Detecting load balancers in the Internet

Felix Hartmond
Advisor: Minoo Rouhi, Dominik Scholz
Seminar Future Internet SS2017
Chair of Network Architectures and Services
Department of Informatics, Technical University of Munich
Email: felix.hartmond@tum.de

ABSTRACT

Most tools for measurements of internet properties assume a model of the internet where only one path exists between a source and a destination which is taken by all packets send to this destination. With load balancers, which distribute traffic to multiple paths this model changes. This paper takes a look at the changes which have to been done to the classical traceroute to extend it to a tool which can discover multipath topologies. Additional we will look at results of mass-scans of routes and their findings about the overall usage of load balancers in the internet.

Keywords

traceroute, measurement, load balancing, topology, multipath

1. INTRODUCTION

In the traditional model of the internet a packet which is sent to a certain destination address always takes the same path through the internet. Every router on the path of the packet has a forwarding table which has a fixed mapping of destination addresses to output interfaces of the routers over which the address is reachable. To deliver the packet to the destination, the packet is sent always to the output interface which is defined in this table. This model does not apply to the internet in its current state. Nowadays there are special routers, called load balancers [14, 17], which induce the requirement of a changed model.

In contrast to normal routers, load balancers have multiple output interfaces over which the same destination address is reachable. It distributes all packets addressed to this destination over these interfaces which results in multiple routes leading to the same destination. Multiple possible links to the destination have the advantage of increased Network speed. Moreover, bandwidth and route bottlenecks are dissolved. The network resilience is improved since the destination is still reachable if a route fails.

Figure 1 shows a network with a load balancer. Routes in such networks split at a load balancer into parallel paths. The parallel paths merge back together before they reach the destination host at a convergence point.

There are many widely-used tools for running analysis on networks. One very common tool for analyzing which routers are on the path to a destination is *traceroute*. Like many other tools, it is designed on the old model of the internet without parallel routes. When using traceroute in a network with load balancers, different problems occur which make the results of the scan inaccurate. Therefore, a new tool is needed which can work in load balanced networks.

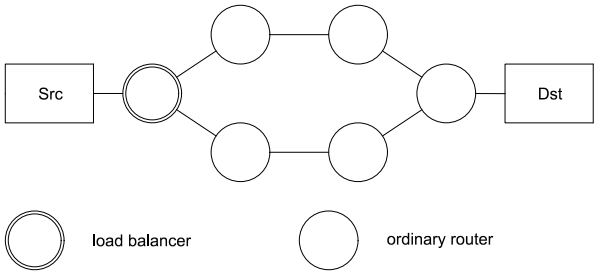

Figure 1: A network with a load balancer

This new tool should be able to identify load balancers on a route and discover the parallel paths behind them. This way the tool can deliver information about the whole topology on the way to the destination address.

Despite analyzing a single path such a tool can also be utilized to investigate the overall usage of load balancers in the internet. This can be done by scanning a lot of different path from different sources to different destinations. If the routes are well distributed over the internet conclusions can be drawn about the overall usage of load balancing in the internet.

There is already a public available implementation of such a multipath detection algorithm called *paris traceroute*. [18]

This paper proceeds as follows. Section 2 defines different types of load balancers. Afterwards in Section 3 we look at the classical traceroute and the necessary steps to extend it to a multipath aware tool like paris traceroute. In the end in Section 4 we look at bulk-scans with such tools which have been done by different researchers and look at their findings about the overall usage of load balancers in the contemporary internet.

