Measuring Multipath Routing in the Internet

Brice Augustin, Timur Friedman, *Member, IEEE*, and Renata Teixeira

***Abstract*—Tools to measure Internet properties usually assume the existence of just one single path from a source to a destination. However, load-balancing capabilities, which create multiple active paths between two end-hosts, are available in most contemporary routers. This paper extends Paris traceroute and proposes an extensive characterization of multipath routing in the Internet. We use Paris traceroute from RON and PlanetLab nodes to collect various datasets in 2007 and 2009. Our results show that the traditional concept of a single network path between hosts no longer holds. For instance, 39% of the source–destination pairs in our 2007 traces traverse a load balancer. This fraction increases to 72% if we consider the paths between a source and a destination network. In 2009, we notice a consolidation of per-flow and per-destination techniques and confirm that per-packet load balancing is rare.**

***Index Terms*—Internet topology, load balancing, multipath, traceroute.**

I. Introduction

THE TRADITIONAL model of the Internet assumes just one single path between a pair of end-hosts at any given time. Internet applications, network simulation models, and measurement tools work under this assumption. However, most commercial routers have load-balancing capabilities [1], [2]. If network administrators turn this feature on, then a stream of packets from a source to a destination will no longer follow a single path. Faced with load balancing, not only traceroute, but also other tools that measure Internet properties (e.g., delays, loss or available bandwidth), might report incomplete results (because they measure the properties of a single path, not all of them) or even inaccurate results.

Load-balancing routers (or *load balancers*) use three different algorithms to split packet streams on outgoing links[1]: *per destination*, which forwards all packets destined to a host to the same output interface (similar to the single-path destination-based forwarding of classic routing algorithms, but this technique assigns each IP address in a prefix to a different outgoing interface); *per flow*, which uses the same output interface for all packets that have the same *flow identifier* (described as a 5-tuple: IP source address, IP destination address, protocol, source port, and destination port); or *per packet*, which makes the forwarding decision independently for each packet (and which has potentially detrimental effects on TCP connections, as packets from the same connection can follow different paths and be reordered).

Our tool, *Paris traceroute* [3] controls the paths that packets take under per-flow load balancing by setting the flow identifiers in packet headers. The *Multipath Detection Algorithm* (MDA) finds, with a low failure probability bound, all paths from a source to a destination under per-flow and per-packet load balancing. We extend it to cover per-destination load balancing and use a refined version of the algorithm based on a more solid theoretical model [4].

We use Paris traceroute to quantify the multipath routes observed from two measurement platforms (15 RON nodes [5] and over 250 PlanetLab nodes [6]) to four destination lists in 2007 and 2009. We also evaluate how measurement parameters (source and destination lists, MDA failure probability bound, probe protocol) may impact our characterization. Finally, we describe multipath routes in terms of their length, width, and asymmetry.[2] The main findings of the paper are the following.

- Per-flow and per-destination load balancing is common in our traces. In our 2007 dataset, the paths between 39% of source–destination pairs traverse a per-flow load balancer, and 72% traverse a per-destination load balancer. We observe a consolidation of this prevalence in 2009.
- Per-packet load balancing affects a small fraction of paths, is deployed in edge networks, and seems to disappear.
- The deployment of load balancing is much less common in academic and research networks than in commercial backbones.
- Multipath routes typically span a few hops in a single autonomous system (AS).

This paper proceeds as follows. Section II presents our tool to measure multipath routes under each type of load balancer. Section III, describes our measurement setup and characterization metrics. Section IV characterizes the load balancers found in our traces, and Section V studies the properties of multipath routes. We discuss the previous work in Section VI, and Section VII ends the paper.

Manuscript received February 26, 2010; revised September 22, 2010; accepted October 16, 2010; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor O. Bonaventure. Date of publication December 17, 2010; date of current version June 15, 2011.

The authors are with the Laboratoire d'Informatique de Paris 6 (LIP6), University Pierre et Marie Curie (UPMC) Sorbonne Universités and Centre National de la Recherche Scientifique (CNRS), Paris 75005, France (e-mail: brice.augustin@lip6.fr; timur.friedman@upmc.fr; renata.teixeira@lip6.fr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNET.2010.2096232

[1]Some variants of per-destination and per-flow load balancers include additional protocol header fields in their hash, but any load balancer falls into one of the three categories used in this work.

[2]This paper extends our previous work [7] in three directions. First, we confirm our previous findings, based on a larger and more accurate set of experiments. Second, in comparing the results between 2007 and 2009, we confirm the wide prevalence of per-destination and per-flow load balancing in Internet paths, while per-packet load balancing tends to disappear. Third, additional experiments show that academic networks have a lower prevalence of load balancing than commercial backbones.

II. Measuring Multipath Routes

This section describes the MDA that Paris traceroute uses to discover multipath routes.[3] Section II-A describes our prior work [4] on enumerating all paths between a source and a destination in the presence of per-flow load balancing. The remaining sections introduce a simple extension to take into account per-destination load balancers and discuss the limitations of our technique.

A. Multipath Detection Algorithm

Our initial work on Paris traceroute [3] largely fixed the problem of the false paths reported by classic traceroute. The problem was that classic traceroute systematically varies the flow identifier for its probe packets. By maintaining a constant flow identifier, Paris traceroute can accurately trace a path across a per-flow load balancer. However, this early version only traced one path at a time.

Our subsequent work suggested a new goal for route tracing: to find the entire set of multipath routes from source to destination. We showed that the classic traceroute practice of sending three probes per hop is inadequate to have even a moderate level of confidence that one has discovered load balancing at a given hop. The MDA uses a stochastic approach to send a sufficient number of probes to find all the paths to a destination, with a given probability to fail. This section provides a brief overview of the algorithm. For more details, refer to our previous papers [4] and [8].

The MDA proceeds hop by hop, eliciting the full set of interfaces for each hop. For a given interface r at hop $h-1$, it generates at random a number of flow identifiers and selects those that will cause probe packets to reach r. It then sends probes with those identifiers, but one hop further, in an effort to discover the successors of r at hop h. The number of probes sent depends on the number of interfaces already discovered and on a tunable parameter of the algorithm: a bound of the probability to fail at discovering all existing interfaces. If the MDA discovers more than one successor interface for r, r is a load balancer. It then sends additional probes so as to classify it as either a per-flow or a per-packet load balancer.

Our first characterization of load balancing was based on an earlier implementation of this algorithm [8]. It uses a simpler model providing only statistical guarantees at the node level, whereas the newer version [4] provides guarantees at the level of the entire end-to-end path. This translates to sending more probes, the number being calibrated by the failure probability bound.

B. Extending the MDA

When tracing toward a single destination with the MDA, Paris traceroute is naturally incapable of detecting instances of per-destination load balancing. In Fig. 1, for example, destinations T and T' belong to the same prefix. They might even be consecutive addresses in this prefix. As a result, L has a single entry in its forwarding table to reach both destinations. However, L might be a per-destination load balancer, sending traffic destined for T along the upper path, and traffic for T' along the lower path. When Paris traceroute uses the MDA to trace to T, it only discovers the upper path. We generalize the MDA to enumerate all of the paths from a source to a given address prefix rather than simply to a given destination. In this example, the generalized MDA detects both paths and flags L_0 as the interface of a per-destination load balancer.

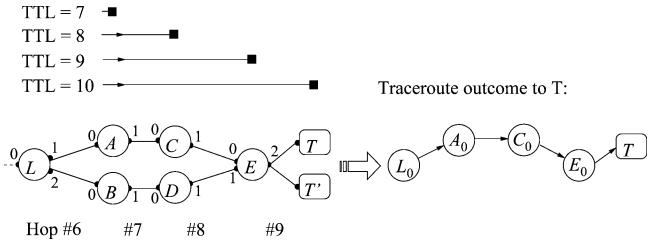

Fig. 1. Traceroute and per-destination load balancing.

