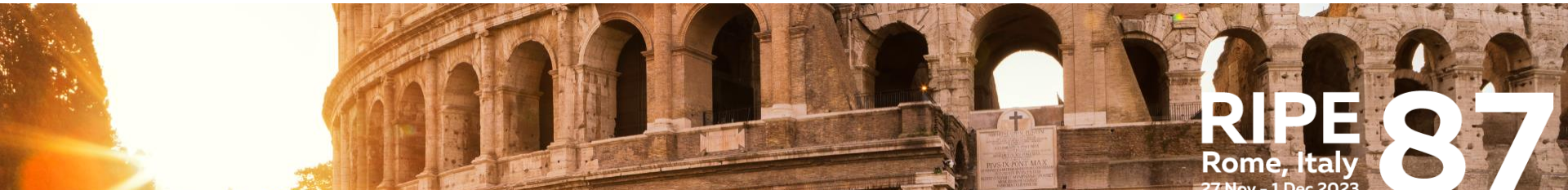
"Where the !?*! are the packets going?"

36 years later… 1987 to RIPE 87

MAT WG – Measurements and Tools

Luca Sani

Senior R&D Software Engineer @ Catchpoint



Whoami?

Luca Sani

Senior R&D Software Engineer at Catchpoint

Università di Pisa / University of Pisa

Fun Fact

Luca & live in Lucca.

Traceroute

Traceroute is one of the most famous and long-lasting diagnostic tools in networking environment

First implementation by Van Jacobson in late 80s to answer the question:

"where the !?*! are the packets going" ?

```
Posted-Date: Tue, 20 Dec 88 05:13:28 PST
Received-Date: Tue, 20 Dec 88 05:14:46 PST
Received: from helios.ee.lbl.gov by venera.isi.edu (5.54/5.51)
        id AA25560; Tue, 20 Dec 88 05:14:46 PST
Received: by helios.ee.lbl.gov (5.59/s2.2)
        id AA03127; Tue, 20 Dec 88 05:13:30 PST
Message-Id: <8812201313.AA03127@helios.ee.lbl.gov>
To: ietf@venera.isi.edu, end2end-interest@venera.isi.edu
Subject: 4BSD routing diagnostic tool available for ftp
Date: Tue, 20 Dec 88 05:13:28 PST
From: Van Jacobson <van@helios.ee.lbl.gov>
Content-Length: 2373
X-Lines: 46
Status: RO

After a frustrating week of trying to figure out "where the !?*!
are the packets going?", I cobbled up a program to trace out
the route to a host.  It works by sending a udp packet with a
ttl of one & listening for an icmp "time exceeded" message.  If
it gets one, it prints the source address from the icmp message,
then bumps the ttl by one & etc.  (As usual, I didn't come up
with this clever idea -- I heard Steve Deering mention it at an
end-to-end task force meeting.)
```

Traceroute implementations

- Many traceroute implementations have been created on different OSes
- Over the years it became one of the most used tools in the Internet measurement and topology discovery fields (multipath, de-aliasing, NAT traversal, …)
  - Paris, Dublin, Pamplona traceroute…

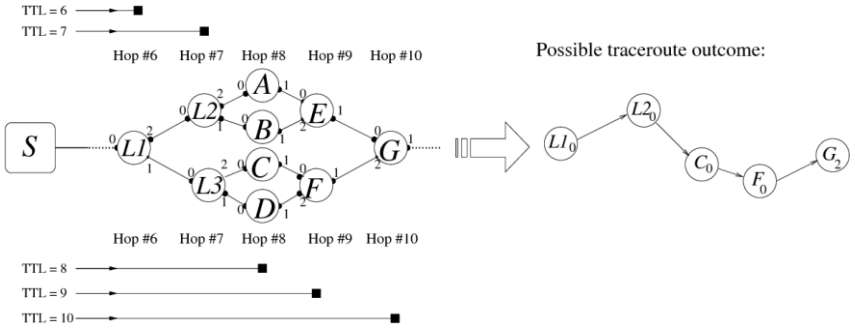


Fig. 1. Traceroute under load balancing

Augustin, Brice, Timur Friedman, and Renata Teixeira. "Multipath tracing with Paris traceroute." Workshop on End-to-End Monitoring Techniques and Services. IEEE, 2007.