2. TYPES OF LOAD BALANCERS

The output interface of a load balancer a packet is sent to is selected based on a so-called *flow identifier*. The flow identifier is calculated from different header fields. Augustin et. al. [6] defined three classes of load balancers which take

doi: 10.2313/NET-2017-09-1_03

different header fields into account:

per-destination. Per-destination load balancers use the destination address of a packet to calculate the flow identifier. This is similar to classic routing however, this balancer assigns different interfaces for different IP addresses in a prefix. All packets addressed to the same host are sent to the same interface. [15]

per-flow. Per-flow load balancers send all packets belonging to the same connection over the same route. Therefore, the IP-5-Tuple [1] is used for the calculation of the flow identifier. Packets from different connections belonging to the same source-destination-pair can be sent over different routes whereas packets belonging to the same connection will always use the same route.

per-packet. Per-packet load balancers do not use any header fields for the calculation of the flow identifier. Every packet is processed independently of the connection it belongs to. The selected output interface can be random or for example based on a round robin principle. Since this processing may lead to packet-reordering inside a connection it can have a bad effect on TCP performance [7]. [15]

When analyzing the usage of load balancing in IPv6 networks Almeida et. al. [2] discovered two additional IPv6-specific types of load balancers:

per-flow with flow label. This type of load balancer works like the already described per-flow balancer. However, instead of using the IP-5-Tupel for the calculation of the flow identifier, it uses the newly introduced flow label field from the IPv6 header. The flow label was introduced as it is less efficient to use the port numbers in IPv6, due to varying offsets of the layer-4-header caused by the new IPv6 extension headers. [9]

per-application. Per-application load balancers are similar to per-destination load balancers. Instead of using the destination addresses they use the destination port to calculate the flow identifier. Thereby the traffic is split according to the application it belongs to.

3. MULTIPATH DETECTION ALGORITHM

To scan networks with load balancers a tool is needed. This section describes the necessary steps to construct an algorithm to analyze a load balanced route. First Section 3.1 looks at the well-known tool traceroute which is widely used to measure paths in networks. The principle of traceroute will be the basis for the tool. Section 3.2 points out different problems of the classic traceroute has when scanning in networks with load balancers. Section 3.3 describes changes which have to be done to the classic traceroute so that consecutive probes for analyzing one possible route are not sent to different routes by a load balancer. Finally in Section 3.4 the changed traceroute is used to reveal all possible paths which packets can take to the scanned destination.

[1]source address, destination address, protocol, source port and destination port

3.1 Classical traceroute

The classical traceroute is a tool to track the route of a packet on its way through a network. To find the intermediate routers between the source and the destination host it sends probe packets addressed to the destination with a low TTL (time to live) header value. It starts with a TTL of one. Every router which forwards a packet decreases the TTL field by one. So the probe packet will have a TTL value of 0 at the first router. This router will drop the probe packet because of that send an ICMP time exceeded back to the source host [11]. Traceroute iteratively increased the TTL of the probe packets until it receives an ICMP port unreachable, which is the answer from the destination host. Traceroute sends per default three probes per hop to get information about each router even when answer packets are lost. Traceroute has different modes which use different types of probes. The traditional mode uses udp packets as probes. There are other methods which use for example ICMP echo requests or TCP SYN packets. [1]

3.2 Problems of the classic traceroute

It is necessary to be able to match the incoming responses to the send probes. The IP header and the first 64 bits of the probe packet are included in an ICMP time exeeded message [11]. The UDP header is only 64 bits long and is completely included in an ICMP time exeeded message. When running with default settings, Traceroute uses the destination port to match the answers and therefore increments the destination port for each sent probe [1].

Like mentioned in Section 2, some types of load balancers include the destination port in the calculation of the flow identifier, that the probe packets of one scan can be distributed to different paths. This leads to different problems and measurement anomalies.

