Working of Traceroute using Wireshark

June 6, 2018 By Raj Chandel

In this Post, we are going to discuss working with traceroute using UDP/ICMP/TCP packets with the help of Wireshark.

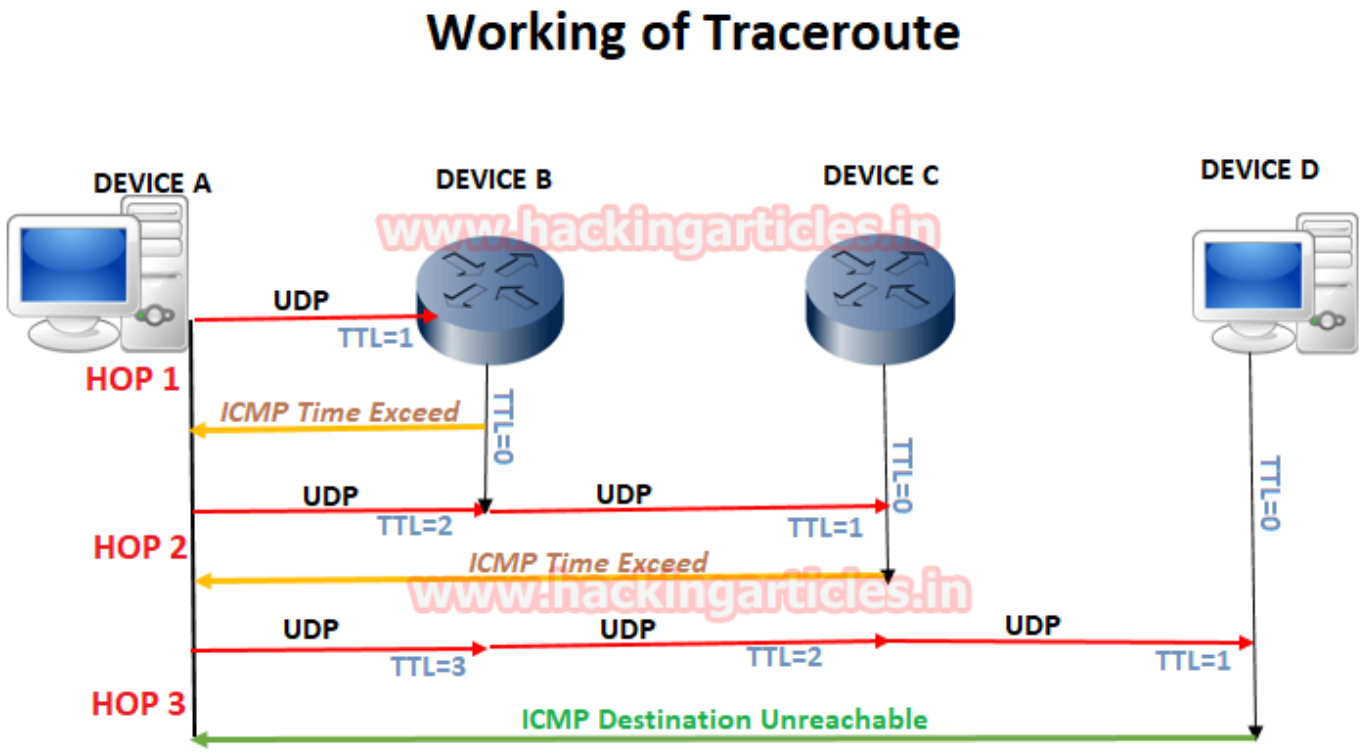
Traceroute or Tracert: It is a CUI based computer network diagnostic tools used in UNIX and Windows-like system respectively. It traces the path of a packet from the source machine to an Internet host such as Google.com by calculating the average time taken each hop. Traceroute sends a UDP packet to the destination by taking benefit of ICMP's messages. It uses the ICMP error-reporting messages –Destination Unreachable and Time exceeded.

TTL: The time-to-live value, also known as the hop limit, is a mechanism that limits the lifespan or lifetime of data in a computer or network.

Hop: A hop is one portion of the path between source and destination. Data packets pass through bridges, routers, and gateways as they travel between source and destination. On the internet, before the data reaches its final destination, it goes through several routers and a hop occurs when an incoming packet is forwarded to the next router.

The asterisk (*): Denotes probe timeout which means that the router at that hop doesn't respond to the packet received from the source used for the traceroute due to firewall filter.

Working of Traceroute



Read the below steps:

- Traceroute sends a UDP packet with a TTL = 1 from the source to destination.
- When the first router receives the UDP packet it reduces the TTL value by 1 (1-1=0) then drop the packet and sends an ICMP message “Time exceeded” to the source. Thus Traceroute makes a list of the router’s address and the time taken for the round-trip.

The TTL time exceeded ICMP message is sent after the TTL value of a UDP packet gets zero. In typical condition, a network doesn’t have such a diameter that lead the TTL=0. This could be possible when there is a routing loop. In this case, as the packet is sent back and forth between the looping points, the TTL keeps getting decrement until it becomes zero. And at last, the source receives ICMP error message sent by the router.

- Again source device sends two more packets, in the same way, to get an average value of the round-trip time and again TTL gets zero when it reaches to the 2nd router and response through ICMP error message time exceeds.
- Traceroute keeps on doing this, and record the IP address and name of every router until the UDP packets reach to the destination address. Once it reaches at the destination address, Time exceeded ICMP message is NOT sent back to the source.
- Since Traceroute uses the random port for sending UDP packets as result destination machine will drop the packet and send a new ICMP error message-Destination Unreachable to the source which indicates the UDP packets has reached to the destination address.

Tracert with Wireshark

As discussed above tracert is CLI utility for windows system to trace the path of a packet from source to destination. So herewith help of the following command, we can observe the path of the packet which travels to reach Google DNS.

Syntax: tracert [options] Host IP

```
tracert 8.8.8.8
```

or

```
tracert -d 8.8.8.8
```

Traceroute generates a list of each hop by entering IP of routers that traversed between source and destination and average round-trip time. As a result **hop 22 denotes** entry of destination i.e. Google DNS.

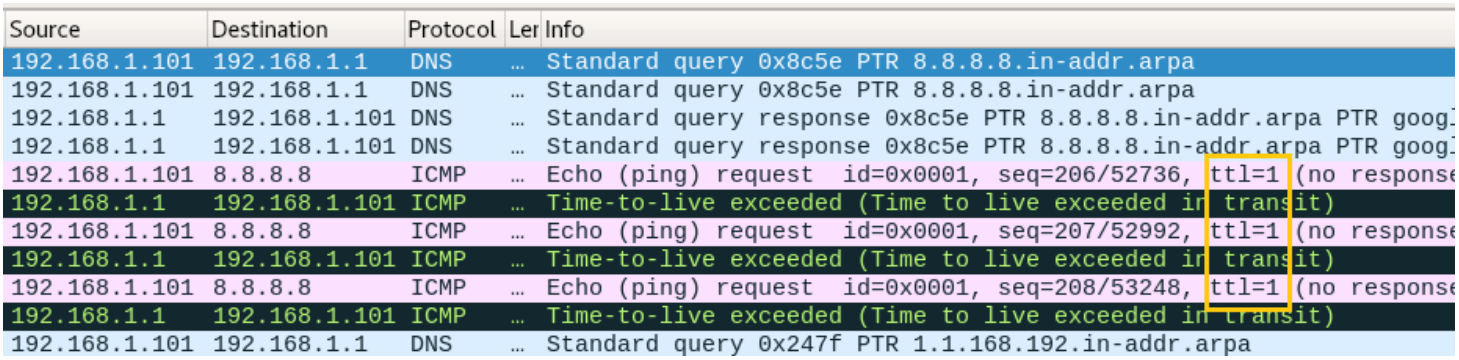
In order to notice the activity of traceroute, we have turned on Wireshark in the background.

