

Esercitazione traceroute

Riferimenti

- <https://manpages.ubuntu.com/manpages/noble/man1/traceroute.db.1.html>

Installazione

Potrebbe essere necessario installare *traceroute*. Molte distribuzioni Linux suggeriscono il pacchetto da installare quando si prova a usare un comando non presente. Su Ubuntu 24.04, occorre installare il pacchetto *inetutils-traceroute*

```
$ sudo apt install inetutils-traceroute
```

Esercizio 1

Avendo avviato la cattura dei pacchetti con Wireshark (da sospendere dopo l'esecuzione del comando *traceroute*), eseguire il seguente comando per determinare un percorso verso www.google.it

```
$ traceroute www.google.it
```

```
traceroute to www.google.it (142.250.180.131), 64 hops max
```

```
1  172.21.240.1  1.073ms  0.828ms  0.771ms
2  172.29.32.1   17.562ms  17.118ms  17.535ms
3  160.80.255.1  18.685ms  18.410ms  17.608ms
4  10.0.253.54   17.365ms  17.392ms  18.615ms
5  10.0.253.82   17.397ms  17.466ms  *
6  160.80.176.1  18.794ms  17.575ms  18.587ms
7  193.206.131.45 19.472ms  18.727ms  18.843ms
8  * * *
9  185.191.180.57 51.127ms  35.175ms  97.332ms
10 142.250.164.230 26.662ms  26.678ms  27.324ms
11 * * *
```

```

12    108.170.233.56    36.082ms    34.650ms    34.861ms
13    108.170.255.210    36.168ms    35.605ms    36.153ms
14    *    *    *
15    142.250.180.131    31.233ms    30.428ms    31.369ms

```

Il comando *traceroute* stampa dapprima la destinazione e il numero massimo di hop (in questo caso 64).

Le righe successive ci mostrano i router attraversati fino all'host di destinazione, che si trova nell'ultima riga.

Per ciascuna riga, viene fornito l'indirizzo IP del router (o host di destinazione) e i valori di RTT misurati attraverso i 3 probe. Un asterisco indica che non è stata ricevuta una risposta: può darsi che sia andata persa, magari bloccata da qualche firewall, oppure mai inviata (non lo possiamo sapere). Potrebbe succedere che su una riga troviamo due o più indirizzi IP, se i probe hanno seguito percorsi differenti. Inoltre, come si può vedere gli RTT non sono crescenti attraverso i vari hop, contrariamente a quello che ci si aspetterebbe, ma ciò si può spiegare considerando che il ritardo include una variabile quale il ritardo di coda: può darsi che un probe che ha compiuto più salti ha un RTT minore di un probe che ne ha compiuti di meno, perché ha subito meno ritardo di coda.

Nell'esempio, il percorso di rete dall'host sorgente a quello di destinazione consta di 15 salti, attraverso 14 router.

Se stampiamo l'exit status del comando *traceroute*, eseguendo subito dopo `echo "$?"` otteniamo 0, perché è stato trovato un percorso verso la destinazione indicata.

In wireshark, notiamo una serie di pacchetti contenenti segmenti UDP inviati dall'host a www.google.it (il cui indirizzo fornito da *traceroute* è 142.250.180.131).

