Towards DisNETPerf: a Distributed Internet Paths Performance Analyzer

S. Wassermann[†‡], P. Casas[†], B. Donnet[‡]
† FTW Vienna, ‡ Université de Liège
sarah.wassermann@student.ulg.ac.be; casas@ftw.at; benoit.donnet@ulg.ac.be [*]

ABSTRACT

For more than 25 years now, `traceroute` has demonstrated its supremacy for network-path measurement, becoming the most widely used Internet path diagnosis tool today. A major limitation of `traceroute` when the destination is not controllable by the user is its inability to measure *reverse* paths, i.e., the path from a destination back to the source. Proposed techniques to address this issue rely on IP address spoofing, which might lead to security concerns. In this paper we introduce and evaluate DisNETPerf, a new tool for locating probes that are the closest to a distant server. Those probes are then used to collect data from the server point-of-view to the service user for path performance monitoring and troubleshooting purposes. We propose two techniques for probe location, and demonstrate that the reverse path can be measured with very high accuracy in certain scenarios.

1. WHY DISNETPERF?

Internet-scale services such as YouTube and Facebook are provisioned from geo-distributed servers, using large Content Delivery Networks (CDNs). While user-requests are normally redirected to the closet servers (in terms of latency), internal CDN load-balancing policies may select servers which lie at hundreds of milliseconds from customers, potentially impacting their Quality of Experience (QoE). For instance, previous work [1] reports a real case study in which a Google load balancing policy results in a drop in the throughput of YouTube flows, impacting the QoE of a large number of customers watching videos at peak-load times in an operational ISP. In this event, the minimum RTT to YouTube servers increases by more than 300%, corresponding to a different load balancing policy selecting servers at much farther locations [2]. The anomaly could be due to either wrongly dimensioned/faulty YouTube servers, or to the presence of heavy network congestion at peak time in the servers-to-customers paths.

Having a tool that can measure the performance of these paths becomes paramount for the ISP to diagnose and troubleshoot the detected performance degradation [3]. Performing `traceroute` measurements from the servers towards the monitoring vantage point cannot be done in practice, as YouTube servers are not under the control of the ISP. A solution to this problem has been proposed in the past, known as *reverse traceroute* [4]; however, the proposed approach heavily relies on IP spoofing and IP Record Route Option, both being not necessarily allowed everywhere [5, 6] and causing potential security concerns.

In this paper we introduce *DisNETPerf*, a Distributed Internet Paths Performance Analyzer, that can monitor any Internet path using the RIPE Atlas framework [7] and standard `traceroute` measurements.

2. DISNETPERF

Given a certain content server with IP address IP_s, and a destination customer with IP address IP_d, DisNETPerf locates the closest RIPE Atlas probe [7] to IP_s, namely IP_c, and periodically runs `traceroute` measurements from IP_c to IP_d, collecting different path performance metrics such as RTT per hop, end-to-end RTT, etc. This data might then be used to troubleshoot paths from the content server (mimicked by IP_c) to the target customer. Fig. 1 depicts the overall idea. It uses a combined topology- and delay-based distance, as probes are located first by AS (BGP routing proximity allowing to select probes in the same AS as IP_s) and then by propagation delay (for selecting the closest probe to IP_s). Note that DisNETPerf is not strictly tied to RIPE Atlas, but can be used with any other distributed measurement framework such as CAIDA's Ark [8] or PlanetLab [9].

We evaluate two probe selection approaches for DisNETPerf, partially proposed in the literature [10, 11,

---

[*] This work has been supported by the European Commission through the Seventh Framework Grant Agreements mPlane (FP7-318627).

CoNEXT Student Workshop'15, December 01 2015, Heidelberg, Germany.
Copyright 2015 ACM. ISBN 978-1-4503-4066-3/15/12 ...$15.00.
DOI: http://dx.doi.org/10.1145/2842665.2843570

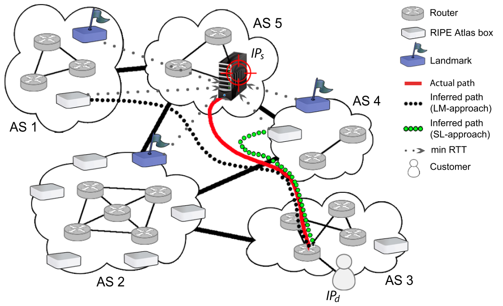
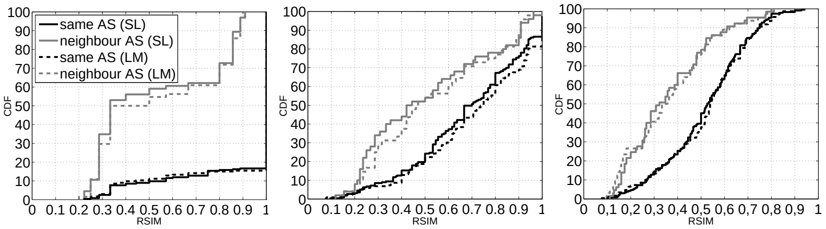


Figure 1: DisNETPerf overview.