Linux traceroute

- We leverage Dmitry Butskoy's "Linux traceroute"
  - Very fast
  - Open source
  - Easily extendible

- During the years we enhanced this traceroute to include new monitor capabilities
- We hope these enhancements can be useful to the community

Pietrasanta Traceroute

"A noble town since 1841 and a city of art"
(and where our Italian office is located!)

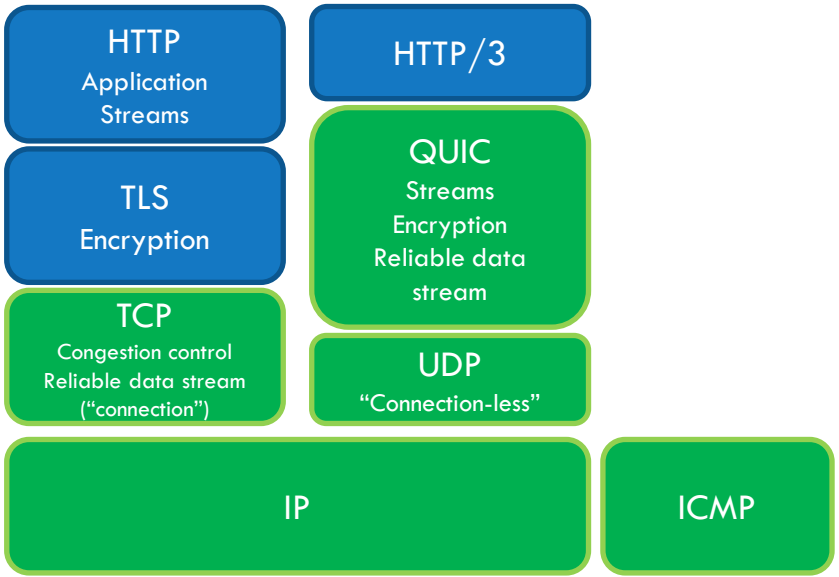
Pietrasanta Traceroute

- QUIC traceroute
- TCP "InSession"
- Work in Azure environment
- … and many more

QUIC support

QUIC

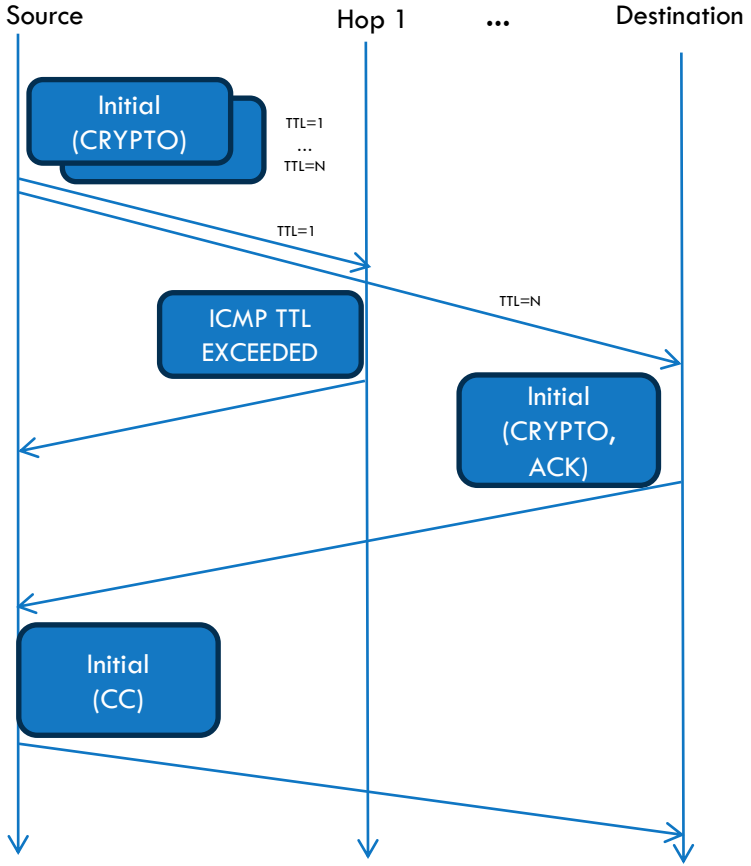
- QUIC is considered a transport layer protocol
  - More than just “UDP”
  - e.g., it is the transport layer of HTTP/3