Apply a display filter ... <Ctrl-/>						
No.	Time	Source	Destination	Protocol	Length	Info
9	7.367734001	172.21.242.109	142.250.180.131	UDP	51	59645 → 33434 Len=9
10	7.368586403	172.21.240.1	172.21.242.109	ICMP	79	Time-to-live exceeded (Time to live exceeded in transit)
11	7.369013204	172.21.242.109	142.250.180.131	UDP	51	59645 → 33434 Len=9
12	7.369665405	172.21.240.1	172.21.242.109	ICMP	79	Time-to-live exceeded (Time to live exceeded in transit)
13	7.370035606	172.21.242.109	142.250.180.131	UDP	51	59645 → 33434 Len=9
14	7.370628007	172.21.240.1	172.21.242.109	ICMP	79	Time-to-live exceeded (Time to live exceeded in transit)
15	7.371013908	172.21.242.109	142.250.180.131	UDP	51	59645 → 33435 Len=9
16	7.388307443	172.29.32.1	172.21.242.109	ICMP	79	Time-to-live exceeded (Time to live exceeded in transit)
17	7.388797844	172.21.242.109	142.250.180.131	UDP	51	59645 → 33435 Len=9
18	7.405616178	172.29.32.1	172.21.242.109	ICMP	79	Time-to-live exceeded (Time to live exceeded in transit)
19	7.406128179	172.21.242.109	142.250.180.131	UDP	51	59645 → 33435 Len=9
20	7.423390514	172.29.32.1	172.21.242.109	ICMP	79	Time-to-live exceeded (Time to live exceeded in transit)
21	7.423901715	172.21.242.109	142.250.180.131	UDP	51	59645 → 33436 Len=9
22	7.442313952	160.80.255.1	172.21.242.109	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
23	7.442807953	172.21.242.109	142.250.180.131	UDP	51	59645 → 33436 Len=9

L'indirizzo mittente coincide l'indirizzo associato alla scheda *eth0*, come restituito da questo comando:

```
$ ip addr
```

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state
UNKNOWN group default qlen 1000
```

```
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
```

```
    inet 127.0.0.1/8 scope host lo
```

```
        valid_lft forever preferred_lft forever
```

```
    inet 10.255.255.254/32 brd 10.255.255.254 scope global lo
```

```
        valid_lft forever preferred_lft forever
```

```
    inet6 ::1/128 scope host
```

```
        valid_lft forever preferred_lft forever
```

```
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state
UP group default qlen 1000
```

```
    link/ether 00:15:5d:08:4c:ed brd ff:ff:ff:ff:ff:ff
```

```
    inet 172.21.242.109/20 brd 172.21.255.255 scope global eth0
```

```
        valid_lft forever preferred_lft forever
```

```
    inet6 fe80::215:5dff:fe08:4ced/64 scope link
```

```
        valid_lft forever preferred_lft forever
```

A ciascun pacchetto UDP, segue un messaggio ICMP TTL Exceeded, il cui mittente sono, in successione, i vari router che troviamo nella stampa di traceroute e il destinatario è l'host.

Se clicchiamo sui pacchetti contenenti i segmenti UDP, nel pannello dei dettagli vediamo nell'intestazione IP valori crescenti del campo di intestazione TTL: 1, 2, 3, ... (per gruppi di 3 pacchetti).

Il pacchetto con TTL = n, può fare n salti, finché arrivato all'n-esimo router lungo il percorso il TTL si azzerà e questo manda il messaggio ICMP *TTL Exceeded* all'host sorgente.

Nell'esempio, i pacchetti con TTL = 15 riescono ad arrivare all'host di destinazione, che invia un messaggio ICMP Destination unreachable (Port unreachable), perché non c'è nessuna socket in ascolto sul numero di porta "improbabile" indicato nei pacchetti UDP (nell'esempio 33434).

```

▶ Frame 9: 51 bytes on wire (408 bits), 51 bytes captured (408 bits) on interface eth0, id 0
▶ Ethernet II, Src: Microsoft 08:4c:ed (00:15:5d:08:4c:ed), Dst: Microsoft_c4:22:ed (00:15:5d:c4:22:ed)
▼ Internet Protocol Version 4, Src: 172.21.242.109, Dst: 142.250.180.131
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
    ▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
      Total Length: 37
      Identification: 0x2aa2 (10914)
    ▶ 010. .... = Flags: 0x2, Don't fragment
      ...0 0000 0000 0000 = Fragment Offset: 0
    ▶ Time to Live: 1
      Protocol: UDP (17)
      Header Checksum: 0x6d25 [validation disabled]
      [Header checksum status: Unverified]
      Source Address: 172.21.242.109
      Destination Address: 142.250.180.131
    ▼ User Datagram Protocol, Src Port: 59645, Dst Port: 33434
      Source Port: 59645
      Destination Port: 33434
      Length: 17
      Checksum: 0xe223 [unverified]
      [Checksum Status: Unverified]
      [Stream index: 3]
      ▶ [Timestamps]
      UDP payload (9 bytes)
    ▶ Data (9 bytes)

```

0000	00 15 5d c4 22 ed 00 15 5d 08 4c ed 08 00 45 00	..]..L..E..
0010	00 25 2a a2 40 00 01 11 6d 25 ac 15 f2 6d 8e fa	..%*.@...m%...m..
0020	b4 83 e8 fd 82 9a 00 11 e2 23 53 55 50 45 52 4d[...#SUPERM
0030	41 4e 00	AN

Nonostante traceroute ci abbia indicato un percorso di lunghezza 15, è possibile "pingare" www.google.it (142.250.180.131) con un limite di hop inferiore, in questo caso 13 (ovviamente ciò dipende da dove si sta eseguendo il test).