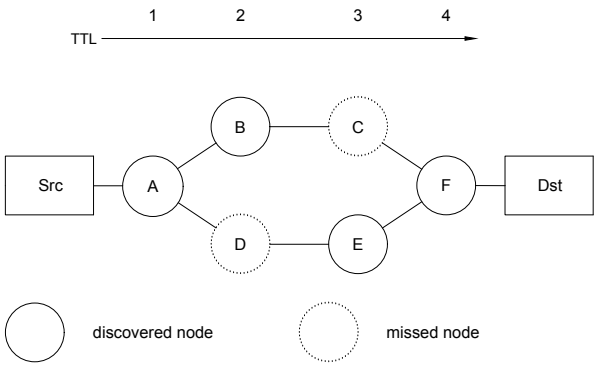

Figure 2: Missing nodes with classical traceroute

One problem is that some of the nodes on the route stay undiscovered. When there are different routers with the same distance from the source only one of them can be reached by a single probe and therefore all other routers at the same distance stay undiscovered. [4]

In the example in Figure 2 the probe with TTL 2 takes the upper route. Router B is detected, but router D, which is also two hops away from the source, stays undiscovered. The same situation appears with the probe with TTL 3 which is

doi: 10.2313/NET-2017-09-1_03

send to the lower route in the example. Router E is found but router C stays undiscovered.

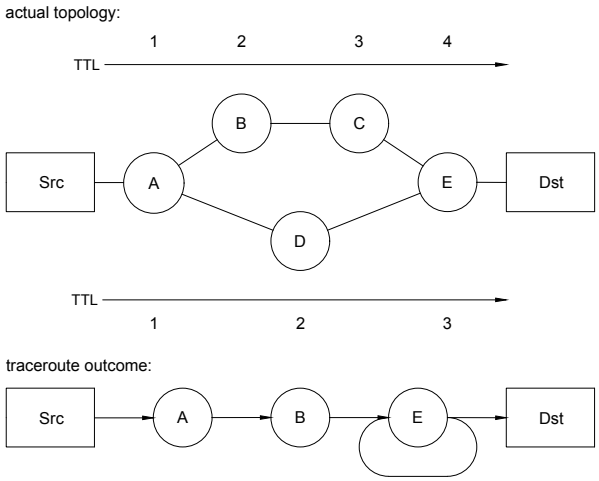

Figure 3: Multiple detection of a router with classical traceroute

Another problem is that router, can be detected multiple times. The result of the traceroute then shows a loop. Such loops can appear if a load balancer distributes probes to routes of different length. With such routes, routers behind the convergence point of the balanced routes are reachable over different distances over the different routes. When a probe takes the short route and the next probe takes a route which is one step longer, the probes end up at the same route which will show up multiple times in the result. In the example in Figure 3 the probe with TTL 3 takes the lower route and the probe with TTL 4 the upper one. They both end up at router E which shows up two times in the result. [4]

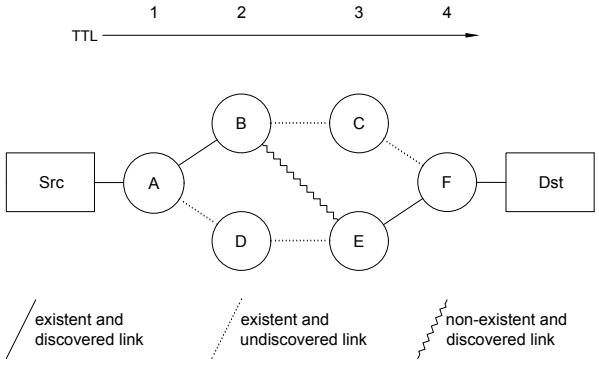

Figure 4: Detection of non-existent links with classical traceroute

A third problem is the detection of non-existing links. The classical traceroute simply draws links between the incrementally discovered routers to show the discovered route. When detected routers are distributed on parallel routes this approach does not work. Routers with a distance difference of one do not have a direct link between them when they are on different routes. In the example in Figure 4 traceroute discovers the routers A, B, E and F. When simply connecting the discovered routers a link is drawn between the routers B and E which does not exist in the topology. [4]

3.3 Controlling the balancers decision

To avoid the problems discussed in the last section traceroute has to be improved to a tool which is aware of the existence of parallel load balanced paths in a scanned network and does not interfere with the existent load balancers without intending this. To find all routers on one of the paths we want all the probes to be sent to the same route by a load balancer, but some fields have to be changed constantly to be able to match the responses to the sent probes.

