Dissecting Round Trip Time on the Slow Path with a Single Packet

Pietro Marchetta*, Alessio Botta*, Ethan Katz-Bassett†, and Antonio Pescapé*

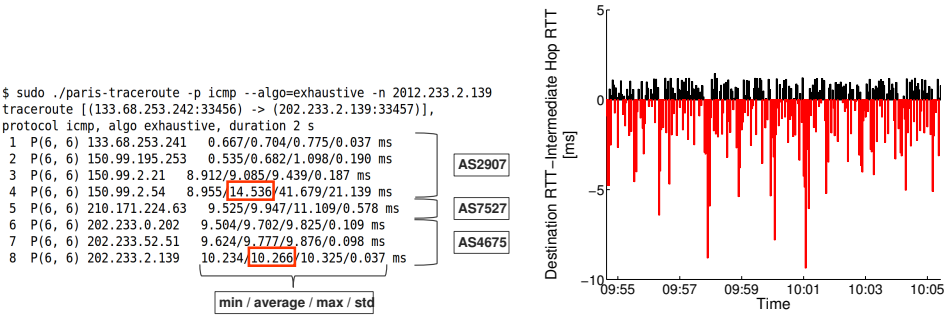
*University of Napoli Federico II, Napoli, Italy
†University of Southern California, Los Angeles, USA

Abstract. Researchers and operators often measure *Round Trip Time* when monitoring, troubleshooting, or otherwise assessing network paths. However, because it combines all hops traversed along both the forward and reverse path, it can be difficult to interpret or to attribute delay to particular path segments.
In this work, we present an approach using a single packet to dissect the RTT in chunks mapped to specific portions of the path. Using the IP Prespecified Timestamp option directed at intermediate routers, it provides RTT estimations along portions of the slow path. Using multiple vantage points (116 PlanetLab nodes), we show that the proposed approach can be applied on more than 77% of the considered paths. Finally, we present preliminary results for two use cases (home network contribution to the RTT and per-Autonomous System RTT contribution) to demonstrate its potential in practical scenarios.

1 Introduction and motivation

A common metric used to estimate the delay over a network path is the Round Trip Time (RTT) [1], defined as the length of time it takes to send a data packet toward a destination and receive its response. Monitoring RTT provides useful information about the network status when managing testbeds and operational networks [28]. However, an RTT sample comprises all the delays experienced by the data packet and its response along the forward and reverse path respectively, and it also includes the time the destination takes to inspect the incoming packet and generate the proper response. As a consequence, it can be difficult to interpret RTT values or tease apart the contributing factors.

From this point of view, dissecting the RTT into chunks related to specific portions of the network path may be helpful, making it possible to evaluate the relative impact of each subpath on the total experienced RTT. This approach is particularly useful in several scenarios. In a home network, one could isolate the impact of the home network on the RTT experienced toward a destination of interest, such as a website or network service. A large corporatation with multiple providers may want to evaluate the impact of its access networks when considering performance optimization and traffic engineering. Service providers may be interested in assessing if the ISP of a particular user has a great impact



(a) A sample traceroute trace from planet1.pnl.nitech.ac.jp.

(b) Difference between the average RTTs up to the destination and up to the last hop within AS2907.

Fig. 1: On the inaccuracy of traditional approaches useful for RTT dissecting.

on the RTT, thus potentially representing the main cause of poor performance perceived by the user.

Unfortunately, accurately dissecting RTT is not a trivial task, especially through active measurements. One possibility is to rely on the RTTs reported by traceroute, i.e. the time it takes to send the TTL-limited probe and receive the ICMP Time Exceeded reply. However, it is not uncommon to observe RTT of intermediate hops higher than the RTT of the destination, as reported in the sample trace of Fig. 1(a)[1]. Another possibility is to use the ping command to monitor both the RTT to an intermediate hop and to the destination. For example, let us assume that our goal is to evaluate the impact of the provider, AS2907 (SINET-AS), on the RTT experienced toward the destination. We monitored the RTTs up to the last hop within AS2907 (150.99.2.54) and the destination by issuing pairs of ICMP Echo Request packet probes closely in time with the ping command. We launched one probe pair every 200 ms for 10 minutes and computed the average RTT obtained in one second bins. Finally, we computed the difference between the average RTT to the destination and to the intermediate hop. Fig. 1(b) presents the results. For about half of the bins, the intermediate hop had an average RTT higher than the RTT of the destination, making it hard to understand how the intermediate hop contributes to overall latency. Preliminary analysis suggests the this problem holds even for sophisticated ping variants that control RTT variance [21].

