Traceroute Anomalies

Martin Erich Jobst
Supervisor: Dipl.-Inf. Johann Schlamp
Seminar Future Internet SS2012
Chair for Network Architectures and Services
Department for Computer Science, Technische Universität München
Email: martin.jobst@tum.de

ABSTRACT

Traceroute is – after ping – one of the most widely used network diagnostic tools, due to its simplicity and yet very wide range of applications. Possible applications for traceroute range from simple error diagnosis to large scans, which reveal the underlying network topology. However, since traceroute was not built with modern network technologies in mind, it faces many difficulties. These difficulties usually manifest themselves in strange or false results, so-called *anomalies*. This drastically affects traceroute's abilities for network diagnosis and analyzation, especially in large-scale networks. The correct use of traceroute and interpretation of its output has therefore become more and more important. Projects trying to map the topology of the Internet are also greatly affected by traceroute anomalies, as they usually have to solely rely on traceroute and similar scans.

This paper gives a systematic overview of the most frequent traceroute anomalies. The main symptoms of each anomaly are examined based on example scenarios and corresponding output. Additionally, the consequences each anomaly has on the diagnosis of network failures, congestion and mapping efforts is analyzed. This also includes typical wrong conclusions drawn from anomalous traceroute results. Finally, several existing and promising future countermeasures against the respective anomalies are presented and analyzed.

Keywords

Traceroute, Anomalies, Load Balancing, Paris Traceroute, Traceroute Extensions

1. INTRODUCTION

Developed by Van Jacobson in 1988 [4], traceroute has become one of the most important tools for diagnosing network problems. It is used to measure the path a packet takes from the local host to a specified destination. Additionally, for each hop on the path, the *round-trip times* or RTTs are recorded.

Since large-scale networks, like the Internet, are usually operated by many different administrative entities, complete and up-to-date information about the network topology and state is usually difficult to obtain. In many cases, measurements taken with traceroute are the only way, to obtain such information. The traceroute user base therefore ranges from end-users in small home LANs to operators of large backbone networks. The conclusions taken from traceroute output are often the only way to effectively diagnose network problems, like link failure and congestion issues, and to analyze traffic flow. As even information about the Internet's global network topology is relatively scarce, some projects have emerged which try to map the topology based on several different scans [8]. These projects also heavily rely on the accuracy of the results taken from traceroute and similar scans to thousands of destinations.

However, the classic traceroute was not built with modern network management technologies in mind. Since it is mostly oblivious to such new developments, it often generates false or strange results, so-called *anomalies*. These anomalies make diagnosing network problems with traceroute much more difficult, if not downright impossible.

Several measurements [1, 2, 10] have in fact shown that, against common belief, traceroute anomalies are actually occurring quite frequently. Hence, traceroute anomalies are a very big obstacle for network administrators, as well as the various efforts to create a complete and accurate map of the Internet. To effectively use traceroute and correctly interpret its output has become more and more of a skill today, as can be seen in additional efforts taken to instruct network operators on these topics, e.g. [9].

In section 2 general background information for this paper is discussed. Section 3 contains an overview of the respective anomalies, with a description of their effects and common causes, as well as example scenarios and corresponding output. The impact on network analysis and diagnosis of each anomaly is also analyzed. In section 4 several existing solutions to mitigate problems related to traceroute anomalies are examined, as well as different existing and future extensions for traceroute. Finally, section 5 summarizes the results of the discussed anomalies and solutions.

2. BACKGROUND

This section presents some background information which is important for the understanding of this paper. It gives a general overview about traceroute, as well as different load balancing principles and routing techniques which affect traceroute.

2.1 Traceroute Basics

The classic traceroute works by sending out ICMP echo requests, so-called *probes,* with a fixed TTL. The TTL value usually starts at one and is incremented on each probe. Each hop then decrements the TTL by one, when forwarding the

doi: 10.2313/NET-2012-08-1_02

packet, and if the TTL reaches zero, it sends back an ICMP TTL exceeded error. This way, traceroute gets an error message from each hop between itself and the destination, containing the IP address of each hop. By subtracting the time when the error is received by the time the probe was sent, traceroute is also able to compute the RTT for each hop. Finally, when the probe reaches the destination, traceroute gets an ICMP echo reply and stops. The basic traceroute message flow is shown is figure 1.

