

MaxCDN Support: Tutorial

Reading a Traceroute

Ivan Dabic

March 10, 2015

Updated
March 10, 2015

When you run a traceroute, it sends a test packet (in turns – cycles) toward the destination in order to “scan” the route between the source and the destination. This process can be used to debug problems that are network-related.

Step by step, a traceroute looks like this:

1

Source sends a test packet to destination with TTL = 1.

2

Every hop (router) between the source and the destination is decrementing this value by 1 until it reaches 0.

3

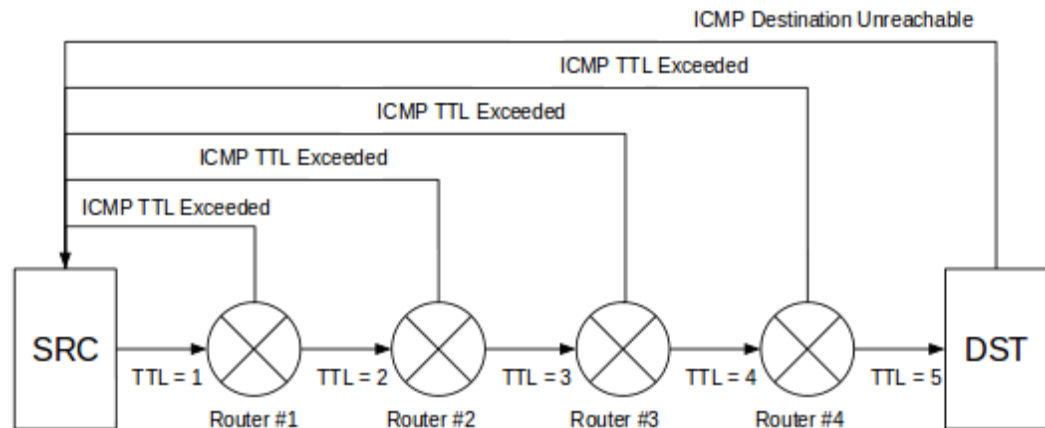
When 0 is reached, router is dropping packet and returning the packet back to source that originally sent the packet.

4

Source is receiving this packet and displaying the “hop” information with Round Trip Time (RTT).

5

This process repeats until the packet reaches the destination and it responds back, closing the cycle.



During the debugging process, you expect to get some response from the destination point. With that in mind, you should be aware that many traceroute implementations are using UDP protocol in the test process.

UDP doesn't require the end point (destination) to respond back to the source, thereby making the debugging process somewhat vague without a "Destination Unreachable" response. The port that traceroute uses is 33434.

Another implementation of traceroute uses the ICMP echo request test that suffers from the same gap as the UDP does on 33434. It can't detect the destination if it doesn't respond with "Destination Unreachable."

Many newer traceroute implementations are able to use ALL protocols, giving you more flexibility during the debugging process. For instance, using TCP requires the server to respond to requests and lets you know for sure if the server is reachable or not.

Example:

```
~$ traceroute google.com
traceroute to google.com (216.58.208.110), 30 hops
max, 60 byte packets
 1  192.168.1.1 (192.168.1.1)  2.371 ms  2.323 ms
```

3	TRUNCATED			
4	90-0.static.ikomline.net (95.180.0.9)	36.218 ms	36.901 ms	
5	peer-AS31042.sbb.rs (82.117.192.77)	44.300 ms	44.190 ms	
6	bg-yb-m-1-vl99.sbb.rs (89.216.5.66)	35.942 ms		
		18.134 ms	25.657 ms	
7	FREE.sbb.rs (89.216.6.17)	21.607 ms	28.993 ms	
		29.120 ms		
8	peer-AS15169.sbb.rs (82.117.192.2)	42.308 ms		
		39.643 ms	40.327 ms	
9	216.239.51.223 (216.239.51.223)	41.979 ms		
		41.943 ms	39.870 ms	
10	sof01s11-in-f14.1e100.net (216.58.208.110)			
		37.624 ms	38.717 ms	38.690 ms

Traceroute results are shown and divided into five columns:

- 1

Hop number
- 2

Hostname or IP address of router
- 3

RTT 1
- 4

RTT 2
- 5

RTT 3

On another note, many are often confused by asterisks in traceroute results. This is mainly because asterisks can be treated differently depending on the scenario. Because of this, traceroute results must be interpreted globally and wholly rather than in piecemeal reading each line independently.

```
8 * * *
```

In this instance, the host in question may be unreachable. But, depending on its position in a trace, it can mean something completely different. For example, it may mean that ICMP echo packets are being blocked by the hop in question.