The inaccuracy of the two methods described above is determined by specific factors: (i) due to path asymmetry [12], the intermediate hop may not be part of the reverse path from the destination, thus its RTT is not part of the RTT of the destination; (ii) the two RTT samples are obtained by employing two distinct packet probes that potentially experience different network conditions or paths;[2] (iii) the two solicited devices may require a different amount of time to inspect the probe and generate the response [11]; finally (iv) when using ping,

[1] This forward path is stable and unique, according to paris-traceroute [2].

[2] For example, due to load balancers located along the reverse paths [2].

the forward path up to the intermediate hop may not represent a subpath of the forward path toward the destination, since fowarding is destination-based.

In this work, we introduce a new approach to dissect the RTT experienced toward a given destination into two distinct chunks, using a single purposely crafted probe packet to avoid the complications introduced in the previous paragraph. Our approach uses the IP Timestamp option and needs an intermediate router that honors the option and appears on both the forward and reverse paths. In these cases, the technique dissects the RTT into (a) the time the probe spends between the source and an intermediate router (in both directions) and (b) the time the probe spends between the intermediate router and the destination (in both directions). While our approach requires a preliminary phase to identify compliant intermediate routers, it uses only widely adopted network diagnostic tools such as traceroute and ping.

Using multiple vantage points (116 PlanetLab nodes), we provide experimental results about the degree of applicability of our approach as well as case studies demonstrating its utility in practical scenarios.

2 Dissecting Round Trip Time

In this section, after a brief recap of the IP Prespecified Timestamp option, we describe the approach we propose to dissect the RTT in chunks.

Background. Although IP options headers [22] are not universally supported on the Internet [5, 9], researchers have used them as the basis for a number of recent measurement techniques [8, 14, 16, 17, 19, 20, 25, 26]. In this work, we use the IP Prespecified Timestamp option [22] (hereafter TS option) to dissect the RTT. This option lets the sender specify up to four IP addresses in the header of the packet, to request timestamps from the corresponding routers. We adopt the notation proposed by Sherry *at al.* [25]: X |ABCD refers to an ICMP Echo Request packet where X is the targeted destination and ABCD is the ordered list of prespecified IPs from which a timestamp is requested. Note that the position of each prespecified address in the ordered list ABCD is essential since it implies that B cannot insert its own timestamp before A, C before B, and so on. Typically, when the packets are not filtered along the path [9], the incoming option is replicated by the destination inside the ICMP Echo Reply. The TS option has been used to infer aliases [19, 25], to infer routers statistics such as traffic shape and CPU load [8], to identify third-party addresses and hidden routers in traceroute trace [17, 20], to reconstruct reverse paths [14], to infer link latency [24], and to identify symmetric link traversal [15].

Dissecting RTT. Our approach makes it possible to dissect the RTT toward a destination that (i) provides at least one timestamp when probed with D |DDDD and (ii) is not an extra-stamper [25], i.e. it does not provide more than one timestamp when probed with D |DXXX where X is an IP address surely not involved on the traversed path. On these paths, we can dissect the RTT into chunks by exploiting a *compliant* router located along the path (see Fig. 2):

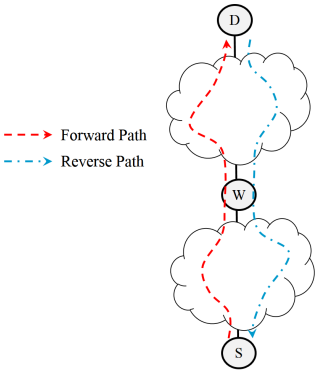


Fig. 2: Baseline scenario (S: source - W: compliant node - D: destination).