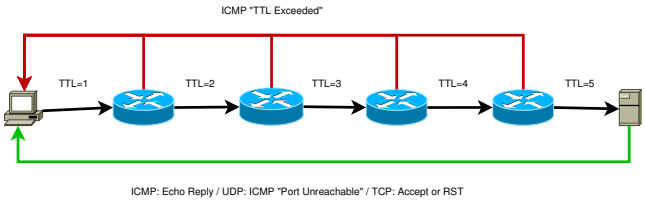


Figure 1: A typical traceroute message flow

Modern traceroute variants also include support for UDP and TCP, as well as IPv6. When using UDP or TCP, the only difference to ICMP is that the packet received from the destination is usually either an ICMP port unreachable or TCP RST packet, respectively. Typical exceptions of this are if the packet is blocked by a firewall or if the port is in use, in which case no error is returned.

2.2 Load Balancing

Load balancing in general is the distribution of packets among several different links or paths. Load balancing mechanisms are usually distinguished in three categories, explained below.

2.2.1 *Per-flow Load Balancing*

Per-flow load balancing tries to distribute packets according to their so-called *flow*. A flow is usually identified by the 5-tuple of the corresponding packets, i.e. IP addresses, protocol and ports. This is done, so that packets belonging to the same connection are delivered in order to the destination, as best as possible.

2.2.2 *Per-packet Load Balancing*

Per-packet load balancing distributes each packet individually among the links available. Normally, the packets are distributed randomly or in a round-robin fashion. This has the advantage of requiring less effort inside the router, but on the other hand often introduces huge jitter to connections, especially if the different routes aren't equal in length. Per-packet load balancing usually presents the most problems to traceroute in general, because of its random nature.

2.2.3 *Per-destination Load Balancing*

Per-destination load balancing distributes packets based on their destination. It is mostly identical to classic routing and normally has little to no impact on the network. Traceroute usually remains completely unaffected by per-destination load balancing.

2.3 MPLS

Multiprotocol Label Switching or *MPLS*, described in RFC 3031 [7], is used to effectively route packets in large-scale networks, e.g. the Internet. Normally, each router has to make its own routing decisions based on the information contained in the IP header. Since IP addresses are spread quite thin in the Internet, this often requires routers to hold very large routing tables. Additionally, since only few fields in the IP header, i.e. the source and destination address, as well as the TTL, are actually used for routing, it introduces a large unnecessary overhead.

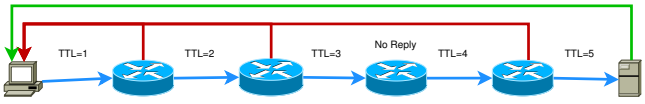
MPLS uses its own header, which encapsulates the original packet. With this, only the first router has to examine the IP header and assigns a *Forwarding Equivalence Class* or *FEC* to the packet in the new header. This designates destinations which are considered equivalent for routing decisions. Since most destinations can actually be grouped together into large blocks, the corresponding tables can be very small. Subsequent routers are then able to base their routing decisions on the much shorter and easier to handle FEC in the MPLS header. Since the TTL values can be copied back and forth between the IP and MPLS header, MPLS routers are also able to honor TTL values set in the original packet. Additionally, RFC 4950 [3] enables the generation and use of ICMP packets in an MPLS context. Hence, MPLS routers may offer basic support for traceroute.

3. TRACEROUTE ANOMALIES

The following is a description of the most frequent traceroute anomalies and their characteristic symptoms. For each anomaly an example message flow is shown, as well as the corresponding output. Additionally, the impact on network diagnosis and analysis is examined.

3.1 Missing Hops

This is the most basic anomaly, where one or more hops are missing from the traceroute output. It usually occurs when a router is protected by a firewall or otherwise configured not to generate ICMP TTL exceeded errors. An example message flow for this anomaly is shown in figure 2, along with the corresponding output. The three asterisks highlighted below, meaning no reply was received for the respective probe, are the key signs for this anomaly.




```
host1.site:/ # traceroute server1.site
traceroute to server1.site (192.168.0.6), 30 hops max, 40 byte
packets using UDP
1 router1.site (192.168.0.2) 1.985 ms 0.961 ms 0.091 ms
2 router2.site (192.168.0.3) 15.899 ms 16.979 ms 17.369 ms
3 * * *
4 router4.site (192.168.0.5) 18.207 ms 17.970 ms 18.431 ms
5 server1.site (192.168.0.6) 17.704 ms 17.671 ms 18.202 ms
```

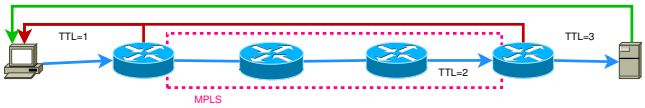
Figure 2: Missing hops example

This anomaly is very easy to notice and of little impact in real life. However, when the network problem is situated exactly on the hop which is not responding, it may actually be quite annoying.

