traces. DRAGO finds the exact position of the 14% of the detected hidden routers and shrinks the area affected by hidden routers to the one fifth of the Traceroute trace.

Acknowledgements

The work of the authors is partially funded by the PLATINO (PON01_01007) and S^2−MOVE (PON04a3_00058) projects financed by MIUR. The authors wish to thank the reviewers for their helpful comments and suggestions.

References

[1] V. Jacobson *et al.*, “Traceroute,” ftp://ftp.ee.lbl.gov/traceroute.tar.gz., 1989.

[2] R. Steenbergen, “A practical guide to (correctly) troubleshooting with traceroute,” *North American Network Operators Group*, pp. 1–49, 2009.

[3] Y. Shavitt and E. Shir, “Dimes: Let the internet measure itself,” *ACM SIGCOMM CCR*, vol. 35, no. 5, pp. 71–74, 2005.

[4] K. Claffy, Y. Hyun, K. Keys, M. Fomenkov, and D. Krioukov, “Internet mapping: from art to science,” in *CATCH’09*. IEEE, 2009, pp. 205–211.

[5] K. Chen, D. Choffnes, R. Potharaju, Y. Chen, F. Bustamante, D. Pei, and Y. Zhao, “Where the sidewalk ends: Extending the internet as graph using traceroutes from p2p users,” in *ACM CONEXT*, 2009.

[6] P. Marchetta, P. Mérindol, B. Donnet, A. Pescapé, and J. Pansiot, “Quantifying and Mitigating IGMP Filtering in Topology Discovery,” in *IEEE GLOBECOM*, 2012.

[7] B. Donnet and T. Friedman, “Internet topology discovery: a survey,” *IEEE Comm. Surveys and Tutorials*, vol. 9, no. 4, 2007.

[8] N. Spring, R. Mahajan, and D. Wetherall, “Measuring isp topologies with rocketfuel,” *ACM SIGCOMM CCR*, vol. 32, no. 4, 2002.

[9] R. Bush, J. Hiebert, O. Maennel, M. Roughan, and S. Uhlig, “Testing the reachability of (new) address space,” in *ACM SIGCOMM INM*, 2007.

[10] R. Beverly, A. Berger, and G. Xie, “Primitives for active internet topology mapping: Toward high-frequency characterization,” in *ACM SIGCOMM IMC*, 2010, pp. 165–171.

[11] Z. Mao, D. Johnson, J. Rexford, J. Wang, and R. Katz, “Scalable and accurate identification of AS-level forwarding paths,” in *INFOCOM 2004*, vol. 3. IEEE, 2004, pp. 1605–1615.

[12] A. Botta, W. De Donato, A. Pescapé, and G. Ventre, “Discovering topologies at router level: Part II,” in *GLOBECOM’07*. IEEE, 2007.

[13] P. Marchetta, P. Mérindol, B. Donnet, A. Pescapé, and J. Pansiot, “Topology discovery at the router level: a new hybrid tool targeting isp networks,” *JSAC*, vol. 29, no. 9, pp. 1776–1787, 2011.

[14] E. Katz-Bassett, H. Madhyastha, J. John, A. Krishnamurthy, D. Wetherall, and T. Anderson, “Studying black holes in the internet with hubble,” in *USENIX NSDI*, 2008, pp. 247–262.

[15] M. Zhang, C. Zhang, V. Pai, L. Peterson, and R. Wang, “Planetseer: Internet path failure monitoring and characterization in wide-area services.” OSDI, CA, 2004.

[16] Y. Zhang, Z. Mao, and M. Zhang, “Effective diagnosis of routing disruptions from end systems,” in *USENIX NSDI*, 2008, pp. 219–232.

[17] R. Mahajan, M. Zhang, L. Poole, and V. Pai, “Uncovering performance differences among backbone isps with netdiff,” in *USENIX NSDI*. USENIX Association, 2008, pp. 205–218.

