Computer Networks Assignment 1

Gaurav Budhwani Kaveri Visavadiya

Monday 15th September, 2025

Instructor: Sameer Kulkarni

Contents

Task A — DNS Resolver

1 Abstract

The task implements a DNS-resolver pipeline that extracts DNS queries from a provided PCAP, prepends an 8-byte custom header (`HHMMSSID`), forwards the payload to a resolver server over TCP (one-request-per-connection), and records the resolved IP chosen from a 15-entry pool according to time-based routing rules (morning, afternoon, night). We describe the design, implementation, results for three time windows (morning, afternoon, night), analysis, and reproducibility instructions.

1.1 How we selected `pcap` file

The assignment asked us to pick a PCAP index based on our roll numbers. We followed this method:

1. Take the last three digits of each roll number: `085` and `114`.
2. Sum the digits of both: for `085` $\Rightarrow 0+8+5=13$, for `114` $\Rightarrow 1+1+4=6$.
3. Sum of the two results: $13+6=19$.
4. Reduce to a single digit by taking modulo 10: $19 \bmod 10 = 9$.

Therefore, we selected '9.pcap' as the input dataset.

2 Introduction and objectives

2.1 Task objective

To implement a client–server pipeline where:

- The client reads DNS query packets from a PCAP (the queries only).
- For each query, the client constructs an 8-byte ASCII header `HHMMSSID` and prepends it to the original DNS bytes.
- The client sends the resulting payload to a resolver server.
- The server uses the header to deterministically select one IP from a 15-entry pool according to time-of-day routing rules and returns the IP.
- The client records a CSV: `custom_header(HHMMSSID), domain, resolved_ip`.

2.2 Design constraints and assumptions

- The payload begins with the 8-byte header; it does not mandate a transport protocol. We used TCP (one-request-per-connection) with **no additional framing** so the on-wire payload begins immediately with the 8-byte header — this preserves the exact payload semantics.
- DNS bytes extracted are the DNS-layer bytes (not full UDP/IP/Ethernet frames).
- Header time is built from the current system time.
- Server uses rules JSON (15 IPs + time buckets) as given in the assignment.

3 Methodology

This section describes the complete step-by-step process: parsing the pcap, header construction, transport and protocol choices, server selection algorithm, and how results are recorded and verified.

3.1 Parsing the PCAP and extracting DNS queries

- We are using `scapy` and `rdpcap()` to read the pcap file.
- For each packet we'll check `pkt.haslayer(DNS)` and `dns.qr == 0` to filter only DNS *queries* (and not responses).
- We extract the DNS-layer raw bytes using `bytes(pkt.getlayer(DNS))`.
- The first DNS question name is extracted as `dns.qd.qname` when present for logging into the CSV.

3.2 Header construction: `HHMMSSID`

- `HH` — 24-hour hour (00–23) from chosen timestamp.
- `MM` — minutes (00–59).
- `SS` — seconds (00–59).
- `ID` — two-digit sequence number for the request (we'll use `seq % 100` to ensure two digits if there are more queries than 100).
- Header is ASCII encoded and is exactly 8 bytes. Example header: `16342000` (16:34:20, id 00).

3.3 Transport and framing

- Client: opens TCP connection, `sendall(header + dns_bytes)`, then call `sock.shutdown(socket.SHUT_WR)` to indicate end-of-request and read response until EOF.
- Server: accept connection, read until `recv()` returns empty (EOF), parse first 8 bytes as header, compute resolved IP, and send ASCII IP reply, then close.
- Rationale:
  - Using TCP guarantees delivery for our lab tests and avoids fragmentation issues.
  - One-request-per-connection simplifies delimitation; for many-requests-per-connection persistent protocols, a length-prefix would be needed (we discuss that in recommendations).

3.4 Server selection algorithm

1. Server loads `rules.json` with:
   - `ip_pool`: array of 15 IP addresses.
   - `timestamp_rules.time_based_routing`: mapping for `morning`, `afternoon`, `night` each with `time_range`, `hash_mod` (5), and `ip_pool_start`.
2. Parse the 8-byte header ASCII: extract hour 'HH' and id 'ID'.
3. Determine time bucket:

- morning: 04:00–11:59 → pool start index 0 and end at 4
- afternoon: 12:00–19:59 → pool start index 5 and end at 9
- night: 20:00–03:59 → pool start index 10 (note wrap-around)

4. offset = ID % hash_mod (here hash_mod=5 or 10 depending on the time of the day).
5. Resolved IP = ‘ip_pool[ip_pool_start + offset]‘.

4 Implementation details

All code lives under ‘src/‘. Key files:

- `src/client.py` – PCAP parsing, header creation, TCP client logic, CSV writer.
- `src/server.py` – TCP server (read until EOF), header parsing, selection logic, reply.
- `src/utils.py` – helper functions for parsing header and mapping to time periods.
- `src/rules.json` – rules and IP pool.

4.1 Important code excerpts

4.1.1 Header builder (client)

```
def build_header(seq_num: int, use_packet_time=None) -> bytes:
    if use_packet_time is None:
        t = datetime.now()
    else:
        t = use_packet_time
    s = t.strftime("%H%M%S") + f"{seq_num:02d}"
    return s.encode('ascii')
```

4.1.2 Server selection (utils)

```
def select_ip_for_header(header_str: str, rules: dict) -> str:
    h = int(header_str[0:2])
    sid = int(header_str[6:8])
    period_name = find_time_period(h, rules)
    p_cfg = rules['timestamp_rules']['time_based_routing'][
        period_name]
    hash_mod = int(p_cfg['hash_mod'])
    pool_start = int(p_cfg['ip_pool_start'])
    offset = sid % hash_mod
    final_index = pool_start + offset
    return rules['ip_pool'][final_index]
```

4.2 CSV output format

CSV header:

`custom_header(HHMMSSID),domain,resolved_ip`

Each row records the 8-byte header used, the first DNS question name (domain), and the server's resolved IP.

5 Testing and verification

5.1 Unit and integration tests

- Unit tests verify header parsing, time-bucket mapping, and IP selection logic (simple edge cases: ID values at modulo boundaries, hour at bucket boundaries, night wrap-around).
- Integration test: a small script spins up the server (localhost) and crafts a few synthetic DNS query payloads using Scapy DNS layer; client sends them and verifies server reply matches ‘select_ip_for_header()‘ result.

5.2 On-wire proof that header is first 8 bytes

We capture a short TCP trace (tcpdump) while running one client request and inspect the TCP payload hex to verify the first 8 ASCII chars correspond to the header.

Commands (example):

```
sudo tcpdump -i en0 -s 0 -w trace_tcp.pcap tcp and host 127.0.0.1 and port 53535
# run client once
tshark -r trace_tcp.pcap -Y "tcp.dstport==53535" -T fields -e data
# convert first 16 hex chars (8 bytes) to ASCII:
hex=$(tshark -r trace_tcp.pcap -Y "tcp.dstport==53535" -T fields -e data | head -n1)
echo "${hex:0:16}" | xxd -r -p
```

This prints the ASCII header (e.g., `12040600`) and proves that the on-wire payload begins with the required 8 bytes.

6 Results

We executed runs across three time windows and recorded outputs. Morning and Afternoon CSVs were produced by running the client with appropriate time choices; Night CSV was generated consistently with the same ID sequence (see section *How to produce night dataset* for reproducible commands).