Another reason for missing hops in the traceroute output are MPLS routers which don't honor the TTL value set in the IP

doi: 10.2313/NET-2012-08-1_02

header. Thus, one or more MPLS hops are simply missing in the resulting output. In figure 3 an example scenario for missing MPLS hops is shown. The line where the MPLS routers should appear is highlighted in the output.



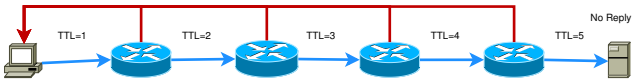

```
host1.site:/ # traceroute server1.site
traceroute to server1.site (192.168.0.6), 30 hops max, 40 byte
packets using UDP
1 router1.site (192.168.0.2) 1.764 ms 0.874 ms 0.218 ms
2 router4.site (192.168.0.3) 15.765 ms 16.847 ms 17.723 ms
3 server1.site (192.168.0.6) 18.934 ms 19.387 ms 19.876 ms
```

Figure 3: Missing MPLS hops example

This anomaly is very hard, if not impossible, to notice and sometimes very annoying, especially if a MPLS related problem is to be diagnosed.

3.2 Missing Destination

Another, also quite trivial, anomaly is when the destination is missing from the traceroute result. In this case traceroute simply continues with the scan, until it reaches the maximum probe TTL value or if it is interrupted by another constraint. An example would be to stop after a certain number of unsuccessful tries. A special side effect of this anomaly is, that there may be an arbitrary number of hops missing at the end of the output. The usual case for a missing destination is a destination which is protected by a firewall. Figure 4 shows an example of this anomaly. The output again contains the typical three asterisks for the unsuccessful probes and then continues on until the maximum TTL is reached.




```
host1.site:/ # traceroute server1.site
traceroute to server1.site (192.168.0.6), 30 hops max, 40 byte
packets using UDP
1 router1.site (192.168.0.2) 2.172 ms 1.845 ms 1.239 ms
2 router2.site (192.168.0.3) 16.874 ms 17.874 ms 16.276 ms
2 router3.site (192.168.0.4) 17.943 ms 18.098 ms 17.764 ms
4 router4.site (192.168.0.5) 18.897 ms 17.974 ms 18.843 ms
5 * * *
[...]
```

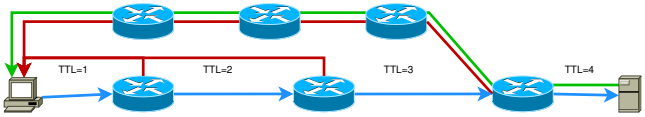
Figure 4: Missing destination example

Again, this anomaly is easily noticeable and merely annoying, for the most part. As it causes scans to take unnecessary long, this anomaly may become a problem, though, especially in cases where the complete topology of a network is to be scanned.

3.3 False Round-Trip Times

This is the case, when the round-trip times reported by traceroute are false. There are usually two reasons for this, either asymmetric packet paths or MPLS routing.

When the respective paths to and from the destination are asymmetric, i.e. the packets are routed on different paths to and from the target, the round-trip times may not reflect the actual time it takes for a packet to reach the destination. The resulting round trip subsequently show misleading values. The actual path may in fact be much shorter or longer than the round-trip time indicates, depending on the situation. In figure 5 such a scenario is shown, with the corresponding times highlighted in the output. If the return path would jump from the longer path to the shorter, the RTTs measured by traceroute would even become shorter, i.e. the output would show a negative increase in the TTL for the last link.