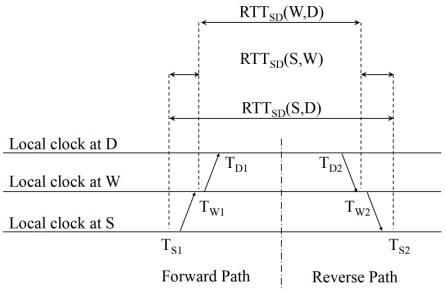


Fig. 3: Timestamps collected with D|WDDW and related RTT chunks.

a compliant node W (i) is part of both the forward and reverse path under investigation; (ii) honors the TS option and provides standard timestamps [22], i.e milliseconds since midnight UT; (iii) provides timestamps both on the forward and reverse path. Hereafter we adopt the following notation: $\mathrm{RTT}_{S,D}$(X, Y) is the time taken by probes sent from the source S to the destination D to travel from X to Y on the forward path and from Y to X on the reverse path. This is a portion of the RTT of the entire path, i.e. $\mathrm{RTT}_{S,D}$(S, D).

Let W be a compliant node between the source S and the destination D. Besides $\mathrm{RTT}_{S,D}$(S, D), our approach estimates $\mathrm{RTT}_{S,D}$(S, W) and $\mathrm{RTT}_{S,D}$(W, D) by using the same single-packet probe. To this end, we send a D|WDDW probe from S to D. Once S receives the reply, six timestamps are available: (a) the sending and receiving time at the source (T_{S1} and T_{S2}); (b) the timestamp provided by W along the forward (T_{W1}) and reverse path (T_{W2}); (c) the two timestamps provided by the targeted destination D (T_{D1} and T_{D2}). These timestamps allow us to easily compute the RTT chunks (see Fig. 3 as reference): $\mathrm{RTT}_{S,D}$(S, D) as T_{S2}-T_{S1}, $\mathrm{RTT}_{S,D}$(W, D) as T_{W2}-T_{W1} and $\mathrm{RTT}_{S,D}$(S, W) as $\mathrm{RTT}_{S,D}$(S, D)-$\mathrm{RTT}_{S,D}$(W, D) .[3] When the destination provides only one timestamp when probed with D|DDDD, we send probe packets formatted like D|WDWW, rather than D|WDDW, to dissect the RTT.

To identify the compliant nodes and to monitor the path, we use widely adopted network diagnostic tools such as traceroute and ping: the ping option `-T tsprespec` sends ICMP Echo Request packets with a customized TS option.

The slow path. Packets can traverse a router either through the *fast* (hardware) or the *slow* (route processor/software) path. The IP option on our probes causes routers to inspect them and process them on the slow path. Previous work showed that IP options traffic experiences higher RTT, jitter, and packet loss,

[3] Note how it would be possible to estimate also several one way delays: from S to D (T_{D1}-T_{S1}), D to S (T_{S2}-T_{D2}), S to W (T_{W1}-T_{S1}), W to D (T_{D1}-T_{W1}), D to W (T_{W2}-T_{D2}) and W to S (T_{S2}-T_{W2}). However, unlike the RTT considered in this paper, one way delays are potentially biased if clocks at the various nodes are not properly synchronized, a common case in the Internet.

compared to traffic without IP options [10]. Ferguson *et al.* [8] recently observed that the processing time of packets with the TS option depends on the status of the router (traffic and CPU load). Accordingly, the estimated RTTs provide insight into the current condition of network links and routers, a different view of network path performance.

Accuracy concerns. Concerns about the accuracy of the estimated RTTs may arise since we exploit timestamps provided by distinct network nodes potentially not synchronized. However, we compute each RTT using only the timestamps provided by a single router's clock. Accordingly, any clock offsets do not affect the estimated RTTs. Our measurements are subject to local clock drift, but we assume this impact is negligible over the short duration of a typical RTT.

3 Evaluation

In this section we first describe the results of an experimental campaign aiming at evaluating the applicability of the proposed approach. Then, we describe two use cases to show the utility of the proposed approach.