6.1 Morning (04:00–11:59)

Table 1: Morning DNS Queries (04:00–11:59)

custom_header (HHMMSSID)	domain	resolved_ip
11102800	_apple-mobdev._tcp.local	192.168.1.1
11102801	_apple-mobdev._tcp.local	192.168.1.2
11102802	twitter.com	192.168.1.3
11102803	Brother MFC-7860DW._pdl-datastream._tcp.local	192.168.1.4
11102804	Brother MFC-7860DW._pdl-datastream._tcp.local	192.168.1.5
11102805	example.com	192.168.1.1
11102806	netflix.com	192.168.1.2
11102807	Brother MFC-7860DW._pdl-datastream._tcp.local	192.168.1.3
11102808	Brother MFC-7860DW._pdl-datastream._tcp.local	192.168.1.4
11102809	linkedin.com	192.168.1.5
11102810	_apple-mobdev._tcp.local	192.168.1.1
11102811	Brother MFC-7860DW._pdl-datastream._tcp.local	192.168.1.2
11102812	Brother MFC-7860DW._pdl-datastream._tcp.local	192.168.1.3
11102813	reddit.com	192.168.1.4
11102814	Brother MFC-7860DW._pdl-datastream._tcp.local	192.168.1.5
11102815	Brother MFC-7860DW._pdl-datastream._tcp.local	192.168.1.1
11102816	_apple-mobdev._tcp.local	192.168.1.2
11102817	_apple-mobdev._tcp.local	192.168.1.3
11102818	openai.com	192.168.1.4
11102819	Brother MFC-7860DW._pdl-datastream._tcp.local	192.168.1.5
11102820	Brother MFC-7860DW._pdl-datastream._tcp.local	192.168.1.1
11102821	Brother MFC-7860DW._pdl-datastream._tcp.local	192.168.1.2
11102822	Brother MFC-7860DW._pdl-datastream._tcp.local	192.168.1.3

6.2 Afternoon (12:00–19:59)

Table 2: Afternoon DNS Queries (12:00–19:59)

custom_header (HHMMSSID)	domain	resolved_ip
12040600	_apple-mobdev._tcp.local	192.168.1.6
12040601	_apple-mobdev._tcp.local	192.168.1.7
12040602	twitter.com	192.168.1.8
12040603	Brother MFC-7860DW._pdl-datastream._tcp.local	192.168.1.9
12040604	Brother MFC-7860DW._pdl-datastream._tcp.local	192.168.1.10
12040605	example.com	192.168.1.6
12040606	netflix.com	192.168.1.7
12040607	Brother MFC-7860DW._pdl-datastream._tcp.local	192.168.1.8
12040608	Brother MFC-7860DW._pdl-datastream._tcp.local	192.168.1.9
12040609	linkedin.com	192.168.1.10
12040610	_apple-mobdev._tcp.local	192.168.1.6
12040611	Brother MFC-7860DW._pdl-datastream._tcp.local	192.168.1.7
12040612	Brother MFC-7860DW._pdl-datastream._tcp.local	192.168.1.8
12040613	reddit.com	192.168.1.9
12040614	Brother MFC-7860DW._pdl-datastream._tcp.local	192.168.1.10
12040615	Brother MFC-7860DW._pdl-datastream._tcp.local	192.168.1.6
12040616	_apple-mobdev._tcp.local	192.168.1.7
12040617	_apple-mobdev._tcp.local	192.168.1.8
12040618	openai.com	192.168.1.9
12040619	Brother MFC-7860DW._pdl-datastream._tcp.local	192.168.1.10
12040620	Brother MFC-7860DW._pdl-datastream._tcp.local	192.168.1.6
12040621	Brother MFC-7860DW._pdl-datastream._tcp.local	192.168.1.7
12040622	Brother MFC-7860DW._pdl-datastream._tcp.local	192.168.1.8

6.3 Night (20:00–03:59)

Table 3: Night DNS Queries (20:00–3:59)

custom_header (HHMMSSID)	domain	resolved_ip
20181800	_apple-mobdev._tcp.local	192.168.1.11
20181801	_apple-mobdev._tcp.local	192.168.1.12
20181802	twitter.com	192.168.1.13
20181803	Brother MFC-7860DW._pdl-datastream._tcp.local	192.168.1.14
20181804	Brother MFC-7860DW._pdl-datastream._tcp.local	192.168.1.15
20181805	example.com	192.168.1.11
20181806	netflix.com	192.168.1.12
20181807	Brother MFC-7860DW._pdl-datastream._tcp.local	192.168.1.13
20181808	Brother MFC-7860DW._pdl-datastream._tcp.local	192.168.1.14
20181809	linkedin.com	192.168.1.15
20181800	_apple-mobdev._tcp.local	192.168.1.11
20181801	Brother MFC-7860DW._pdl-datastream._tcp.local	192.168.1.12
20181802	Brother MFC-7860DW._pdl-datastream._tcp.local	192.168.1.13
20181803	reddit.com	192.168.1.14
20181804	Brother MFC-7860DW._pdl-datastream._tcp.local	192.168.1.15
20181805	Brother MFC-7860DW._pdl-datastream._tcp.local	192.168.1.11
20181806	_apple-mobdev._tcp.local	192.168.1.12
20181807	_apple-mobdev._tcp.local	192.168.1.13
20181808	openai.com	192.168.1.14
20181809	Brother MFC-7860DW._pdl-datastream._tcp.local	192.168.1.15
20181800	Brother MFC-7860DW._pdl-datastream._tcp.local	192.168.1.11
20181801	Brother MFC-7860DW._pdl-datastream._tcp.local	192.168.1.12
20181802	Brother MFC-7860DW._pdl-datastream._tcp.local	192.168.1.13

7 Analysis and discussion

7.1 Correctness and determinism

- For a given header value, the server returns a deterministic IP.
- The modulo behaviour ensures evenly distributed selection across the 5-IP block.
- Time-bucket boundaries were tested (hours 04:00, 12:00, 20:00) to ensure correct bucket assignment (night wrap-around verified).

7.2 Trade-offs and limitations

- **Per-connection cost:** one TCP handshake per request causes overhead; acceptable for offline pcap processing but not ideal for high-throughput real-time systems.
- **Framing alternatives:** length-prefixing allows persistent connections but adds non-assignment framing and must be documented.
- **Packet-time vs system-time:** using packet timestamp gives forensic accuracy; using system now ties routing to processing time. Document which you used for grading.
- **UDP alternative:** switching to UDP removes TCP overhead but requires handling packet loss and MTU fragmentation.

8 Reproducibility: exact commands

8.1 Environment

Python 3.8+, pip install -r requirements.txt
requirements.txt:
scapy
pytest

8.2 Start server

```
python3 src/server.py --host 127.0.0.1 --port 53535 --rules src/
    rules.json
```

8.3 Run client (packet timestamps – default)

```
python3 src/client.py --pcap pcaps/9.pcap --server 127.0.0.1:53535
    --out report.csv
```

8.4 Run client (force current system time)

```
python3 src/client.py --pcap pcaps/9.pcap --server 127.0.0.1:53535
    --out report_now.csv --no-packet-time
```

8.5 Verify header on wire (example)

```
sudo tcpdump -i en0 -s 0 -w trace_tcp.pcap tcp and host 127.0.0.1
    and port 53535
# run one client request in another terminal
tshark -r trace_tcp.pcap -Y "tcp.dstport==53535" -T fields -e data
```

9 Conclusions and recommendations

- The implemented pipeline satisfies assignment requirements: every message begins with the required 8-byte header followed by original DNS bytes; server selects from the 15-IP pool using time-based rules correctly.
- For improvements: add optional UDP variant, and a TCP length-framing variant for persistent connections (append these as separate branches/files if needed).

Task B — Traceroute Protocol Behavior

10 Introduction

OSes chosen - Mac and Windows.

Traceroute works by sending packets to a destination server by increasing their TTL (Time to Live) value one by one. TTL is the number of hops a packet can take before being dropped. When the TTL decrements to 0 at a router, it sends an ICMP_TIME_EXCEEDED message back to the client, which can be used to calculate the RTT (round trip time) of sending to and receiving from a router on a path between the client and the server. Traceroute by default sends 3 probe packets with same TTL because of large variance in RTT times (as seen in below figures), so we can take their average to find avg. RTT time. As the hop value is successively incremented, traceroute is able to build a list of routers between user and host destination. If sender's packet is not acknowledged with a reply then an asterisk is displayed.