QUIC assumes responsibility for the confidentiality and integrity protection of packets. For this it uses keys derived from a TLS handshake, but instead of carrying TLS records over QUIC (as with TCP), TLS handshake and alert messages are carried directly over the QUIC transport, which takes over the responsibilities of the TLS record layer.

RFC9001 - Using TLS to Secure QUIC

QUIC support



- Packets sent are QUIC compliant, so the header is protected and the payload (frames) are encrypted
  - We leverage openssl3
- Nice “side effects”
  - Check whether the path filters QUIC
  - Determine if the destination supports QUIC
  - Check whether ECN is supported
    - Set IP-ECN in probes

QUIC traceroute

- Like "TCP half open"
- Do a QUIC handshake then closes the session (if opened)
  - Send QUIC "Initial" packet
    - Include a CRYPTO frame with TLS "Server Hello"
  - Intermediate hops will return ICMP TTL Exceeded
  - Destination may return
    - QUIC packet
    - ICMP port unreachable (still good, dest reached)
    - Nothing (timedout)
  - Close the session if it is the case
    - Send QUIC Initial packet including a CONNECTION_CLOSE frame

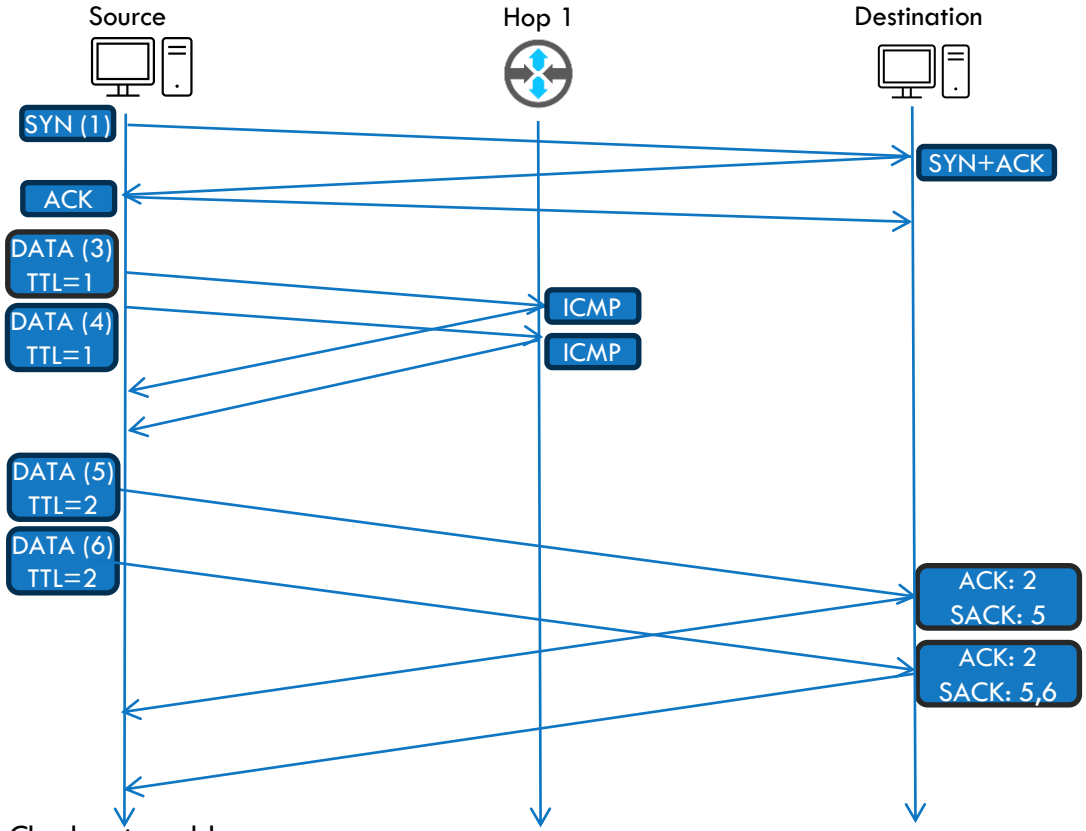
TCP InSession

TCP "InSession"

- Classic TCP traceroute sends a different SYN for each hop
  - Different SYNs can take different paths
    - No consistency within a single traceroute
  - Many SYNs are sent per traceroute
    - Trigger firewall rules (SYN flood?)

- TCP InSession firstly opens a TCP session with the destination
- Then tracerouting is performed by sending 1-byte data packets within the session (with incremental TTL)
  - Inspired by TCP Sidecar

TCP "InSession"

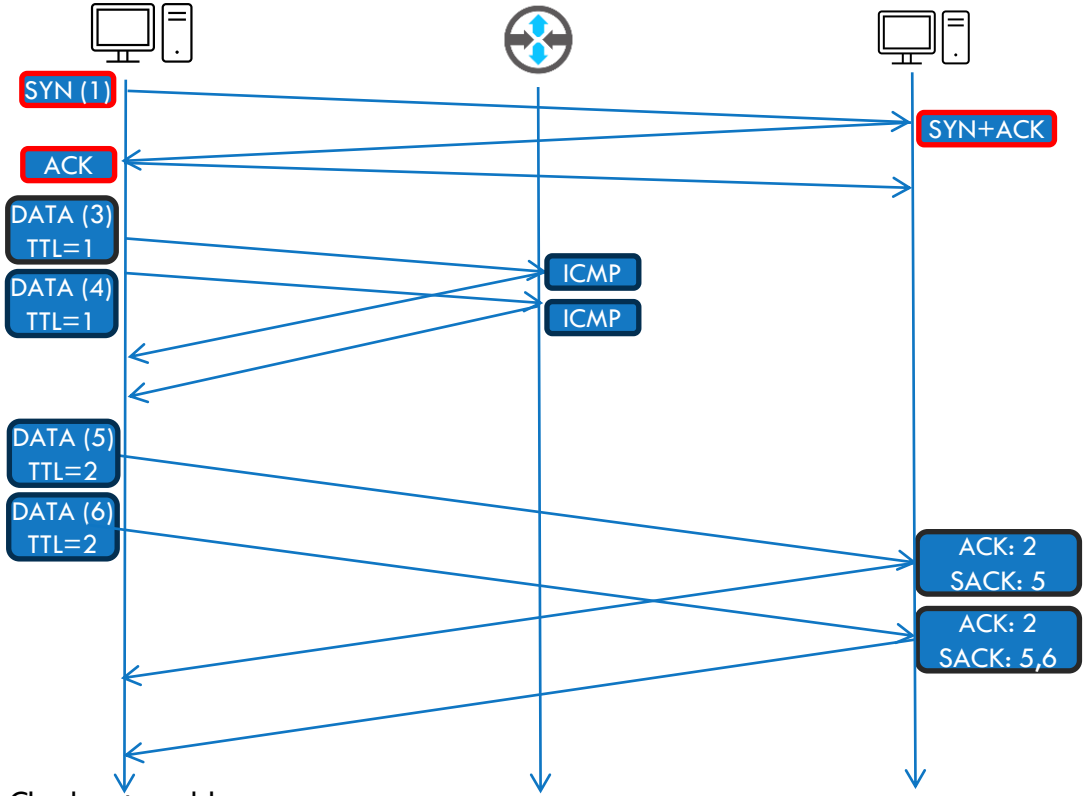


- Two hops
- Two probes per hop sent in parallel

Check out our blog:

https://www.catchpoint.com/blog/traceroute-insession-catchpoints-effort-towards-a-more-reliable-network-diagnostic-tool