Note: Result of tracert can vary each time for hop count but does not go beyond 30 hops because it is the maximum hop limit.

```
C:\Users\singh>tracert 8.8.8.8

Tracing route to google-public-dns-a.google.com [8.8.8.8]
over a maximum of 30 hops:

  1    <1 ms    <1 ms    <1 ms  192.168.1.1
  2    13 ms    20 ms    15 ms  120.57.48.1
  3    14 ms    13 ms    13 ms  triband-del-59.180.212.202.bol.net.in [59.180.212.202]
  4    14 ms    14 ms    14 ms  triband-del-59.180.210.150.bol.net.in [59.180.210.150]
  5    14 ms    13 ms    13 ms  125.20.37.21
  6    14 ms    16 ms    14 ms  182.79.181.230
  7    60 ms    59 ms    60 ms  182.79.190.57
  8    67 ms   101 ms    92 ms  182.79.198.162
  9    63 ms    63 ms    62 ms  72.14.197.166
 10    55 ms    55 ms    54 ms  108.170.253.121
 11   122 ms    89 ms    88 ms  216.239.63.213
 12    87 ms    86 ms    86 ms  216.239.47.109
 13     *        *        *     Request timed out.
 14     *        *        *     Request timed out.
 15     *        *        *     Request timed out.
 16     *        *        *     Request timed out.
 17     *        *        *     Request timed out.
 18     *        *        *     Request timed out.
 19     *        *        *     Request timed out.
 20     *        *        *     Request timed out.
 21     *        *        *     Request timed out.
 22    88 ms    88 ms    87 ms  google-public-dns-a.google.com [8.8.8.8]

Trace complete.
```

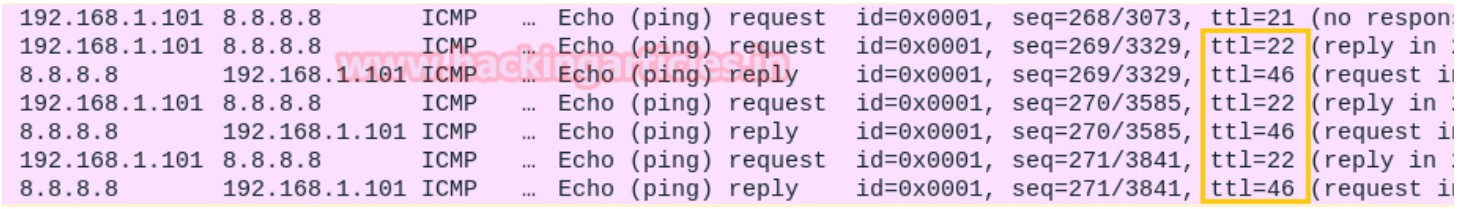
At Wireshark we notice the following points:

- ICMP echo request packet is used instead of UDP to send DNS query.
- The packet first goes from source 192.168.1.101 to first router 192.168.1.1 having ICMP echo request packet with TTL=1
- The router will drop that packet and send ICMP Time Exceeded error message to the source.
- All this happens 3 times before the source machine sends next packet by incrementing TTL value by 1 i.e. TTL=2.



Source	Destination	Protocol	Ler	Info
192.168.1.101	192.168.1.1	DNS	…	Standard query 0x8c5e PTR 8.8.8.8.in-addr.arpa
192.168.1.101	192.168.1.1	DNS	…	Standard query 0x8c5e PTR 8.8.8.8.in-addr.arpa
192.168.1.1	192.168.1.101	DNS	…	Standard query response 0x8c5e PTR 8.8.8.8.in-addr.arpa PTR goog
192.168.1.1	192.168.1.101	DNS	…	Standard query response 0x8c5e PTR 8.8.8.8.in-addr.arpa PTR goog
192.168.1.101	8.8.8.8	ICMP	…	Echo (ping) request id=0x0001, seq=206/52736, ttl=1 (no response
192.168.1.1	192.168.1.101	ICMP	…	Time-to-live exceeded (Time to live exceeded in transit)
192.168.1.101	8.8.8.8	ICMP	…	Echo (ping) request id=0x0001, seq=207/52992, ttl=1 (no response
192.168.1.1	192.168.1.101	ICMP	…	Time-to-live exceeded (Time to live exceeded in transit)
192.168.1.101	8.8.8.8	ICMP	…	Echo (ping) request id=0x0001, seq=208/53248, ttl=1 (no response
192.168.1.1	192.168.1.101	ICMP	…	Time-to-live exceeded (Time to live exceeded in transit)
192.168.1.101	192.168.1.1	DNS	…	Standard query 0x247f PTR 1.1.168.192.in-addr.arpa

Form this image we can observe ICMP echo reply message is sent from 8.8.8.8 (destination) to 192.168.1.101 (source) for TTL 22.



192.168.1.101	8.8.8.8	ICMP	…	Echo (ping) request	id=0x0001, seq=268/3073,	ttl=21 (no respon
192.168.1.101	8.8.8.8	ICMP	…	Echo (ping) request	id=0x0001, seq=269/3329,	ttl=22 (reply in
8.8.8.8	192.168.1.101	ICMP	…	Echo (ping) reply	id=0x0001, seq=269/3329,	ttl=46 (request i
192.168.1.101	8.8.8.8	ICMP	…	Echo (ping) request	id=0x0001, seq=270/3585,	ttl=22 (reply in
8.8.8.8	192.168.1.101	ICMP	…	Echo (ping) reply	id=0x0001, seq=270/3585,	ttl=46 (request i
192.168.1.101	8.8.8.8	ICMP	…	Echo (ping) request	id=0x0001, seq=271/3841,	ttl=22 (reply in
8.8.8.8	192.168.1.101	ICMP	…	Echo (ping) reply	id=0x0001, seq=271/3841,	ttl=46 (request i

Traceroute with Wireshark (via UDP packets)

As discussed above traceroute in utility for Unix -like the system to trace the path of a packet from source to destination. So here with the help of the following command, we can observe the path of packet travels to reach Google DNS.

Syntax: traceroute [options] Host IP

```
traceroute 8.8.8.8
```

```
root@kali:~# traceroute 8.8.8.8
traceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 60 byte packets
 1  _gateway (192.168.1.1)  0.911 ms  1.590 ms  1.547 ms
 2  120.57.48.1 (120.57.48.1)  15.927 ms  22.933 ms  27.992 ms
 3  triband-del-59.180.212.202.bol.net.in (59.180.212.202)  21.901 ms  24.841 ms
 27.157 ms
 4  triband-del-59.180.210.202.bol.net.in (59.180.210.202)  29.744 ms  32.287 ms
 34.415 ms
 5  219.65.112.105.static-delhi.vsnl.net.in (219.65.112.105)  82.049 ms  84.255 m
s triband-del-59.180.211.226.bol.net.in (59.180.211.226)  41.617 ms
 6  219.65.112.105.static-delhi.vsnl.net.in (219.65.112.105)  87.818 ms  61.802 m
s  57.288 ms
 7  172.29.250.33 (172.29.250.33)  71.198 ms 172.23.183.134 (172.23.183.134)  66.
030 ms 172.29.250.33 (172.29.250.33)  75.236 ms
 8  182.79.190.57 (182.79.190.57)  69.426 ms 182.79.198.59 (182.79.198.59)  157.2
21 ms  158.060 ms
 9  182.79.239.199 (182.79.239.199)  78.136 ms 182.79.177.241 (182.79.177.241)  8
9.109 ms 182.79.189.227 (182.79.189.227)  78.776 ms
10  72.14.197.166 (72.14.197.166)  89.442 ms  91.175 ms 108.170.253.121 (108.170.
253.121)  55.042 ms
11  209.85.249.195 (209.85.249.195)  98.052 ms 209.85.248.251 (209.85.248.251)  1
01.366 ms 209.85.249.195 (209.85.249.195)  98.769 ms
12  216.239.51.57 (216.239.51.57)  112.781 ms 216.239.48.209 (216.239.48.209)  10
0.491 ms 216.239.51.57 (216.239.51.57)  126.290 ms
13  * * *
14  * * *
15  * * *
16  * * *
17  * * *
18  * * *
19  * * *
20  * * *
21  google-public-dns-a.google.com (8.8.8.8)  121.727 ms  103.554 ms  111.293 ms
```

Traceroute generates a list of each hop by entering IP of routers that comes between source and destination and average round-trip time. As a result **hop 21 denotes** entry of destination i.e. Google DNS.