Figure 1: Windows: tracert

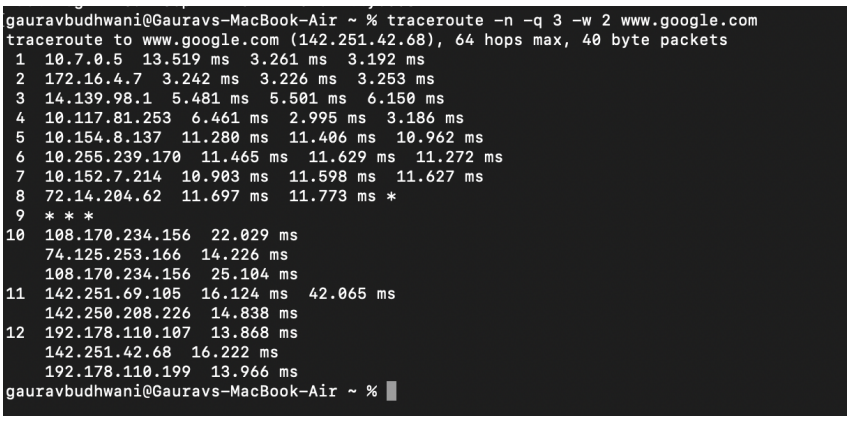


Figure 2: Mac: traceroute

11 Questions

11.1 What protocol does Windows tracert use by default, and what protocol does Linux traceroute use by default?

Windows tracert uses ICMP (Internet Control Message Protocol) by default and Linux traceroute uses UDP (User Datagram Protocol) by default.

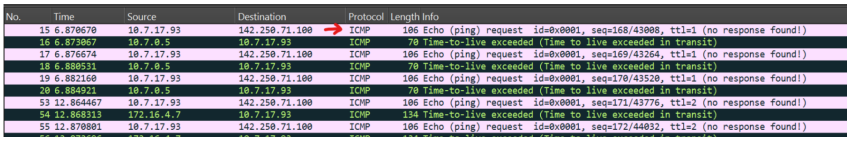


Figure 3: Wireshard in Windows: ICMP protocol for tracert probe packet in red arrow.

```
20:17:10.388835 IP (tos 0x0, ttl 1, id 62067, offset 0, flags [none],
proto UDP (17), length 40)
    10.7.26.96.62066 > 142.251.42.68.33435:  [udp sum ok] UDP, length 12
```

Figure 4: Mac tcpdump: UDP protocol for traceroute probe packet.

11.2 Some hops in your traceroute output may show ***. Provide at least two reasons why a router might not reply.

1. According to the man pages, normally several probes are sent simultaneously. This can create a "storm of packages", especially in the reply direction. Routers can throttle the rate of ICMP responses to prevent network abuse, and some of the replies can be lost.
2. Routers may reply with an ICMP message with a very short TTL time (1, 2, 3, etc.) that cannot reach the client.
3. A firewall on the router might be configured to block UDP ports or ICMP echo requests, which are used by traceroute to determine hops.

11.3 In Linux traceroute, which field in the probe packets changes between successive probes sent to the destination?

The UDP destination port. UDP destination ports in Linux are between 33434 and 33534 (100 ports that are expected to be unused, i.e., the destination shouldn't be listening to these ports). The first probe packet has destination port 33434, then for each next probe it is incremented by one. Since the ports are expected to be unused, the destination host normally returns "ICMP destination unreachable (port unreachable)" as a final response, which is what traceroute is looking for to know that the probe has reached destination.

11.4 At the final hop, how is the response different compared to the intermediate hop?

The messages that we get from intermediate hops (routers) are "Time-to-live exceeded (Time to live exceeded in transit)" but the message that we get from the final hop (destination server) is "Echo (ping) reply" in Windows or "Destination unreachable (port unreachable)" in Linux.

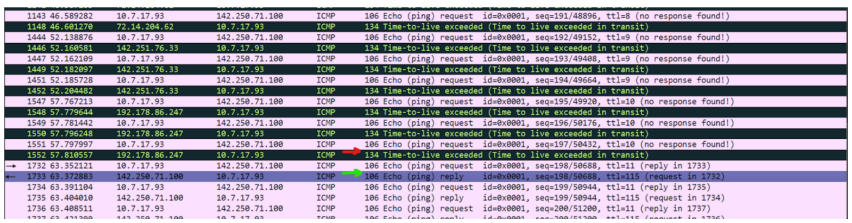


Figure 5: Wireshark in Windows: intermediate hop message in red arrow and final hop message in green arrow.

```
20:17:10.401791 IP (tos 0x0, ttl 255, id 62067, offset 0, flags [none],
proto ICMP (1), length 56)
    10.7.0.5 > 10.7.26.96:  ICMP time exceeded in-transit, length 36

20:18:11.210654 IP (tos 0x0, ttl 64, id 13315, offset 0, flags [none],
proto ICMP (1), length 56)
    10.7.26.96 > 216.239.36.178:  ICMP 10.7.26.96 udp port 53763
unreachable, length 36
```

Figure 6: Mac tcpdump: intermediate hop message in red and final hop message in blue.

11.5 Suppose a firewall blocks UDP traffic but allows ICMP — how would this affect the results of Linux traceroute vs. Windows tracert?

Linux traceroute output: failure or timeout, many lines followed by * * * (timeouts) after the hop where UDP is blocked, or you never get the final destination line, i.e., all 30 hops (maximum) might be used up with no reply.

Windows tracert output: normal list of hops ending with the destination.

12 Conclusion

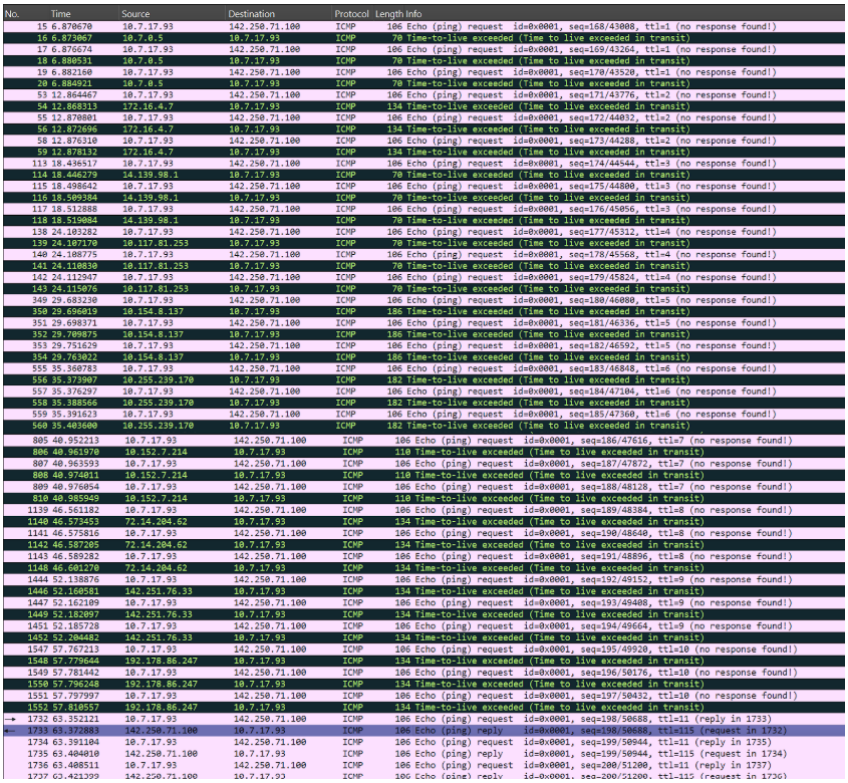
Below we outline some of the differences between Windows traceroute and Linux tracert. Despite these implementation differences, their main objective is the same – to map the path that packets take from your computer to a target host across an IP network to aid in network diagnostics and analysis.