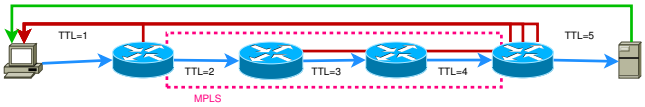



```
host1.site:/ # traceroute server1.site
traceroute to server1.site (192.168.0.6), 30 hops max, 40 byte
packets using UDP
1 router1.site (192.168.0.2) 1.823 ms 0.984 ms 0.872 ms
2 router2.site (192.168.0.3) 15.784 ms 17.188 ms 17.525 ms
2 router3.site (192.168.0.4) 29.658 ms 31.165 ms 34.235 ms
4 server1.site (192.168.0.5) 32.574 ms 33.984 ms 33.654 ms
```

Figure 5: Asymmetric path example

This anomaly is especially problematic, since it may lead to wrong conclusions related to congestion. A sudden and overly large increase in the RTT is usually a very accurate sign for congestion on that link or hop. In this case, however, it may simply be a result of returning packets taking a different route. As this is not visible in the output, it may lead to the wrong conclusion that a link or hop is congested.

A similar result occurs on MPLS links, where the response packet has to travel to the end of the MPLS path, until it is returned to the sender of the probe. Since pure MPLS routers only know about the next hop of a packet, they can't send ICMP errors back right away. Instead, they have to use the path where the original packet would have gone. The result of this is, that all packets are travelling to the last MPLS hop first. Therefore the round-trip times shown in traceroute for the hops in the MPLS path all reflect roughly the round-trip time for the last MPLS router. An example for this anomaly is shown in figure 6. The characteristic signs for this anomaly are the almost equivalent round-trip times for multiple hops in the traceroute output, highlighted below.




```
host1.site:/ # traceroute server1.site
traceroute to server1.site (192.168.0.8), 30 hops max, 40 byte
packets using UDP
1 router1.site (192.168.0.2) 2.644 ms 1.945 ms 0.671 ms
2 router2.site (192.168.0.3) 18.856 ms 17.654 ms 17.897 ms
3 router3.site (192.168.0.4) 18.872 ms 17.984 ms 17.978 ms
4 router4.site (192.168.0.5) 18.723 ms 17.352 ms 17.872 ms
5 server1.site (192.168.0.6) 19.874 ms 19.635 ms 20.698 ms
```

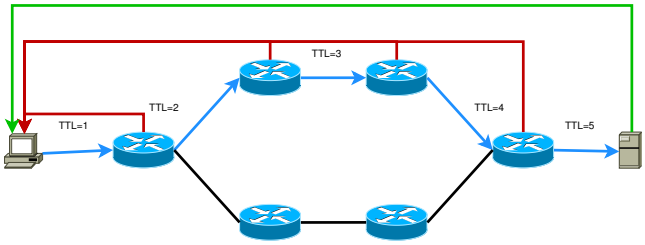
Figure 6: MPLS path example

doi: 10.2313/NET-2012-08-1_02

This can also lead to the wrong conclusion, that a link or hop is congested, for the same reasons as above. In case there actually is congestion on the MPLS-routed path, this anomaly additionally obscures the link or hop which is congested. Since the RTTs reflect the time it takes to reach the last MPLS router, it may be any router in the MPLS path that is congested. However, the output would suggest, that it is the first router or link where the congestion issue is located, if any.

3.4 Missing Links

This anomaly means that the traceroute output is missing links, which are present in the actual topology. The usual reason for this is load balancing, in this case, when all packets are routed on a single path. Figure 7 shows an example of this anomaly. The other link should appear at the two highlighted lines in the output.




```
host1.site:/ # traceroute server1.site
traceroute to server1.site (192.168.0.8), 30 hops max, 40 byte
packets using UDP
1 router1.site (192.168.0.2) 0.133 ms 0.059 ms 0.096 ms
2 router2.site (192.168.0.3) 16.281 ms 15.936 ms 16.319 ms
3 router3.site (192.168.0.4) 17.380 ms 17.811 ms 17.431 ms
4 router6.site (192.168.0.7) 17.579 ms 17.972 ms 18.905 ms
5 server1.site (192.168.0.8) 18.201 ms 17.847 ms 18.269 ms
```

Figure 7: Missing links example

This anomaly is quite problematic, in the sense that it obscures the actual topology. This is a concern, if the network topology itself is to be scanned, as well as if an error is to be diagnosed on the missing link. In the latter case, an error wouldn't show up on the traceroute output, even when it may actually have a great impact on the network.