Degree of Applicability. We conducted a study to evaluate how many nodes per path will allow our approach to dissect the RTT (i.e. are compliant). To identify compliant nodes on a path between a source S and a destination D, we first need to discover all the nodes along the path. To this end, we collect an ICMP traceroute from S toward D. Let us suppose that the destination D provides two timestamps when probed with D |DDDD. For each discovered address Y, we send two packet probes D |YDDY and D |DYYY: if D |YDDY collects four timestamps, then Y is a compliant node. Indeed, four timestamps imply that Y inserted the first timestamp along the forward path (otherwise, D would not have been able to insert its own timestamp), and Y inserted its second timestamp along the reverse path (because the destination D inserted its timestamp before).[4] Non-compliant nodes (i) simply ignore the TS option (D |YDDY and D |DYYY collect none and one timestamp, respectively) or (ii) provide a timestamp only on the forward path (D |YDDY and D |DYYY collect between two and three timestamps and one timestamp respectively) or (iii) provide a timestamp only on the reverse path (D |YDDY and D |DYYY collect one and more than one timestamp, respectively). We refer to the latter two cases as *forward* and *backward stampers*. Forward stampers are nodes that do not appear on the reverse path while backward stampers are more challenging to explain: these nodes are discovered along the forward path but insert a timestamp only when traversed on the reverse path. Load balancing and off-path addresses [13, 17, 18] may explain this behavior.[5] When the destination provides only one timestamp,

[4] Previous work exploited a similar approach to assess symmetric link traversal [15,16].

[5] Standard-compliant routers set as source address of Time Exceeded replies the address associated to the outgoing interface causing Traceroute to report addresses associated to interfaces not actually traversed by the traffic sent to the Traceroute destination [13, 17, 18].

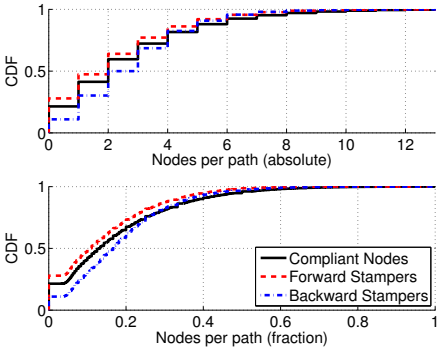


Fig. 4: Compliant nodes per path.

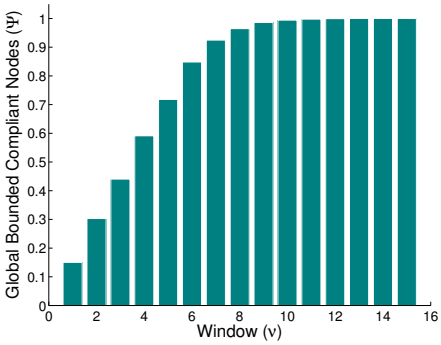


Fig. 5: Compliant nodes relative position.

we make use of D |YDYY probes instead of D |YDDY. In this case, a node is compliant when D |YDYY collects at least three timestamps.

To generate a hitlist of suitable destinations, we extracted the addresses that provided at least one timestamp when probed with D |DDDD in a large-scale experimental campaign from our previous work [5]. Of 1.7M IP addresses probed, 36% replied providing timestamps. From these addresses, we randomly selected one representative IP for each AS [4]. The final hitlist comprises 3,133 distinct ASes, including all Tier-1 ISP networks[6] and 35 out of 50 top-10 ASes for each region, according to the APNIC weekly routing table report. We then performed another experimental campaign using 116 PlanetLab nodes [3] as vantage points (VPs). Each VP made the following steps for each destination of the hitlist: first, it sent two probes, D |DDDD and D |DXXX, to check if the destination is still responsive and is not an extra-stampers (see Sec. 2). Second, it performed a traceroute toward the destination. Third, for each address Y discovered along the path, it sent a D |YDDY (or D |YDYY depending on the number of timestamps provided by the destination) and D |DYYY. After removing about 90 K paths toward extra-stamping destinations and 50 K paths toward addresses unresponsive for a subset of vantage points due to in-transit filtering, our final dataset comprises 223,548 distinct paths.

Fig. 4 reports the compliant nodes observed per path. Ideally, we would like all intermediate routers to be compliant, in order to split the RTT into all the available chunks. On the other hand, just a single compliant node (W) allows us to split the RTT into $\text{RTT}_{S,D}$(S, W) and $\text{RTT}_{S,D}$(W, D), thus providing much more information on the network status than a classic RTT estimation. We found that about 77.4% of the paths contain at least one compliant node and 27.3% contain more than four compliant nodes. On average, we observed 2.5 compliant nodes, 2.1 forward stampers, and 2.7 backward stampers per path. This result means that, on average, about 17% of the nodes in each scanned path are compliant.