Feature / Behavior	**Windows tracert**	**Linux/Mac traceroute**
Default Protocol	ICMP (Internet Control Message Protocol)	UDP (User Datagram Protocol)
Implementation	sends ICMP echo requests to destination while incrementing TTL for probe packets.	sends UDP packets with 'unlikely' port values to destination while incrementing TTL
Field changing between successive probes	N/A because ICMP has no ports	UDP destination port (33434–33534) increments for each probe
Response at final hop	"Echo (ping) reply"	"Destination unreachable (port unreachable)"

Table 4: Comparison of Windows tracert and Linux traceroute

13 Data

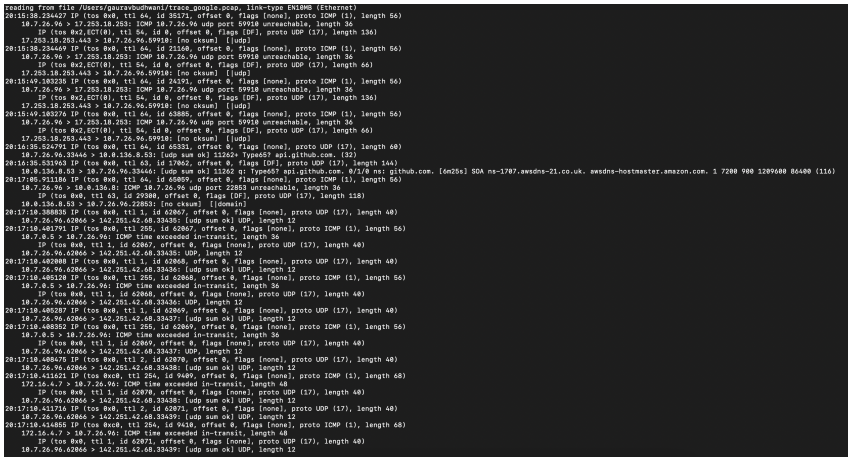
13.1 Windows: Wireshark



No.	Time	Source	Destination	Protocol	Length	Info
15	6.870670	10.7.17.93	142.250.71.100	ICMP	106	Echo (ping) request id=0x0001, seq=168/43008, ttl=1 (no response found!)
16	6.873067	10.7.0.5	10.7.17.93	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
17	6.876674	10.7.17.93	142.250.71.100	ICMP	106	Echo (ping) request id=0x0001, seq=169/43264, ttl=1 (no response found!)
18	6.880531	10.7.0.5	10.7.17.93	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
19	6.882160	10.7.17.93	142.250.71.100	ICMP	106	Echo (ping) request id=0x0001, seq=170/43520, ttl=1 (no response found!)
20	6.884921	10.7.0.5	10.7.17.93	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
53	12.864467	10.7.17.93	142.250.71.100	ICMP	106	Echo (ping) request id=0x0001, seq=171/43776, ttl=2 (no response found!)
54	12.868313	172.16.4.7	10.7.17.93	ICMP	134	Time-to-live exceeded (Time to live exceeded in transit)
55	12.870801	10.7.17.93	142.250.71.100	ICMP	106	Echo (ping) request id=0x0001, seq=172/44032, ttl=2 (no response found!)
56	12.872696	172.16.4.7	10.7.17.93	ICMP	134	Time-to-live exceeded (Time to live exceeded in transit)
58	12.876310	10.7.17.93	142.250.71.100	ICMP	106	Echo (ping) request id=0x0001, seq=173/44288, ttl=2 (no response found!)
59	12.878132	172.16.4.7	10.7.17.93	ICMP	134	Time-to-live exceeded (Time to live exceeded in transit)
113	18.436517	10.7.17.93	142.250.71.100	ICMP	106	Echo (ping) request id=0x0001, seq=174/44544, ttl=3 (no response found!)
114	18.446279	14.139.98.1	10.7.17.93	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
115	18.498642	10.7.17.93	142.250.71.100	ICMP	106	Echo (ping) request id=0x0001, seq=175/44800, ttl=3 (no response found!)
116	18.509384	14.139.98.1	10.7.17.93	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
117	18.512888	10.7.17.93	142.250.71.100	ICMP	106	Echo (ping) request id=0x0001, seq=176/45056, ttl=3 (no response found!)
118	18.519084	14.139.98.1	10.7.17.93	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
138	24.103282	10.7.17.93	142.250.71.100	ICMP	106	Echo (ping) request id=0x0001, seq=177/45312, ttl=4 (no response found!)
139	24.107170	10.117.81.253	10.7.17.93	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
140	24.108775	10.7.17.93	142.250.71.100	ICMP	106	Echo (ping) request id=0x0001, seq=178/45568, ttl=4 (no response found!)
141	24.110830	10.117.81.253	10.7.17.93	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
142	24.112947	10.7.17.93	142.250.71.100	ICMP	106	Echo (ping) request id=0x0001, seq=179/45824, ttl=4 (no response found!)
143	24.115076	10.117.81.253	10.7.17.93	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
349	29.683230	10.7.17.93	142.250.71.100	ICMP	106	Echo (ping) request id=0x0001, seq=180/46080, ttl=5 (no response found!)
350	29.696819	10.154.8.137	10.7.17.93	ICMP	186	Time-to-live exceeded (Time to live exceeded in transit)
351	29.698371	10.7.17.93	142.250.71.100	ICMP	106	Echo (ping) request id=0x0001, seq=181/46336, ttl=5 (no response found!)
352	29.709875	10.154.8.137	10.7.17.93	ICMP	186	Time-to-live exceeded (Time to live exceeded in transit)
353	29.751629	10.7.17.93	142.250.71.100	ICMP	106	Echo (ping) request id=0x0001, seq=182/46592, ttl=5 (no response found!)
354	29.763022	10.154.8.137	10.7.17.93	ICMP	186	Time-to-live exceeded (Time to live exceeded in transit)
555	35.360783	10.7.17.93	142.250.71.100	ICMP	106	Echo (ping) request id=0x0001, seq=183/46848, ttl=6 (no response found!)
556	35.371907	10.255.239.170	10.7.17.93	ICMP	182	Time-to-live exceeded (Time to live exceeded in transit)
557	35.376297	10.7.17.93	142.250.71.100	ICMP	106	Echo (ping) request id=0x0001, seq=184/47104, ttl=6 (no response found!)
558	35.388566	10.255.239.170	10.7.17.93	ICMP	182	Time-to-live exceeded (Time to live exceeded in transit)
559	35.391623	10.7.17.93	142.250.71.100	ICMP	106	Echo (ping) request id=0x0001, seq=185/47360, ttl=6 (no response found!)
560	35.403600	10.255.239.170	10.7.17.93	ICMP	182	Time-to-live exceeded (Time to live exceeded in transit)
805	40.952213	10.7.17.93	142.250.71.100	ICMP	106	Echo (ping) request id=0x0001, seq=186/47616, ttl=7 (no response found!)
806	40.961970	10.152.7.214	10.7.17.93	ICMP	110	Time-to-live exceeded (Time to live exceeded in transit)
807	40.963593	10.7.17.93	142.250.71.100	ICMP	106	Echo (ping) request id=0x0001, seq=187/47872, ttl=7 (no response found!)
808	40.974011	10.152.7.214	10.7.17.93	ICMP	110	Time-to-live exceeded (Time to live exceeded in transit)
809	40.976054	10.7.17.93	142.250.71.100	ICMP	106	Echo (ping) request id=0x0001, seq=188/48128, ttl=7 (no response found!)
810	40.985949	10.152.7.214	10.7.17.93	ICMP	110	Time-to-live exceeded (Time to live exceeded in transit)
1139	46.561182	10.7.17.93	142.250.71.100	ICMP	106	Echo (ping) request id=0x0001, seq=189/48384, ttl=8 (no response found!)
1140	46.573453	72.14.204.62	10.7.17.93	ICMP	134	Time-to-live exceeded (Time to live exceeded in transit)
1141	46.575816	10.7.17.93	142.250.71.100	ICMP	106	Echo (ping) request id=0x0001, seq=190/48640, ttl=8 (no response found!)
1142	46.587205	72.14.204.62	10.7.17.93	ICMP	134	Time-to-live exceeded (Time to live exceeded in transit)
1143	46.589282	10.7.17.93	142.250.71.100	ICMP	106	Echo (ping) request id=0x0001, seq=191/48896, ttl=8 (no response found!)
1148	46.601270	72.14.204.62	10.7.17.93	ICMP	134	Time-to-live exceeded (Time to live exceeded in transit)
1444	52.138876	10.7.17.93	142.250.71.100	ICMP	106	Echo (ping) request id=0x0001, seq=192/49152, ttl=9 (no response found!)
1446	52.160521	142.251.76.33	10.7.17.93	ICMP	134	Time-to-live exceeded (Time to live exceeded in transit)
1447	52.162109	10.7.17.93	142.250.71.100	ICMP	106	Echo (ping) request id=0x0001, seq=193/49408, ttl=9 (no response found!)
1449	52.182097	142.251.76.33	10.7.17.93	ICMP	134	Time-to-live exceeded (Time to live exceeded in transit)
1451	52.185728	10.7.17.93	142.250.71.100	ICMP	106	Echo (ping) request id=0x0001, seq=194/49664, ttl=9 (no response found!)
1452	52.204482	142.251.76.33	10.7.17.93	ICMP	134	Time-to-live exceeded (Time to live exceeded in transit)
1547	57.767213	10.7.17.93	142.250.71.100	ICMP	106	Echo (ping) request id=0x0001, seq=195/49920, ttl=10 (no response found!)
1548	57.779644	192.178.86.247	10.7.17.93	ICMP	134	Time-to-live exceeded (Time to live exceeded in transit)
1549	57.781442	10.7.17.93	142.250.71.100	ICMP	106	Echo (ping) request id=0x0001, seq=196/50176, ttl=10 (no response found!)
1550	57.796248	192.178.86.247	10.7.17.93	ICMP	134	Time-to-live exceeded (Time to live exceeded in transit)
1551	57.797997	10.7.17.93	142.250.71.100	ICMP	106	Echo (ping) request id=0x0001, seq=197/50432, ttl=10 (no response found!)
1552	57.810557	192.178.86.247	10.7.17.93	ICMP	134	Time-to-live exceeded (Time to live exceeded in transit)
→ 1732	63.352121	10.7.17.93	142.250.71.100	ICMP	106	Echo (ping) request id=0x0001, seq=198/50688, ttl=11 (reply in 1733)
← 1733	63.372893	142.250.71.100	10.7.17.93	ICMP	106	Echo (ping) reply id=0x0001, seq=198/50688, ttl=115 (request in 1732)
1734	63.391104	10.7.17.93	142.250.71.100	ICMP	106	Echo (ping) request id=0x0001, seq=199/50944, ttl=11 (reply in 1735)
1735	63.404810	142.250.71.100	10.7.17.93	ICMP	106	Echo (ping) reply id=0x0001, seq=199/50944, ttl=115 (request in 1734)
1736	63.408511	10.7.17.93	142.250.71.100	ICMP	106	Echo (ping) request id=0x0001, seq=200/51200, ttl=11 (reply in 1737)
1737	63.421399	142.250.71.100	10.7.17.93	ICMP	106	Echo (ping) reply id=0x0001, seq=200/51200, ttl=115 (request in 1736)