For each mode of traceroute (UPD, ICMP Echo and TCP) a way is needed to put an identifier into the first 64 bit of the transport header without touching fields which are used to calculate the flow indentifier for any of the load balancer types which were introduces in Section 2.

When scanning the route to a fixed destination, per-destination load balancers do not have to considered because they only use the destination address which is never modified.

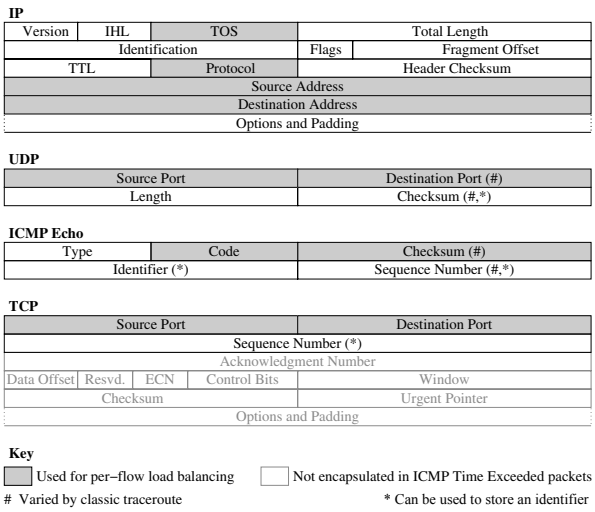

Figure 5: Overview over header fields [4]

Figure 5 summarizes which header fields of the different transport headers are used for per-flow load balancing, which are varied by the classical traceroute and which can be used for storing an identifier without interfering with per-flow load balancers.

For TCP probes the sequence number and for ICMP Echo request probes the indentifier and the sequence number can be easily be set to store an identifier. For UPD probes there are no fields which can take user-defined content, but the checksum field can be used. To vary this field the payload has to be varied to cause different checksums.

Per-packet load balancers can not be controlled due to their nature of handling packets independently from any header fields. Therefore, the tool cannot completely reconstruct a topology which contains per-packet load balancers.

doi: 10.2313/NET-2017-09-1_03

Per-flow balancers which use the IPv6 flow label do not cause problems. When looking at per-flow balancers for IPv4 there already header fields found which can also be used for IPv6 as well. So the flow label field can be left untouched without further problems. The flow label should be initialized with a non-zero value because any router can initialize a flow label which is zero but has to leave a non-zero flow label untouched [3].

To also cover per-applicatoin load balancers no additional changes have to be done. Per-application load blancers use only the destination port field to calculate the flow identifier which was already excluded from the varying fields when covering per-flow balancers.

3.4 Revealing the whole topology

With the changes mentioned in Section 3.3, the modified traceroute is now able to track a route without having probes distributed to different paths by non-per-packet load-balancers. But when tracking a route to a destination we do not want to know only one possible path. Instead, we want to know all possible paths which traffic to the destination can take. To achieve this, the tool has to recognize if there are load balancers on the route and have to track the multiple routes behind them separately. This is only possible for non-per-packet load balancers. Per-packet load balancers can only be detected but due to their non-controllable behavior it is impossible to send probes to a certain path behind them.

Keeping header fields which influence the load balancer's decision steady is very straightforward. But inciting the balancer to send a probe to another output as the previous probe is not. It is not possible to force the probe to be sent to a chosen output interface due to missing information about how the header fields are used and how many outputs the load balancer distributes traffic to.

Additionally, it is unknown which routers are load balancers at all. To identify a router as a load balancer and find all its output interfaces a stochastic approach can be used when sending probes with varying flow identifier.

Because we do not know the exact algorithm of the balancer we can not say if we have found every path for sure. So we have to set a level of confidence which we want to reach with the probing. For probing we assume that all load balancers distribute traffic equally to all output interfaces.

The classical traceroute sends three probes per hop on default [1]. To test if the router is a load balancer this is not enough. With three probes the probability is 25% that all three probes take the same path if the probed router is a load balancer with two output interfaces.