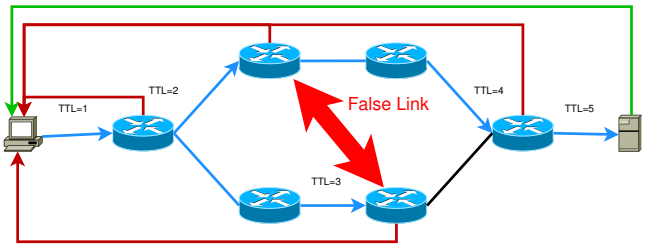
3.5 False Links

In this case, traceroute implies a false link between hops. It usually occurs in load-balanced links, when some packets are routed via one path and some are routed on another path. An example of this can be seen in figure 8. The false link shows up at the two lines highlighted in the output.

This anomaly is actually a huge problem modern networks, especially since it is not obvious to users without knowledge of the actual topology. Hence, it may lead people to wrong conclusions about the network or a problem to be solved.

3.6 Loops and Circles

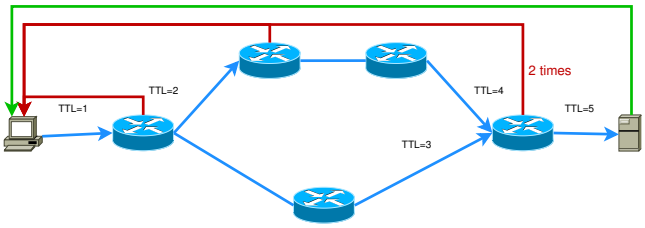
This is one of the more complex anomalies, where some hops are missing and other hops are shown multiple times, i.e. the packets seem to travel in loops or circles. The most common case for loops is when load balancing is used for paths of unequal length. Another example may be MPLS links, if




```
host1.site:/ # traceroute server1.site
traceroute to server1.site (192.168.0.8), 30 hops max, 40 byte
packets using UDP
1 router1.site (192.168.0.2) 2.380 ms 1.357 ms 0.313 ms
2 router2.site (192.168.0.3) 16.567 ms 17.862 ms 17.334 ms
3 router5.site (192.168.0.6) 17.837 ms 17.480 ms 17.350 ms
4 router6.site (192.168.0.7) 17.957 ms 17.804 ms 19.717 ms
5 server1.site (192.168.0.8) 18.665 ms 18.287 ms 19.281 ms
```

Figure 8: False links example

the address of the last MPLS router is used for ICMP errors, e.g. when intermediate routers lack an IP address. A rarer example is, when packets with a TTL of zero are forwarded to the next hop, e.g. by a faulty router. Cycles usually occur only on load-balanced links, where the difference in length is greater than one. An example message flow is depicted in figure 9. The two lines highlighted below show the hop which is probed twice. However, the corresponding output may also be justified, in case there is an actual forwarding loop or cycle.




```
host1.site:/ # traceroute server1.site
traceroute to server1.site (192.168.0.8), 30 hops max, 40 byte
packets using UDP
1 router1.site (192.168.0.2) 2.644 ms 1.945 ms 0.671 ms
2 router2.site (192.168.0.3) 16.324 ms 17.547 ms 17.640 ms
3 router5.site (192.168.0.6) 17.235 ms 17.724 ms 17.365 ms
4 router5.site (192.168.0.6) 18.132 ms 17.971 ms 18.590 ms
5 server1.site (192.168.0.8) 19.874 ms 19.635 ms 20.698 ms
```

Figure 9: Loops example

This anomaly is normally quite obvious, but still a serious problem. Since some links are missing, the actual topology is yet again obscured. To an unsuspecting user, it may even seem, that packets are actually moving in loops or circles. This is especially the case, if the missing destination anomaly above occurs in conjunction with this anomaly. In that case, it may seem like a valid network problem, when it is only an unfortunate combination of different anomalies.

3.7 Diamonds

Diamonds belong to the most complex anomalies, where some additional links are shown, while others are missing. This anomaly only occurs, when sending out multiple probes for one hop. It is usually caused by load balancing, when

doi: 10.2313/NET-2012-08-1_02

some probes are forwarded on one path and some on other paths. This leads to a complete chaos in the traceroute output, as seen in figure 10. In this case, instead of the textual output, the resulting links which would be inferred by traceroute are shown in figure 11, for brevity.