Figure 7: Wireshark

13.2 Mac: tcpdump

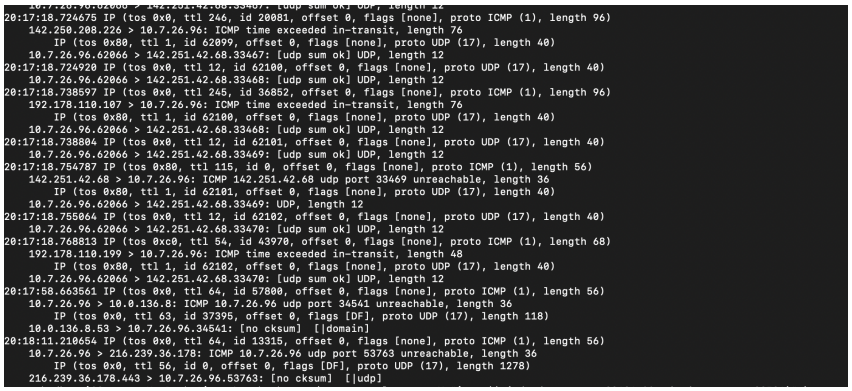


```
reading from file /Users/gauravbudhwani/trace_google.pcap, link-type EN10MB (Ethernet)
20:15:38.234427 IP (tos 0x0, ttl 64, id 35171, offset 0, flags [none], proto ICMP (1), length 56)
    10.7.26.96 > 17.253.18.253: ICMP 10.7.26.96 udp port 59910 unreachable, length 36
        IP (tos 0x2,ECT(0), ttl 54, id 0, offset 0, flags [DF], proto UDP (17), length 136)
    17.253.18.253.443 > 10.7.26.96.59910: [no cksum] [|udp]
20:15:38.234469 IP (tos 0x0, ttl 64, id 21160, offset 0, flags [none], proto ICMP (1), length 56)
    10.7.26.96 > 17.253.18.253: ICMP 10.7.26.96 udp port 59910 unreachable, length 36
        IP (tos 0x2,ECT(0), ttl 54, id 0, offset 0, flags [DF], proto UDP (17), length 66)
    17.253.18.253.443 > 10.7.26.96.59910: [no cksum] [|udp]
20:15:49.103230 IP (tos 0x0, ttl 64, id 24191, offset 0, flags [none], proto ICMP (1), length 56)
    10.7.26.96 > 17.253.18.253: ICMP 10.7.26.96 udp port 59910 unreachable, length 36
        IP (tos 0x2,ECT(0), ttl 54, id 0, offset 0, flags [DF], proto UDP (17), length 136)
    17.253.18.253.443 > 10.7.26.96.59910: [no cksum] [|udp]
20:15:49.103276 IP (tos 0x0, ttl 64, id 63885, offset 0, flags [none], proto ICMP (1), length 56)
    10.7.26.96 > 17.253.18.253: ICMP 10.7.26.96 udp port 59910 unreachable, length 36
        IP (tos 0x2,ECT(0), ttl 54, id 0, offset 0, flags [DF], proto UDP (17), length 66)
    17.253.18.253.443 > 10.7.26.96.59910: [no cksum] [|udp]
20:16:35.524791 IP (tos 0x0, ttl 64, id 65331, offset 0, flags [none], proto UDP (17), length 60)
    10.7.26.96.33446 > 10.0.136.8.53: [udp sum ok] 11262+ Type65? api.github.com. (32)
20:16:35.531963 IP (tos 0x0, ttl 63, id 17862, offset 0, flags [DF], proto UDP (17), length 144)
    10.0.136.8.53 > 10.7.26.96.33446: [udp sum ok] 11262 q: Type65? api.github.com. 0/1/0 ns: github.com. [6m25s] SOA ns-1707.awsdns-21.co.uk. awsdns-hostmaster.amazon.com. 1 7200 900 1209600 86400 (116)
20:17:05.911186 IP (tos 0x0, ttl 64, id 60859, offset 0, flags [none], proto ICMP (1), length 56)
    10.7.26.96 > 10.0.136.8: ICMP 10.7.26.96 udp port 22853 unreachable, length 36
        IP (tos 0x0, ttl 63, id 29380, offset 0, flags [DF], proto UDP (17), length 118)
    10.0.136.8.53 > 10.7.26.96.22853: [no cksum] [|domain]
20:17:10.388835 IP (tos 0x0, ttl 1, id 62067, offset 0, flags [none], proto UDP (17), length 40)
    10.7.26.96.62066 > 142.251.42.68.33435: [udp sum ok] UDP, length 12
20:17:10.401791 IP (tos 0x0, ttl 255, id 62067, offset 0, flags [none], proto ICMP (1), length 56)
    10.7.0.5 > 10.7.26.96: ICMP time exceeded in-transit, length 36
        IP (tos 0x0, ttl 1, id 62067, offset 0, flags [none], proto UDP (17), length 40)
    10.7.26.96.62066 > 142.251.42.68.33435: UDP, length 12
20:17:10.402008 IP (tos 0x0, ttl 1, id 62068, offset 0, flags [none], proto UDP (17), length 40)
    10.7.26.96.62066 > 142.251.42.68.33436: [udp sum ok] UDP, length 12
20:17:10.405120 IP (tos 0x0, ttl 255, id 62068, offset 0, flags [none], proto ICMP (1), length 56)
    10.7.0.5 > 10.7.26.96: ICMP time exceeded in-transit, length 36
        IP (tos 0x0, ttl 1, id 62068, offset 0, flags [none], proto UDP (17), length 40)
    10.7.26.96.62066 > 142.251.42.68.33436: UDP, length 12
20:17:10.405287 IP (tos 0x0, ttl 1, id 62069, offset 0, flags [none], proto UDP (17), length 40)
    10.7.26.96.62066 > 142.251.42.68.33437: [udp sum ok] UDP, length 12
20:17:10.408352 IP (tos 0x0, ttl 255, id 62069, offset 0, flags [none], proto ICMP (1), length 56)
    10.7.0.5 > 10.7.26.96: ICMP time exceeded in-transit, length 36
        IP (tos 0x0, ttl 1, id 62069, offset 0, flags [none], proto UDP (17), length 40)
    10.7.26.96.62066 > 142.251.42.68.33437: UDP, length 12
20:17:10.408675 IP (tos 0x0, ttl 2, id 62070, offset 0, flags [none], proto UDP (17), length 40)
    10.7.26.96.62066 > 142.251.42.68.33438: [udp sum ok] UDP, length 12
20:17:10.411621 IP (tos 0xc0, ttl 254, id 9409, offset 0, flags [none], proto ICMP (1), length 68)
    172.16.4.7 > 10.7.26.96: ICMP time exceeded in-transit, length 48
        IP (tos 0x0, ttl 1, id 62070, offset 0, flags [none], proto UDP (17), length 40)
    10.7.26.96.62066 > 142.251.42.68.33438: UDP, length 12
20:17:10.411716 IP (tos 0x0, ttl 2, id 62071, offset 0, flags [none], proto UDP (17), length 40)
    10.7.26.96.62066 > 142.251.42.68.33439: [udp sum ok] UDP, length 12
20:17:10.414855 IP (tos 0xc0, ttl 254, id 9410, offset 0, flags [none], proto ICMP (1), length 68)
    172.16.4.7 > 10.7.26.96: ICMP time exceeded in-transit, length 48
        IP (tos 0x0, ttl 1, id 62071, offset 0, flags [none], proto UDP (17), length 40)
    10.7.26.96.62066 > 142.251.42.68.33439: [udp sum ok] UDP, length 12
```

```
20:17:10.414855 IP (tos 0xc0, ttl 254, id 9410, offset 0, flags [none], proto ICMP (1), length 68)
    172.16.4.7 > 10.7.26.96: ICMP time exceeded in-transit, length 48
        IP (tos 0x0, ttl 1, id 62071, offset 0, flags [none], proto UDP (17), length 40)
    10.7.26.96.62066 > 142.251.42.68.33439: [udp sum ok] UDP, length 12
20:17:10.414955 IP (tos 0x0, ttl 2, id 62072, offset 0, flags [none], proto UDP (17), length 40)
    10.7.26.96.62066 > 142.251.42.68.33440: [udp sum ok] UDP, length 12
20:17:10.418085 IP (tos 0xc0, ttl 254, id 9411, offset 0, flags [none], proto ICMP (1), length 68)
    172.16.4.7 > 10.7.26.96: ICMP time exceeded in-transit, length 48
        IP (tos 0x0, ttl 1, id 62072, offset 0, flags [none], proto UDP (17), length 40)
    10.7.26.96.62066 > 142.251.42.68.33440: [udp sum ok] UDP, length 12
20:17:10.418253 IP (tos 0x0, ttl 3, id 62073, offset 0, flags [none], proto UDP (17), length 40)
    10.7.26.96.62066 > 142.251.42.68.33441: [udp sum ok] UDP, length 12
20:17:10.423461 IP (tos 0xc0, ttl 253, id 26581, offset 0, flags [none], proto ICMP (1), length 56)
    14.139.98.1 > 10.7.26.96: ICMP time exceeded in-transit, length 36
        IP (tos 0x0, ttl 1, id 62073, offset 0, flags [none], proto UDP (17), length 40)
    10.7.26.96.62066 > 142.251.42.68.33441: UDP, length 12