To test if a router is a load balancer it is assumed that the router is a balancer has two output interfaces. Then the tool tries to disprove this assumption with the set level of confidence. For a level of confidence of 95% this needs six probes. If all the six probes take the same way, the tool assumes that the router is a normal router and continues with the next hop. If the probes take more than one route the assumption is set to a load balancer with three output interfaces and more probes are sent to this router. This is continued until an assumption is disproved at the set confidentiality. Scans on the internet have shown that load balancers have up to 16 output interfaces. In such a case, 96 probes are needed to find all interfaces at the confidentiality of 95%. [5]

After identifying a router as a load balancer, it can be tested if the router is a per-flow or a per-packet one. This is possible by assuming a per-packet balancer and sending 6 probes with fixed flow indentifier for a 95% confidentiality of the decision. If all packets go to the same interface the balancer is labeled as a per-flow balancer, otherwise it is labeled as a per-packet balancer. [5]

When a per-packet load balancer is detected it is not possible to reveal the whole topology. But at least all output interfaces of the known routers are revealed.

The tool iteratively probes all new discovered routers and draws the topology of these routers and their paths. When probing a router on a path behind a load balancer first flow identifiers have to be found which reach the tested router. Therefore, a number of identified and those are selected who reach the router which should be probed [6].

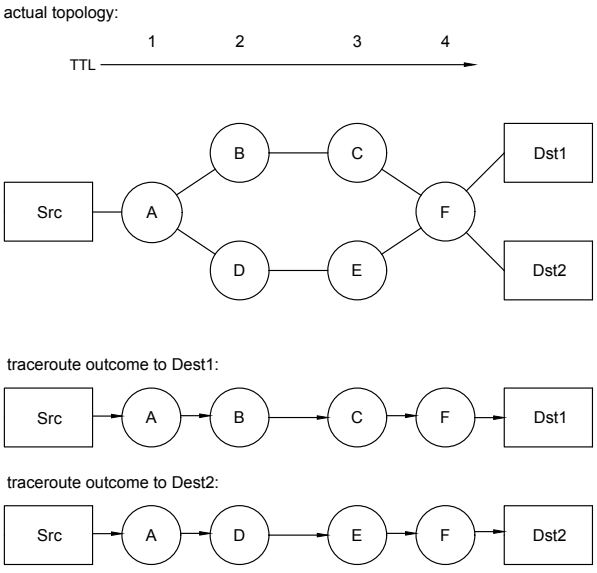

Figure 6: Per-destination load balancing [5]

So far, the tool does not detect per-destination load balancers. To extend the tool to also cover per-packet load balancers it has to trace towards a subnet instead of a single destination address. In the example in Figure 6 node A is a per-destination load balancer. If tracing toward address *Dst1* only the upper path is discovered and only the lower path if tracing towards destination *Dst2*.

To also detect per-destination load balancers the tool creates also flow identifiers which differ in the destination inside the destination prefix address when creating probes for testing if a router is a load balancer.

doi: 10.2313/NET-2017-09-1_03

4. USAGE OF LOAD BALANCERS

To analyze the overall usage of load balancing in the internet different scans [2, 5] have been done with multipath detection algorithms.

Metrics which can be used to describe topologies which are caused by load balancers will be covered in Section 4.1. Section 4.2 takes a look at a method which can be used to realize scans to gain information about the overall usage of load balancing in the internet. Finally, Section 4.3 presents the results the researchers got when executing and evaluating their scans.