We achieve this goal by refining the techniques previously described. When testing the hypothesis that there are n next-hops for an interface r, the MDA initially chooses flow identifiers that differ only in their destination address. It chooses destination addresses that share a long prefix with the destination of interest. Two addresses sharing a prefix longer than /29 are unlikely to have different entries in a core router, so any path differences should purely be the result of load balancing. Analyzing a BGP table provided by the RouteViews project in June 2009, we found that less than 2% of the prefixes were longer than /24. Nevertheless, long prefixes might be more common in edge networks for intradomain destinations, causing an overestimation of the prevalence of per-destination load balancing. The analysis of our collected data (described in Section III-A) reveals, however, that the majority of per-destination load balancers are located in core ASs rather than in destination ASs.

The MDA initially chooses addresses that share a /29 prefix, allowing the choice of up to eight different addresses. As before, the MDA sends a number of probes (determined by the failure probability bound) one hop past r in order to enumerate its next-hops. As this nex-thop enumeration technique is designed to work when r belongs to a per-destination load balancer, it will *a fortiori* also work when r belongs to a per-flow or a per-packet load balancer. If the MDA has found a load balancer, it must then classify it. It sends additional probes, maintaining the destination address and ports, in order to distinguish the per-destination load balancers from per-flow ones.

C. Limitations

The Multipath Detection Algorithm may return incomplete or inaccurate results in the following cases.

1) MPLS: Multiprotocol Label Switching (MPLS) represents a challenge for all traceroute-like measurements because some Internet service provider (ISP) networks deploy MPLS tunnels in which routers do not necessarily decrement the IP time to live (TTL) of packets. Under this configuration, the TTL will never expire while in a tunnel, and traceroute will observe the path through the tunnel as a single link, causing an underestimation of the network layer hop length of the path. Furthermore, if a load balancer splits traffic across several

[3]Note that our technique detects load sharing performed by routers. It is not our goal to measure load balancing at server farms, where dedicated boxes distribute incoming requests to a set of replicated servers.

TABLE I
SUMMARY OF EXPERIMENTS

Experiment	RON07	PL09	COMMON	WEB07	PL09-allpref	PL09-lowfail	PL09-proto	PL09-PL
Date	2007	2009	2007/09	2007	2009	2009	2009	2009
Sources	RON	PlanetLab	RON/PlanetLab	LIP6	PlanetLab	PlanetLab	PlanetLab	PlanetLab
Targets list	MIT	MIT	MIT	WEB	ALLPREF	MIT	MIT	PL
Failure probability bound	.79	.79	.79	.79	.79	.05	.79	.79
Probe protocol	UDP	UDP	UDP	UDP	UDP	UDP	ICMP, TCP	UDP

MPLS paths sharing the same entry and exit points, the MDA will not detect the existence of multipath routing. Just like classic traceroute, Paris traceroute reports ICMP extensions for MPLS [9]. Perhaps these extensions might be used to detect load balancing across MPLS paths.

2) Nonresponding Routers [10]: When routers do not respond to probes even after retransmissions, we cannot accurately enumerate a given next-hop set. This is a fundamental limit to traceroute-style measurements, and the amount of load balancing will be underestimated in these instances.

3) Uneven Load Balancing: If a load balancer distributes load with nonuniform probability across its next-hop interfaces, the algorithm risks not discovering a low-probability next-hop interface. The solution, if we expect probabilities to be possibly skewed up to some maximum extent, is to send more probes in order to regain the desired failure probability bound. Despite having seen some examples in which a router does not distribute load evenly, our informal experience tells us that this is rare. However, we would need to run a specific experiment to confirm this insight. We have not yet adjusted the MDA to catch all such cases, leading to another small source of underestimation of multipath routes.

4) Routing Changes: Routing changes during a traceroute can lead to the inference of false links. They may cause an overestimation of load balancing, or the incorrect classification of a routing change as per-packet load balancing. Fortunately, routing changes are relatively infrequent [11], especially on the time scale of an individual traceroute. The MDA has an extension to reprobe a path to try to determine if the route has changed, but we did not use it for our data collection.

III. METHOD

This section presents our measurement setup, the datasets we collected, and the metrics we use to characterize load balancing.

A. Main Datasets

We collected a total of seven datasets in 2007 and 2009. Table I summarizes their setup in terms of the measurement platform, destination list, and measurement parameters.

We collected the "RON07" dataset in 2007 from 15 sources: 13 RON nodes (the other RON nodes were not available) plus a host at our laboratory in Paris, France, and another in Bucharest, Romania. Eleven of the sources are in the U.S., the others in Europe. Table II summarizes the geographic locations of the nodes. Although the sources do not exhibit great geographic diversity (most of them are on the U.S. East and West Coasts), they connect to the Internet through many different providers [7]. We used a destination list, called MIT, which contains 68 629 addresses. It was generated by researchers at the Massachusetts Institute of Technology (MIT), Cambridge, from the BGP table

TABLE II
SUMMARY OF MEASUREMENT PLATFORMS

Platform	RON	PlanetLab
Sources	15	234
ASes	13	166
Countries	5	23

of a router located there. They randomly selected a couple of addresses from each classless interdomain routing (CIDR) block of the BGP table and ran classic traceroute from MIT toward each address. The basis of the MIT list consists of the last responding hop from each trace, which explains the relatively small number of addresses in the list, compared to total number of prefixes in a full BGP table. From this, they removed addresses that appeared in any of several blacklists, as well as any host from which they received complaints during their experiments. We updated this list by adding all our source nodes.

We collected our initial datasets over the months of February–April 2007 using Paris traceroute adapted to run in 32 parallel threads of a single process. We limit the overall bandwidth to 200 probes per second. Each thread takes the next address d in the destination list and uses the MDA to enumerate all of the paths to d. We use the following parameters: 50 ms of delay between each probe sent, abandon after three consecutive unresponsive hops. We use the old version of the MDA, with a 0.05 per-hop failure probability, to find the next-hops of an interface. This per-hop probability translates to a global per-path failure probability as high as 0.79. We use UDP probes. We avoided ICMP probes because some per-flow load balancers do not perform load balancing on ICMP packets, thus hiding part of the multipath routes. We did not use TCP probes to avoid triggering alarms at intrusion detection systems (IDSs). We collected data from all 15 sources, but due to disk space restrictions, we were able to collect per-destination load balancing data from only 11 of them. Our traces with the MIT list, for all sources, cover 9506 ASs, including all nine tier-1 networks and 96 of the 100 top-20 ASs of each region according to APNIC's weekly routing table report.[4]

Because many RON nodes used in 2007 were unavailable in 2009, we ran Paris traceroute from PlanetLab nodes instead toward the same destination list (the "PL09" dataset in Table I). PlanetLab's acceptable use policy prevents from probing random addresses, so we took a number of precautions to set up this experiment. The successful large-scale experiment performed in 2007 from the RON nodes, the testing of Paris traceroute's implementation, and the repeated use of our destination lists without triggering any abuse reports made us believe that experimenting from PlanetLab was now safe.

[4]APNIC automatically generates reports describing the state of the Internet routing table. It ranks ASs per region according to the number of networks announced.

Table II compares the characteristics of both measurement infrastructures.

Although we initially kept the experiment parameters unchanged (MIT destination list and same failure probability bound), the number and location of the sources differ significantly between RON07 and PL09. As a result, one must be careful when comparing the 2007 and 2009 datasets, as changes might not only be caused by an evolution of load-balancing deployment. To allow a fair comparison, we built the "COMMON" dataset, which groups the five sources that were both available in 2007 and 2009 (Paris, Cornell, Intel Berkeley, NYU, and MIT). The measurement parameters used in 2007 and 2009 from these sources were exactly the same. Analyzing this dataset allows us to detect an evolution of load balancing.