The 3 most common latency interpretation scenarios include:

- Latency high but consistent
- Latency existing on only one part of the route (isolated group)
- Latency accumulates throughout the route

Scenario #1: Latency High But Consistent

First, let's define what high latency is.

When testing the edge servers of a global network within a single continent, a round trip time (RTT) of 100ms indicates a potential problem. But for the same number of hops connecting Europe to the United States, that number is fine, even typical.

RTT is nothing else but a measure of how long it takes for hops to receive a traceroute test packet and respond back. Taking intercontinental routing into consideration, 100ms is safe to ignore on a first hop that connects two continents. 100ms can also be treated as the highest number to tolerate latency for. Although, as you can see, it can potentially come down to arbitrary opinion based on a particular traceroute test result.

If the latency is consistently high throughout the route, it's possible there is no problem. In this case, your local network may be appending latency time, or the RTT is simply justifiable based on the protocol you've chosen to run this traceroute test with. If you weren't already using TCP the first time around, it's recommended that you do so.

```
~$ traceroute google.com
traceroute to google.com (216.58.208.110), 30 hops
max, 60 byte packets
 1  192.168.1.1 (192.168.1.1)  2.371 ms  2.323 ms
   3.586 ms
 2  TRUNCATED
 3  TRUNCATED
 4  90-0.static.ikomline.net (95.180.0.90)  96.238 ms
   96.218 ms  96.901 ms
 5  peer-AS31042.sbb.rs (82.117.192.77)  94.318 ms
   94.300 ms  94.190 ms
 6  bg-yb-m-1-vl99.sbb.rs (89.216.5.66)  95.942 ms
   98.134 ms  95.657 ms
 7  FREE.sbb.rs (89.216.6.17)  91.607 ms  98.993 ms
   99.120 ms
 8  peer-AS15169.sbb.rs (82.117.192.2)  92.308 ms
   99.643 ms  90.327 ms
 9  216.239.51.223 (216.239.51.223)  91.979 ms
   91.943 ms  99.870 ms
10  sof01s11-in-f14.1e100.net (216.58.208.110)
   97.624 ms  98.717 ms  98.690 ms
```

NOTE: Hops 4 through 10 indicate approximately the same latency. Don't mistake this route as faulty!

Scenario #2: Latency Exists Only on One Part of Route (Isolated Group)

This most likely means you are using ICMP echo request protocol and the hops in question are lowering the priority of ICMP packets. The hops are doing this in order to give the highest throughput for actual traffic and TCP requests, thereby causing the traceroute to show a hop or isolated group of hosts to have high RTT's.

```
~$ traceroute google.com
traceroute to google.com (216.58.208.110), 30 hops
```

```

max, 60 byte packets
 1  192.168.1.1 (192.168.1.1)  2.371 ms  2.323 ms
3.586 ms
 2  TRUNCATED
 3  TRUNCATED
 4  90-0.static.ikomline.net (95.180.0.90)  36.238 ms
36.218 ms  36.901 ms
 5  peer-AS31042.sbb.rs (82.117.192.77)  44.318 ms
44.300 ms  44.190 ms
 6  bg-yb-m-1-vl99.sbb.rs (89.216.5.66)  235.942 ms
218.134 ms  225.657 ms
 7  FREE.sbb.rs (89.216.6.17)  221.607 ms  228.993 ms
229.120 ms
 8  peer-AS15169.sbb.rs (82.117.192.2)  242.308 ms
239.643 ms  240.327 ms
 9  216.239.51.223 (216.239.51.223)  41.979 ms
41.943 ms  39.870 ms
10  sof01s11-in-f14.1e100.net (216.58.208.110)
37.624 ms  38.717 ms  38.690 ms

```

NOTE: Hops 6 through 8 are isolated groups. Most likely, ICMP packets are downgraded in priority within these hops.

