Classification of Load Balancing in the Internet

M.Sc. Rafael Luis Caldas Almeida
Advisor: Ítalo Fernando Scotá Cunha

Department of Computer Science
Universidade Federal de Minas Gerais

{rlca, cunha}@dcc.ufmg.br

Abstract. *Recent advances in programmable data plans, software-defined networks and the adoption of IPv6 support new and more complex load balancing strategies. In this work, we introduce the Multipath Classification Algorithm (MCA), a probing algorithm that extends traceroute to identify and classify load balancing on Internet routes. We generalize the current formalism to describe load balancing and extend existing measurement techniques to consider that load balancing can use arbitrary combinations of packet header fields. We propose optimizations to reduce the probing cost, applicable to both MCA and existing load balancing measurement techniques. Through large-scale measurement campaigns, we characterize and study the evolution of load balancing on the IPv4 and IPv6 Internet, using various transport protocols. Our results show that load balancing is more prevalent and that load balancing strategies currently in use are more mature than the previous characterizations have found. We share our measurement and analysis software as well as datasets with the Internet measurement community.*

1. Introduction

Internet traffic load balancing helps with increasing bandwidth, reliability, and reducing maximum link utilization. Load balancing can be configured in routers by mechanisms such as Equal-Cost Multi-Path (ECMP). Routers that perform load balancing, which we call *load balancers*, compute which network link a packet should be forwarded to as a hash function of the packet's *flow identifier*, a subset of fields in the packet's headers (e.g., IP addresses and port numbers). Network operators widely deploy load balancing, impacting most Internet routes.

Understanding the properties of Internet routes serves multiple goals. Operators use Internet route measurements to troubleshoot failures, routing anomalies, bad performance, and misconfigurations. Researchers measure and study properties of Internet routes to develop new models of its topology and design solutions to address limitations.

The standard tool for measuring Internet routes is traceroute. Classic implementations of traceroute, however, may miss routers and links as well as infer links that do not exist when measurements traverse load balancers. The Multipath Detection Algorithm (MDA) extends traceroute and systematically varies probes' flow identifiers to identify links used for load balancing at a router (if any).

Recent advances in programmable data planes, software-defined networking (SDN), and the adoption of IPv6 lead to increasingly complex load balancing deployments. Moreover, new router implementations support load balancing using flow identifiers composed

of arbitrary set of packet header fields. Existing MDA implementations, however, are unable to identify load balancers when they do not consider port numbers for load balancing. Load balancing misidentification may be aggravated over time as router implementations evolve and the adoption of programmable data planes, SDN, and IPv6 increases.

Given the applicability and importance of route measurements, researchers and network operators need route measurement tools to characterize load balancing in the Internet. In this thesis, we extend the existing formalism [Veitch et al. 2009] to consider that load balancers may use arbitrary combinations of packet header fields for load balancing.

We introduce the Multipath Classification Algorithm (MCA), a probing algorithm that identifies the set of bits in the packet header used by load balancers. We design and evaluate multiple optimizations to reduce the additional probing cost incurred for classification. We conduct large-scale campaigns of MCA measurements of IPv4 and IPv6 routes in the Internet using a diverse set of vantage points to characterize the prevalence and deployment of load balancing in the Internet today, and reappraise previous characterizations of load balancing [Augustin et al. 2011, Almeida et al. 2017].

Our results indicate that load balancing is more prevalent in Internet routes and their configurations are more complex than previously reported. Our reappraisal of previous results also reveals that IPv4 load balancing is still more prevalent than IPv6 load balancing. We observe, for the first time, load balancers considering ToS/traffic class and flow label fields for load balancing. We also detect and report on networks employing inter-domain load balancing. Finally, we make our tools and datasets available to the research community.