B. Specific Datasets

We performed experiments with three additional destination lists in order to verify the impact of the probed destinations on our characterization of load balancing. Furthermore, we modified several experiment parameters to understand how they may influence our results.

Since the MIT list doubtless includes targets that are routers or middleboxes, we are not certain that we can trace all the way through the network for all destinations. As a result, in 2007 we used a second destination list composed by end-hosts only. This consists of the 500 most popular Web sites, as reported by the commercial service Alexa.[5] We call this the WEB list. We could only use it from the Paris node, as the RON acceptable use policy forbids tracing toward arbitrary destinations. We call the corresponding dataset "WEB07."

The "ALLPREF" list is similar to the MIT list, but generated from a more recent BGP table dump collected by the RouteViews project in June 2009. It contains 120 K addresses, compared to the 68 K addresses in the original MIT list built in 2007. By using a more recent list, one makes sure that we trace all allocated prefixes, including the recently allocated ones, not present in the older list. In those new networks and prefixes, we can expect to find different practices in terms of load-balancing deployment. This experiment, called "PL09-allpref," aims at detecting such emerging practices.

Finally, we characterize load balancing on the paths between PlanetLab nodes. As most PlanetLab nodes are located at universities, the paths between them are known to have properties that do not reflect the paths properties observed in the commercial Internet [12]. This dataset will enable a comparison of load-balancing deployment in academic and commercial networks.

We also study the effect of experiment parameters on our results. In particular, we evaluated the impact of the following parameters.

1) MDA Failure Probability Bound: Our initial Paris traceroute implementation, used in 2007, used a somewhat high failure probability bound (0.79) and included a small probing bias, subsequently corrected [4]. To build the "PL09-lowfail" dataset, we used a lower failure probability (0.05). Our previous work [4] provides a precise comparison of the completeness of the collected traces to different failure probability bounds, so we will not repeat it here. This dataset will help us understand how the failure probability parameter may affect our characterization of load balancing.

2) Protocol: We used UDP probes by default, but we also measured ICMP and TCP multipaths (the "PL09-proto" dataset). While it is reasonable to expect that load balancers will treat TCP probes like UDP probes, we surprisingly notice that ICMP probes are often load-balanced [8], which was confirmed later [13].

As some measurements failed on some PlanetLab nodes, we discard 15 nodes that completed less than 50% of the experiment. To comply with PlanetLab's use policy, we took the following precautions. We first ran the experiments on a private machine in our lab and waited for any complaint before deploying it on PlanetLab nodes. We decreased the number of parallel traceroutes to decrease the probing load. The detection of per-destination load balancing requires to trace to several addresses in the same prefix. Such a traffic could be confused with network scanning. For this reason, we did not run a full experiment, but traced 5000 addresses randomly selected in the MIT and ALLPREF lists. We repeated this experiment with different lists to verify that the sampling does not affect the results. We took the same approach to evaluate the impact of a lower failure probability (because it increases the network load) and of other protocols like TCP and ICMP (especially because TCP probes may be confused with network scanning).

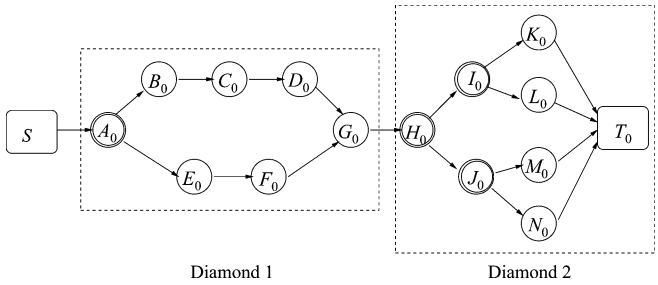

Fig. 2. Two diamonds in a set of paths to a destination.

C. Metrics

This section describes the metrics we use to characterize load balancing. We characterize load-balancing behavior at the IP level, making no attempt to resolve the router-level graph. Although Section V provides some insight on load-balancing deployment at the router level, a more thorough analysis will be necessary to characterize parallel links between routers (see Section V-A.2) and load balancing across router-disjoint paths.

Fig. 2 illustrates the metrics in use. This is a real topology we discovered when tracing from a US source, S, towards a Google web server, T. We use the following terminology in the context of IP-level directed graphs generated by the MDA:

1) Load Balancer: A node with out-degree $d > 1$ is an interface of a load balancer. For instance, A_0, H_0, I_0, and J_0 are interfaces of load balancers.

2) Diamond: A diamond is a subgraph delimited by a *divergence point* followed, two or more hops later, by a *convergence point*, with the requirement that all flows from source to destination flow through both points. Fig. 2 has two diamonds, shown

[5]See http://www.alexa.com/site/ds/top_sites?ts_mode=global&lang=none.

TABLE III
OCCURRENCES OF LOAD BALANCING

Experiment	RON07	PL09	COMMON		WEB07	PL09-PL
			2007	2009		
per-flow	39%	50%	51.2%	54.8%	35%	20%
per-packet	2.1%	1%	2%	1%	0%	<1%
per-dest	72%	83%	75%	78%	n.a.	n.a.
any	89%	91%	89%	92%	n.a.	n.a.

in dashed boxes. Note that this differs from definitions of diamonds we have employed in other work, in which we restricted their length to 2 hops, or allowed just a subset of flows to pass through them (as between I_0 and T in Fig. 2).

3) Diamond Width: We use two metrics to describe the width of a diamond. The *min-width* counts the number of link-disjoint paths between the divergence and convergence points. This gives us a lower bound on the path diversity in a diamond. For instance, diamonds 1 and 2 in Fig. 2 have the same min-width of 2, although diamond 2 offers a greater diversity with more branching points. Thus, in addition, we also use the *max-width* metric, which indicates the maximum number of interfaces that one can reach at a given hop in a diamond. In our example, diamond 1 has a max-width of 2, whereas diamond 2 has a max-width of 4.

4) Diamond Length: This is the maximum number of hops between the divergence and convergence points. In our example, diamond 1 has length 4, and diamond 2 has length 3.

5) Diamond Symmetry: If all the parallel paths of a diamond have the same number of hops, we say that the diamond is symmetric. Otherwise, it is asymmetric. The *diamond asymmetry* is the difference between the longest and the shortest path from the divergence point to the convergence point. Diamond 1 has an asymmetry of 1 since the longest path has 4 hops and the shortest one has 3 hops. Diamond 2 is symmetric.

IV. LOAD BALANCERS

This section characterizes load balancers. We show that per-flow and per-destination load balancing are very common in our traces. This high frequency is due to the fact that per-flow and per-destination load balancers are located in core networks, and thus are likely to affect many paths. We also observe that the majority of load balancing happens within a single network.

A. Occurrences of Multipaths

Table III summarizes the occurrence of load balancing in our traces. Let us start with the main findings from our 2007 dataset (RON07). Per-destination load balancers are the most common in these traces: The paths between 72% of the 771 795 source–destination pairs traverse a per-destination load balancer. This percentage is still considerable for per-flow load balancers, 39%, but fairly small, only 2.1%, for per-packet load balancers. Our measurements for per-flow and per-packet load balancers had 1 010 256 source–destination pairs in total. (This difference is because our dataset for per-destination load balancers uses only 11 sources, whereas the per-flow and per-packet dataset uses 15 sources.) The fraction of per-flow load balancers generalizes the results of our preliminary study [3], in which we found that per-flow load balancing was common from the Paris source. This result comes from the widespread availability of load balancing in routers. For instance, Cisco and Juniper routers can be configured to perform any of the three types of load balancing [1], [2], [14]. Even though per-packet load balancing is widely available, network operators avoid this technique because it can cause packet reordering [15].