Scenario #3: Latency Accumulates Throughout Route

Of the main types of scenarios, accumulative latency most likely indicates an actual problem. The hops where latency starts to accumulate is the source of the problem and needs to be addressed properly.

```

~$ traceroute google.com
traceroute to google.com (216.58.208.110), 30 hops
max, 60 byte packets
 1  192.168.1.1 (192.168.1.1)  2.371 ms  2.323 ms
3.586 ms
 2  TRUNCATED

```

```
3  TRUNCATED
4  90-0.static.ikomline.net (95.180.0.90)  36.238 ms
36.218 ms  36.901 ms
5  peer-AS31042.sbb.rs (82.117.192.77)  44.318 ms
44.300 ms  44.190 ms
6  bg-yb-m-1-vl99.sbb.rs (89.216.5.66)  135.942 ms
118.134 ms  125.657 ms
7  FREE.sbb.rs (89.216.6.17)  121.607 ms  128.993 ms
129.120 ms
8  peer-AS15169.sbb.rs (82.117.192.2)  142.308 ms
139.643 ms  140.327 ms
9  216.239.51.223 (216.239.51.223)  141.979 ms
141.943 ms  139.870 ms
10 sof01s11-in-f14.1e100.net (216.58.208.110)
137.624 ms  138.717 ms  138.690 ms
```

NOTE: Hop 4 is where the issue started!

How to Read Asterisks

Traceroute sends the “probe” in cycles of three and requires all three to return RTT values. If you get three asterisks, try testing the same destination with TCP as you are most likely using UDP/ICMP.

In general, all three asterisks can mean a disaster or simple ICMP echo packets block. As previously mentioned, you can confirm what the case truly is by switching to TCP traceroute. If TCP returns the same result, chances are the problem lies in the last hop you saw RTT and other details for. you should see all (default) 30 hops traceroute tried to use (all asterisks).

If so, you’ve found the point that has broken the route. Now it must be addressed accordingly.

```
~$ traceroute google.com
```

```

traceroute to google.com (216.58.208.110), 30 hops
max, 60 byte packets
 1  192.168.1.1 (192.168.1.1)  2.371 ms  2.323 ms
   3.586 ms
 2  TRUNCATED
 3  TRUNCATED
 4  90-0.static.ikomline.net (95.180.0.90)  36.238 ms
   36.218 ms  36.901 ms
 5  peer-AS31042.sbb.rs (82.117.192.77)  44.318 ms
   44.300 ms  44.190 ms
 6  bg-yb-m-1-vl99.sbb.rs (89.216.5.66)  35.942 ms
   18.134 ms  25.657 ms
 7  FREE.sbb.rs (89.216.6.17)  21.607 ms  28.993 ms
   29.120 ms
 8  peer-AS15169.sbb.rs (82.117.192.2)  42.308 ms
   39.643 ms  40.327 ms
 9  216.239.51.223 (216.239.51.223)  41.979 ms
   41.943 ms  39.870 ms
10  * * *
...
30  * * *

```

NOTE: Assuming this is the TCP traceroute test result, you have found a problem on hop 10! If not TCP, it's probably safe to ignore it and re-test via TCP.

On the other hand, below is the example of traceroute that is probably clean, just showing ICMP echo block before the end of the route:

```

~$ traceroute google.com
traceroute to google.com (216.58.208.110), 30 hops
max, 60 byte packets
 1  192.168.1.1 (192.168.1.1)  2.371 ms  2.323 ms
   3.586 ms
 2  TRUNCATED

```

```

3  TRUNCATED
4  90-0.static.ikomline.net (95.180.0.90)  36.238 ms
36.218 ms  36.901 ms
5  peer-AS31042.sbb.rs (82.117.192.77)  44.318 ms
44.300 ms  44.190 ms
6  bg-yb-m-1-vl99.sbb.rs (89.216.5.66)  35.942 ms
18.134 ms  25.657 ms
7  FREE.sbb.rs (89.216.6.17)  21.607 ms  28.993 ms
29.120 ms
8  * * *
9  * * *
10 sof01s11-in-f14.1e100.net (216.58.208.110)
37.624 ms  38.717 ms  38.690 ms

```

NOTE: Hops 8 and 9 shouldn't be seen as an issue. It's just a ICMP echo block!

