

Computer Networks Assignment 1

Manoj (23110025), Eshwar (23110215)

Task-1

Both of our roll numbers ended with a '5', so we took the 0.pcap file, upon filtering out the DNS queries, changing the headers, these were the DNS resolutions received:

Custom Header	Domain	Resolved IP
18041600	bing.com	192.168.1.6
18041601	example.com	192.168.1.7
18041602	amazon.com	192.168.1.8
18041603	yahoo.com	192.168.1.9
18041604	google.com	192.168.1.10
18041605	github.com	192.168.1.6

Task-2

To understand the traceroute utility, we have run the `tracert google.com` in Windows and `traceroute google.com` in Linux. To observe the packets information and the protocols being used, we have used the Wireshark application.

This is a snapshot of the packet information (observed via Wireshark) when `tracert google.com` was run on the Windows terminal.

654	21.950643	107.6.7.215	ICMP	34 [RST] keep-alive ACK seq=2497787333 win=255 len=0
655	21.594643	107.6.7.215	ICMP	60 Echo (ping) request id=0x0001, seq=629/29954, ttl=2 (no response found!)
656	21.598036	172.16.4.7	ICMP	134 Time-to-live exceeded (Time to live exceeded in transit)
657	21.599321	107.6.7.215	ICMP	60 Echo (ping) request id=0x0001, seq=630/30210, ttl=2 (no response found!)
658	21.602715	172.16.4.7	ICMP	134 Time-to-live exceeded (Time to live exceeded in transit)
659	21.606563	107.6.7.215	ICMP	60 Echo (ping) request id=0x0001, seq=631/30466, ttl=2 (no response found!)
660	21.609560	172.16.4.7	ICMP	134 Time-to-live exceeded (Time to live exceeded in transit)
661	21.618794	107.6.7.215	DNS	83 Standard query 0x527e PTR 7.4.16.172.in-addr.arpa
662	21.623145	108.136.7	DNS	145 Standard query response 0x527e No such name PTR 7.4.16.172.in-addr.arpa SQA dns.itgnet.ac.in
663	21.624144	107.6.7.215	NBNS	92 Name query NBSTAT *0x0<00>0x00<00>0x00<00>0x00<00>0x00<00>0x00<00>
664	21.975234	107.6.7.215	TCP	55 58050 -> 443 [ACK] Seq=1 Ack=1 Win=25 Len=1
665	22.029158	104.18.32.47	TCP	66 443-> 58050 [ACK] Seq=1 Ack=2 Win=0 Len=0 SIF=1 SRF=2

This is a snapshot of the packet information (observed via Wireshark) when traceroute google.com was run on the Linux terminal.

No.	Time	Source	Destination	Protocol	Length	Info
97	10.063368980	10.7.20.214	142.251.220.14	UDP	74	35774 → 33443 Len=32
98	10.064018125	10.7.20.214	142.251.220.14	UDP	74	34225 → 33444 Len=32
99	10.064169280	10.7.20.214	142.251.220.14	UDP	74	33617 → 33445 Len=32
100	10.064251665	10.7.20.214	142.251.220.14	UDP	74	37931 → 33446 Len=32
101	10.064292551	10.7.20.214	142.251.220.14	UDP	74	38381 → 33447 Len=32
102	10.064324256	10.7.20.214	142.251.220.14	UDP	74	56333 → 33448 Len=32
103	10.064357276	10.7.20.214	142.251.220.14	UDP	74	60282 → 33449 Len=32
104	10.066481081	10.7.0.5	10.7.20.214	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
105	10.066481342	172.16.4.7	10.7.20.214	ICMP	102	Time-to-live exceeded (Time to live exceeded in transit)
106	10.066481378	172.16.4.7	10.7.20.214	ICMP	102	Time-to-live exceeded (Time to live exceeded in transit)
107	10.066481409	172.16.4.7	10.7.20.214	ICMP	102	Time-to-live exceeded (Time to live exceeded in transit)
108	10.066481449	10.7.0.5	10.7.20.214	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
109	10.066481472	10.7.0.5	10.7.20.214	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
110	10.066481507	10.117.81.253	10.7.20.214	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
111	10.066854366	10.7.20.214	10.0.136.7	DNS	81	Standard query 6xebe6 STD 5 0 7 10 in-addr.arpa

Traceroute information in the Linux terminal:

```
(synjar@synjar)~[/etc]$ traceroute google.com
traceroute to google.com (142.251.220.14), 30 hops max, 60 byte packets
 1  10.7.0.5 (10.7.0.5)  3.545 ms  3.442 ms  3.402 ms
 2  172.16.4.7 (172.16.4.7)  3.368 ms  3.336 ms  3.266 ms
 3  14.139.98.1 (14.139.98.1)  6.971 ms  5.762 ms  6.895 ms
 4  10.117.81.253 (10.117.81.253)  3.114 ms  3.199 ms  3.038 ms
 5  10.154.8.137 (10.154.8.137)  11.659 ms  11.616 ms  11.584 ms
 6  10.255.239.170 (10.255.239.170)  11.551 ms  10.499 ms  11.331 ms
 7  10.152.7.214 (10.152.7.214)  10.100 ms  10.956 ms  10.922 ms
 8  * * *
 9  * * *
10  142.250.208.148 (142.250.208.148)  17.142 ms  142.250.228.50 (142.250.228.50)  59.648 ms  142.250.238.198 (142.250.238.198)  13.924 ms
11  142.251.64.13 (142.251.64.13)  13.890 ms  13.856 ms  192.178.110.106 (192.178.110.106)  12.937 ms
12  pnbomb-ay-in-f14.1e100.net (142.251.220.14)  13.858 ms  192.178.110.249 (192.178.110.249)  22.420 ms  142.251.77.69 (142.251.77.69)  12.776 ms
```