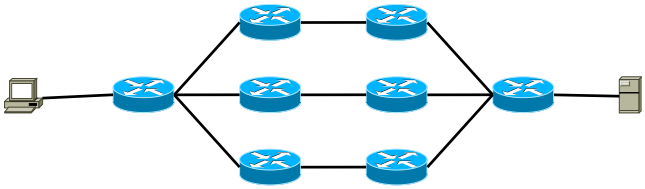


Figure 10: Diamond example

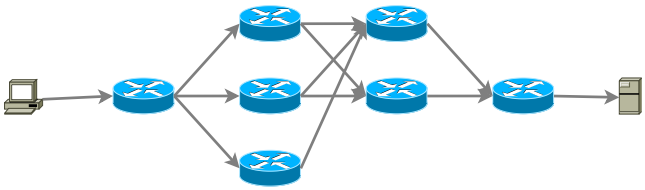


Figure 11: Diamond results

This is yet another case, when load-balanced links cause the traceroute output to be completely useless, even if the correct topology may be known. It can be seen as a combination of the anomalies regarding missing and false links, as well as loops and circles.

4. SOLUTIONS

The following are several solutions for the various anomalies presented above. These solutions can, of course, only limit the impact of said anomalies most of the time. A summary to the anomalies and their respective possible solutions is shown in table 1 further below.

4.1 Paris Traceroute

Paris traceroute was developed to correct most of the deficiencies found in classic traceroute, especially in regard to load-balanced networks. The distinguishing feature of Paris traceroute is, that it tries to actively influence routing decisions in per-flow load-balanced links. It does this by carefully setting header fields in the sent probe packets, which are taken into account by per-flow load balancing [1]. The respective header fields are depicted in tables 2, 3, 4 and 5. The fields used for per-flow load balancing and thus set by Paris traceroute are underlined. The fields used by traceroute to match replies to sent probes are double underlined. A special case is the identifier field in the UDP header, which is specifically modified to produce the desired checksum.

Version	IHL	*TOS*	Total Length	
Identification			Flags	Fragment Offset
TTL	Protocol		Header Checksum	
Source Address				
Destination Address				
Options and Padding				

Table 2: IP Header fields used by Paris traceroute

Source Port	Destination Port
Length	Checksum

Table 3: UDP Header fields used by Paris traceroute

Type	Code	Checksum
Identifier		Sequence Number

Table 4: ICMP Header fields used by Paris traceroute

Source Port	Destination Port
Sequence Number	
...	

Table 5: TCP Header fields used by Paris traceroute

By keeping the necessary fields constant, Paris traceroute is able to scan a single path. To scan all paths, the fields are intentionally varied and several scans are conducted to, hopefully, traverse all possible links. Thus, it is able to accurately scan single paths, as well as all load-balanced paths to a destination in case of per-flow load balancing. Per-packet load balancing may only be detected by current traceroute versions, due to the randomness of the packet's distribution. Future versions are supposed to include statistical algorithms to accurately distinguish per-packet load-balanced links, too [10]. Paris traceroute additionally includes support for limited control over the return path by influencing the flow information of returned ICMP error packets [2].

4.2 Traceroute Extensions

The following is a list of important traceroute extensions related to the anomalies examined above. Most of them can be found in all modern traceroute variants by now.

4.2.1 UDP and TCP probes

Modern variants of traceroute also support sending of UDP or TCP probes, instead of ICMP echo requests, as described before. Since most routers and firewalls block ICMP echo requests, most modern traceroute implementations in fact use UDP by default. Another advantage of UDP probes is, that they don't require root privileges for sending probes on Linux systems. TCP probes are normally only used in very special cases, usually either to circumvent very restrictive firewalls or to traverse NAT gateways. The main reason against TCP is that it tries to create a connection which subsequently introduces state into the network. Additionally, an application listening on TCP is more likely than for UDP. In fact, to more easily traverse firewalls, most implementations use TCP port 80 as default. To clear up pending connections an additional TCP RST packet is then required. All in all, by using either UDP or TCP instead of ICMP echo requests, missing hops or missing destination anomalies may be somewhat mitigated.