Lacking one or two asterisks indicates there is a potential problem with latency that will lead to packet loss. With one or two asterisks, we are facing timeout for one or two “probe packets” that traceroute sent from the source. If there is a timeout, you should consider this hop as a potential problem.

```

~$ traceroute google.com
traceroute to google.com (216.58.208.110), 30 hops
max, 60 byte packets
1  192.168.1.1 (192.168.1.1)  2.371 ms  2.323 ms
3.586 ms
2  TRUNCATED
3  TRUNCATED
4  90-0.static.ikomline.net (95.180.0.90)  36.238 ms
36.218 ms  36.901 ms
5  peer-AS31042.sbb.rs (82.117.192.77)  44.318 ms
44.300 ms  44.190 ms
6  bg-yb-m-1-vl99.sbb.rs (89.216.5.66)  35.942 ms
18.134 ms  25.657 ms

```

```

 7  FREE.sbb.rs (89.216.6.17)  21.607 ms  28.993 ms
29.120 ms
 8  peer-AS15169.sbb.rs (82.117.192.2)  42.308 ms
39.643 ms  40.327 ms
 9  216.239.51.223 (216.239.51.223)  *  41.943 ms
39.870 ms
10  sof01s11-in-f14.1e100.net (216.58.208.110)
37.624 ms  38.717 ms  38.690 ms

```

```

~$ traceroute google.com
traceroute to google.com (216.58.208.110), 30 hops
max, 60 byte packets
 1  192.168.1.1 (192.168.1.1)  2.371 ms  2.323 ms
3.586 ms
 2  TRUNCATED
 3  TRUNCATED
 4  90-0.static.ikomline.net (95.180.0.90)  36.238 ms
36.218 ms  36.901 ms
 5  peer-AS31042.sbb.rs (82.117.192.77)  44.318 ms
44.300 ms  44.190 ms
 6  bg-yb-m-1-vl99.sbb.rs (89.216.5.66)  35.942 ms
18.134 ms  25.657 ms
 7  FREE.sbb.rs (89.216.6.17)  21.607 ms  28.993 ms
29.120 ms
 8  peer-AS15169.sbb.rs (82.117.192.2)  42.308 ms
39.643 ms  40.327 ms
 9  216.239.51.223 (216.239.51.223)  *  *  39.870 ms
10  sof01s11-in-f14.1e100.net (216.58.208.110)
37.624 ms  38.717 ms  38.690 ms

```

Aside from what's already been mentioned, using common sense and taking geographics into consideration is key when trying to determine if the traceroute response makes sense.

The following traceroute shows a bigger problem than what appears:

```
~$ traceroute google.com
traceroute to google.com (216.58.208.110), 30 hops
max, 60 byte packets
 1  192.168.1.1 (192.168.1.1)  2.371 ms  2.323 ms
   3.586 ms
 2  TRUNCATED
 3  TRUNCATED
 4  90-0.static.ikomline.net (95.180.0.90)  36.238 ms
   36.218 ms  36.901 ms
 5  peer-AS31042.sbb.rs (82.117.192.77)  44.318 ms
   44.300 ms  44.190 ms
 6  bg-yb-m-1-vl99.sbb.rs (89.216.5.66)  35.942 ms
   18.134 ms  25.657 ms
 7  ae0.bbr02.eq01.chi01.networklayer.com
   (173.192.18.130)  52.455 ms  52.196 ms  52.458 ms
 8  peer-AS15169.sbb.rs (82.117.192.2)  42.308 ms
   39.643 ms  40.327 ms
 9  216.239.51.223 (216.239.51.223)  *  *  139.870 ms
10  sof01s11-in-f14.1e100.net (216.58.208.110)
   37.624 ms  38.717 ms  38.690 ms
```

On hop 7, we can see an obvious jump from Europe to the United States and back to Europe on hop 8. In practice, this indicates that the ISP that forced this route (sbb.rs) found it to be the best possible route at the given moment. ISP must address this issue accordingly.

Topics

Debugging

Related Articles

Performing Traceroute on Mac and Windows

Solving Mixed Content Problems

SSL Debugging

403 Forbidden

Comments

Community

1

Login

Recommend

3

Tweet

Share

Sort by Best

Start the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS ?

Name

Be the first to comment.

ALSO ON MAXCDN ONE

Allowing Blank Referers for Specific ...

1 comment • 4 years ago

Toing — This is really good and working for me.

Pseudo Streaming with nginx

3 comments • 4 years ago

Sergey — Where to download nginx mp4 module with support

What is SSL?

1 comment • 4 years ago

sadanand — what is ssl, you can understand here.Get

What is Page Load Time?

1 comment • 3 years ago

Riaz Khan — http://www.pkshop.pk

Subscribe

Add Disqus to your siteAdd DisqusAdd

Disqus' Privacy PolicyPrivacy PolicyPrivacy