```
$ ping -t 13 -c 1 142.250.180.131
```

```
PING 142.250.180.131 (142.250.180.131) 56(84) bytes of data.
```

```
64 bytes from 142.250.180.131: icmp_seq=1 ttl=113 time=31.0 ms
```

```
--- 142.250.180.131 ping statistics ---
```

```
1 packets transmitted, 1 received, 0% packet loss, time 0ms
```

```
rtt min/avg/max/mdev = 30.982/30.982/30.982/0.000 ms
```

Come mai??? Vediamo l'esercizio successivo.

Esercizio 2

L'opzione -M permette di indicare il protocollo da usare per inviare i probe: udp (default) oppure icmp (impiegando gli stessi messaggi ICMP *Echo Request* e *Echo Reply* impiegati dal comando *ping*).

```
$ traceroute -M icmp www.google.it
```

```
traceroute to www.google.it (142.250.180.131), 64 hops max
```

```
1  172.21.240.1  0.337ms  0.244ms  0.189ms
2  172.29.32.1   16.601ms  16.087ms  16.208ms
3  160.80.255.1  16.558ms  17.507ms  21.339ms
4  10.0.253.54   17.245ms  17.624ms  18.851ms
5  10.0.253.82   18.627ms  17.854ms  17.664ms
6  * * *
7  193.206.131.45 19.236ms  19.576ms  19.084ms
8  * * *
9  185.191.180.57 35.016ms  35.387ms  34.808ms
10 142.250.174.46 34.615ms  34.791ms  34.847ms
11 72.14.238.234 34.702ms  35.239ms  36.167ms
12 142.250.211.29 30.824ms  30.976ms  30.956ms
13 142.250.180.131 31.260ms  30.556ms  30.974ms
```

Notiamo che traceroute ha trovato un percorso differente, segno che i pacchetti ICMP sono stati instradati in maniera diversa. Sii può notare che la lunghezza del percorso è 13, spiegando perché ping (che usa gli stessi messaggi) era in grado di raggiungere l'host di destinazione con un limite di hop pari a 13.

Invece, con un limite di hop pari a 12, come si aspettava, non riusciamo a raggiungere l'host:

```
$ ping -t 12 -c 1 142.250.180.131 ; echo "exit status: $?"
```

```
PING 142.250.180.131 (142.250.180.131) 56(84) bytes of data.
```

```
From 142.250.211.29 icmp_seq=1 Time to live exceeded
```

```
--- 142.250.180.131 ping statistics ---
```

```
1 packets transmitted, 0 received, +1 errors, 100% packet loss,
time 0ms
```

```
exit status: 1
```

Esercizio 3

Con l'opzione `-m` si può fornire un limite di hop:

```
$ traceroute -m 2 www.google.it
```

```
traceroute to www.google.it (142.250.180.131), 2 hops max
```

```
1  172.21.240.1  0.461ms  0.407ms  0.466ms
2  172.29.32.1  16.340ms  17.155ms  18.089ms
```

Si noti come l'ultima riga non sia relativa all'host di destinazione, segno che non è stato trovato un percorso non più lungo di 2 salti.

Infatti, stampando l'exit status ci viene mostrato 1:

```
$ echo "$?"
```

```
1
```