In order to notice the activity of traceroute, we have turned on Wireshark in the background.

Note: Result of traceroute can vary each time for hop count but does not go beyond 30 hops because it is maximum hop limit.

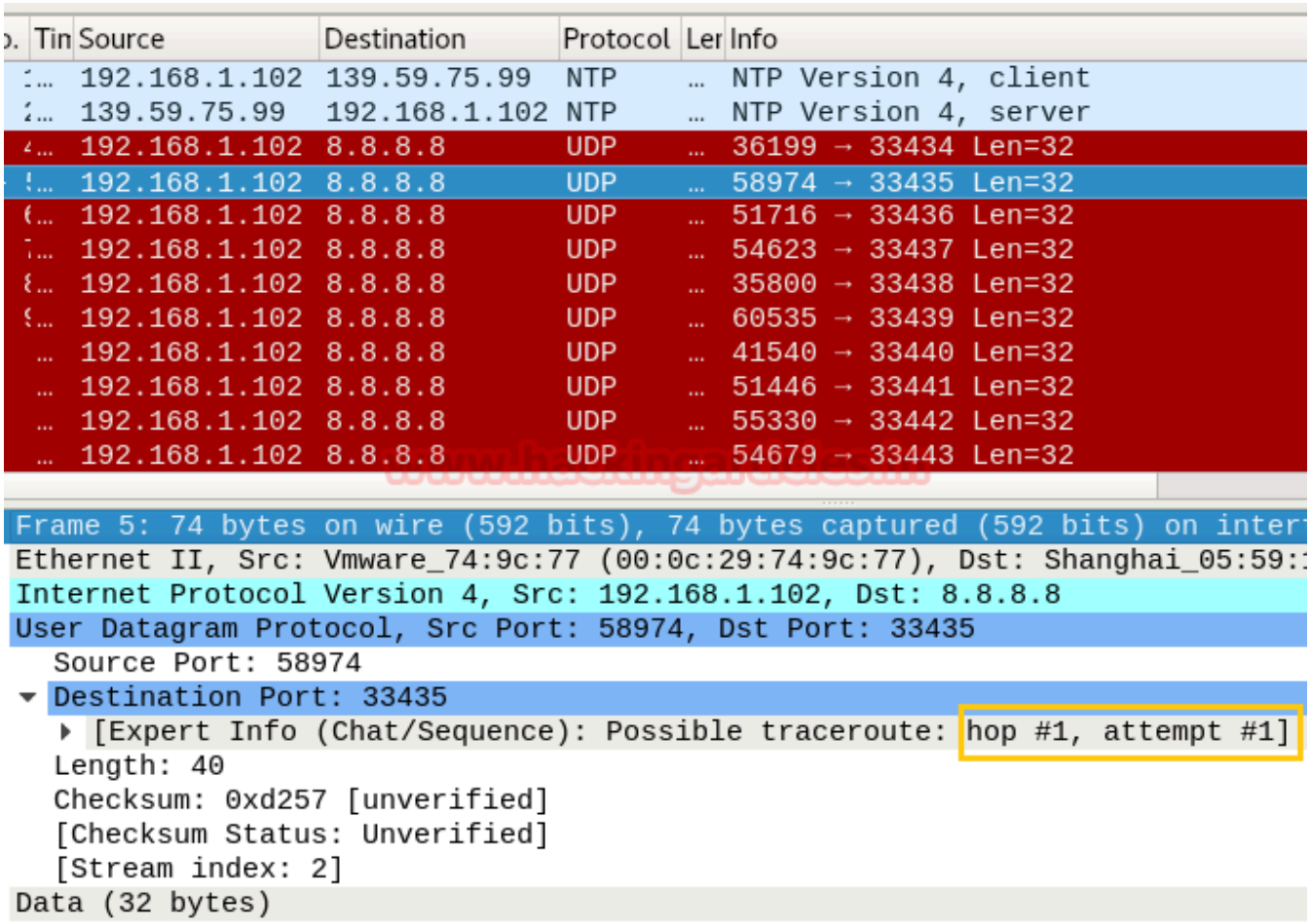
At Wireshark we notice the following points:

- UDP packet is used to send DNS query with help of 32-bit payload.
- The packet first goes from source 192.168.1.101 to first router 192.168.1.1 having ICMP request packet with TTL=1
- The router will drop that packet and send ICMP Time Exceeded error message to the source.
- All this happens 3 times before the source sent next packet with increment TTL value by 1 i.e. TTL=2.

Source	Destination	Protocol	Ler	Info
192.168.1.102	139.59.75.99	NTP	…	NTP Version 4, client
139.59.75.99	192.168.1.102	NTP	…	NTP Version 4, server
192.168.1.102	8.8.8.8	UDP	…	36199 → 33434 Len=32
192.168.1.102	8.8.8.8	UDP	…	58974 → 33435 Len=32
192.168.1.102	8.8.8.8	UDP	…	51716 → 33436 Len=32
192.168.1.102	8.8.8.8	UDP	…	54623 → 33437 Len=32
192.168.1.102	8.8.8.8	UDP	…	35800 → 33438 Len=32
192.168.1.102	8.8.8.8	UDP	…	60535 → 33439 Len=32
192.168.1.102	8.8.8.8	UDP	…	41540 → 33440 Len=32
192.168.1.102	8.8.8.8	UDP	…	51446 → 33441 Len=32
192.168.1.102	8.8.8.8	UDP	…	55330 → 33442 Len=32
192.168.1.102	8.8.8.8	UDP	…	54679 → 33443 Len=32
192.168.1.102	8.8.8.8	UDP	…	34975 → 33444 Len=32
192.168.1.102	8.8.8.8	UDP	…	46706 → 33445 Len=32
192.168.1.102	8.8.8.8	UDP	…	56440 → 33446 Len=32
192.168.1.102	8.8.8.8	UDP	…	46824 → 33447 Len=32
192.168.1.102	8.8.8.8	UDP	…	37066 → 33448 Len=32
192.168.1.102	8.8.8.8	UDP	…	36065 → 33449 Len=32
192.168.1.1	192.168.1.102	ICMP	…	Time-to-live exceeded (Time to live exceeded in tr
192.168.1.102	192.168.1.1	DNS	…	Standard query 0x3481 PTR 1.1.168.192.in-addr.arpa
192.168.1.1	192.168.1.102	ICMP	…	Time-to-live exceeded (Time to live exceeded in tr
192.168.1.1	192.168.1.102	ICMP	…	Time-to-live exceeded (Time to live exceeded in tr
120.57.48.1	192.168.1.102	ICMP	…	Time-to-live exceeded (Time to live exceeded in tr
59.180.212.2…	192.168.1.102	ICMP	…	Time-to-live exceeded (Time to live exceeded in tr
120.57.48.1	192.168.1.102	ICMP	…	Time-to-live exceeded (Time to live exceeded in tr

In tracert we have seen that each TTL value between source to the first router proceeds 3 times, similarly, same technique is followed by UDP. To demonstrate this we have explored UDP packets 5,6,7 and 8th continuously.