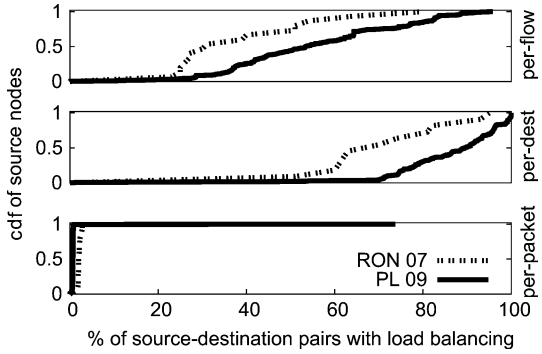

Fig. 3. Fraction of source–destination pairs affected by (Top) per-flow, (Middle) per-destination, and (Bottom) per-packet load balancing.

In the PL09 dataset, 50% of the source–destination pairs traverse a per-flow load balancer (83% for per-destination). The significant difference compared to 2007 does not necessarily reflect a wider deployment of load balancing in the Internet because the measurements were collected from a different set of sources. When we only compare the prevalence observed from sources both available in 2007 and 2009 (referred as the "COMMON" dataset), the results are mitigated. For two sources, the fraction of per-flow load balancing increased; it decreased for two other sources, and remained stable for the last one. Overall, the numbers are 51.2% in 2007 and 54.8% in 2009. In addition, the fraction of ASs using load balancing remained stable during these two years, around 6%–7%. Because of the small increase and the very small number of sources in this dataset, it is not clear if the prevalence of load balancing increased from 2007 to 2009.

We notice that less than 1% of the paths traverse a per-packet load balancer in PL09. This number is lower than the fraction observed in 2007, which may indicate the disappearance of this type of load balancing.

Fig. 3 plots the cumulative distribution function (cdf) of the fraction of paths affected by load balancing for each source in the RON07 and PL09 datasets. Each graph plots the distribution for a given type of load balancing (from top to bottom: per-flow, per-destination, per-packet). The frequency of per-flow and per-destination load balancers greatly varies according to the source (in 2007, it varies from 23% to 80% for per-flow, and from 51% to 95% for per-destination load balancing), whereas the frequency of per-packet load balancers is more stable across all sources (around 2%). The frequency of per-flow and per-destination load balancers depends on the location and upstream connectivity of the source. For instance, for two sources in the same location and having the same upstream connectivity, we observe the same fraction of per-flow load balancers. In 2009, the fraction varies from 20% to 95% for per-flow, depending on the PlanetLab node. The figures are even higher for per-destination load balancing, confirming our findings of 2007.

On the other hand, the frequency of per-packet load balancing depends mostly on the destination list used: always around 2% for RON07, using the MIT list, zero for the WEB07 dataset, and 1% for PL09. Furthermore, it is relatively constant from all our sources, which suggests that per-packet load balancers tend to be close to destinations. However, the larger number of vantage points in PL09 allows us to observe a previously unseen scenario. The PlanetLab node at the National Chengchi University (NCCU) in Taiwan is very close to a per-packet load balancer, which results in a very high fraction of source–destination pairs with per-packet load balancing (60%), for this particular source. The real fraction is actually larger, but a manual inspection revealed that this load balancer forwards packet on an uneven basis. With the setup we used, we did not send enough probes at each hop to detect uneven load balancing with a low failure probability.

As already noticed, the prevalence of per-flow and per-destination load balancing depends on the source location rather than on the destination list. Using an up-to-date destination list (PL09-allpref), we observed similar numbers than in the PL09 dataset. We noticed that, with the new list, the overall fraction of source–destination pairs affected by load balancing was around 2% higher than with the old list. However, we do not believe that this increase is significant enough to draw any conclusions (in particular, whether new networks employ load balancing more heavily).

Interestingly, per-flow load balancing is much less common in the paths between PlanetLab nodes: Only 20% of pairs are affected, which is half of the fraction observed in PL09. Many PlanetLab nodes are located in universities, so they communicate through academic backbones, whose characteristics are different from commercial backbones [12]. Schwartz *et al.* [16] also report results indicating that academic networks are more averse to load balancing than commercial ones. This result implies that testing services and protocols on PlanetLab will not observe multipath routes as in commercial backbones. Researchers should take this result into account for future experiments and evaluation on PlanetLab.

When tracing with different TCP and UDP probes between the same set of source–destination pairs, we found a similar prevalence of per-flow load balancers, which suggests that TCP probes are load-balanced just the same way as UDP probes. However, using ICMP probes revealed a much lower prevalence, indicating that ICMP probes are much less subject to load balancing (the fraction of source–destination pairs affected is 28%, versus 50% with UDP probes). Our early experiments ran in 2006 and 2007 had revealed a much higher fraction. This change may indicate a router implementation update during the last two years.

Using a lower failure probability bound and a refined implementation of the MDA did not affect our estimations of load-balancing prevalence, which leads us to the conclusion that a relatively high failure probability bound (like the one used in our 2007 experiments) is enough to get accurate estimates.

B. Occurrences of Load Balancers

We now study how load balancers affect paths to verify whether there are a few routers responsible for most load balancing. In a typical trace, we find from each source around 1000 distinct per-flow load balancers, 2500–3000 per-destination load balancers, and 500 per-packet load balancers.

There is a clear disparity between the relatively small number of load balancers and the large number of multipath routes for the per-flow and per-destination cases. Indeed, the top-50 load balancers affect at least 78% of the paths that exhibit load balancing. For instance, the most frequent per-flow load balancer affects 38% of the multipath routes from the Paris source. We studied this load balancer in detail and found that it is a router in Level3's network that provides connectivity to RENATER. Similarly, nearly all the paths from the Intel source have per-destination load balancing caused by a load balancer in AT&T's network, which is Intel's provider.

We noticed that some ASs have a wider load-balancing deployment than others. Among these, one can cite Level3, which seems to use load balancing at all its points of presence (PoPs). As a result, any source having Level3 as an upstream provider will exhibit a high fraction of paths affected by load balancing (e.g., our Paris source).

In contrast, we do not find any predominant per-packet load balancer in the RON07 data. The 50 most frequently found ones affect at most 60% of the paths with per-packet load balancing. We find that the most frequently encountered per-packet load balancers are in Sprint's network. This finding is puzzling given that large ISPs often avoid per-packet load balancing for fear of the negative impact on TCP connections. We studied these load balancers more closely and verified that they are located at peering points between Sprint and other domains. For instance, we found one per-packet load balancer between Sprint and the China169 backbone in RON07. The load-balanced interfaces after this load balancer all belong to the same router and have DNS names such as sl-china7-5-0.sprintlink.net, a name that indicates that it is, indeed, at a peering point. In PL09, we find similar situations at the edges of tier-1 ISPs such as AT&T, Sprint, and NTT. If this is being done purposefully, perhaps it is a situation where high link utilization is especially important, such as when load balancing over a bundle of parallel low capacity links is preferred to a single, more expensive, high-capacity link. Some other instances may also correspond to misconfigured routers using the per-packet technique instead of the per-flow or per-destination one.

Most of the per-packet load balancers affect just a few paths because they are located far from the source and close to the destination. Indeed, 85% of those load balancers are located at less than 3 hops from the destination.

C. Routing Protocols and Load Balancing

Multipath routes can be contained in one AS, which we define as *intradomain load balancing*, or span multiple ASs, defined as *interdomain load balancing*. Although forwarding in both cases is done in the same way, the routing mechanism behind them is very different. A router can install multiple intradomain routes in its forwarding table because of the equal-cost multipath capability of common intradomain routing protocols such as IS-IS [17] and OSPF [18]. In this case, the paths will diverge after entering the AS and reconverge before exiting it.

