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

FOSDEM 2024

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" ?
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...

Linux traceroute

• We leverage Dmitry Butskoy’s "Linux traceroute"
  • Very fast
  • Open source
  • Easily extendible

This is a new modern implementation of traceroute(8) utility for Linux systems.
It has replaced the old one in the majority of distributions now, including Fedora, RHEL, Debian, Mandriva, Gentoo, Ubuntu.

• 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
• 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:

TCP "InSession"

Open the TCP Session

Check out our blog:
TCP "InSession"

Insert a gap into the stream of sequence numbers

Check out our blog:
TCP "InSession"

We leverage SACK to match the ACK with the probe
- Negotiated in handshake

Caused by probe with seq num 5

Caused by probe with seq num 6

Check out our blog:
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

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

Classic TCP traceroute

- Almost each hop has multiple IP addresses
- The destination replied only once
Work in Azure environment
Azure environment

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

```
sudo traceroute -I google.com
```

- (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).

1) Probe sent
2) ICMP TTL Exceeded

mismatch
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) 2.687 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...