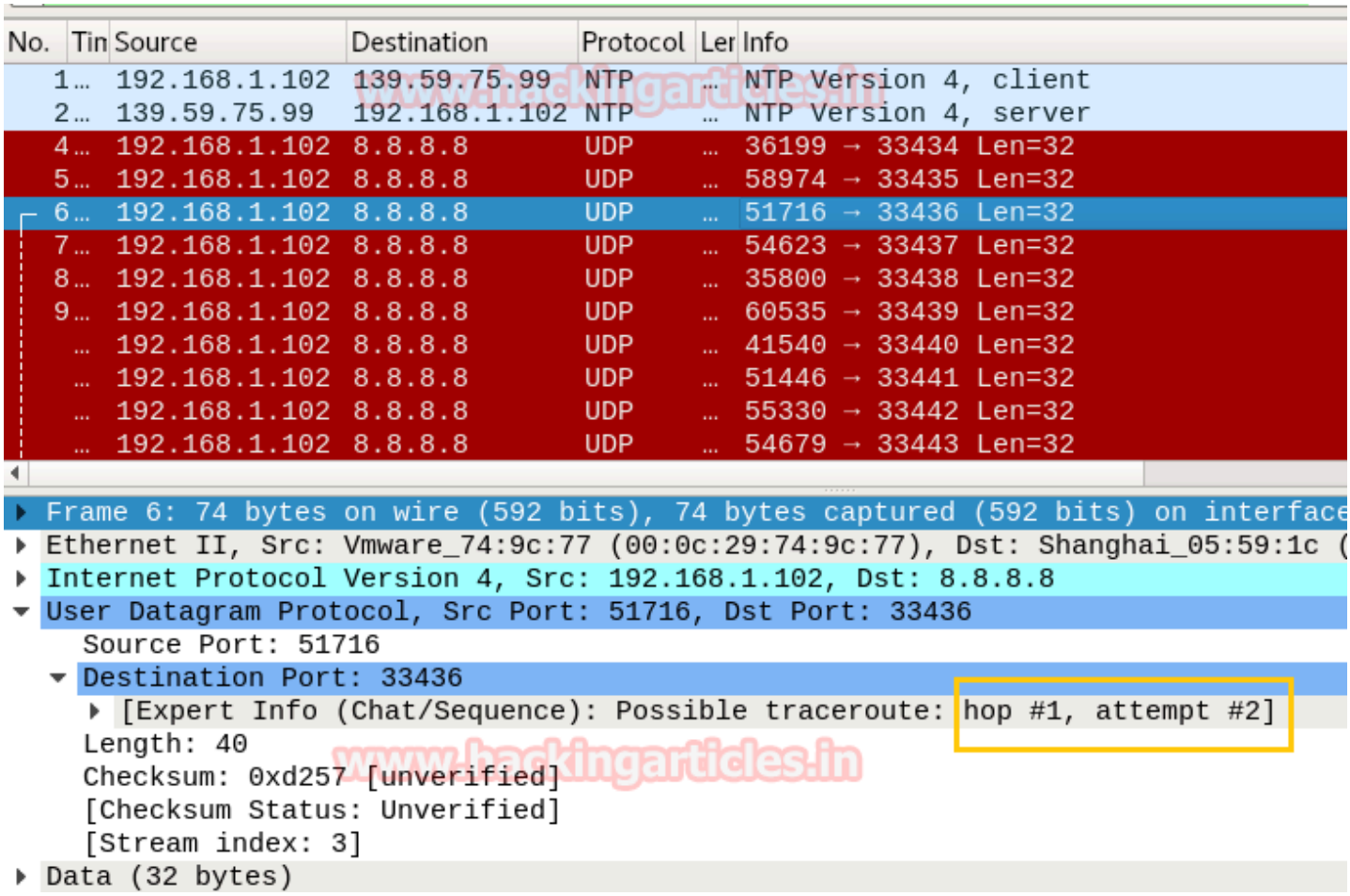
In the 5th packet, we observe the UDP packet sent by source (192.168.1.102) to destination 8.8.8.8 on port 33435 and count as **Hop #1, attempt #1**.



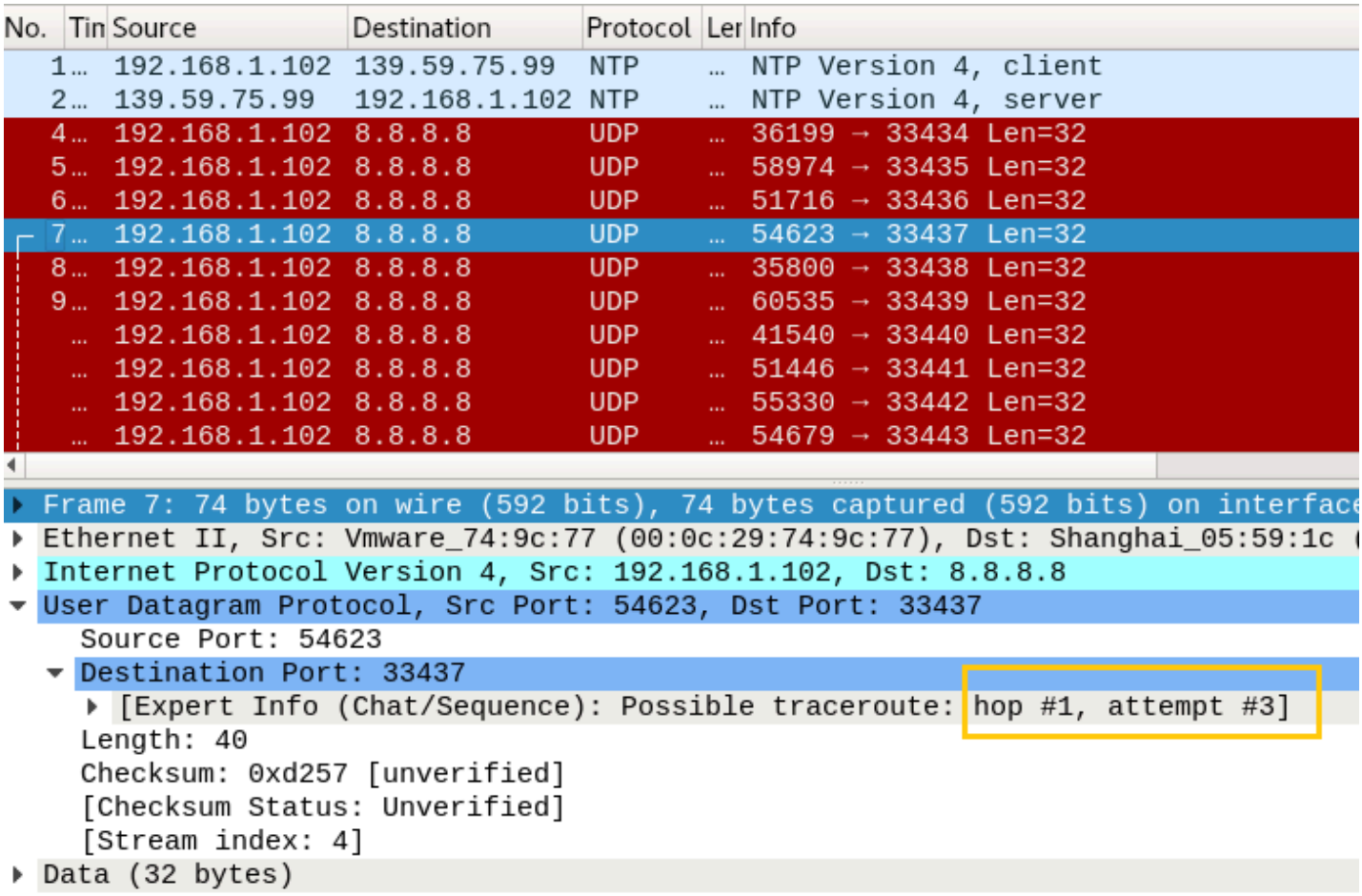
No.	Time	Source	Destination	Protocol	Length	Info
3	…	192.168.1.102	139.59.75.99	NTP	…	NTP Version 4, client
4	…	139.59.75.99	192.168.1.102	NTP	…	NTP Version 4, server
5	…	192.168.1.102	8.8.8.8	UDP	…	36199 → 33434 Len=32
6	…	192.168.1.102	8.8.8.8	UDP	…	58974 → 33435 Len=32
7	…	192.168.1.102	8.8.8.8	UDP	…	51716 → 33436 Len=32
8	…	192.168.1.102	8.8.8.8	UDP	…	54623 → 33437 Len=32
9	…	192.168.1.102	8.8.8.8	UDP	…	35800 → 33438 Len=32
10	…	192.168.1.102	8.8.8.8	UDP	…	60535 → 33439 Len=32
11	…	192.168.1.102	8.8.8.8	UDP	…	41540 → 33440 Len=32
12	…	192.168.1.102	8.8.8.8	UDP	…	51446 → 33441 Len=32
13	…	192.168.1.102	8.8.8.8	UDP	…	55330 → 33442 Len=32
14	…	192.168.1.102	8.8.8.8	UDP	…	54679 → 33443 Len=32


```
Frame 5: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on inter
Ethernet II, Src: Vmware_74:9c:77 (00:0c:29:74:9c:77), Dst: Shanghai_05:59:
Internet Protocol Version 4, Src: 192.168.1.102, Dst: 8.8.8.8
User Datagram Protocol, Src Port: 58974, Dst Port: 33435
    Source Port: 58974
  ▾ Destination Port: 33435
    ▸ [Expert Info (Chat/Sequence): Possible traceroute: hop #1, attempt #1]
    Length: 40
    Checksum: 0xd257 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 2]
Data (32 bytes)
```