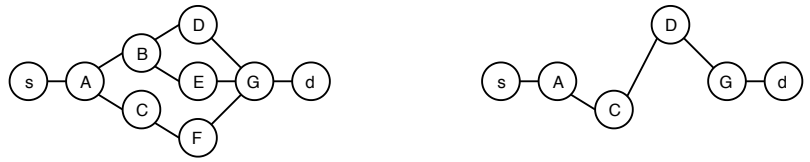
This work improves the *foundations* upon which we build tools to identify load balancing in the Internet. Our implementation of MCA provides more accurate information than existing tools, and can be useful in network characterization studies, particularly those concerned with traffic engineering and reliability to failures.

2. Definitions

At any given time, the connectivity between a fixed source s and a destination d is realized by its *current route*. A *route* can be *simple*, consisting of a sequence of IP interfaces from s toward d, or *branched*, when one or more *load balancing* routers (LB) are present, giving rise to multiple overlapping sequences (called "multi-paths" in [Veitch et al. 2009]). Thus a route is a directed graph with interface-labelled nodes. A route can be a sequence that terminates before reaching d due to routing changes or the absence of a complete route to the destination.

We define a *diamond* as the set of all interfaces between an LB and its *join point*, the interface where all the sub-branches originating from the LB first rejoin. An *outer* diamond is one that contains other *nested* LBs, and therefore their own diamonds. We define an *outermost diamond* as one which is not contained in some other diamond. Figure 1 shows an example branched route where the LB A defines an outermost diamond with join point G, containing a diamond defined by nested LB B, also with join point G. A branched route can have multiple (nonoverlapping) outermost diamonds.

By *hop-set* or *hop* we define the set of interfaces found at some fixed distance, or *radius*, from the source. We denote the hop-set at radius r in route p as $p[r]$. Conversely,



Route with load balancing Possible traceroute output

Figure 1. Traceroute and load balancing

for some valid hop-set h in route p, we let $p\langle h\rangle$ denote the radius of h. Thus $h = p[p\langle h\rangle]$. In Figure 1, $p = [s, A, \{B, C\}, \{D, E, F\}, G, d]$ is a route with three branches. The first hop is $p[1] = \{A\}$, the second hop is $p[2] = \{B, C\}$, and hop $\{G\}$ is found at radius $p\langle\{G\}\rangle = 4$. Assuming loop-free routes, hops are unique, although individual interfaces can be found in multiple hops when route branches have different lengths.

3. Load Balancer Model

Load balancing at a given router has two main aspects: (i) *when* load balancing is performed, and (ii) the *type* of load balancing when it is performed.

Aspect (i) is a function of whether multiple routes (through different next hops) to a destination prefix are known, the incoming link at the router, and the policy in place. To capture these dependencies, we test for LB behavior at the granularity of ⟨interface IP, destination prefix, upstream IP⟩ triples, where the interface and upstream IPs are those returned by traceroute, used, as usual, as surrogates for the true, unknown interfaces.

Aspect (ii) is a function of the router's forwarding decision algorithm. We model this as an ideal hash function over a *hash domain* consisting of a subset, in general not contiguous, of packet header bits. Each possible bit-field value over the domain is mapped to one of the available next hops. We consider *hash domains* are LB-specific.

The goal of the above model is to capture very general LB behavior. For (i), it allows load balancing to be triggered on some paths but not others, and the load balancing type to be path dependent. For (ii), it includes the classic LB types mentioned earlier, but allows for much richer structure that router vendors or SDNs may be implementing today or in the future. Per-packet LBs are included as the special case of an empty hash domain. The generality of arbitrary hash domains may seem intractable at a practical level. There are two important reasons why this is not the case which we exploit in our work:

1. *LB discovery:* By sending probes which vary all header bits (within some given target set), the presence of a LB can be detected *without needing to know* the exact hash domain. Moreover, if we assume the hash function maps uniformly over the next hops, then the same stopping rules and statistical guarantees used for MDA [Veitch et al. 2009] still apply *regardless* of the unknown hash domain.