[18] H. Madhyastha, T. Isdal, M. Piatek, C. Dixon, T. Anderson, A. Krishnamurthy, and A. Venkataramani, “iplane: An information plane for distributed services,” in *USENIX OSDI*, 2006, pp. 367–380.

[19] B. Gueye, A. Ziviani, M. Crovella, and S. Fdida, “Constraint-based geolocation of internet hosts,” *Networking, IEEE/ACM Transactions on*, vol. 14, no. 6, pp. 1219–1232, 2006.

[20] B. Wong, I. Stoyanov, and E. Sirer, “Octant: A comprehensive framework for the geolocalization of internet hosts,” in *USENIX NSDI*, 2007.

[21] M. Gunes and K. Sarac, “Resolving anonymous routers in internet topology measurement studies,” in *INFOCOM 2008*. IEEE, 2008.

[22] M. Luckie, A. Dhamdhere, D. Murrell *et al.*, “Measured impact of crooked traceroute,” *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 1, pp. 14–21, 2011.

[23] X. Jin, W. Yiu, S. Chan, and Y. Wang, “Network topology inference based on end-to-end measurements,” *JSAC*, vol. 24, no. 12, 2006.

[24] B. Yao, R. Viswanathan, F. Chang, and D. Waddington, “Topology inference in the presence of anonymous routers,” in *INFOCOM 2003*, vol. 1. IEEE, 2003, pp. 353–363.

[25] B. Augustin, X. Cuvellier, B. Orgogozo, F. Viger, T. Friedman, M. Latapy, C. Magnien, and R. Teixeira, “Avoiding traceroute anomalies with paris traceroute,” in *ACM SIGCOMM IMC*, 2006, pp. 153–158.

[26] P. Marchetta, W. de Donato, and A. Pescapé, “Detecting third-party addresses in traceroute ip paths,” in *ACM SIGCOMM*, 2012.

[27] R. Sherwood, A. Bender, and N. Spring, “Discarte: a disjunctive internet cartographer,” *ACM SIGCOMM CCR*, vol. 38, no. 4, 2008.

[28] E. Rosen, A. Viswanathan, R. Callon *et al.*, “Multiprotocol label switching architecture,” *RFC 3031*, 2001.

[29] IP Address Hitlist, “PREDICT ID USC-LANDER internet-address-hitlist-it47w-20120427. 2010-03-29 to 2012-05-30,” http://www.isi.edu/ant/lander.

[30] J. Postel, “Internet Protocol,” RFC 791 (Standard), Internet Engineering Task Force, Sep. 1981. [Online]. Available: http://www.ietf.org/rfc/rfc791.txt

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

[32] E. Katz-Bassett, H. V. Madhyastha, V. K. Adhikari, C. Scott, J. Sherry, P. van Wesep, T. E. Anderson, and A. Krishnamurthy, “Reverse traceroute,” ser. NSDI’10, 2010, pp. 219–234.

[33] A. Ferguson and R. Fonseca, “Inferring router statistics with ip timestamps,” in *ACM CoNEXT Student Workshop*, 2010.

[34] W. de Donato, P. Marchetta, and A. Pescapé, “A Hands-on Look at Active Probing using the IP Prespecified Timestamp Option,” in *PAM 2012*. Springer, 2012, pp. 189–199.

[35] P. Marchetta and A. Pescapè, “Drago, source code,” *http://traffic.comics.unina.it/drago/*, 2012.

[36] T. Cymru, “IP to ASN mapping,” http://www.team-cymru.org/Services/ip-to-asn.html, 2012.

[37] Cisco Systems, “Asa/pix/fwsm: Handling icmp pings and traceroute.” http://www.cisco.com/image/gif/paws/15246/31.pdf.

[38] S. Zander, G. Armitage, and P. Branch, “Dynamics of the IP Time To Live field in Internet traffic flows,” *CAIA Tech. Rep. 070529A*.

[39] R. Sherwood and N. Spring, “Touring the internet in a tcp sidecar,” in *ACM SIGCOMM IMC*, 2006, pp. 339–344.