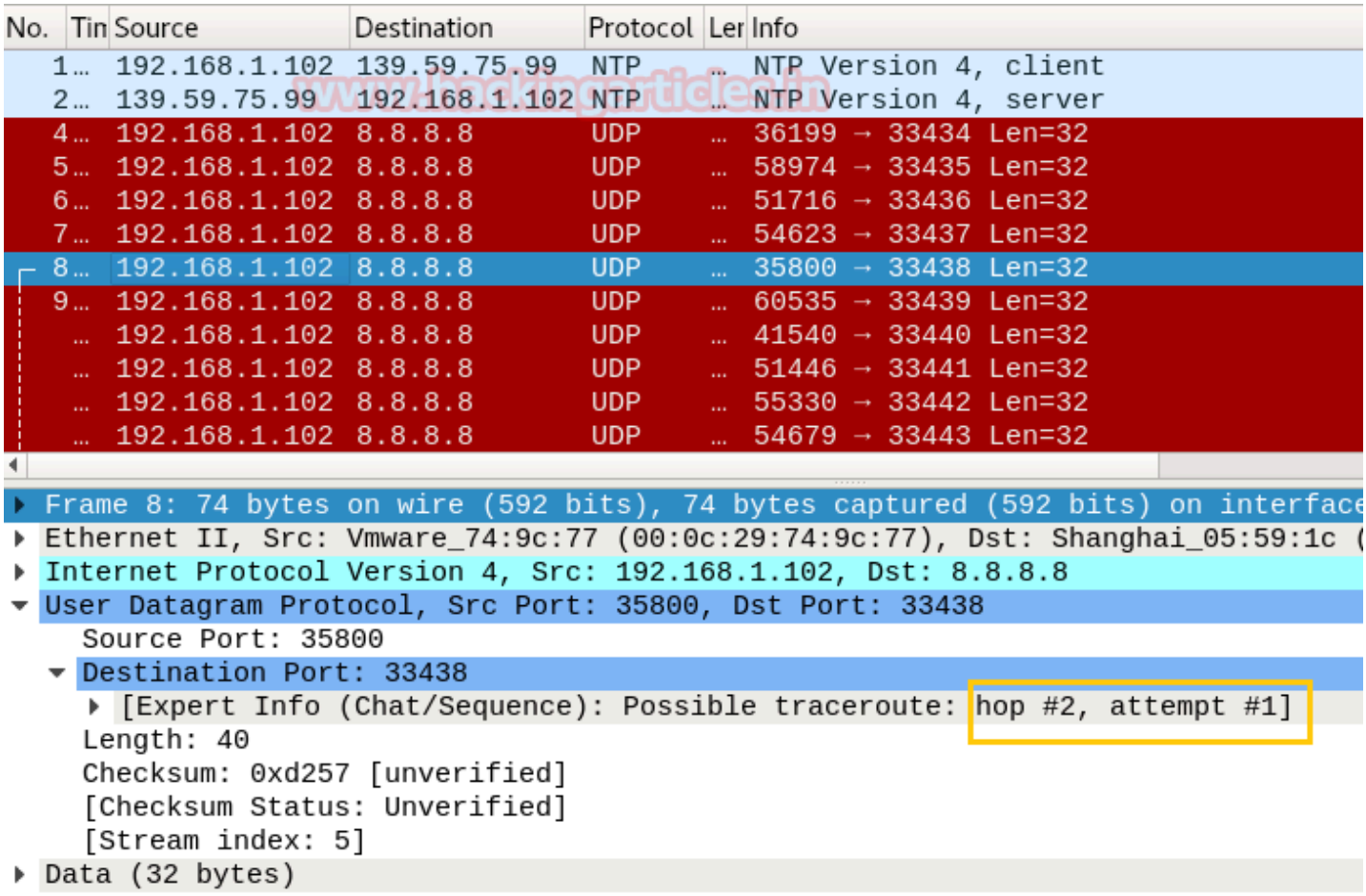
In the 6th packet, we observe the UDP packet sent by source (192.168.1.102) to destination 8.8.8.8 on port 33436 and count as **Hop #1, attempt #2**.



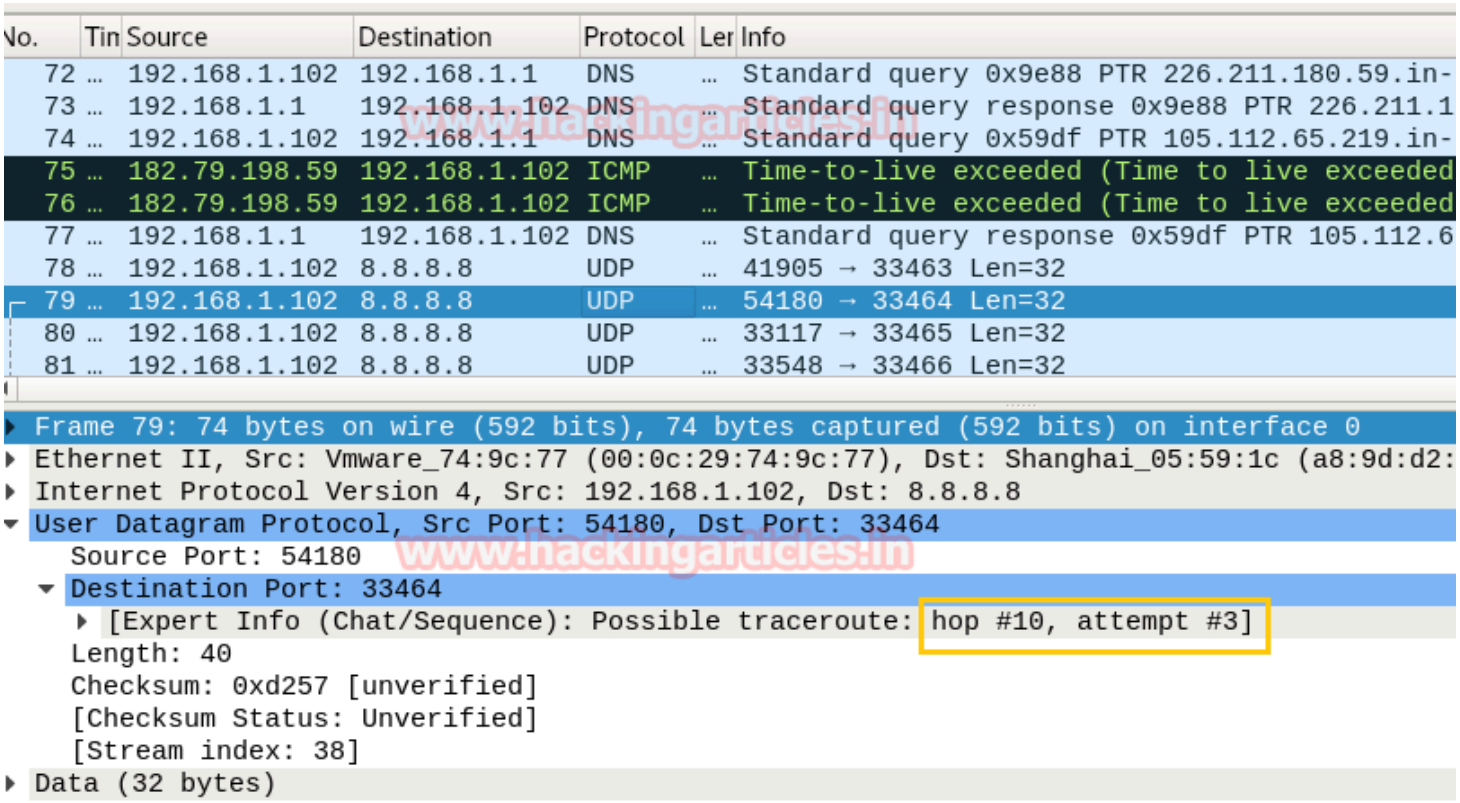
Similarly, in the 7th packet, we observe the UDP packet sent by source (192.168.1.102) to destination 8.8.8.8 on port 33437 and count as **Hop #1, attempt #3**.