(a) AS level (b) PoP level (c) IP level

Figure 2: Probe selection evaluation, based on RSIM.

12]: the *smallest latency* (SL) approach and the *landmark* (LM) approach. In both cases, we start by determining whether RIPE Atlas probes are located and available in the same AS as the content server IP_s. If this is not the case, we look for probes in the neighbor ASes of IP_s. Neighborhood information is obtained through AS relationships [13]. Once RIPE Atlas probes have been identified, the selection of IP_c can start.

In the SL-approach, we run 10 standard `ping` measurements from the probes towards IP_s. The probe with the smallest minimum RTT to IP_s is then elected as the representative probe of the content server, i.e., IP_c. In the LM-approach, we first start by grouping the identified RIPE Atlas probes in two groups: the *landmarks* and the probes that can be elected as IP_c. Landmarks represent a base location reference for all the probes, as inter-probe distances are measured using them as origin. Landmarks are chosen randomly among all the candidate-probes. We then run 10 `ping` measurements from those landmarks towards IP_s and all the other identified probes belonging to the other group. For each pinged IP address, we build a feature vector d containing the minimum RTT from each landmark to this IP address. We select IP_c as the probe which feature vector is the most similar to the one of IP_s, according to the normalized distance $D_{ij} = \frac{1}{K}\sum_{l=1}^{K} |d_{il} - d_{jl}|$, where K is the number of landmarks providing a RTT for both IP_i and IP_j, and d_{il} is the minimum RTT between IP_i and landmark l. When D_{ij} is small, we assume that IP_i and IP_j are close to each other. In the evaluations next, we select 20 landmarks for each IP_s.

We say that IP_c is a *good* probe w.r.t. IP_s and IP_d if the path from IP_c to IP_d is highly similar to the path from IP_s to IP_d. Similarly to Hu and Steenkiste [14], we define path similarity as the fraction of common links among both paths, using the *Route Similarity* index (RSIM), a value in the interval [0, 1], defined as $RSIM(IP_c, IP_s, IP_d) = \frac{2 \times C_{links}(IP_c,\, IP_s,\, IP_d)}{T_{links}(IP_c,\, IP_s,\, IP_d)}$, where C_{links} refers to the number of links shared in common by both paths, and T_{links} to the total number of links. A high RSIM indicates a high similarity between the considered paths. We consider links at the AS level (IP2AS mapping based on Maxmind [15]), PoP level (IP2PoP mapping based on iPlane [16]), and IP level.

Fig. 2 reports evaluation results in terms of path similarity. We use RIPE Atlas probes as source and destination (i.e., IP_s and IP_d) so as to compute the real path (i.e., the ground-truth) between servers and customers. We randomly select 300 RIPE Atlas source probes IP_{s_i}, and consider a single fixed destination probe IP_d. For each source IP_{s_i} we run DisNETPerf to locate the closest probe IP_{c_i}, obtain both the ground truth path $IP_{s_i} \rightarrow IP_d$ and the DisNETPerf path $IP_{c_i} \rightarrow IP_d$, and compute the RSIM index $RSIM(IP_{c_i}, IP_{s_i}, IP_d)$.

We compute RSIM at the AS level, PoP level, and IP level, and plot the resulting CDFs. Results are reported for the two approaches and for two different groups, the first one in which IP_{c_i} and IP_{s_i} are located in the same AS, and the second one in which IP_{c_i} is located in a neighbor AS. At the AS-level, the case of same AS colocation results in near optimal results. Nevertheless, we observe that about 40% of the tests carried out for the SL-approach and 45% of the tests performed for the LM-approach in the non-collocated scenario yield a RSIM index ≥ 0.5. Note that the most relevant segment of the path to monitor for troubleshooting purposes is the one closer to the customer (where problems generally occur), thus a RSIM of 0.5 is actually very good.