4.2.2 AS-number lookup

This feature makes it possible to automatically query AS-numbers from databases, e.g. the RIPE database, for IP

doi: 10.2313/NET-2012-08-1_02

Anomaly	Solutions	Comments
Missing Hops	none	usually impossible to solve from the user's end
Missing Destination	UDP/TCP probes	some hosts also block UDP/TCP probes
False Round-Trip Times	(Reverse traceroute), MPLS Label-decoding (if caused by MPLS)	helps for a more accurate interpretation of the results
Missing Links	Paris Traceroute	only partially helps for per-packet load balancing
False Links	Paris Traceroute	only partially helps for per-packet load balancing
Loops and Cycles	Paris Traceroute, MPLS Label-decoding (if caused by MPLS)	only partially helps for per-packet load balancing
Diamonds	Paris Traceroute	only partially helps for per-packet load balancing

Table 1: Summary of solutions to the respective traceroute anomalies

addresses encountered by traceroute. It is especially useful to identify network operators, as well as to detect network boundaries. This information may subsequently used to contact administrators, in case of network failure. There is also a modern algorithm, which combines BGP information with information from several databases to produce even more accurate results [6].

4.2.3 Path-MTU discovery

Path-MTU discovery in traceroute enables users to identify the MTU until each hop. This can ease the identification of "MTU-bottlenecks", i.e. links where the MTU suddenly drops. It may also help to identify the ideal default MTU to set for outgoing packets, in case automatic detection yields unsatisfactory results.

4.2.4 MPLS-label decoding

This is used to decode MPLS labels, i.e. FECs, returned in extended ICMP error packets, as defined in RFC 4950 [3]. It makes diagnosing MPLS related problems much easier and additionally allows for a more accurate interpretation of the traceroute output. This is especially useful if MPLS-related anomalies, like the one causing false round-trip times or loops resulting from MPLS routing, are suspected.

4.2.5 Reverse traceroute

Reverse traceroute techniques are used to track the path a packet takes from a remote source to the local host. By inspecting the packet return path, additional network problems may be diagnosed and the interpretation of existing output may be eased. This is especially of interest concerning the anomalies related to asymmetric paths.

There is a proposal which would actually achieve this without interaction from the target system [5]. However, it requires the use of the *record-route* or *RR* IP option, which records the hops traversed by the packet. This is done to record the return path of each packet. The RR option was invented to be an alternative to traceroute, but since it requires interaction by the respective routers, it is often not supported and was abandoned. It is also only able to record 8 hops in both directions, which is why the proposal requires multiple hosts, so-called "vantage points", between the target and the local host. The proposal is therefore not feasible for users without the necessary resources. Finally, it is necessary to spoof the source IP address in sent probes to redirect the responses to said hosts, which is prevented by most routers.

5. CONCLUSION

Some of the described anomalies have little to no impact on network analysis and diagnosis, while others pose a huge problem. Paris traceroute solves or at least limits the impact of traceroute anomalies in load balancing contexts. Several other traceroute extensions also contribute to counteracting several problems for traceroute found in modern networks. There is also quite some research on this topic, to further improve the situation. Especially reverse traceroute is a promising candidate to solve at least some of the remaining anomalies.

6. REFERENCES

[1] 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.

[2] 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.

[3] R. Bonica, D. Gan, D. Tappan, and C. Pignataro. ICMP Extensions for Multiprotocol Label Switching. RFC 4950 (Proposed Standard), August 2007.

[4] V. Jacobson. Original traceroute announcement, 1988.

[5] E. Katz-Bassett, H. V. Madhyastha, V. K. Adhikari, C. Scott, J. Sherry, P. Van Wesep, T. Anderson, and A. Krishnamurthy. Reverse traceroute. In *Proceedings of the 7th USENIX conference on Networked systems design and implementation*, NSDI'10, pages 15–15, Berkeley, CA, USA, 2010. USENIX Association.

[6] Z. M. Mao, J. Rexford, J. Wang, and R. H. Katz. Towards an accurate AS-level traceroute tool. In *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '03, pages 365–378, New York, NY, USA, 2003. ACM.

[7] E. Rosen, A. Viswanathan, and R. Callon. Multiprotocol Label Switching Architecture. RFC 3031 (Proposed Standard), January 2001.

[8] N. Spring, R. Mahajan, D. Wetherall, and T. Anderson. Measuring ISP topologies with rocketfuel. *IEEE/ACM Trans. Netw.*, 12(1):2–16, Feb. 2004.

[9] R. Steenbergen. A Practical Guide to (Correctly) Troubleshooting with Traceroute. NANOG 47, 2009.

[10] F. Viger, B. Augustin, X. Cuvellier, C. Magnien, M. Latapy, T. Friedman, and R. Teixeira. Detection, understanding, and prevention of traceroute measurement artifacts. *Comput. Netw.*, 52(5):998–1018, Apr. 2008.