In the 8th packet, we observe the UDP packet sent by source (192.168.1.102) to destination 8.8.8.8 on port 33436 and count as **Hop #2, attempt #1** and repeat so on process till reaches the destination.



In packet 79th we observe that the last hop captured was hop #10 attempt #3 when the UDP packet sent by source (192.168.1.102) to destination 8.8.8.8 on port 33464 and Time exceeded ICMP message is NOT sent back to the source after this.



As a result, at last, source received ICMP message Destination Port Unreachable which means our UDP packet reaches on the destination address.

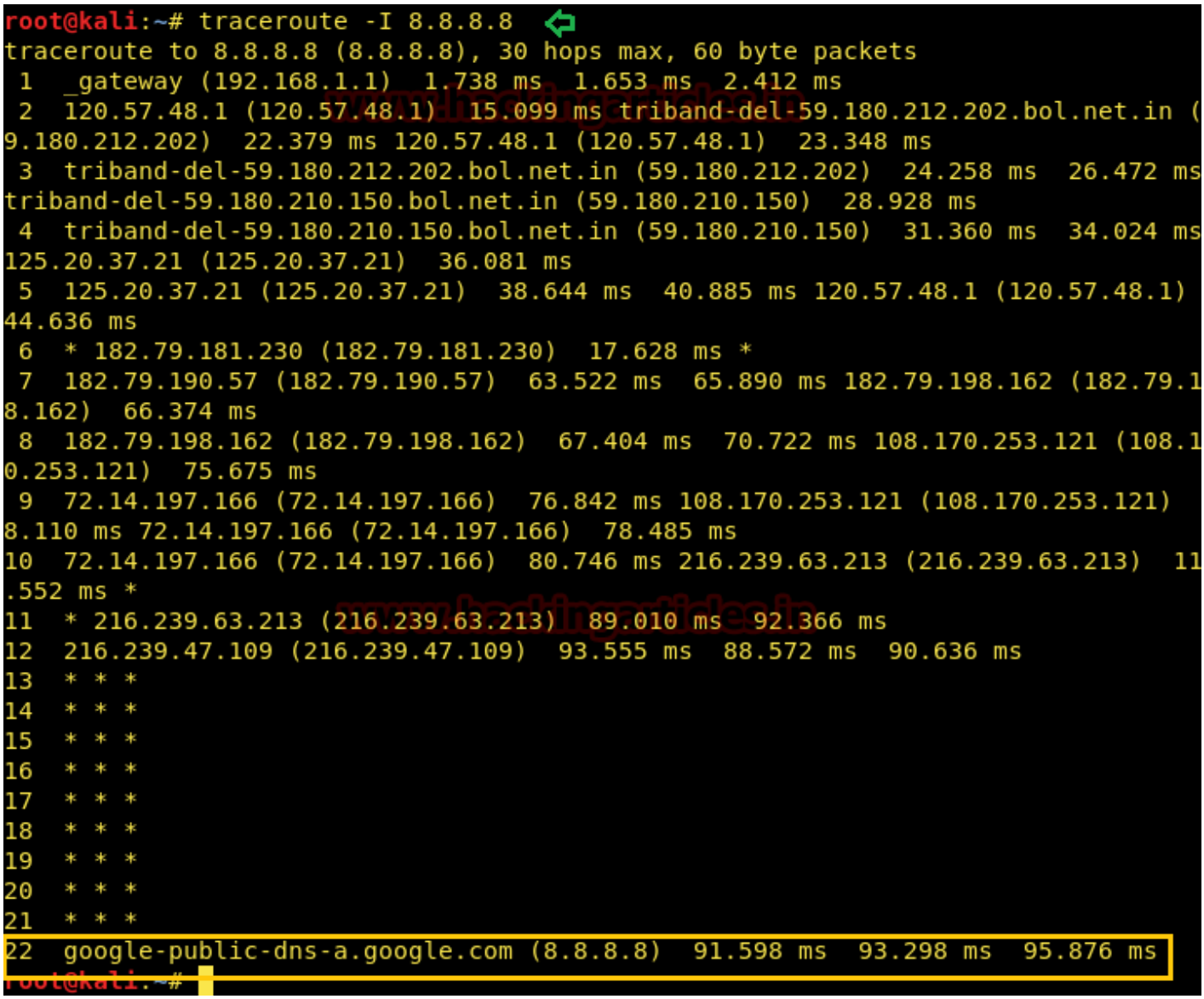
At last from given below image we observed the following:

- Source sent DNS query to the router for DNS lookup 8.8.8.8
- Router sent a response to source as the answer of DNS Name Google-Public-DNS-google.com

Traceroute with Wireshark (via ICMP packets)

As you know by default traceroute use UDP packet but with help of **-I option** you can make it work as tracert which uses ICMP request packet.

```
traceroute -I 8.8.8.8
```



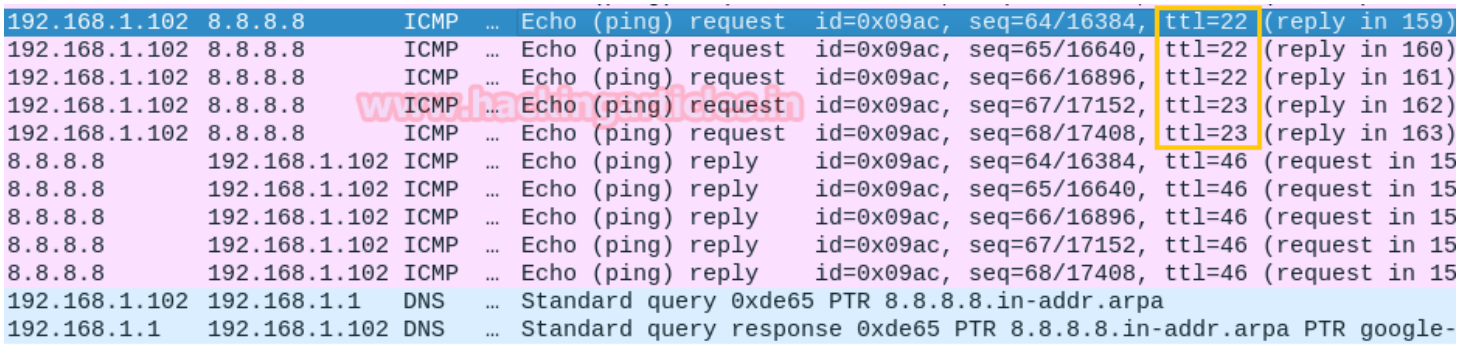
It generates a list of each hop by entering IP of routers that comes between source and destination and average round-trip time. As a result **hop 22 denotes** entry of destination i.e. Google DNS. In order to notice the activity of traceroute, we have turned on Wireshark in the background.

At Wireshark we notice the following points:

First ICMP echo request packet will be sent to the first router with TTL 1 and it will send back an ICMP error message time exceed which follow the same technique as explained above in tracert with Wireshark.