TCP "InSession"

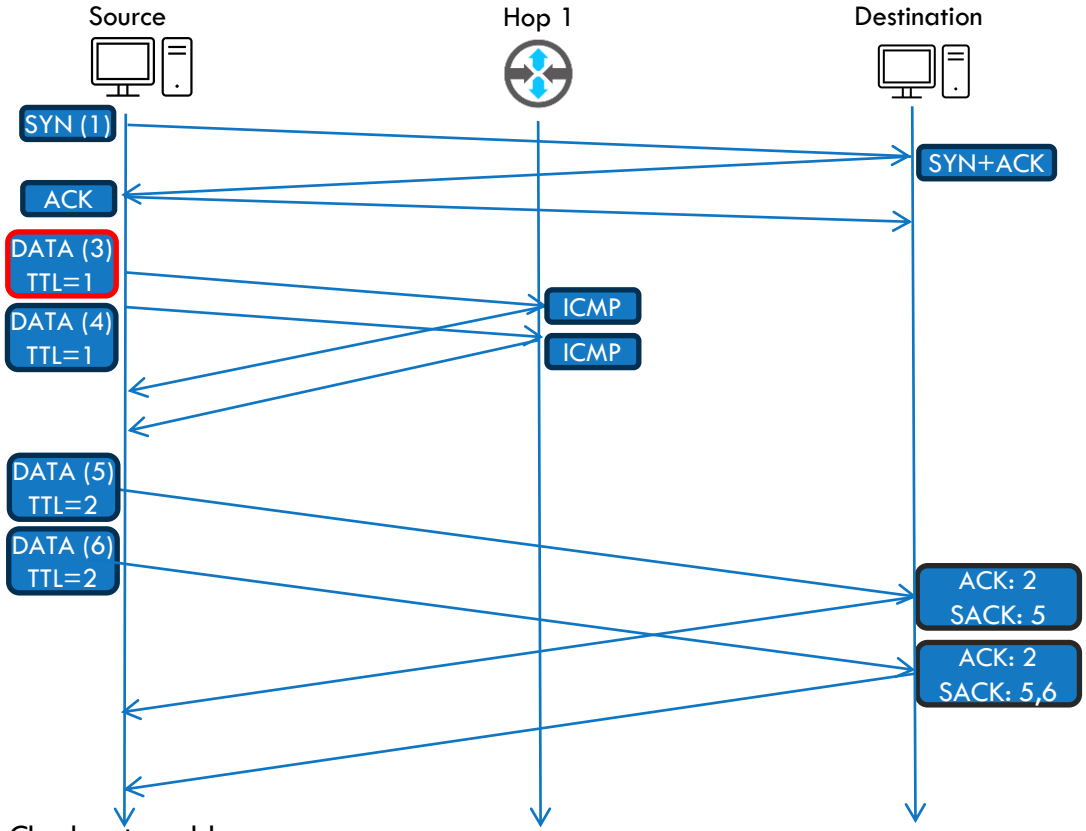


Open the TCP Session

Check out our blog:

https://www.catchpoint.com/blog/traceroute-insession-catchpoints-effort-towards-a-more-reliable-network-diagnostic-tool

TCP "InSession"