On the other hand, BGP [19], the Internet's interdomain routing protocol, does not allow a router to install more than one next-hop for a destination prefix. Given this restriction, there should be no interdomain load balancing. However, some

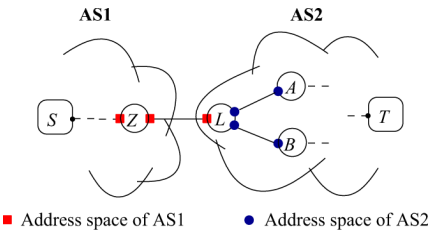

Fig. 4. Domain boundary delimitation can be inaccurate.

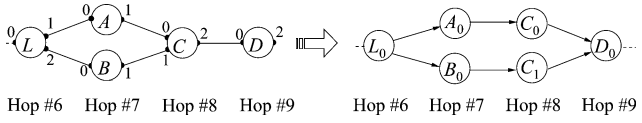

Fig. 5. IP-level multipath route inferred by Paris traceroute may not correspond to the router-level multipath in the real topology.

router vendors now provide BGP-multipath capabilities (for instance, Juniper [20] and Cisco [21]). If two BGP routes for a prefix are equivalent (same local preference, AS-path length, etc.) and the multipath capability is on, then BGP can install more than one next-hop for a prefix. Another scenario in which we could observe interdomain load balancing is when BGP routes are injected into the intradomain routing protocol. Then, BGP routes would be subject to the OSPF or IS-IS equal-cost multipath mechanism. Injecting BGP routes into intradomain routing is, we believe, rare, so this scenario should not often arise. However, injecting only the default route(s) to upstream provider(s) is a more practicable scenario that is often used by network operators.

To make the distinction between the two types of load balancing, we need to map each IP address in our traces to an AS. We use a public IP-to-AS mapping service [22]. This service builds its mapping from a collection of BGP routing tables. There are well-known issues with this type of mapping [23], so for one of the traces we manually verified each instance of supposed interdomain load balancing.

Our automated classification does not consider the convergence or the divergence point of a diamond to label load balancers. In so doing, we avoid false positives (classification of intradomain load balancing as interdomain), but may generate false negatives. This technique is important because it is very common that an interface in the boundary between two ASs is numbered from the address space of one AS, but belongs in fact to the other. Fig. 4 illustrates this scenario. It shows two domains, AS1 and AS2, and a load balancer L. Square interfaces are numbered from AS1's address space, whereas circular ones belong to AS2's address space. We observe that the interfaces of the link Z–L are numbered from AS1's address space. A traceroute from S to T discovers the "square" interface of L. In this case, we could mistakenly label L as an interdomain load balancer because L belongs to AS1 and balances traffic to routers A and B, which belong to AS2. If we ignore the divergence point when computing the AS path in a diamond, then L would be correctly labeled as an intradomain load balancer in AS2.

We also ignore the convergence point because it may not be involved in load balancing. Indeed, the IP-level multipath route inferred by Paris traceroute may not correspond to the router-level multipath route in the real topology. Fig. 5 illustrates how this phenomenon arises. The left side represents the router-level topology, and the right side the IP-level topology inferred with the MDA. The two paths merge at two different interfaces of router C. The probing of the upper path reveals C_0, and the lower path reveals C_1. Since we do not conduct alias resolution, we treat those two interfaces as if they belonged to different routers. The consequences are twofold. First, the length of the measured diamond does not reflect the length of the multipath route in the router-level topology. Second, we may consider some parts of the topology as being involved in load balancing, whereas they are not. More precisely, the convergence point in the inferred topology, D_0, has actually nothing to do with load balancing. The left side of the figure shows that router D is not part of the real multipath route at all. As a result, we may misclassify some diamonds as interdomain if router D belongs to a different autonomous system. Note that this bias arises because the parallel paths merge through different interfaces of a router. If they merge through a level-2 device such as a switch and then connect to a single interface, then the inferred topology maps to the router-level one. Although we do not perform systematic alias resolution on the discovered interfaces, our partial observations of IP IDs [24] and DNS names indicate that all the penultimate interfaces of a diamond generally belong to the same router.

The manual verification step is very time-consuming, so we only classified intra- and interdomain load balancers observed from the Paris source in 2007. In most cases, diamonds are created by intradomain load balancing. From the Paris source, 86% of the per-flow diamonds fit in a single AS. Fig. 2 illustrates this case. Diamond 1 exactly spans Savvis's network, and diamond 2 spans Google's network. The parallel paths in diamond 1 diverge at the entry point of Savvis's domain, and then reconverge before they reach its exit point, because routers selected a single peering link between the two domains. We found rarer cases of diamonds crossing multiple ASs. Most of them involve two ASs, but extremely rare diamonds cross three networks. We found such diamonds in the paths toward 37 destinations. They always involved Level3 as the first domain, peering with Verizon, Bellsouth, and some smaller networks like Road Runner. Interestingly, when we include the divergence point of a diamond to label load balancers, our results do not change. Thus, it seems that very few core networks enable BGP multipath capabilities in their routers, and the false positives induced by a diamond's divergence point are negligible.

Most per-destination diamonds are also created by intradomain load balancers (at least 80% from the Paris source), but we did not conduct any of the manual verification on this dataset.

V. Characteristics of Mutipath Routes

Having described the mechanisms behind multipath routes, we now study their properties and characterize them in terms of the widths and lengths of diamonds. The statistics presented here are for the RON07 and PL09 datasets. Other datasets present very similar trends.

A. Diamond Width

We use two metrics defined in Section III-C to describe the number of paths available in a given diamond: A diamond's

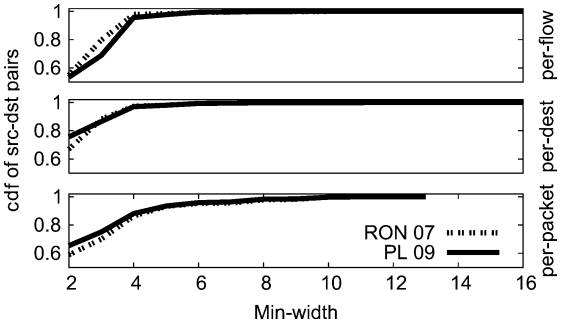

Fig. 6. Min-width distributions for (Top) per-flow, (Middle) per-destination, and (Bottom) per-packet load balancing.

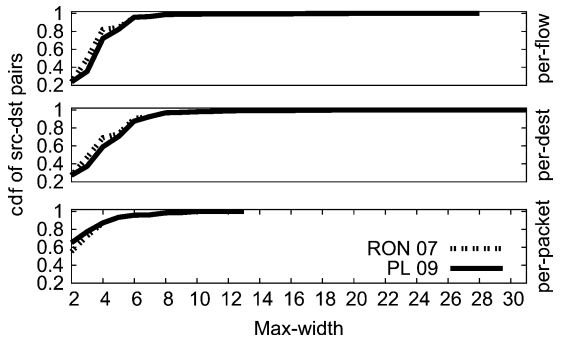

Fig. 7. Max-width distributions for (Top) per-flow, (Middle) per-destination, and (Bottom) per-packet load balancing.

min-width provides a lower bound, and the *max-width* provides a richer picture of a diamond's path diversity. If there should be two or more diamonds in a multipath route, we take the lowest min-width and the highest max-width. It is fairly common to see two diamonds in a path: 22% of the pairs have two per-flow diamonds, and 21% have two per-destination diamonds in PL09. Any more than two is extremely rare, less than 1% of the paths.

Fig. 6 presents the min-width distribution for multipath routes in the RON07 and PL09 datasets. Each subfigure plots the distribution for a given type of load balancing: per-flow (top), per-destination (middle), per-packet (bottom). The results for both datasets are very similar.