2. *LB classification:* It is not necessary to fully determine the hash domain to gain useful results. For example, under the uniform hash assumption, if we vary a single field (e.g., source port) uniformly, we will induce a uniform sampling of the next hops provided the hash domain and the field have a non-empty intersection. Thus, the involvement of fields in LB decisions can be determined without a full resolution at the bit level.

4. Multipath Classification Algorithm

MCA receives as input the destination IP address d to trace; a confidence level α_d to bound the probability of detection errors, a confidence level α_c to bound the probability of classification errors; and a set $\mathcal{B}$ of *target* bits in the packet header defining the scope of LB detection and classification. We call the set of header fields that intersect $\mathcal{B}$ the *target* header fields, denoted $\mathcal{F}$.

MCA measures a route p hop-by-hop in increasing radius order. At each radius r, MCA first executes a modified version of MDA to enumerate next hops of routers in $p[r]$ that perform load balancing involving bits in $\mathcal{B}$ (§§4.1). MCA then sends additional probes to classify the type of any LBs found (§4.2). During both detection and classification phases, MCA employs optimizations to reduce the probing cost (omitted due to space constraints). After reaching the destination or stopping conditions, MCA outputs a directed graph representing the route discovered and the classification of each LB.

4.1. Load Balancer Discovery

MCA computes the *number* of probes to send through an interface to enumerate next hops following MDA [Veitch et al. 2009]. The novelty of MCA is the application of a broad definition of an LB search 'universe' based on $\mathcal{B}$, operating in the context of an arbitrary LB operating on the hash domain $\mathcal{H}$. In theory, if the *flow identifiers* (the value of the bits in $\mathcal{B}$) carried by probes are chosen uniformly, then so will be the bits in the intersection $\mathcal{H} \cap \mathcal{B}$, resulting in uniformity over the next hops. In practice, as $|\mathcal{H} \cap \mathcal{B}|$ may be small, it is important for flow identifiers to vary all bit combinations as much as possible, to ensure variety over (arbitrary and unknown) $\mathcal{H} \cap \mathcal{B}$.

Let $\phi_1, \phi_2, \ldots, \phi_n$ be the flow identifiers generated for the first n probes. We generate the new flow identifier ϕ_{n+1} greedily bit-by-bit in random order. The value of each bit is set such that it maximizes the Shannon entropy of the distribution of values seen over the $n + 1$ identifiers, restricted to the bits considered so far, with ties broken randomly. If the generated ϕ_{n+1} repeats an earlier identifier, bits are randomly flipped until uniqueness is obtained. This greedy heuristic enforces variability across all bit combinations and generates each flow identifier with $O(|\mathcal{B}|)$ operations.

4.2. Load Balancer Classification

For each interface identified as a LB in the discovery phase, our goal is to identify which header bits in $\mathcal{B}$ overlap its hash domain. MCA chooses one flow identifier known from the detection phase to traverse i, and sends multiple probes with that *fixed* identifier to i's next hop. We compute the number of probes to send as in MDA [Veitch et al. 2009]. MCA classifies i as a per-packet LB if responses arrive from multiple next hops.

If interface i is not classified as per-packet, MCA sends additional probes to infer which bits in $\mathcal{B}$ overlap with i's hash domain and classify the LB type. We define the classification of an LB interface i as correct when *all* target bits in $\mathcal{B}$ are correctly labelled as overlapping or not with i's hash domain $\mathcal{H}_i$. The number n of trials to perform for each bit is a function of a configurable confidence level α_c that determines the acceptable probability of classification error.

In each trial, MCA finds a flow identifier ϕ that traverses i and a flow identifier ϕ' that also traverses i and is identical to ϕ except for the value of one bit b. MCA

sends probes with ϕ and ϕ' to i's next hop (if not yet sent) and waits for the responses. This process is repeated until MCA identifies flow identifiers ϕ and ϕ' that i forwards to different interfaces, or until enough trials are performed to infer that bit b does not overlap with i's hash domain at the configured confidence level.