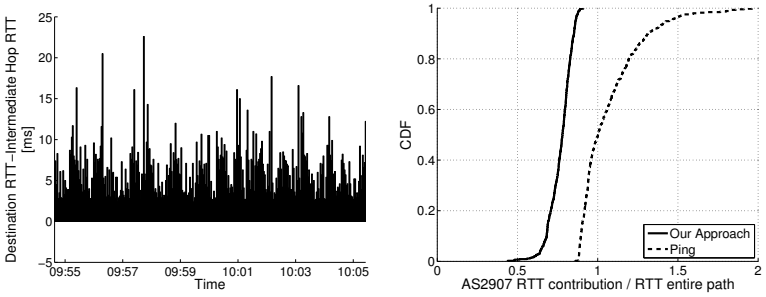
[6] `http://en.wikipedia.org/wiki/Tier_1_network#List_of_tier_1_networks`. August 1, 2013.

Since compliant nodes represent meeting points between the forward and reverse path and most paths in the Internet are asymmetric at the router level [12, 23], we expect most compliant nodes to appear close to the source or the destination. Our experimental results partially confirm this hypothesis. Let Ω be the set of traceroute traces and p a particular trace comprising n nodes (a_1, .. , a_i, .. , a_n). Also, let C be the overall number of compliant nodes contained in the dataset. To investigate the position of the compliant nodes, we used a *window* ν to compute the *bounded compliant nodes* $\Phi(p, \nu)$ representing the number of compliant nodes on the path p appearing within ν hops from the source *or* the destination, i.e the compliant nodes contained in (a_1, .. a_ν) and ($a_{n-\nu}$, .. a_n). The *global bounded compliant nodes* $\Psi(\nu) = \frac{\sum_{p \in \Omega} \Phi(p,\nu)}{C}$ represent the global fraction of compliant nodes contained within ν hops from the source or the destination when considering all the paths. Fig. 5 depicts how the global bounded compliant nodes varies with ν. If the hypothesis is true, then the global bounded compliant nodes should quickly tend to one. The figure shows evident though not sharp growth: about 72% of all the compliant nodes occur within 5 hops from the source or the destination, with about 15% appearing just one hop after the source or before the destination. These results confirm that the majority of the compliant nodes are located near the two end points of the paths, while there is also a significant percentage of compliant nodes in the middle of the paths.

Applications. We now report preliminary potential use cases of the proposed approach.

Per-Autonomous System RTT contribution. Our approach can isolate the RTT contribution of entire ASes. Consider again the trace in Fig. 1(a). Our goal is to isolate the RTT contribution of the provider network, AS2907. To this end, we monitored the path by using both the ping command and our approach (the last hop within AS2907, 150.99.2.54, is a compliant node). As anticipated in Sec. 1, when using ping to estimate the RTT up to the last hop within AS2907 and up to the destination with packet probes sent closely in time, we observed inconsistent results, as reported in Fig. 1(b). Often, the average RTT up to the intermediate hop is higher than the RTT up to the destination (see the negative difference values in Fig. 1(b)). Our approach, instead, always provides coherent results. As shown in Fig. 6(a), the estimated contribution of the AS2907 is always a fraction of the whole RTT. Results obtained with ping do not provide any meaningful information about the impact of the AS2907 on the end-to-end performance. As shown in Fig. 6(b), according to ping, the AS2907 RTT contribution represents on average 106% of the whole RTT, an unreasonable result. On the other hand, thanks to our approach, we can conclude that the AS2907 RTT contribution on the slow path is on average 76.8% of the whole RTT. The packet probes spent more than two-third of the time within the provider network.

Our approach also isolates the RTT contribution of a target AS network when the first hop within this AS is a compliant node. In the dataset collected to evaluate the applicability, the last hop within the provider AS (the first hop within the targeted AS) is a compliant node in 44,846 (22,236) paths, about 20% (9.95%) of the paths.



(a) Difference between the average RTTs up to the destination and up to the last hop within AS2907.