1) Narrow Diamonds: These plots show that multipath routes are generally narrow. In 2009, for per-flow load balancing, 55% of the pairs encounter a diamond with only two link-disjoint paths, and 99% of the pairs encounter diamonds with five or fewer link-disjoint paths. For per-destination load balancing, the figures are 77% and 98%, and for per-packet load balancing, they are 63% and 90%.

The max-width distribution (Fig. 7) is, of course, less skewed toward diamonds of width 2. Only 24% of per-flow multipath routes and 27% of per-destination multipath routes traverse a diamond with just two interfaces at the widest hop distance. Nonetheless the diamonds tend to be narrow by this metric as well: 85% of the per-flow diamonds and 90% of the per-destination diamonds have five or fewer interfaces at the widest hop. Because most of per-packet diamonds have a length equal to 2, their max-width distribution is similar to their min-width distribution.

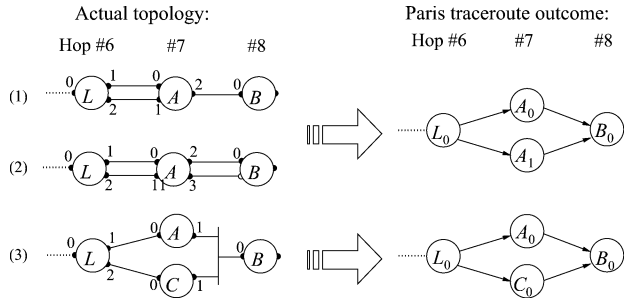

Fig. 8. Three different router topologies that lead to the same diamond.

2) Wide Diamonds: The maximum width that we encounter, by either metric, is 16. For instance, in the RON07 dataset, we discovered a diamond of max-width 16 for per-flow load balancing at a peering point between a tier-1 and a Brazilian ISP. This may correspond to many low-capacity links that are bundled because the next higher capacity link is unavailable, unaffordable, or unsuitable. That we do not see anything wider can be explained by a built-in limit to the number of entries that a router can install in the forwarding table for a given prefix. For instance, Juniper [2] allows one to configure at most 16 load-balanced interfaces.

Almost all of the diamonds of width 10 and greater are 2 hops long. One obvious explanation for a diamond of this length is that we are seeing multiple parallel links between a pair of routers. As routers typically respond to traceroute probes using the address of the incoming interface [25], a pair of routers with parallel links will appear as a diamond of length 2 at the IP level. Case (1) in Fig. 8 shows an example with two parallel links. The figure also illustrates two other actual topologies that lead to the same diamond in Paris traceroute reports. Case (2) shows two parallel links between routers A and B, but B responds to traceroute probes using only one interface, B_0, while B_1 remains invisible to Paris traceroute. Case (3) is an example of load balancing over two different routers, A and C, that are connected to B through a shared media (e.g., an Ethernet).

There are rare cases (67 source–destination pairs in the RON07 dataset) of very wide and short per-packet diamonds at the ends of paths (i.e., close to the destinations). For instance, all multipath routes to a few hosts in Egypt traverse a per-packet diamond of length 2, having 11 interfaces in parallel. Alias resolution techniques (DNS names and checking the IP Identifier values returned by probes [24]) confirm that all 11 interfaces belong to the same router, and thus that the network operator configured 11 parallel links between two routers. Per-packet load balancing typically appears to take place at the boundary of a small AS and its provider. Customers may use such load balancers on access links for resilience and traffic engineering.

B. Diamond Length

Recall that Section III-C defines the length of a diamond as the maximum number of hops between its divergence point and convergence point. We define the diamond length for a multipath route to be the length of the longest diamond found in that path.

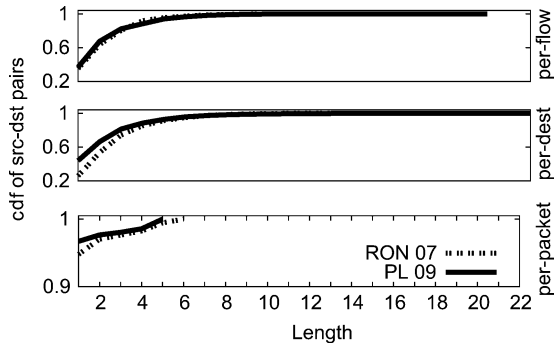

Fig. 9. Diamond length distributions for (Top) per-flow, (Middle) per-destination, and (Bottom) per-packet load balancing.

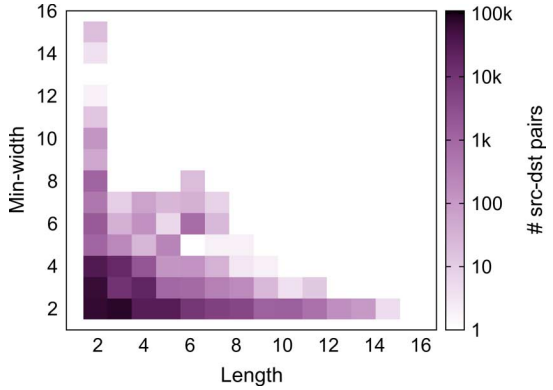

Fig. 10. Diamond length and min-width of per-flow multipath routes (RON07 dataset).

Fig. 9 shows the cdf of the diamond lengths for the multipath routes between all source–destination pairs in the RON07 and PL09 datasets. Overall, diamonds tend to be short, with a significant portion being of length 2 or 3.

1) Short Diamonds: In PL09, 37% of the source–destination pairs with per-flow load balancing have a diamond of length 2. Per-destination diamonds also tend to be short. Of the paths with per-destination load balancing, 44% of them have a diamond of length 2. Diamond length is most skewed toward the short end for per-packet load balancing, with 96% of paths having a diamond of length 2.

As discussed earlier, diamonds of length 2 should typically correspond to multiple links between two routers. Operators use load balancing between two routers not only for load sharing, but also as active backup in case of single-link failures.

2) Long Diamonds: Multipath routes with longer diamonds are less frequent. For instance, fewer than 1% of per-destination multipath routes have diamonds longer than 8. We observe per-flow diamonds of lengths up to 15 and per-destination diamonds with up to 17 hops. The longest per-packet diamonds have lengths up to 6.

Per-destination diamonds tend to be longer than per-flow. Around 37% of multipath routes traverse a per-flow diamond of length greater than 3; this percentage is 46% for per-destination diamonds. There are few long per-packet diamonds (only 3% have a length greater than 3).

We looked at the 25 cases of per-packet diamonds of length 5 and 6 in detail in RON07. Most of them appear in core networks in Asian ISPs (Thailand and China). Given the general practice of avoiding per-packet load balancing in core networks, perhaps these are cases of misconfigured load balancing. If so, then we see how Paris traceroute could help operators detect such misconfigurations. We observe similar cases in PL09, but they are less frequent, as per-packet load balancing tends to disappear.

3) Length and Width: We now study the relationship between the min-width and length. Fig. 10 presents the number of per-flow multipath routes in the RON07 dataset with a given diamond length and min-width. The vertical axis represents the number of source–destination pairs whose diamond length and min-width are given by the horizontal axis.

As discussed in Section V-A, there may be several diamonds for the same source–destination pair. If so, we select the min-width and length of the diamond with the smallest min-width. There is a clear peak in the number of diamonds with length and min-width equal to 2 (multipath routes between 17% of the source–destination pairs with per-flow load balancing are in this category). Multipath routes between 53% of source–destination pairs traverse a diamond with a length less or equal to 2 and min-width 2 or 3. This result confirms that the vast majority of the diamonds are both short and narrow.

There are no wide and long diamonds. There is a bipartition of the remaining diamonds into two categories. The first category contains wide but short diamonds. It is extremely rare to observe wide diamonds (whose width is greater than 2) with more than 3 hops. The second one corresponds to narrow but long parallel paths. In this case, the min-width is always 2. Wide but short diamonds probably correspond to multiple links between routers. Operators may introduce new links between routers to upgrade capacity. Long and narrow diamonds likely correspond to paths between the ingress and egress routers in a network, which are useful for traffic engineering.