At last from given below image we observed the following:

- ICMP echo reply message is sent from 8.8.8.8 (destination) to 192.168.1.101 (source) for TTL 22.
- Source sent DNS query to the router for DNS lookup 8.8.8.8
- Router sent the response to source as the answer of DNS Name Google-Public-DNS-google.com

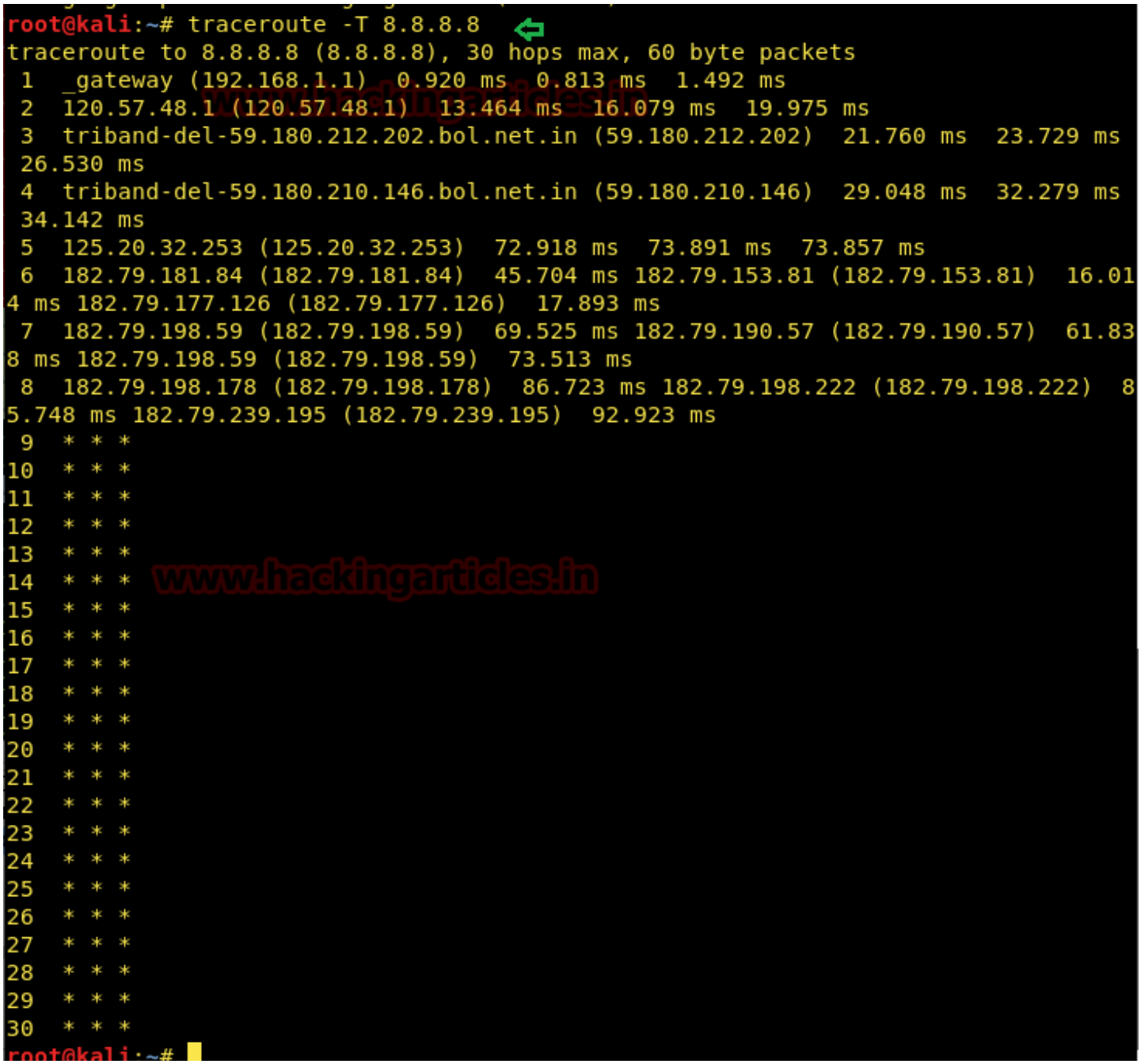


Traceroute with Wireshark (via TCP packets)

As you know by default traceroute use UDP packet with the use of ICMP error message for generating a response but with the help of **-T option**, you can use TCP packet, which uses syn request packet via port 80. It is most useful in diagnosing connection issues to a specific service eg. Web server.

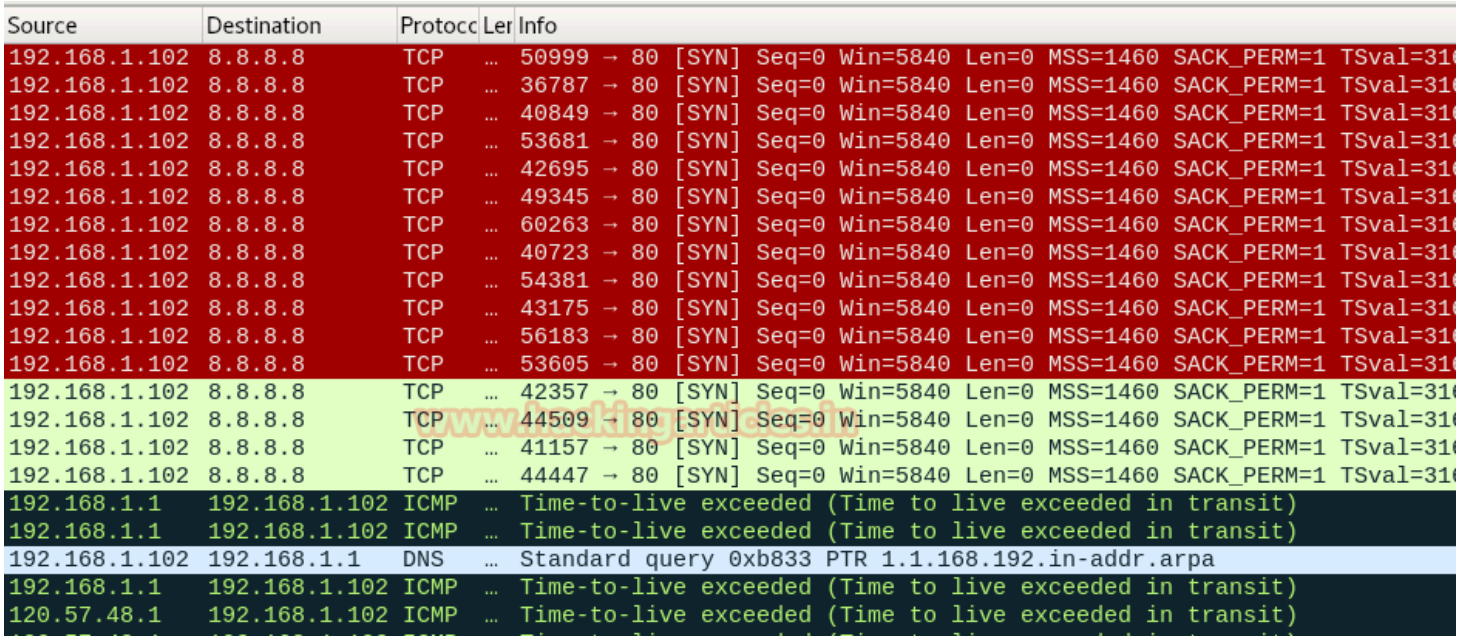
```
tcptraceroute - 8.8.8.8
or
traceroute -T 8.8.8.8
```

As we know the maximum hop limit is 30 and but here till 30th hop we didn't find desirable output. TCP traceroute basically follow TCP half communication and waits for the sys-ack packet from destination till the last hop.



In order to notice the activity of tcp traceroute, we have turned on Wireshark in the background where we noticed that it works same as UDP but here the syn packets are used to send the requests to the destination. Tcptraceroute does not measure the time it takes to complete the three-way handshake because that never occurs in such a situation. It only measures the time from the initial SYN to the SYN/ACK.

Since Wireshark also didn't notice any syn-ack packet from destination to source, therefore, Tcptraceroute didn't edit destination response in its record list this is due to because it is useful while diagnosing web server.