(b) AS2907 RTT contribution as fraction of the entire RTT.

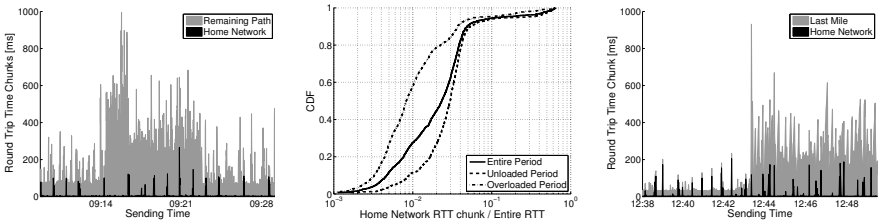
Fig. 6: Isolating the RTT contribution of AS2907 over the path of Fig.1(a).

Home network contribution to the RTT. The impact of home networks on Internet performance has recently attracted an increasing interest from the research community [6, 27]. However, classic diagnostic tools or simply probing the home gateway are not always able to reliably state if the home network is the cause of the performance degradation [7].

When the home gateway behaves as a compliant node, our approach allows us to evaluate the RTT toward any destination, as well as the contribution of the home network, by using a single packet probe. [7] As a case study, we monitored the RTT toward a top-ranked Italian journal website (repubblica.it). The monitored home network is connected to the Internet via an ADSL connection provided by Telecom Italia. The laptop in charge of monitoring is connected via Wi-Fi to a NETGEAR DGN2200v3, a common commercial modem-router compliant with our approach. To monitor the RTT, we used D $\big|$WDDW packet probes where W is the private address of the modem-router: We approximate the home network contribution as $\mathrm{RTT}_{S,D}$(S, W).

Fig. 7(a) shows the trend over time of the RTT chunks. In the beginning, the home network is unloaded. However, from 9:14 to 9:23, another Wi-Fi connected host started downloading and uploading large files through the Internet. During the overloaded period, the RTT grows in median by 356% (from 69.8 ms to 249 ms) but the home network played just a marginal role (see Fig. 7(b)). On average, packets spent 4.7% and 2.6% of the entire RTT within the home network during the unloaded and overloaded period, respectively. At the same time, we observed spurious latency spikes inside the home network probably caused by the packet-by-packet impact of contention-induced transmission delays over the wireless link (these spikes disappear on the wired connection). In the worst cases, the spikes represent more than 60% of the total RTT experienced in both

[7] In these experiments, the precise border of the home network clearly depends on when and how the home router handles the IP option. For instance, if the home router inserts its own timestamp before putting the probe on an overloaded buffer (an instance of home network bufferbloat), such buffering delay is not included in the home network contribution.



(a) RTT chunks over time. Another host transferred large files from 9:14 to 9:23.

(b) Home network RTT contribution as a fraction of the entire RTT.

(c) Home network RTT contribution over last mile.

Fig. 7: Home network RTT contribution toward repubblica.it monitored through a wireless link and an ADSL connection.

the unloaded and overloaded period. These results suggest that the stable performance degradation observed during the overloaded period is not caused by the home network but by congestion of the last mile.[8] Indeed, by replicating the experiment while monitoring the RTT on the last mile and isolating the home network contribution, we observed that downloading and uploading large files through the Internet does not affect the intra-home network delay while it determines a dramatic growth of the delay on the last mile (see Fig. 7(c)).

4 Conclusion

We presented an approach using a single packet to accurately dissect the RTT on the slow path in chunks mapped to specific portions of the end-to-end path. We observed how using other techniques based on ping and traceroute to this end may provide misleading results. Our approach uses the IP Timestamp option and a compliant router along the path. A large-scale measurement study we performed from 116 vantage points comprising 223K traced paths showed that 2.5 router per path on average are compliant. As preliminary evidence of the use of our approach, we presented two case studies, showing how it allows us to isolate the RTT contribution of the home network and of an entire AS.

Acknowledgements. This work is partially funded by the MIUR projects: PLATINO (PON01_01007), SMART HEALTH (PON04a2_C), and S^2−MOVE (PON04a3_00058).