4.1 Metrics

Augustin et. al. defined diamonds to describe the structure of load balanced paths. A diamond is defined as a subgraph of the path which begins at a divergence point and ends, two or more hops later, at a convergence point. All possible flows from the source to the destination have to traverse both divergence and convergence point. [6]

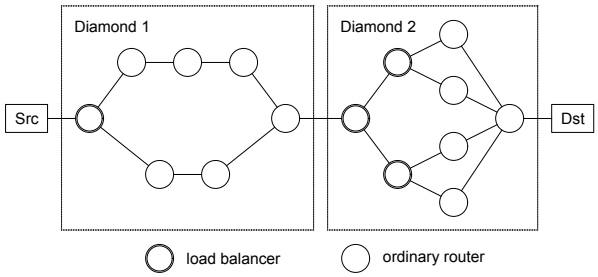

Figure 7: A Network with two diamonds and four load balancers [6]

Figure 7 shows an example of a route with two diamonds. A route can contain multiple diamonds and a diamond can contain multiple load balancers. In the example diamond 2 contains three load balancers.

Further Augustin et. al. [6] defined three metrics to quantize such diamonds:

Diamond width. To describe the diamond width two metrics are used. The min-width describes the amount of link-disjoint paths between the divergence and convergence points. This defines a lower limit for the path diversity inside the diamond. In the example, both diamonds have a min-width of 2. Additionally the max-width describes the maximum number of interfaces which can be reached at a given distance form the source. Diamond 1 has a max-width of 2 and diamond 2 has a max-width of 4.

Diamond length. The diamond length describes the maximum number of hops between divergence and convergence point. In the example Diamond 1 has the length 4 and diamond 2 has the length 3.

Diamond symmetry. A diamond is considered symmetric if all possible path are of the same length. If not, the diamond is asymmetric and the asymmetry describes the difference between the longest and the shortest path. In the example, diamond 1 is asymmetric. The longest path has a length of 4 and the shortest a length of 3 so the diamond has an asymmetry of 1. Diamond 2 is symmetric.

4.2 Scanning procedure

To get results which are as close as possible to the actual usage across the entire Internet, they choose different locations as their sources for their scans which are spread as widely as possible around the world. Due to the non-availability of nodes in some regions and restrictions on tracing arbitrary devices from some nodes [5], it is not possible to reach an ideal distribution.

Due to the huge size of the IP address space and the costly scanning procedure, it is not feasible to scan routes towards all possible addresses. Because of that a preselection of destination addresses has to be done. Different approaches for generating destination lists have been used. For example a list of the addresses of the 500 most popular websites was used. Alternatively a list was used which was created by MIT researchers by running classical traceroutes to addresses from BGP tables of one of their routers and added the last responding host to the list [6].

For deeper analysis of the context in which the discovered load balancers are used, more information about their surrounding is needed. For example it is possible to analyze in which autonomous systems (AS) load balancers are used in. To find the AS for a discovered node IP to AS mapping services can be used [19].

4.3 Results

In this section, we will look at the results of two research groups. The first one around Augustin [6, 5] analyzed the usage of load balancers in the IPv4 internet. The second one around Almeida [2] analyzed the usage in the IPv6 internet.

4.3.1 Prevalence of load balancer types

	Overall			Filtered	
	Fraction of Balancers	% Routes IPv6	% Routes IPv4	Fraction of Balancers	% Routes IPv6
Per-destination	29.3%	43.5%	78.0%	29.2%	11.1%
Per-flow	50.0%	30.0%	54.8%	50.1%	17.7%
Per-packet	10.7%	30.1%	1.0%	10.6%	7.7%
Per-flow with TC	3.2%	14.8%	—	3.2%	3.3%
Per-application	6.0%	5.1%	—	6.0%	3.3%
Others	0.8%	1.2%	—	0.9%	0.6%
Total	100%	74%	92%	100%	29%

Figure 8: Overview of the prevalence of load balancers in IPv4 [6] and IPv6 [2]

Figure 8 shows an overview over their results about the prevalence of the different types of load balancers. The results Almeida et. al. received from their measurements showed very different results depending from the source of the scans. When analyzing their results they noticed that the sources which showed a very high prevalence of load balancers had a balancer one or two hops away from the source so almost all traces traverse this balancer. When ignoring these balancers all their sources showed very similar results. The values in the filtered column have these balancers removed if they are in the same AS as the source. [2]

doi: 10.2313/NET-2017-09-1_03

Augustin et. al. found out that per-destination load balancing is the most prevalent type in the IPv4 internet followed by per-flow load balancing. They found out that per-packet load balancing is used very rarely and seems to disappear. This can be explained with the negative effects of per-packet load balancing on TCP connections due to a high risk of packet reordering [7, 6].