At the PoP-level, the RSIM index computed using both approaches is ≥ 0.5 for about 50% of the tests when IP_{c_i} and IP_{s_i} are in different ASes, and for more than 80% of the tests when they are in the same AS. At the IP-level, only about 20% of the tests yield an RSIM index ≥ 0.5 when IP_{c_i} and IP_{s_i} are not colocated in the same AS. However, IP-level paths are generally less relevant, and we plan to evolve to router-based paths [17]. In general, we note that the results for both SL- and LM-approaches are comparable and highly similar. Finally, we observed that probes selected by DisNETPerf using the SL-approach generally correspond to paths with the highest similarity to the ground-truth ones: in more than 80% of the performed tests, $RSIM(IP_{c_i}, IP_{s_i}, IP_d)$ results in the highest RSIM index among all the selected candidates when considering the AS-level, which is the most relevant level for our purposes. Indeed, we want to detect which AS on the Internet path from servers to customers might be responsible for the performance degradation, and thus do not take into account the intra-AS routing.

3. REFERENCES

[1] P. Casas, A. D'Alconzo, P. Fiadino, A. Bär, A. Finamore, and T. Zseby, "When YouTube does not work. analysis of QoE-relevant degradation in Google CDN traffic," *IEEE Transactions on Network and Service Management*, vol. 11, no. 4, pp. 441–457, December 2014.

[2] P. Fiadino, A. D'Alconzo, A. Bär, a. Finamore, and P. Casas, "On the detection of network traffic anomalies in content delivery network services," in *Proc. International Teletraffic Congress (ITC)*, September 2014.

[3] R. Krishnan, H. V. Madhyastha, S. Srinivasan, S. Jain, A. Krishnamurthy, T. Anderson, and J. Gao, "Moving beyond end-to-end information to optimize CDN performance," in *Proc. ACM Internet Measurement Conference (IMC)*, November 2009.

[4] E. Katz-Bassett, H. Madhyastha, V. Adhikari, C. Scott, J. Sherry, P. van Wesep, A. Krishnamurthy, and T. Anderson, "Reverse traceroute," in *Proc. USENIX Symposium on Networked Systems Design and Implementations (NSDI)*, June 2010.

[5] R. Beverly, A. Berger, Y. Hyun, and k. claffy, "Understanding the efficacy of deployed Internet source address validation filtering," in *Proc. ACM Internet Measurement Conference (IMC)*, November 2010.

[6] R. Fonseca, G. M. Porter, R. H. Katz, S. Shenker, and I. Stoica, "IP options are not an option," University of California at Berkeley, Technical Report UCB/EECS-2005-24, December 2005.

[7] RIPE, "RIPE Atlas distributed active measurements framework," September 2015, see https://atlas.ripe.net/.

[8] CAIDA, "Archipelago (ark) measurement infrastructure," August 2015, see http://www.caida.org/projects/ark/.

[9] PlanetLab, "PlanetLab, an open platform for developing, deploying, and accessing planetary-scale services," August 2015, see https://www.planet-lab.org/.

[10] E. Katz-Bassett, J. P. John, T. Anderson, A. Krishnamurthy, Y. Chawathe, and D. Wetherall, "Towards IP geolocaiton using delay and topology measurements," in *Proc. ACM Internet Measurement Conference (IMC)*, October 2006.

[11] V. Padmanabhan and L. Subramanian, "An investigation of geographic mapping techniques for internet hosts," in *Proc. ACM SIGCOMM*, August 2001.

[12] Y. Liao, W. Du, and G. Leduc, "A lightweight network proximity service based on neighborhood models," in *Proc. IEEE Symposium on Communications and Vehicular Technology in the Benelux (SCVT)*, November 2015.

[13] The CAIDA UCSD, "AS relationships," June 2015, http://data.caida.org/datasets/as-relationships/serial-1/20150601.as-rel.txt.bz2.

[14] N. Hu and P. Steenkiste, "Quantifying Internet end-to-end route similarity," in *Proc. Passive and Active Measurement Workshop (PAM)*, April 2006.

[15] MaxMind, "Geolocation and online fraud prevention from MaxMind," June 2015, see http://download.maxmind.com/download/geoip/database/asnum/GeoIPASNum.dat.gz.

[16] H. V. Madhyastha, T. Isdal, M. Piatek, C. Dixon, T. Anderson, A. Krishnamurthy, and A. Venkataramani, "iPlane: An information plane for distributed services," in *Proc. USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, November 2006.

[17] K. Keys, "Internet-scale IP alias resolution techniques," *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 1, pp. 50–55, January 2010.