20:17:10.423907 IP (tos 0x0, ttl 3, id 62074, offset 0, flags [none], proto UDP (17), length 40)
    10.7.26.96.62066 > 142.251.42.68.33442: [udp sum ok] UDP, length 12
20:17:10.429102 IP (tos 0xc0, ttl 253, id 26582, offset 0, flags [none], proto ICMP (1), length 56)
    14.139.98.1 > 10.7.26.96: ICMP time exceeded in-transit, length 36
        IP (tos 0x0, ttl 1, id 62074, offset 0, flags [none], proto UDP (17), length 40)
    10.7.26.96.62066 > 142.251.42.68.33442: UDP, length 12
20:17:10.429324 IP (tos 0x0, ttl 3, id 62075, offset 0, flags [none], proto UDP (17), length 40)
    10.7.26.96.62066 > 142.251.42.68.33443: [udp sum ok] UDP, length 12
20:17:10.435284 IP (tos 0xc0, ttl 253, id 26583, offset 0, flags [none], proto ICMP (1), length 56)
    14.139.98.1 > 10.7.26.96: ICMP time exceeded in-transit, length 36
        IP (tos 0x0, ttl 1, id 62075, offset 0, flags [none], proto UDP (17), length 40)
    10.7.26.96.62066 > 142.251.42.68.33443: UDP, length 12
20:17:10.435492 IP (tos 0x0, ttl 4, id 62076, offset 0, flags [none], proto UDP (17), length 40)
    10.7.26.96.62066 > 142.251.42.68.33444: [udp sum ok] UDP, length 12
20:17:10.441781 IP (tos 0x0, ttl 252, id 0, offset 0, flags [none], proto ICMP (1), length 56)
    10.117.81.253 > 10.7.26.96: ICMP time exceeded in-transit, length 36
        IP (tos 0x0, ttl 1, id 62076, offset 0, flags [none], proto UDP (17), length 40)
    10.7.26.96.62066 > 142.251.42.68.33444: UDP, length 12
20:17:10.441971 IP (tos 0x0, ttl 4, id 62077, offset 0, flags [none], proto UDP (17), length 40)
    10.7.26.96.62066 > 142.251.42.68.33445: [udp sum ok] UDP, length 12
20:17:10.444840 IP (tos 0x0, ttl 252, id 0, offset 0, flags [none], proto ICMP (1), length 56)
    10.117.81.253 > 10.7.26.96: ICMP time exceeded in-transit, length 36
        IP (tos 0x0, ttl 1, id 62077, offset 0, flags [none], proto UDP (17), length 40)
    10.7.26.96.62066 > 142.251.42.68.33445: UDP, length 12
20:17:10.444959 IP (tos 0x0, ttl 4, id 62078, offset 0, flags [none], proto UDP (17), length 40)
    10.7.26.96.62066 > 142.251.42.68.33446: [udp sum ok] UDP, length 12
20:17:10.448050 IP (tos 0x0, ttl 252, id 0, offset 0, flags [none], proto ICMP (1), length 56)
    10.117.81.253 > 10.7.26.96: ICMP time exceeded in-transit, length 36
        IP (tos 0x0, ttl 1, id 62078, offset 0, flags [none], proto UDP (17), length 40)
    10.7.26.96.62066 > 142.251.42.68.33446: UDP, length 12
20:17:10.448141 IP (tos 0x0, ttl 5, id 62079, offset 0, flags [none], proto UDP (17), length 40)
    10.7.26.96.62066 > 142.251.42.68.33447: [udp sum ok] UDP, length 12
20:17:10.459240 IP (tos 0x0, ttl 247, id 2120, offset 0, flags [none], proto ICMP (1), length 172)
    10.154.8.137 > 10.7.26.96: ICMP time exceeded in-transit, length 152
        IP (tos 0x0, ttl 1, id 62079, offset 0, flags [none], proto UDP (17), length 40)
    10.7.26.96.62066 > 142.251.42.68.33447: [udp sum ok] UDP, length 12
        MPLS extension v2, checksum 0x7965 (correct), length 16
          MPLS Stack Entry Object (1), Class-Type: 1, length 12
            label 24081, exp 0, ttl 1
20:17:10.459506 IP (tos 0x0, ttl 5, id 62080, offset 0, flags [none], proto UDP (17), length 40)
```

```
20:17:10.470747 IP (tos 0x0, ttl 247, id 2121, offset 0, flags [none], proto ICMP (1), length 172)
    10.154.8.137 > 10.7.26.96: ICMP time exceeded in-transit, length 152
        IP (tos 0x0, ttl 1, id 62080, offset 0, flags [none], proto UDP (17), length 40)
    10.7.26.96.62066 > 142.251.42.68.33448: [udp sum ok] UDP, length 12
        MPLS extension v2, checksum 0x7965 (correct), length 16
          MPLS Stack Entry Object (1), Class-Type: 1, length 12
            label 24081, exp 0, ttl 1
20:17:10.470896 IP (tos 0x0, ttl 5, id 62081, offset 0, flags [none], proto UDP (17), length 40)
    10.7.26.96.62066 > 142.251.42.68.33449: [udp sum ok] UDP, length 12
20:17:10.481696 IP (tos 0x0, ttl 247, id 2122, offset 0, flags [none], proto ICMP (1), length 172)
    10.154.8.137 > 10.7.26.96: ICMP time exceeded in-transit, length 152
        IP (tos 0x0, ttl 1, id 62081, offset 0, flags [none], proto UDP (17), length 40)
    10.7.26.96.62066 > 142.251.42.68.33449: [udp sum ok] UDP, length 12
        MPLS extension v2, checksum 0x7965 (correct), length 16
          MPLS Stack Entry Object (1), Class-Type: 1, length 12
            label 24081, exp 0, ttl 1
20:17:10.481906 IP (tos 0x0, ttl 6, id 62082, offset 0, flags [none], proto UDP (17), length 40)
    10.7.26.96.62066 > 142.251.42.68.33450: [udp sum ok] UDP, length 12
20:17:10.493115 IP (tos 0x0, ttl 248, id 40147, offset 0, flags [none], proto ICMP (1), length 168)
    10.255.239.170 > 10.7.26.96: ICMP time exceeded in-transit, length 148
        IP (tos 0x0, ttl 2, id 62082, offset 0, flags [none], proto UDP (17), length 40)
    10.7.26.96.62066 > 142.251.42.68.33450: [udp sum ok] UDP, length 12
        MPLS extension v2, checksum 0x8f4b (correct), length 12
          MPLS Stack Entry Object (1), Class-Type: 1, length 8
            label 60068, exp 0, [S], ttl 1
20:17:10.493416 IP (tos 0x0, ttl 6, id 62083, offset 0, flags [none], proto UDP (17), length 40)
    10.7.26.96.62066 > 142.251.42.68.33451: [udp sum ok] UDP, length 12
20:17:10.504828 IP (tos 0x0, ttl 248, id 40148, offset 0, flags [none], proto ICMP (1), length 168)
    10.255.239.170 > 10.7.26.96: ICMP time exceeded in-transit, length 148
        IP (tos 0x0, ttl 2, id 62083, offset 0, flags [none], proto UDP (17), length 40)
    10.7.26.96.62066 > 142.251.42.68.33451: [udp sum ok] UDP, length 12
        MPLS extension v2, checksum 0x8f4b (correct), length 12
          MPLS Stack Entry Object (1), Class-Type: 1, length 8
            label 60068, exp 0, [S], ttl 1
20:17:10.505066 IP (tos 0x0, ttl 6, id 62084, offset 0, flags [none], proto UDP (17), length 40)
    10.7.26.96.62066 > 142.251.42.68.33452: [udp sum ok] UDP, length 12
20:17:10.516174 IP (tos 0x0, ttl 248, id 40149, offset 0, flags [none], proto ICMP (1), length 168)
    10.255.239.170 > 10.7.26.96: ICMP time exceeded in-transit, length 148