5. Dataset

We deployed MCA in 31 vantage points (VPs) in 6 platforms (cloud providers, monitoring testbeds, and one university) that provide IPv4 and IPv6 connectivity. The vantage points are spread across 16 countries in 5 continents. We run MCA toward a list of 19,866 IPv4 and 16,674 IPv6 addresses built to capture different classes of destinations. We run six measurement campaigns to cover all combinations of IP protocol (v4, v6) and transport (TCP, UDP, and ICMP). We augment our MCA measurements with IP-to-AS mapping information from Team Cymru, reverse DNS entries (PTR records), and network types from PeeringDB. Our dataset totals 766 GiB and is publicly available.

6. Characterization of Load Balancing

We characterize IPv4 and IPv6 load balancing on the Internet. We study the prevalence of LBs (§6.1), identify common hash domains and classes of LB behavior (§6.2), and revisit previous work and discuss the evolution of load balancing and path diversity (§6.3).

6.1. Occurrence of Load Balancing

Load balancing is prevalent: Even when ignoring load balancing in the vantage point's network, we find that 74% of IPv4 routes and 56% of IPv6 routes traverse at least one LB, and some traverse many more. When considering LBs in the origin network, we find that 86% of IPv4 routes traverse at least one LB, a fraction similar to that observed in 2009 [Augustin et al. 2011], and that 77% of IPv6 routes traverse at least one LB, not significantly larger than the 74% observed in 2016 [Almeida et al. 2017]. (See the 'prevalence' line in Table 1.) These results indicate the extent of traffic engineering used in the Internet and the need for considering LBs in Internet measurement efforts. Even though our measurements are collected from different vantage points to different destinations than measurements in previous work, and thus are not directly comparable, the observations are consistent and indicate prevalence of load balancers.

IPv4 load balancing is more widespread: We find 55% of ASes traversed in our IPv4 measurements employ IPv4 load balancing; compared to 41% in IPv6 measurements. Moreover, although 58% of ASes that appear in both IPv4 and IPv6 measurements employ both IPv4 and IPv6 load balancing, ASes more often employ IPv4 load balancing exclusively. We find that 82% of the ASes that employ IPv6 load balancing and are traversed in IPv4 measurements employ IPv4 load balancing, but that only 66% of the ASes that employ IPv4 load balancers and are traversed in IPv6 measurements employ IPv6 load balancing. These results indicate richer traffic engineering in IPv4 than in IPv6, possibly due to IPv4 still carrying most traffic or simply due to being around (and engineered) for longer. We find that load balancing is applied similarly across all transport protocols; the 'prevalence' row in Table 1 shows the fraction of routes in the dataset with load balancing, and that it does not depend on transport protocols.

	IPv4			IPv6		
	UDP	TCP	ICMP	UDP	TCP	ICMP
Per-flow	69.6%	69.8%	1.5%	77.6%	78.5%	0.3%
Per-dest	24.4%	24.1%	94.2%	13.3%	13.5%	90.1%
Per-app	1.8%	2.1%	0.0%	0.9%	0.8%	0.0%
Per-packet	0.1%	0.1%	0.1%	0.1%	0.1%	0.3%
Per-flow + flow label	—	—	—	2.9%	2.4%	0.0%
Per-dest + flow label	—	—	—	0.2%	0.3%	3.2%
Other	2.3%	2.6%	2.7%	3.2%	2.8%	3.9%
Not classified	1.8%	1.4%	1.5%	1.7%	1.6%	2.2%
Total	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%
Prevalence	74.9%	74.2%	72.9%	56.1%	55.8%	55.4%

Table 1. Breakdown of load balancer classifications and prevalence by protocol

6.2. Classes of Load Balancing

As a measure of the accuracy of our classifications, we check whether multiple measurements of the same LB identify identical hash domains. When using TCP probes, we find that 9.7% of IPv4 and 8.8% of IPv6 LBs are measured multiple times (this ratio is similar across TCP, UDP, and ICMP), and we infer that 98.1% of these LBs have the same hash domain, which is within our 95% confidence threshold.