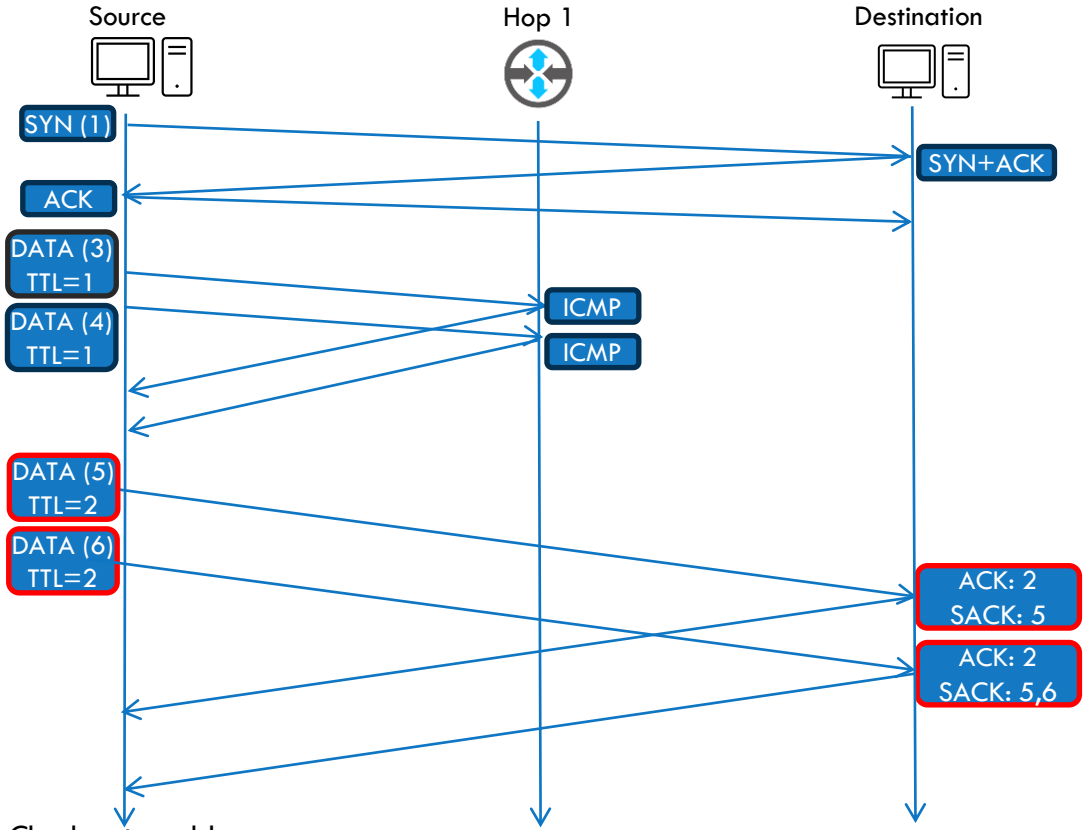


Insert a gap into the stream of sequence numbers

Check out our blog:

https://www.catchpoint.com/blog/traceroute-insession-catchpoints-effort-towards-a-more-reliable-network-diagnostic-tool

TCP "InSession"



We leverage SACK to match the ACK with the probe

- Negotiated in handshake

Check out our blog:

https://www.catchpoint.com/blog/traceroute-insession-catchpoints-effort-towards-a-more-reliable-network-diagnostic-tool

TCP InSession

- CONS
  - Requires SACK mechanism
  - Works only if the endpoint is listening in TCP on the target port
  - Opens a TCP session with the target host
- PROS
  - Sends 1 SYN per traceroute
    - Avoid to cause SYN flood
  - Traceroute probes are seen as part of a data flow
    - Bypass firewalls
    - Data packets are "more likely" to follow the same path