References

1. G. Almes, S. Kalidindi, and M. Zekauskas. A round-trip delay metric for IPPM. Technical report, RFC 2681, september, 1999.
2. B. Augustin et al. Avoiding traceroute anomalies with Paris traceroute. In *ACM SIGCOMM IMC*, pages 153–158. ACM, 2006.
3. A. Bavier et al. Operating system support for planetary-scale network services. In *NSDI*, 2004.
4. T. Cymru. `http://www.team-cymru.org/Services/ip-to-asn.html`, 2012.
5. W. de Donato, P. Marchetta, and A. Pescapé. A hands-on look at active probing using the IP prespecified timestamp option. In *PAM*, 2012.

[8] The physical connection between a customer's home and the DSLAM or the CMTS.

6. L. DiCioccio, R. Teixeira, M. May, and C. Kreibich. Probe and pray: Using UPnP for home network measurements. In *PAM*, 2012.
7. L. DiCioccio, R. Teixeira, and C. Rosenberg. Impact of home networks on end-to-end performance: controlled experiments. In *ACM HomeNets*, 2010.
8. A. Ferguson and R. Fonseca. Inferring router statistics with IP timestamps. In *ACM CoNEXT Student Workshop*, 2010.
9. R. Fonseca, G. Porter, R. Katz, S. Shenker, and I. Stoica. IP options are not an option. *Univ. of California, Berkeley*, 2005.
10. P. Fransson and A. Jonsson. End-to-end measurements on performance penalties of IPv4 options. In *IEEE GLOBECOM*, 2004.
11. R. Govindan and V. Paxson. Estimating router ICMP generation delays. In *PAM*, 2002.
12. Y. He, M. Faloutsos, and S. Krishnamurthy. Quantifying routing asymmetry in the Internet at the AS level. In *IEEE GLOBECOM*, 2004.
13. Y. Hyun, A. Broido, et al. On third-party addresses in traceroute paths. 2003.
14. E. Katz-Bassett et al. Reverse traceroute. In *NSDI*, 2010.
15. H. V. Madhyastha. *An information plane for Internet applications*. UW dissertation, 2008.
16. H. V. Madhyastha, E. Katz-Bassett, T. Anderson, A. Krishnamurthy, and A. Venkataramani. iPlane Nano: Path prediction for peer-to-peer applications. In *NSDI*, 2009.
17. P. Marchetta, W. de Donato, and A. Pescapé. Detecting third-party addresses in traceroute traces with IP timestamp option. In *PAM*, 2013.
18. P. Marchetta, V. Persico, E. Katz-Bassett, and A. Pescapé. Don't trust traceroute (completely). In *ACM CoNEXT Student workshop*, 2013.
19. P. Marchetta, V. Persico, and A. Pescapé. Pythia: yet another active probing technique for alias resolution. In *ACM CoNEXT*, pages 229–234, 2013.
20. P. Marchetta and A. Pescapè. Drago: Detecting, quantifying and locating hidden routers in traceroute IP paths. In *IEEE Global Internet Symposium*, 2013.
21. C. Pelsser, L. Cittadini, S. Vissicchio, and R. Bush. From Paris to Tokyo: On the suitability of ping to measure latency. In *IMC 2013*, pages 427–432. ACM, 2013.
22. J. Postel. Internet protocol: DARPA Internet program protocol specification. *RFC 791*, 1981.
23. Y. Schwartz, Y. Shavitt, and U. Weinsberg. On the diversity, stability and symmetry of end-to-end Internet routes. In *IEEE INFOCOM Workshops*, 2010.
24. J. Sherry. Applications of the IP timestamp option to Internet measurement. *Undergraduate Honor Thesis*, 2010.
25. 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.
26. R. Sherwood and N. Spring. Touring the Internet in a TCP sidecar. In *ACM SIGCOMM IMC*, pages 339–344. ACM, 2006.
27. S. Sundaresan, W. de Donato, N. Feamster, R. Teixeira, S. Crawford, and A. Pescapè. Broadband Internet performance: A view from the gateway. *SIGCOMM 2011*, 41(4):134, 2011.
28. H. Zeng, P. Kazemian, G. Varghese, and N. McKeown. A survey on network troubleshooting. Technical report, TR12-HPNG-061012, Stanford University, 2012.