Table 1 shows the percentage of LBs of each class per protocol for each of the most common types of load balancing (§3) in Internet routes, ignoring LBs in the vantage point's network. As explained before, load balancing is more prevalent on IPv4 routes and there is no significant difference in prevalence between transport protocols. We observe that per-flow LBs are the most common, followed by per-destination. We observe a few LBs considering only transport port numbers in their hash domain (we found this can be configured in MikroTik's RouterOS as "per-application load balancing") and some IPv6 load balancers employing the IPv6 flow label field.

We note that existing MDA implementations would correctly identify and classify only per-flow, per-destination, and per-packet load balancers. In particular, existing implementations would misclassify per-application load balancers as per-flow, and entirely miss load balancing using the DSCP, traffic class, and flow label fields.

Load balancing behavior is improving: Per-packet load balancing is at an all-time low relative to previous characterizations. As per-packet LBs induce packet reordering and TCP performance degradation, this is a positive trend. We find few LBs whose hash domains include the ICMP checksum field (shown as per-flow in Table 1). Augustin et al. [Augustin et al. 2011] reported a considerable decrease in the number of routers performing per-flow load balancing for ICMP between 2006 and 2009; we find this trend has continued, with an all-time low of IPv4 LBs considering the ICMP checksum field (3.2% of load balancers, down from 20% in 2009). This behavior may follow from more mature implementations defaulting to per-destination load balancing for ICMP packets.

New types of load balancing: We find that an average of 2.8% of IPv6 LBs consider the flow label field in their hash domain, when measuring with TCP. We also observe the use of flow label in conjunction with per-flow or per-destination load balancing classes: we classify 2.7% of IPv6 LBs as having this behavior. A significant fraction (85%) of LBs including the flow label in their hash domains are in content and infrastructure providers'

networks (e.g. Facebook's AS32934 and Google's AS15169), which adopt IPv6 and rely on advanced traffic engineering.

Traffic engineering triggers load balancing: IPv6's traffic class and IPv4's DSCP fields serve a similar purpose of classifying packets. Although their use is not widespread, we identified a non-negligible number of LBs with the IPv6 traffic class (2.7%) and IPv4 DSCP fields (2.6%) in their hash domains when measuring with TCP.

6.3. Diamonds and Branched Route Properties

We characterize LB outermost diamonds (called simply *diamonds* in this section) in Internet routes to better understand how LBs are used. We revisit Augustin et al.'s original diamond metrics and consider new ones [Augustin et al. 2011], with similar findings. Our measurements indicate that IPv4 diamonds are more prevalent and complex than IPv6 diamonds, traversing a higher number of LBs and resulting in greater path diversity.

For most metrics, we observe no significant differences between diamonds measured using IPv4 or IPv6; transport protocols (TCP, UDP, or ICMP); or diamonds from LBs with different hash domains. In the following we point out the significant differences we identified. We find that diamonds are usually short (80% of diamonds span at most 5 hops), that per-destination diamonds to be shorter than per-flow diamonds, and that IPv6 diamonds are shorter than IPv4 diamonds. Asymmetry is rare and small for both IPv4 and IPv6 (83% of the diamonds are symmetric). Diamonds are narrow: 86% of diamonds have only two edge-disjoint branches. IPv4 diamonds are slightly wider than IPv6 diamonds. Less than 1% of LBs have more than 16 next hops, which may be a limitation of router implementations. We find that diamonds usually have few LBs, but a few diamonds have more than 20. Diamonds often have between 1 and 3 LBs, which often appear in a symmetric fan-out pattern. IPv4 diamonds have slightly more LBs than IPv6 diamonds.