C. Diamond Asymmetry

We say that a diamond is asymmetric when one can reach its convergence point with different hop counts. There might be some concern that asymmetric diamonds are the result of misconfiguration. However, the equal-cost multipath mechanisms of OSPF and IS-IS require only that paths have the same cost in terms of link weight, not hop count [17], [18]. Network operators can configure two paths of different hop counts to have the same sum of link weights. In addition, some new mechanisms [26] allow load-balancing over paths with small cost differences. From the point of view of performance, asymmetry might cause delay differences [7].

Fig. 11 presents the cdf of source–destination pairs in the RON07 and PL09 datasets that have per-flow, per-destination, and per-packet diamonds with a given asymmetry.

Most paths with per-flow load balancing, 90%, traverse symmetric diamonds in PL09. Paths under per-destination load balancing are also very symmetric: 87% of the paths under per-destination load balancing traverse symmetric diamonds. That still leaves a significant number of destinations that can be reached with different numbers of hops.

Similarly, over 92% of the paths with per-packet load balancing traverse a symmetric diamond. This is consistent with the observation that the majority of such diamonds are short and

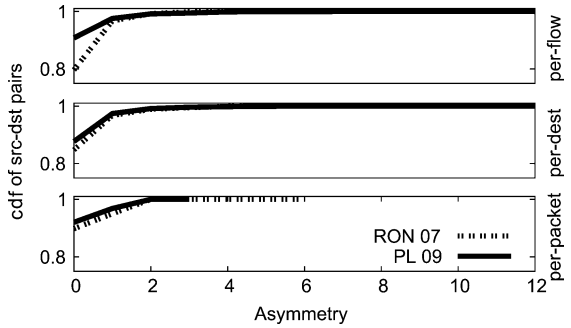

Fig. 11. Asymmetry distributions for (Top) per-flow, (Middle) per-destination, and (Bottom) per-packet load balancing.

thus have less possibility for asymmetry. Nonetheless, the 8% of such paths with asymmetry are a concern for TCP connections, in so far as asymmetry may cause different delays and thus greater chances for packet reordering.

1) Low Asymmetry: When asymmetry is present, it is typically low. For instance, out of the per-flow multipath routes with asymmetric diamonds, 82% only differ by 1 hop. For per-destination multipath routes, this fraction is 79%, and 65% for per-packet.

2) High Asymmetry: Per-flow diamonds with more than 3 hops of difference are extremely rare . For instance, from the Paris source in RON07, we observe only 63 such diamonds. We found 2549 such per-destination, and only 11 such per-packet multipath routes.

We examined the per-flow diamond with the maximum asymmetry in RON07, which has 8-hops difference. One path traverses eight routers between the divergence and convergence points, while the other directly reaches the end of the diamond. We believe that the latter path is an MPLS tunnel, maybe even traversing the same routers as the one traversed on the first path. This hypothesis is supported by the observation that routers in the diamond append MPLS extensions [9] to the ICMP responses. This example suggests that some of the shorter diamonds may also result from MPLS tunnels.

For per-destination load balancing, there are 71 cases of asymmetry between 8 and 10. We examined some of these diamonds with very distinct paths. For instance, there is one asymmetric diamond that spans the U.S. and Europe in Cogent's network. By inspecting the interface names, we concluded that the parallel paths each traverse different numbers of PoPs, which causes the asymmetry. Depending upon which address is probed inside the destination prefix, packets either cross the ocean through a link via London, U.K., or another via Paris.

VI. Related Work

A typical ISP builds redundancy into its physical infrastructure. To use the infrastructure efficiently, the ISP will split traffic load across multiple links, which introduces much of the path diversity that we measure here. The research community has looked at the question of how best to design load-balancing routers, for instance to adaptively split the traffic according to network conditions [27]–[29]. We have not systematically looked for adaptive load balancing, but our familiarity with our own data leads us to believe that most current routers use a static mapping of flows to load-balanced paths. Other studies focus on the network operator's interest in path diversity. Giroire *et al.* [30] show how to exploit an ISP's underlying physical diversity in order to provide robustness at the IP layer by having as many disjoint paths as possible.

Early work on path diversity in the Internet [31], [32] looked at the known topology of the large ISP Sprint and the paths between PoPs in Sprint's network. It found that between any given pair of PoPs there were typically several link-disjoint and several PoP-disjoint paths. It also looked at topologies inferred from traceroute-style probing conducted by Rocketfuel [24] and CAIDA [33], concluding that while there is evidence of significant path diversity in the core of the network, the measurements are particularly sensitive to errors that were inherent to active probing techniques at that time. Furthermore, when looking at path diversity in the router-level graph, the measurements are sensitive to insufficiencies in alias resolution techniques, which infer router-level graphs from IP-level information, and to multipath routers.

Multipath routing in the Internet has received growing attention since then. Luckie *et al.* [13] compare the topologies obtained when using traceroute with different probe protocols. They notice that ICMP probing tend to reveal less IP links than UDP probing and show that ICMP packets are not load-balanced as widely as UDP probes. We already noticed this behavior in our previous work [7] and its impact on the individual measured multipath routes. Sherwood *et al.* [34], [35] use the Record Route option in traceroute probes to detect multipath routing. This method is complementary to our MDA. Their work mainly focuses on using this method to get accurate topology measurements, but it does not provide the in-depth characterization of load balancing that this paper proposes.

VII. Conclusion

This paper characterizes multipath routing in the Internet. We measured the end-to-end multipath routes from RON and PlanetLab nodes to several destination lists in 2007 and 2009. We conclude that multipath routes are common in the Internet, but their prevalence strongly depends on the location and the upstream connectivity of the source node. It also depends on the type of network that end-to-end paths traverse. Indeed, our measurements between PlanetLab nodes reveal a lower prevalence, indicating that multipath routing is less common in academic networks than in commercial ones. While we observe a lower prevalence of per-packet load balancing in 2009, the per-flow and per-destination techniques are still widely used, typically creating short multipath routes within a single AS.

Future work includes a more thorough analysis of load balancing at the router level, taking advantage of state-of-the-art alias resolution techniques, and a validation of our results against real trusted topologies supplied by some selected ISPs. Finally, we envisage to carry out a longer-term analysis of load-balancing practices by performing regular measurements from PlanetLab nodes.

Acknowledgment

The authors are grateful to M. Crovella, J. Rexford, and V. Paxson for suggestions on the early versions of this work,

and to A. Krishnamurthy and the anonymous reviewers for their useful remarks. They are indebted to D. Andersen for the access to the RON nodes, S. Kandula for the MIT destination list, and A. Crivat for providing the node at Bucharest. They also thank M. Latapy, C. Magnien, and F. Viger for their thoughtful comments. X. Cuvellier wrote the base implementation of Paris traceroute. Team Cymru gave the authors access to their AS mapping database.

REFERENCES

[1] Cisco, "How does load balancing work?," San Jose, CA, Doc. ID 5212, Aug. 2005 [Online]. Available: http://www.cisco.com/en/US/tech/tk365/technologies_tech_note09186a0080094820.shtml

[2] Juniper, "Configuring load-balance per-packet action," Sunnyvale, CA, Aug. 2010 [Online]. Available: http://www.juniper.net/techpubs/software/junos/junos70/swconfig70-policy/html/policy-actions-config11.html

[3] F. Viger, B. Augustin, X. Cuvellier, B. Orgogozo, T. Friedman, M. Latapy, C. Magnien, and R. Teixeira, "Detection, understanding, and prevention of traceroute measurement artifacts," *Comput. Netw.*, vol. 52, no. 5, pp. 998–1018, 2008.