TCP InSession

```
[redacted] sudo ./traceroute/traceroute -n -T www.bing.com
traceroute to www.bing.com(204.79.197.200), 30 hops max, 60 byte packetsoverall timeout not set
   1  192.168.0.1  0.669 ms  0.621 ms  0.604 ms
   2  192.168.1.1  0.827 ms  0.571 ms  0.527 ms
   3  * * *
   4  172.18.33.212  5.071 ms  5.548 ms 172.18.33.196  5.530 ms
   5  172.18.33.226  6.503 ms  6.080 ms 172.18.33.228  6.060 ms
   6  172.19.184.88  12.701 ms 172.19.184.92  10.165 ms 172.19.184.90  10.119 ms
   7  172.19.177.66  10.087 ms 172.19.177.24  10.058 ms 172.19.177.20  12.082 ms
   8  195.22.192.144  12.046 ms 195.22.205.98  10.502 ms  10.047 ms
   9  195.22.208.79  10.003 ms 195.22.196.69  12.290 ms 195.22.208.79  12.248 ms
  10  195.22.196.129  22.943 ms 195.22.196.81  13.345 ms 195.22.196.129  11.678 ms
  11  13.104.182.192  11.451 ms 13.104.182.193  11.406 ms *
  12  * * *
  13  204.79.197.200 <MSS:1440>  13.649 ms * *
   Timedout: false
   Duration: 163.111 ms
   DestinationReached: true
```

Classic TCP traceroute

- Almost each hop has multiple IP addresses
- The destination replied only once

TCP InSession

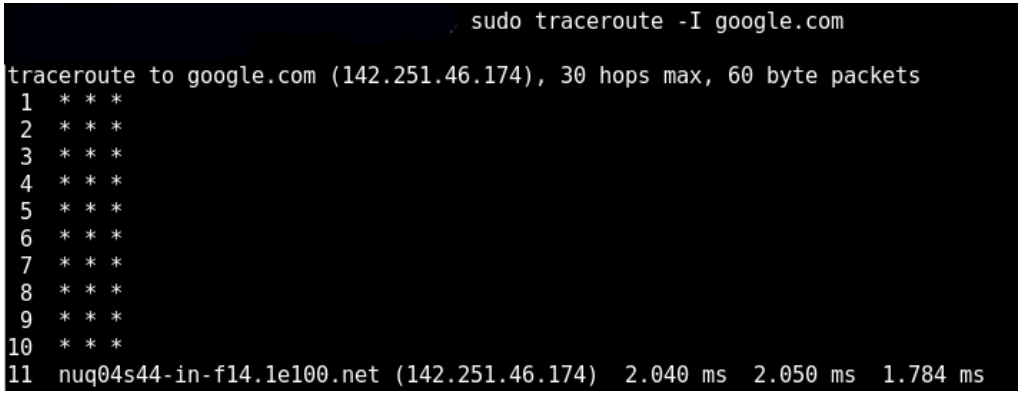
- Each hop has one IP address
- The destination replied to all probes

```
[redacted] sudo ./traceroute/traceroute -n --tcpinsession www.bing.com
traceroute to www.bing.com(95.101.20.193), 30 hops max, 53 byte packetsoverall timeout not set
   1  192.168.0.1  0.490 ms  0.452 ms  0.440 ms
   2  192.168.100.1  5.520 ms  5.511 ms  5.672 ms
   3  * * *
   4  172.18.33.212  5.640 ms  5.632 ms  6.129 ms
   5  172.18.33.230  6.124 ms  6.776 ms  6.774 ms
   6  172.19.184.92  10.463 ms  10.996 ms  10.975 ms
   7  172.19.177.60  10.612 ms  9.895 ms  9.872 ms
   8  151.99.72.197  26.336 ms  26.313 ms  26.302 ms
   9  23.210.57.131  10.219 ms  10.192 ms  10.887 ms
  10  95.101.20.193  9.663 ms  12.033 ms  9.457 ms
   Timedout: false
   Duration: 71.904 ms
   DestinationReached: true
   MSS: 1452
```