7. Relationship to Related Work

Addressing traceroute limitations: Previous work has shown that traceroute measurements may be incomplete or incorrect, e.g., due to routers that rate-limit responses or that fail to respond to probes with ICMP Time Exceeded messages; that do not decrement the TTL field of packets; that encapsulate packets using MPLS; or that respond to probes using off-path interfaces. Load balancing, which our work improves on, is a major source of artifacts in traceroute measurements.

Load balancing measurement and characterization: MDA was conceived for the IPv4 Internet. IPv6 implementations include Scamper and a new version of Paris traceroute. These implementations directly adapt MDA from IPv4 to IPv6, with the same underlying assumptions. Although these assumptions often hold (§6.2), we quantify how often and, more importantly, provide a principled approach to more accurately identify and classify both IPv4 and IPv6 LBs. Previous work used MDA to characterize load balancing on the IPv4 [Augustin et al. 2011] and IPv6 Internet [Almeida et al. 2017]. In this work, we use MCA to revisit these results and show how load balancing evolved over the years.

8. Resulting Artifacts

The contributions of this thesis will appear in the IEEE's 2020 International Conference on Computer Communications (INFOCOM), IEEE's flagship networking confer-

ence [Almeida et al. 2020].[1]

We implement MCA in a command-line tool to identify and classify load balancers. Our MCA implementation supports TCP, UDP, and ICMP measurements using both IPv4 and IPv6. It supports detection and classification covering bits in the DSCP, traffic class, flow label, destination address, source port, destination port, and ICMP checksum fields. The measurement community has expressed interest and we are considering integrating MCA in CAIDA's Scamper, a measurement tool already deployed in multiple monitoring systems.

Understanding traceroutes is not straightforward, and often requires combining multiple source of information such as IP-to-AS mapping, reverse DNS names, inter-AS relationships and AS business types. To facilitate our analysis we implement Route Explorer, a front-end for MCA measurements that allows inspection of probes and responses in an interactive graphical visualizations integrating multiple data sources.

We make our tools as well as our dataset, composed of more than 3 million MCA traces, publicly available to the community.[2]

9. Conclusion

In this thesis we presented a more general model for load balancing that considers that load balancers can use arbitrary bits in packet header fields for load balancing. We designed and implemented MCA, a probing algorithm that identifies and classifies load balancing, alongside optimizations to reduce its overall probing cost. We collected a large dataset to characterize load balancing practices for all combinations of IP protocol version (v4 and v6) and transport protocol (UDP, TCP, and ICMP). Our results show that load balancing is more prevalent and load balancing strategies more mature than previously reported. We identified that existing measurement tools cannot identify 4.7% of load balancers, and will deterministically misclassify an additional 2.3%. Given the rise of IPv6 traffic, programmable data planes, and software-defined networking, these percentages may increase and previous tools may become increasingly inadequate over time.

References

[Almeida et al. 2020] Almeida, R., Cunha, I., Teixeira, R., Veitch, D., and Diot, C. (2020). Classification of Load Balancing in the Internet. In *Proc. IEEE INFOCOM*.

[Almeida et al. 2017] Almeida, R., Fonseca, O., Fazzion, E., Guedes, D., Meira, W., and Cunha, Í. (2017). A Characterization of Load Balancing on the IPv6 Internet. In *Proc. PAM*.

[Augustin et al. 2011] Augustin, B., Friedman, T., and Teixeira, R. (2011). Measuring Multipath Routing in the Internet. *IEEE/ACM Trans. Netw.*, 19(3):830–840.

[Veitch et al. 2009] Veitch, D., Augustin, B., Friedman, T., and Teixeira, R. (2009). Failure Control in Multipath Route Tracing. In *Proc. IEEE INFOCOM*.

[1] This work has been developed exclusively by Rafael during his masters in collaboration with a team of researchers that provided guidance.

[2] `https://dcc.ufmg.br/~rlca/mca/`