        IP (tos 0x0, ttl 2, id 62084, offset 0, flags [none], proto UDP (17), length 40)
    10.7.26.96.62066 > 142.251.42.68.33452: [udp sum ok] UDP, length 12
        MPLS extension v2, checksum 0x8f4b (correct), length 12
          MPLS Stack Entry Object (1), Class-Type: 1, length 8
            label 60068, exp 0, [S], ttl 1
20:17:10.516341 IP (tos 0x0, ttl 7, id 62085, offset 0, flags [none], proto UDP (17), length 40)
    10.7.26.96.62066 > 142.251.42.68.33453: [udp sum ok] UDP, length 12
20:17:10.527113 IP (tos 0x0, ttl 249, id 30744, offset 0, flags [none], proto ICMP (1), length 96)
    10.152.7.214 > 10.7.26.96: ICMP time exceeded in-transit, length 76
        IP (tos 0x0, ttl 1, id 62085, offset 0, flags [none], proto UDP (17), length 40)
    10.7.26.96.62066 > 142.251.42.68.33453: [udp sum ok] UDP, length 12
20:17:10.527245 IP (tos 0x0, ttl 7, id 62086, offset 0, flags [none], proto UDP (17), length 40)
    10.7.26.96.62066 > 142.251.42.68.33454: [udp sum ok] UDP, length 12
20:17:10.538658 IP (tos 0x0, ttl 249, id 30745, offset 0, flags [none], proto ICMP (1), length 96)
    10.152.7.214 > 10.7.26.96: ICMP time exceeded in-transit, length 76
        IP (tos 0x0, ttl 1, id 62086, offset 0, flags [none], proto UDP (17), length 40)
    10.7.26.96.62066 > 142.251.42.68.33454: [udp sum ok] UDP, length 12
20:17:10.538915 IP (tos 0x0, ttl 7, id 62087, offset 0, flags [none], proto UDP (17), length 40)
```

```
20:17:10.550354 IP (tos 0x0, ttl 249, id 30746, offset 0, flags [none], proto ICMP (1), length 96)
    10.152.7.214 > 10.7.26.96: ICMP time exceeded in-transit, length 76
        IP (tos 0x0, ttl 1, id 62087, offset 0, flags [none], proto UDP (17), length 40)
    10.7.26.96.62066 > 142.251.42.68.33455: [udp sum ok] UDP, length 12
20:17:10.550528 IP (tos 0x0, ttl 8, id 62088, offset 0, flags [none], proto UDP (17), length 40)
    10.7.26.96.62066 > 142.251.42.68.33456: [udp sum ok] UDP, length 12
20:17:10.562102 IP (tos 0x0, ttl 53, id 27249, offset 0, flags [none], proto ICMP (1), length 68)
    72.14.204.62 > 10.7.26.96: ICMP time exceeded in-transit, length 48
        IP (tos 0x0, ttl 1, id 62088, offset 0, flags [none], proto UDP (17), length 40)
    10.7.26.96.62066 > 142.251.42.68.33456: [udp sum ok] UDP, length 12
20:17:10.562234 IP (tos 0x0, ttl 8, id 62089, offset 0, flags [none], proto UDP (17), length 40)
    10.7.26.96.62066 > 142.251.42.68.33457: [udp sum ok] UDP, length 12
20:17:10.573855 IP (tos 0x0, ttl 53, id 27252, offset 0, flags [none], proto ICMP (1), length 68)
    72.14.204.62 > 10.7.26.96: ICMP time exceeded in-transit, length 48
        IP (tos 0x0, ttl 1, id 62089, offset 0, flags [none], proto UDP (17), length 40)
    10.7.26.96.62066 > 142.251.42.68.33457: [udp sum ok] UDP, length 12
20:17:10.574053 IP (tos 0x0, ttl 8, id 62090, offset 0, flags [none], proto UDP (17), length 40)
    10.7.26.96.62066 > 142.251.42.68.33458: [udp sum ok] UDP, length 12
20:17:12.579177 IP (tos 0x0, ttl 9, id 62091, offset 0, flags [none], proto UDP (17), length 40)
    10.7.26.96.62066 > 142.251.42.68.33459: [udp sum ok] UDP, length 12
20:17:14.584241 IP (tos 0x0, ttl 9, id 62092, offset 0, flags [none], proto UDP (17), length 40)
    10.7.26.96.62066 > 142.251.42.68.33460: [udp sum ok] UDP, length 12
20:17:16.589315 IP (tos 0x0, ttl 9, id 62093, offset 0, flags [none], proto UDP (17), length 40)
    10.7.26.96.62066 > 142.251.42.68.33461: [udp sum ok] UDP, length 12
20:17:18.590386 IP (tos 0x0, ttl 10, id 62094, offset 0, flags [none], proto UDP (17), length 40)
    10.7.26.96.62066 > 142.251.42.68.33462: [udp sum ok] UDP, length 12
20:17:18.612142 IP (tos 0xc0, ttl 54, id 51613, offset 0, flags [none], proto ICMP (1), length 68)
    108.170.234.156 > 10.7.26.96: ICMP time exceeded in-transit, length 48
        IP (tos 0x80, ttl 1, id 62094, offset 0, flags [none], proto UDP (17), length 40)
    10.7.26.96.62066 > 142.251.42.68.33462: [udp sum ok] UDP, length 12
20:17:18.612408 IP (tos 0x0, ttl 10, id 62095, offset 0, flags [none], proto UDP (17), length 40)
    10.7.26.96.62066 > 142.251.42.68.33463: [udp sum ok] UDP, length 12
20:17:18.626409 IP (tos 0x0, ttl 54, id 40135, offset 0, flags [none], proto ICMP (1), length 68)
    74.125.253.166 > 10.7.26.96: ICMP time exceeded in-transit, length 48
        IP (tos 0x80, ttl 1, id 62095, offset 0, flags [none], proto UDP (17), length 40)
    10.7.26.96.62066 > 142.251.42.68.33463: [udp sum ok] UDP, length 12
20:17:18.626676 IP (tos 0x0, ttl 10, id 62096, offset 0, flags [none], proto UDP (17), length 40)
    10.7.26.96.62066 > 142.251.42.68.33464: [udp sum ok] UDP, length 12
20:17:18.651529 IP (tos 0xc0, ttl 54, id 51623, offset 0, flags [none], proto ICMP (1), length 68)
    108.170.234.156 > 10.7.26.96: ICMP time exceeded in-transit, length 48
        IP (tos 0x80, ttl 1, id 62096, offset 0, flags [none], proto UDP (17), length 40)
    10.7.26.96.62066 > 142.251.42.68.33464: [udp sum ok] UDP, length 12
20:17:18.651817 IP (tos 0x0, ttl 11, id 62097, offset 0, flags [none], proto UDP (17), length 40)
    10.7.26.96.62066 > 142.251.42.68.33465: [udp sum ok] UDP, length 12
20:17:18.667664 IP (tos 0xc0, ttl 53, id 48789, offset 0, flags [none], proto ICMP (1), length 68)
    142.251.69.105 > 10.7.26.96: ICMP time exceeded in-transit, length 48
        IP (tos 0x80, ttl 1, id 62097, offset 0, flags [none], proto UDP (17), length 40)
    10.7.26.96.62066 > 142.251.42.68.33465: [udp sum ok] UDP, length 12
20:17:18.667967 IP (tos 0x0, ttl 11, id 62098, offset 0, flags [none], proto UDP (17), length 40)
    10.7.26.96.62066 > 142.251.42.68.33466: [udp sum ok] UDP, length 12
20:17:18.709765 IP (tos 0xc0, ttl 53, id 48797, offset 0, flags [none], proto ICMP (1), length 68)
    142.251.69.105 > 10.7.26.96: ICMP time exceeded in-transit, length 48
        IP (tos 0x80, ttl 1, id 62098, offset 0, flags [none], proto UDP (17), length 40)
    10.7.26.96.62066 > 142.251.42.68.33466: [udp sum ok] UDP, length 12
20:17:18.710060 IP (tos 0x0, ttl 11, id 62099, offset 0, flags [none], proto UDP (17), length 40)
```