Almeida et. al. also discovered per-destination, per-flow and per-packet, which were already used with IPv4, as the most prevalent types. Surprisingly they found a significant higher amount of per-packet load balancing than Augustin et. al. found for IPv4. Additionally, they found per-flow balancers which utilize the traffic class field in addition to the source and the destination port. They also found a small amount of load balancers which only use the TCP ports for load balancing [2].

4.3.2 Diamond characteristics

This section studies the properties of the discovered diamonds measured with the metrics introduced in Section 4.1. The measurements of Augistin et al. and Almeida et. al. have shown very similar results. So the characteristics are similar for IPv4 and IPv6 load balancing.

Diamond length. Most diamonds are short. Augustin et. at. found out that depending on the load balancer type the amount of diamonds with length two is between 26% (per-destination) and 90% (per-packet) [5, 2].

Diamond width. Most diamonds are narrow. Most diamonds have a max-width of less than 5. Additionally, Augustin et. al. discovered a few very wide diamonds with a max-width of 16 which is the maximum value for some vendors [17]. Such balancers are for example used to bundle a lot of low capacity links [5, 2].

Diamond symmetry. Most diamonds are symmetric. If asymmetry is present it is usually very low. Almeida et. al. discovered that most of the asymmetric diamonds they found belong to per-destination load balancers, so the asymmetry has no negative effects [5, 2].

4.3.3 Intradomain and Interdomain load balancing

Augustin et. al. also looked at the relation between diamonds and ASes. They defined load balancing with diamonds which are entirely inside one single AS as intradomain load balancing. In contrast, they defined load balancing with diamonds spread across multiple ASes as interdomain load balancing. [6]

Even if these two types do forwarding the same way they differ when looking at the routing protocols. For intradomain routing multiple intradomain routes can be installed in the forwarding table due to the equal-cost multipath capabilities of common intradomain routing protocols such as IS-IS [8] and OSPF [10]. In contrast interdomain routing which is done with the internet's interdomain routing protocol BGP does not allow to install multiple routes. Despite this limitation some vendors like Cisco [13] and Juniper [16] have build multipath capabilities for BGP [12] into their routers.

Augustin et. al. found out that most diamonds are created by intradomain load balancing. They analyzed their measurements from one of their sources, which is located in Paris, in detail towards intra- and interdomain load balancing and found out that 86% of all per-flow load balancers fit into a single AS. Diamonds which cross two ASes were rare and diamonds which cross three ASes were extremely rare. They discovered that it seems that the BGP multipath capabilities are disabled in most core routers. [6]

5. CONCLUSION

We have seen why the classical traceroute fails to deliver usable results in networks with load balancers. In detail we have seen that the distribution of packets to different links can lead to classical traceroute to leave nodes undiscovered, to detects nodes multiple times and to show nonexistent links in its result. Because these effects make the resulte unreliable the need for an improved tool, a multipath detection algrithm, is there.

Then we looked at the process of constructing such a mulitpath detection algorithm. First, we looked at nessessary changes for the classial traceroute. Due to these chages header fields which are evealuated by load balancers stay uncanged and the required identifier for a probe is stored in unused fields. Then, we covered how this modified traceroute can be extendet to a tool to reveal the complete topology of a route instead of only a single path. Therefore the routers on the route are tested to be load balancers with a stocastic approach. If a router is identified as a load balancer the different pathes behind it are examined separately. Addionally the type of the load balancer is examined with additonal probes.

Finally, we have looked at measurements which have been done with multipath detection algorithms and their results about the overall usage of load balancers in the internet. We have seen that load balancing is very widely used nowadays and most load balancing topologies are short, small, symmetric and do not span across multiple ASes.