Work in Azure environment

Azure environment

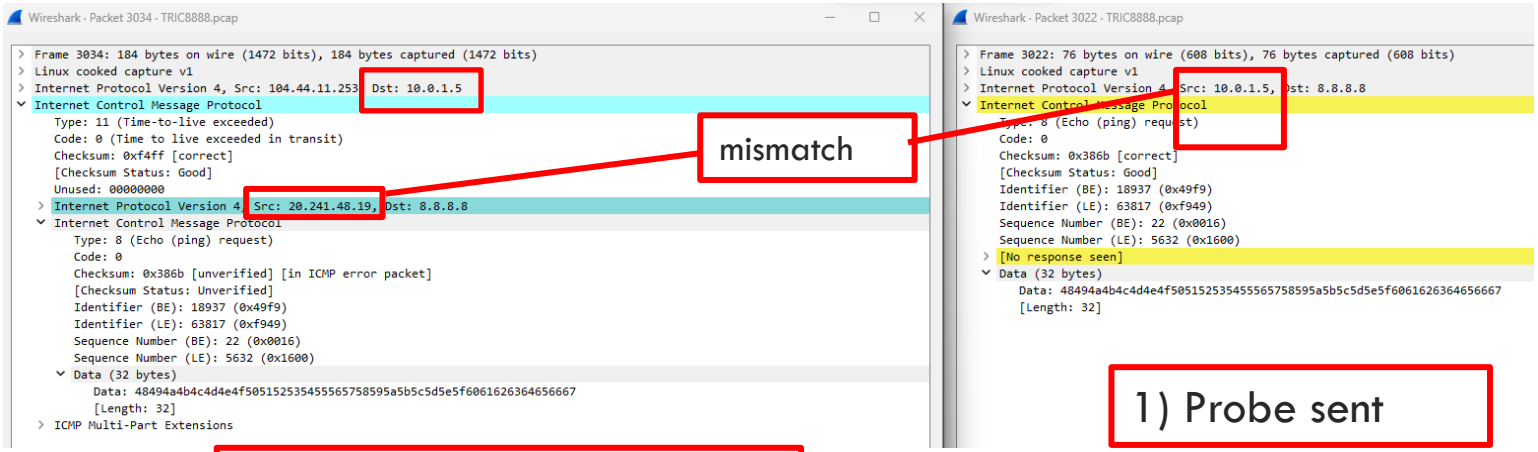
- Intermediate hops are all *
- This happens for all traceroute protocols



- (Linux) VM with private IP
- Inbound ICMP packets are allowed

Azure environment

- This happens because the source IP of the original probe encapsulated into the ICMP TTL Exceeded is left with the node public IP
- Thus, the ICMP reply is discarded by the kernel (not by traceroute)



Work in Azure environment

- We enhanced traceroute to work in "loose match mode"
- Open an additional raw ICMP socket to receive all ICMP packets and do the "kernel checks" at user level...
  - … but do not check the source address of the encapsulated probe

```
                                                traceroute --loose-match -I google.com

traceroute to google.com (142.251.46.174), 30 hops max, 60 byte packets, overall timeout not set
 1  * * * D=5.003980
 2  * * * D=5.003996
 3  * * * D=5.004010
 4  * * * D=5.004024
 5  * * * D=5.004039
 6  * * * D=5.030275
 7  ae31-0.sjc-96cbe-1b.ntwk.msn.net (104.44.238.247)  1.617 ms  1.617 ms  1.611 ms D=0.001641
 8  google.sjc-96cbe-1b.ntwk.msn.net (207.46.219.195)  1.927 ms  1.924 ms  1.919 ms D=0.001939
 9  142.251.69.83 (142.251.69.83)  4.132 ms  4.128 ms  4.124 ms D=0.004141
10  142.251.224.189 (142.251.224.189)  2.087 ms  2.082 ms  2.081 ms D=0.002101
11  nuq04s44-in-f14.1e100.net (142.251.46.174)  2.050 ms  2.046 ms  1.871 ms D=0.003395
```

And many more

And many more!

- ECN-awareness (IP and transport level)
- Path MTU performance improvements
- Report ToS/DSCP hop by hop
- Report MSS when running in TCP mode
- Handle print in a separate thread (speed up)
- Overall timeout
- Compile and run on Alpine
- Avoid UDP standard filtering

Thank you!

- Feel free to check/use/ & contribute!
  https://github.com/catchpoint/Networking.traceroute/ (GPL!)
- And come by to meet us!
  - Pietrasanta is a nice town on Tuscany seaside…