10.7.26.96.62066 > 142.251.42.68.33467: [udp sum ok] UDP, length 12
20:17:18.724675 IP (tos 0x0, ttl 246, id 20081, offset 0, flags [none], proto ICMP (1), length 96)
142.250.208.226 > 10.7.26.96: ICMP time exceeded in-transit, length 76
IP (tos 0x80, ttl 1, id 62099, offset 0, flags [none], proto UDP (17), length 40)
10.7.26.96.62066 > 142.251.42.68.33467: [udp sum ok] UDP, length 12
20:17:18.724920 IP (tos 0x0, ttl 12, id 62100, offset 0, flags [none], proto UDP (17), length 40)
10.7.26.96.62066 > 142.251.42.68.33468: [udp sum ok] UDP, length 12
20:17:18.738597 IP (tos 0x0, ttl 245, id 36852, offset 0, flags [none], proto ICMP (1), length 96)
192.178.110.107 > 10.7.26.96: ICMP time exceeded in-transit, length 76
IP (tos 0x80, ttl 1, id 62100, offset 0, flags [none], proto UDP (17), length 40)
10.7.26.96.62066 > 142.251.42.68.33468: [udp sum ok] UDP, length 12
20:17:18.738804 IP (tos 0x0, ttl 12, id 62101, offset 0, flags [none], proto UDP (17), length 40)
10.7.26.96.62066 > 142.251.42.68.33469: [udp sum ok] UDP, length 12
20:17:18.754787 IP (tos 0x80, ttl 115, id 0, offset 0, flags [none], proto ICMP (1), length 56)
142.251.42.68 > 10.7.26.96: ICMP 142.251.42.68 udp port 33469 unreachable, length 36
IP (tos 0x80, ttl 1, id 62101, offset 0, flags [none], proto UDP (17), length 40)
10.7.26.96.62066 > 142.251.42.68.33469: UDP, length 12
20:17:18.755064 IP (tos 0x0, ttl 12, id 62102, offset 0, flags [none], proto UDP (17), length 40)
10.7.26.96.62066 > 142.251.42.68.33470: [udp sum ok] UDP, length 12
20:17:18.768813 IP (tos 0xc0, ttl 54, id 43970, offset 0, flags [none], proto ICMP (1), length 68)
192.178.110.199 > 10.7.26.96: ICMP time exceeded in-transit, length 48
IP (tos 0x80, ttl 1, id 62102, offset 0, flags [none], proto UDP (17), length 40)
10.7.26.96.62066 > 142.251.42.68.33470: [udp sum ok] UDP, length 12
20:17:58.663561 IP (tos 0x0, ttl 64, id 57800, offset 0, flags [none], proto ICMP (1), length 56)
10.7.26.96 > 10.0.136.8: ICMP 10.7.26.96 udp port 34541 unreachable, length 36
IP (tos 0x0, ttl 63, id 37395, offset 0, flags [DF], proto UDP (17), length 118)
10.0.136.8.53 > 10.7.26.96.34541: [no cksum] [|domain]
20:18:11.210654 IP (tos 0x0, ttl 64, id 13315, offset 0, flags [none], proto ICMP (1), length 56)
10.7.26.96 > 216.239.36.178: ICMP 10.7.26.96 udp port 53763 unreachable, length 36
IP (tos 0x0, ttl 56, id 0, offset 0, flags [DF], proto UDP (17), length 1278)
216.239.36.178.443 > 10.7.26.96.53763: [no cksum] [|udp]