6. REFERENCES

[1] traceroute(8) - traceroute for linux. `http://man7.org/linux/man-pages/man8/traceroute.8.html`. Accessed: 2017-04-07.

[2] R. Almeida, O. Fonseca, E. Fazzion, D. Guedes, W. Meira, and Í. Cunha. *A Characterization of Load Balancing on the IPv6 Internet*, pages 242–254. Springer International Publishing, Cham, 2017.

[3] S. Amante, B. Carpenter, S. Jiang, and J. Rajahalme. Ipv6 flow label specification. RFC 6437, RFC Editor, November 2011.

[4] B. Augustin, X. Cuvellier, B. Orgogozo, F. Viger, T. Friedman, M. Latapy, C. Magnien, and R. Teixeira. Avoiding traceroute anomalies with paris traceroute. In *Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement*, IMC '06, pages 153–158, New York, NY, USA, 2006. ACM.

[5] B. Augustin, T. Friedman, and R. Teixeira. Measuring load-balanced paths in the internet. In *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*, IMC '07, pages 149–160, New York,

NY, USA, 2007. ACM.

[6] B. Augustin, T. Friedman, and R. Teixeira. Measuring multipath routing in the internet. *IEEE/ACM Transactions on Networking*, 19(3):830–840, June 2011.

[7] E. Blanton and M. Allman. On making tcp more robust to packet reordering. *SIGCOMM Comput. Commun. Rev.*, 32(1):20–30, Jan. 2002.

[8] R. Callon. Use of osi is-is for routing in tcp/ip and dual environments. RFC 1195, RFC Editor, December 1990. http://www.rfc-editor.org/rfc/rfc1195.txt.

[9] S. E. Deering and R. M. Hinden. Internet protocol, version 6 (ipv6) specification. RFC 2460, RFC Editor, December 1998. http://www.rfc-editor.org/rfc/rfc2460.txt.

[10] J. Moy. Ospf version 2. STD 54, RFC Editor, April 1998. http://www.rfc-editor.org/rfc/rfc2328.txt.

[11] J. Postel. Internet control message protocol. STD 5, RFC Editor, September 1981. http://www.rfc-editor.org/rfc/rfc792.txt.

[12] Y. Rekhter, T. Li, and S. Hares. A border gateway protocol 4 (bgp-4). RFC 4271, RFC Editor, January 2006. http://www.rfc-editor.org/rfc/rfc4271.txt.

[13] Cisco Systems, Inc. Bgp best path selection algorithm. http://www.cisco.com/c/en/us/support/docs/ip/border-gateway-protocol-bgp/13753-25.html. Accessed: 2017-04-12.

[14] Cisco Systems, Inc. How does load balancing work? http://www.cisco.com/c/en/us/support/docs/ip/border-gateway-protocol-bgp/5212-46.html. Accessed: 2017-04-10.

[15] Cisco Systems, Inc. *Configuring a Load-Balancing Scheme*, chapter 4, page 59. Cisco Systems, Inc., San Jose, CA 95134-1706, USA, 2015.

[16] Juniper Networks. Configuring bgp to select multiple bgp paths. http://www.juniper.net/techpubs/software/junos/junos94/swconfig-routing/configuring-bgp-to-select-multiple-bgp-paths.html. Accessed: 2017-04-12.

[17] Juniper Networks. Configuring load-balance per-packet action. http://www.juniper.net/techpubs/software/junos/junos70/swconfig70-policy/html/policy-actions-config11.html. Accessed: 2017-04-10.

[18] Paris Traceroute. Paris traceroute. https://paris-traceroute.net/. Accessed: 2017-04-11.

[19] Team Cymru Inc. Ip to asn mapping. http://www.team-cymru.org/IP-ASN-mapping.html. Accessed: 2017-04-11.

doi: 10.2313/NET-2017-09-1_03