Source	Destination	Protocc	Ler	Info
192.168.1.102	8.8.8.8	TCP	…	50999 → 80 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM=1 TSval=31
192.168.1.102	8.8.8.8	TCP	…	36787 → 80 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM=1 TSval=31
192.168.1.102	8.8.8.8	TCP	…	40849 → 80 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM=1 TSval=31
192.168.1.102	8.8.8.8	TCP	…	53681 → 80 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM=1 TSval=31
192.168.1.102	8.8.8.8	TCP	…	42695 → 80 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM=1 TSval=31
192.168.1.102	8.8.8.8	TCP	…	49345 → 80 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM=1 TSval=31
192.168.1.102	8.8.8.8	TCP	…	60263 → 80 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM=1 TSval=31
192.168.1.102	8.8.8.8	TCP	…	40723 → 80 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM=1 TSval=31
192.168.1.102	8.8.8.8	TCP	…	54381 → 80 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM=1 TSval=31
192.168.1.102	8.8.8.8	TCP	…	43175 → 80 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM=1 TSval=31
192.168.1.102	8.8.8.8	TCP	…	56183 → 80 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM=1 TSval=31
192.168.1.102	8.8.8.8	TCP	…	53605 → 80 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM=1 TSval=31
192.168.1.102	8.8.8.8	TCP	…	42357 → 80 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM=1 TSval=31
192.168.1.102	8.8.8.8	TCP	…	44509 → 80 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM=1 TSval=31
192.168.1.102	8.8.8.8	TCP	…	41157 → 80 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM=1 TSval=31
192.168.1.102	8.8.8.8	TCP	…	44447 → 80 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM=1 TSval=31
192.168.1.1	192.168.1.102	ICMP	…	Time-to-live exceeded (Time to live exceeded in transit)
192.168.1.1	192.168.1.102	ICMP	…	Time-to-live exceeded (Time to live exceeded in transit)
192.168.1.102	192.168.1.1	DNS	…	Standard query 0xb833 PTR 1.1.168.192.in-addr.arpa
192.168.1.1	192.168.1.102	ICMP	…	Time-to-live exceeded (Time to live exceeded in transit)
120.57.48.1	192.168.1.102	ICMP	…	Time-to-live exceeded (Time to live exceeded in transit)

Therefore let's check the path of Google.com and notice the behavior of tcptraceroute. And you compare both result and behaviour of TCP in case of Google DNS server and Google web server.

```
tcptraceroute google.com
```

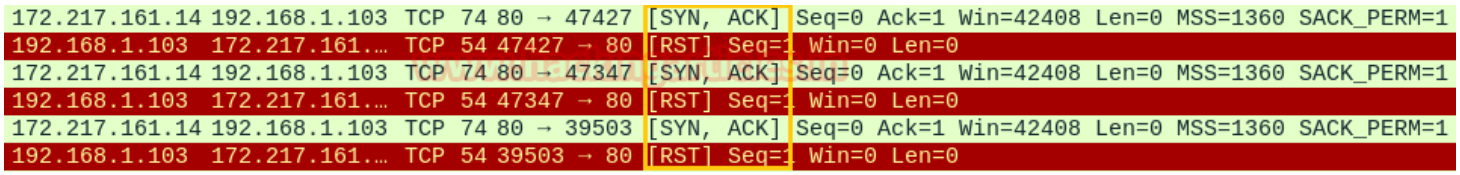
Here we can clearly observe the response of destination machine through SYN, ACK and a complete entry recorded by traceroute.

```
root@kali:~# tcptraceroute google.com
Running:
        traceroute -T -O info google.com
traceroute to google.com (172.217.161.14), 30 hops max, 60 byte packets
 1  _gateway (192.168.1.1)  1.095 ms  1.030 ms  1.802 ms
 2  120.57.48.1 (120.57.48.1)  14.690 ms  19.333 ms  19.909 ms
 3  triband-del-59.180.212.6.bol.net.in (59.180.212.6)  22.104 ms  24.763 ms  26.701 m
s
 4  triband-del-59.180.210.202.bol.net.in (59.180.210.202)  29.071 ms  31.773 ms  34.4
42 ms
 5  219.65.112.105.static-delhi.vsnl.net.in (219.65.112.105)  77.178 ms  79.562 ms  81
.931 ms
 6  182.79.176.59 (182.79.176.59)  43.954 ms 182.79.177.126 (182.79.177.126)  14.571 m
s 182.79.205.145 (182.79.205.145)  16.960 ms
 7  182.79.198.33 (182.79.198.33)  79.827 ms  80.485 ms  81.242 ms
 8  182.79.177.69 (182.79.177.69)  66.840 ms  75.431 ms 182.79.239.197 (182.79.239.197
)  73.252 ms
 9  72.14.211.198 (72.14.211.198)  68.435 ms  69.471 ms  73.917 ms
10  74.125.242.147 (74.125.242.147)  87.522 ms 108.170.253.121 (108.170.253.121)  80.2
00 ms 74.125.242.146 (74.125.242.146)  61.516 ms
11  66.249.94.90 (66.249.94.90)  64.369 ms 108.170.253.122 (108.170.253.122)  53.509 m
s 74.125.242.147 (74.125.242.147)  68.532 ms
12  216.239.41.152 (216.239.41.152)  69.898 ms 209.85.247.252 (209.85.247.252)  59.942
 ms  65.398 ms
13  108.170.251.97 (108.170.251.97)  55.566 ms  62.012 ms 209.85.246.165 (209.85.246.1
65)  71.861 ms
14  108.170.251.97 (108.170.251.97)  82.571 ms 64.233.174.71 (64.233.174.71)  66.277 m
s  65.393 ms
15  108.170.251.113 (108.170.251.113)  74.233 ms del03s10-in-f14.1e100.net (172.217.16
1.14) <syn,ack>  77.222 ms  72.135 ms
```

It is as similar as above, the source sent the TCP-SYN packet to the destination machine on port 80 and received ICMP error message from the router for a time exceeded and repeat the process till it receives ACK_SYN from the destination.

Source	Destination	Prot	Len	Info
192.168.1.103	192.168.1.1	DNS	70	Standard query 0xc47e A google.com
192.168.1.103	192.168.1.1	DNS	70	Standard query 0x1888 AAAA google.com
192.168.1.1	192.168.1.103	DNS	86	Standard query response 0xc47e A google.com A 172.217.161.1
192.168.1.1	192.168.1.103	DNS	98	Standard query response 0x1888 AAAA google.com AAAA 2404:68
192.168.1.103	172.217.161.…	TCP	74	60051 → 80 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM=1
192.168.1.103	172.217.161.…	TCP	74	33049 → 80 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM=1
192.168.1.103	172.217.161.…	TCP	74	42891 → 80 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM=1
192.168.1.103	172.217.161.…	TCP	74	37591 → 80 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM=1
192.168.1.103	172.217.161.…	TCP	74	40119 → 80 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM=1
192.168.1.103	172.217.161.…	TCP	74	34125 → 80 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM=1
192.168.1.103	172.217.161.…	TCP	74	59607 → 80 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM=1
192.168.1.103	172.217.161.…	TCP	74	55253 → 80 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM=1
192.168.1.103	172.217.161.…	TCP	74	53943 → 80 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM=1
192.168.1.103	172.217.161.…	TCP	74	41675 → 80 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM=1
192.168.1.103	172.217.161.…	TCP	74	35679 → 80 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM=1
192.168.1.103	172.217.161.…	TCP	74	40945 → 80 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM=1
192.168.1.103	172.217.161.…	TCP	74	36241 → 80 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM=1
192.168.1.103	172.217.161.…	TCP	74	45125 → 80 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM=1
192.168.1.103	172.217.161.…	TCP	74	57317 → 80 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM=1
192.168.1.103	172.217.161.…	TCP	74	35325 → 80 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM=1
192.168.1.1	192.168.1.103	IC…	70	Time-to-live exceeded (Time to live exceeded in transit)

Here we can observe ACK-SYN packet from the destination (172.168.161.14) is sent to source (192.1681.103) from port 80 and source again sent RST packet to the destination via port 80.



At last from given below image we observed the following:

- Source sent DNS query to the router for DNS lookup 172.161.217.14
- Router sent the response to source as the answer of DNS Name del03s10-in-f14.1e100.net

This entry will get recorded by traceroute in its record list.