[4] D. Veitch, B. Augustin, T. Friedman, and R. Teixeira, "Failure control in multipath route tracing," in *Proc. IEEE INFOCOM*, Apr. 2009, pp. 1395–1403.

[5] D. Andersen, H. Balakrishnan, M. F. Kaashoek, and R. Morris, "Resilient overlay networks," in *Proc. 18th ACM SOSP*, Oct. 2001, pp. 131–145.

[6] PlanetLab, "PlanetLab: An open platform for developing, deploying, and accessing planetary-scale services," Aug. 2010 [Online]. Available: http://www.planet-lab.org/

[7] B. Augustin, T. Friedman, and R. Teixeira, "Measuring load-balanced paths in the Internet," in *Proc. ACM SIGCOMM IMC*, Oct. 2007, pp. 149–160.

[8] B. Augustin, T. Friedman, and R. Teixeira, "Multipath tracing with Paris Traceroute," in *Proc. E2EMON*, May 2007, pp. 1–8.

[9] R. Bonica, D. Gan, D. Tappan, and C. Pignataro, "ICMP extensions for multiprotocol label switching," IETF RFC 4950, Aug. 2007.

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

[11] V. Paxson, "End-to-end routing behavior in the Internet," *IEEE/ACM Trans. Netw.*, vol. 5, no. 5, pp. 601–615, Oct. 1997.

[12] S. Banerjee, T. G. Griffin, and M. Pias, "The interdomain connectivity of PlanetLab nodes," in *Proc. PAM*, May 2004, pp. 73–82.

[13] M. Luckie, Y. Hyun, and B. Huffaker, "Traceroute probe method and forward IP path inference," in *Proc. ACM SIGCOMM IMC*, Oct. 2008, pp. 311–324.

[14] Cisco, "Cisco 7600 series routers command references," San Jose, CA, Aug. 2010 [Online]. Available: http://www.cisco.com/en/US/products/hw/routers/ps368/prod_command_reference_list.html

[15] J. Bellardo and S. Savage, "Measuring packet reordering," in *Proc. ACM SIGCOMM IMW*, Nov. 2002, pp. 97–105.

[16] Y. Schwartz, Y. Shavitt, and U. Weinsberg, "On the diversity, stability and symmetry of end-to-end Internet routes," in *Proc. Global Internet*, Mar. 2010, pp. 1–6.

[17] R. Callon, "Use of OSI IS-IS for routing in TCP/IP and dual environments," IETF RFC 1195, Dec. 1990.

[18] J. Moy, "OSPF version 2," IETF RFC 2328, Apr. 1998.

[19] Y. Rekhter, T. Li, and S. Hares, "A Border Gateway Protocol 4 (BGP-4)," IETF RFC 4271, Jan. 2006.

[20] Juniper, "Configuring BGP to select multiple BGP paths," Sunnyvale, CA, Aug. 2010 [Online]. Available: http://www.juniper.net/techpubs/software/junos/junos94/swconfig-routing/configuring-bgp-to-select-multiple-bgp-paths.html

[21] Cisco, "BGP best path selection algorithm," San Jose, CA, Doc. ID 13753, May 2006 [Online]. Available: http://www.cisco.com/en/US/tech/tk365/technologies_tech_note09186a0080094431.shtml

[22] Team Cymru, "IP to BGP ASN lookup and prefix mapping services," Burr Ridge, IL, Aug. 2010 [Online]. Available: http://www.team-cymru.org/Services/ip-to-asn.html

[23] Z. M. Mao, D. Johnson, J. Rexford, J. Wang, and R. H. Katz, "Scalable and accurate identification of AS-level forwarding paths," in *Proc. IEEE INFOCOM*, Mar. 2004, vol. 3, pp. 1605–1615.

[24] N. Spring, R. Mahajan, and D. Wetherall, "Measuring ISP topologies with Rocketfuel," in *Proc. ACM SIGCOMM*, Aug. 2002, pp. 133–145.

[25] Z. M. Mao, J. Rexford, J. Wang, and R. H. Katz, "Towards an accurate AS-level traceroute tool," in *Proc. ACM SIGCOMM*, Aug. 2003, pp. 365–378.

[26] S. F. Shamim, "How does unequal cost path load balancing (variance) work in IGRP and EIGRP?," Cisco, San Jose, CA, Doc. ID 13677, Jul. 2007 [Online]. Available: https://learningnetwork.cisco.com/servlet/JiveServlet/previewBody/3204-102-1-8463/How%20Does%20Unequal%20Cost%20Path%20Load%20Balancing.pdf

[27] S. Kandula, D. Katabi, B. Davie, and A. Charny, "Walking the tightrope: Responsive yet stable traffic engineering," in *Proc. ACM SIGCOMM*, Aug. 2005, pp. 253–264.

[28] S. Sinha, S. Kandula, and D. Katabi, "Harnessing TCPs burstiness using flowlet switching," in *Proc. 3rd ACM SIGCOMM HotNets*, Nov. 2004, pp. 253–264.

[29] A. Elwalid, C. Jin, S. H. Low, and I. Widjaja, "MATE: MPLS adaptive traffic engineering," in *Proc. IEEE INFOCOM*, Apr. 2001, vol. 3, pp. 1300–1309.

[30] F. Giroire, A. Nucci, N. Taft, and C. Diot, "Increasing the robustness of IP backbones in the absence of optical level protection," in *Proc. IEEE INFOCOM*, Mar. 2003, vol. 1, pp. 1–11.

[31] R. Teixeira, K. Marzullo, S. Savage, and G. M. Voelker, "Characterizing and measuring path diversity of Internet topologies," in *Proc. ACM SIGMETRICS*, Jun. 2003, pp. 304–305.

[32] R. Teixeira, K. Marzullo, S. Savage, and G. M. Voelker, "In search of path diversity in ISP networks," in *Proc. ACM SIGCOMM IMC*, Oct. 2003, pp. 313–318.

[33] B. Huffaker, D. Plummer, D. Moore, and K. Claffy, "Topology discovery by active probing," in *Proc. SAINT*, Jan. 2002, p. 90.

[34] R. Sherwood and N. Spring, "Touring the Internet in a TCP sidecar," in *Proc. ACM SIGCOMM IMC*, Oct. 2006, pp. 339–344.

[35] R. Sherwood, A. Bender, and N. Spring, "Discarte: A disjunctive Internet cartographer," in *Proc. ACM SIGCOMM*, 2008, pp. 303–314.

Brice Augustin received the Ph.D. degree in computer science from UPMC Sorbonne Universités, Paris, France, in 2010.

Since 2006, he worked under the supervision of Timur Friedman and Renata Teixeira on the development of the Paris traceroute tool. He is now a Teacher and a Researcher with Université Paris-Est Créteil (UPEC), Paris, France. His research focuses on Internet topology measurement.

Timur Friedman (S'96–A'02–M'04) received the B.A. degree in philosophy from Harvard University, Cambridge, MA, the M.S. degree in management from Stevens Institute of Technology, Hoboken, NJ, in 1991, and the M.S. and Ph.D. degrees in computer science from the University of Massachusetts, Amherst, in 1995 and 2001, respectively.

He is currently a Maître de Conférences (Assistant Professor) with the Department of Engineering at UPMC Sorbonne Universités, Paris, France, and a Researcher at the LIP6 computer science laboratory. His research interests include network measurement systems and networking test-beds.

Renata Teixeira received the B.Sc. degree in computer science and the M.Sc. degree in electrical engineering from Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brazil, in 1997 and 1999, respectively, and the Ph.D. degree in computer science from the University of California, San Diego, in 2005.

During her Ph.D. studies, she was with AT&T Research, Florham Park, NJ. She is currently a Researcher with the Centre National de la Recherche Scientifique (CNRS), LIP6, UPMC Sorbonne Universités, Paris, France.