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

Ans- We can observe that by default, Windows sends **ICMP Echo Request** packets to the destination and receives **ICMP replies** from routers or the target. Whereas Linux sends **UDP** packets and expects **ICMP replies**

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

Ans- Some hops in a traceroute may show ******* because the router does not send a reply, it can be due to the router being overloaded and packets being dropped, or the router is configured not to send ICMP replies (time exceeded, echo replies) due to a firewall/security settings

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

Ans- The **TTL** (time to live) changes every 3 probes, starting with a TTL of 1 and increasing by 1 every 3 probes. We can also observe that the **source port** changes every probe, and the **destination port** is also being increased by 1 every UDP probe

UDP	74	53924 → 33434	Len=32
UDP	74	49799 → 33435	Len=32
UDP	74	41820 → 33436	Len=32
UDP	74	44147 → 33437	Len=32
UDP	74	36732 → 33438	Len=32
UDP	74	40443 → 33439	Len=32
UDP	74	58352 → 33440	Len=32
UDP	74	58851 → 33441	Len=32
UDP	74	58344 → 33442	Len=32
UDP	74	35774 → 33443	Len=32
UDP	74	34225 → 33444	Len=32
UDP	74	33617 → 33445	Len=32

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

In Windows, the destination sends an ICMP Echo reply, whereas the intermediate routers send Time-to-live exceeded (Time to live exceeded in transit), which can be observed from this:

1388	81.564841	192.178.86.239	10.7.62.215	ICMP	134 Time-to-live exceeded (Time to live exceeded in transit)
1389	81.568317	10.7.62.215	142.250.70.110	ICMP	106 Echo (ping) request id=0x0001, seq=654/36354, ttl=10 (no response found!)
1390	81.582662	192.178.86.239	10.7.62.215	ICMP	134 Time-to-live exceeded (Time to live exceeded in transit)
1391	81.589767	10.7.62.215	142.250.70.110	ICMP	106 Echo (ping) request id=0x0001, seq=655/36610, ttl=10 (no response found!)
1392	81.604367	192.178.86.239	10.7.62.215	ICMP	134 Time-to-live exceeded (Time to live exceeded in transit)
1417	87.160438	10.7.62.215	142.250.70.110	ICMP	106 Echo (ping) request id=0x0001, seq=656/36866, ttl=11 (reply in 1418)
1418	87.176140	142.250.70.110	10.7.62.215	ICMP	106 Echo (ping) reply id=0x0001, seq=656/36866, ttl=115 (request in 1417)
1419	87.177533	10.7.62.215	142.250.70.110	ICMP	106 Echo (ping) request id=0x0001, seq=657/37122, ttl=11 (reply in 1420)
1420	87.193000	142.250.70.110	10.7.62.215	ICMP	106 Echo (ping) reply id=0x0001, seq=657/37122, ttl=115 (request in 1419)
1421	87.194071	10.7.62.215	142.250.70.110	ICMP	106 Echo (ping) request id=0x0001, seq=658/37378, ttl=11 (reply in 1422)
1422	87.207518	142.250.70.110	10.7.62.215	ICMP	106 Echo (ping) reply id=0x0001, seq=658/37378, ttl=115 (request in 1421)

In Linux, Intermediate routers: When a probe's TTL expires, the router sends back ICMP Time Exceeded (Time to live exceeded in transit) messages. Final destination: The probe actually reaches the host, but because Linux traceroute sends UDP packets to high, unused port numbers, the host replies with ICMP Port Unreachable messages.

182	10.114918585	10.7.20.214	142.251.220.14	UDP	74 38955 → 33475 Len=32
183	10.114956948	10.7.20.214	142.251.220.14	UDP	74 36037 → 33476 Len=32
184	10.115593260	142.251.220.14	10.7.20.214	ICMP	70 Destination unreachable (Port unreachable)
185	10.115593340	142.251.64.15	10.7.20.214	ICMP	102 Time-to-live exceeded (Time to live exceeded in transit)
186	10.124192427	142.251.220.14	10.7.20.214	ICMP	70 Destination unreachable (Port unreachable)
187	10.124192695	142.251.220.14	10.7.20.214	ICMP	70 Destination unreachable (Port unreachable)
188	10.124192763	192.178.110.249	10.7.20.214	ICMP	102 Time-to-live exceeded (Time to live exceeded in transit)
189	10.124192791	142.251.220.14	10.7.20.214	ICMP	70 Destination unreachable (Port unreachable)
190	10.126198202	142.251.220.14	10.7.20.214	ICMP	70 Destination unreachable (Port unreachable)
191	10.127264754	142.251.220.14	10.7.20.214	ICMP	70 Destination unreachable (Port unreachable)
192	10.127265074	142.251.220.14	10.7.20.214	ICMP	70 Destination unreachable (Port unreachable)

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

Ans- For Windows tracert, there won't be any change as the tracert uses ICMP echo requests and those are allowed, whereas for Linux, the UDP probes will not reach the routers, so TTL messages will not be sent to the source, which means we cannot track the route via Linux traceroute if UDP traffic is blocked