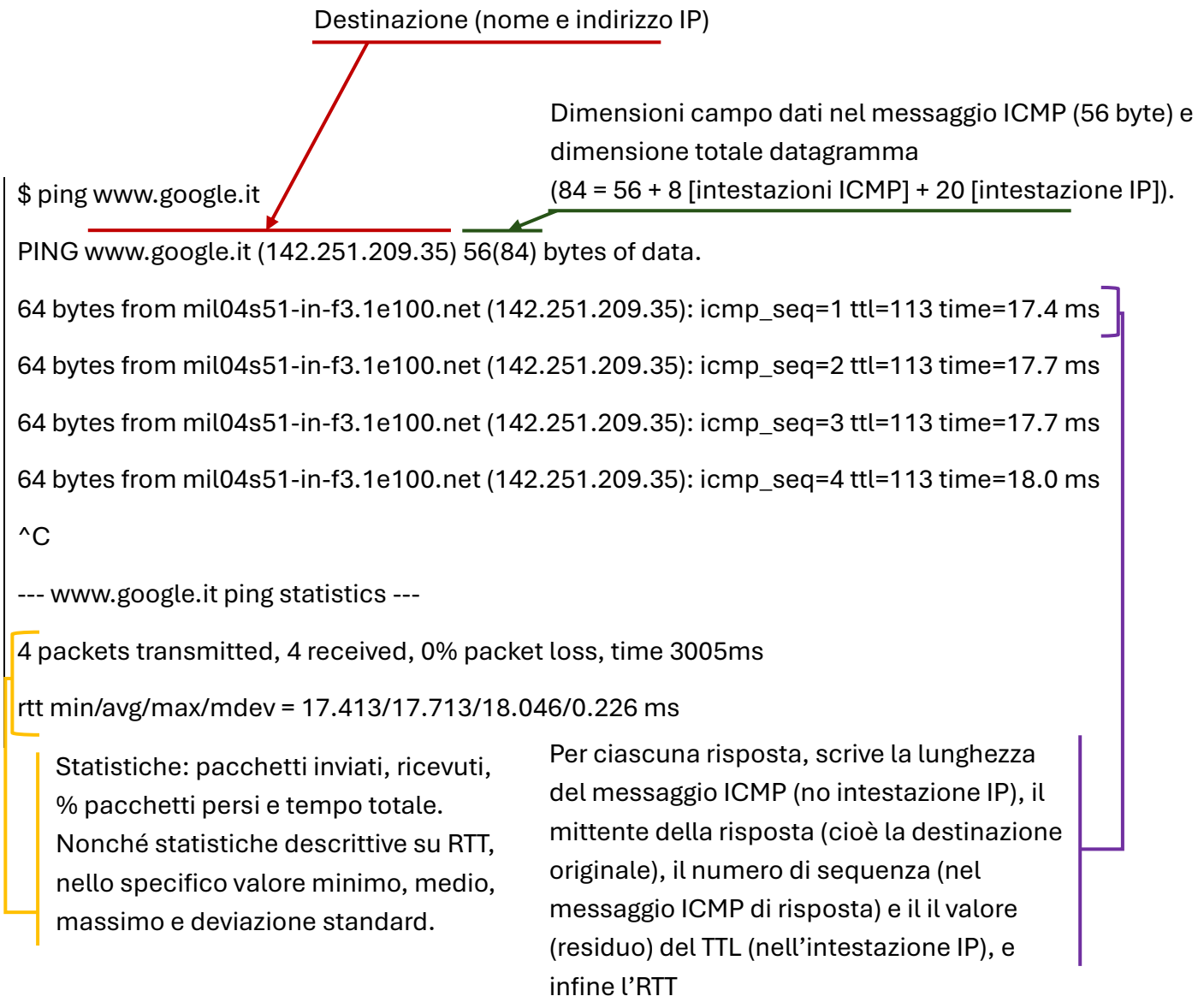
Esercitazione ping

Riferimenti:

- https://manpages.ubuntu.com/manpages/noble/man8/ping.8.html

Esercizio 1

Avviate la cattura dei pacchetti su *eth0* con Wireshark (ricordarsi di interrompere la cattura dei pacchetti). Quindi, eseguire il comando ping, terminandone l'esecuzione dopo qualche istante inviando al processo il segnale *SIGINT* (per esempio, premendo la sequenza di tasti *CTRL + C*). Il comando avrò inviato nel frattempo un messaggio *ICMP Echo Request* al secondo. La destinazione dovrebbe aver risposto a ciascuno di questi con un messaggio *ICMP Echo Reply*.



Il comando ping può essere usato per testare la raggiungibilità di una destinazione e stimare l'RTT. La documentazione ci dice questo per l'*exit status* del comando.

> Se il ping non riceve alcun pacchetto di risposta, esce con il codice 1. Se vengono specificati un numero di pacchetti e una e la scadenza sono entrambi specificati, e al momento della scadenza sono stati ricevuti meno pacchetti del conteggio scadenza, uscirà anch'esso con il codice 1. In caso di altri errori, esce con il codice 2. Altrimenti esce con il codice 0. In questo modo è possibile utilizzare il codice di uscita per vedere se un host è vivo o meno.

Vedere negli esercizi successivi la discusse sulle opzioni *-c* e *-w*.

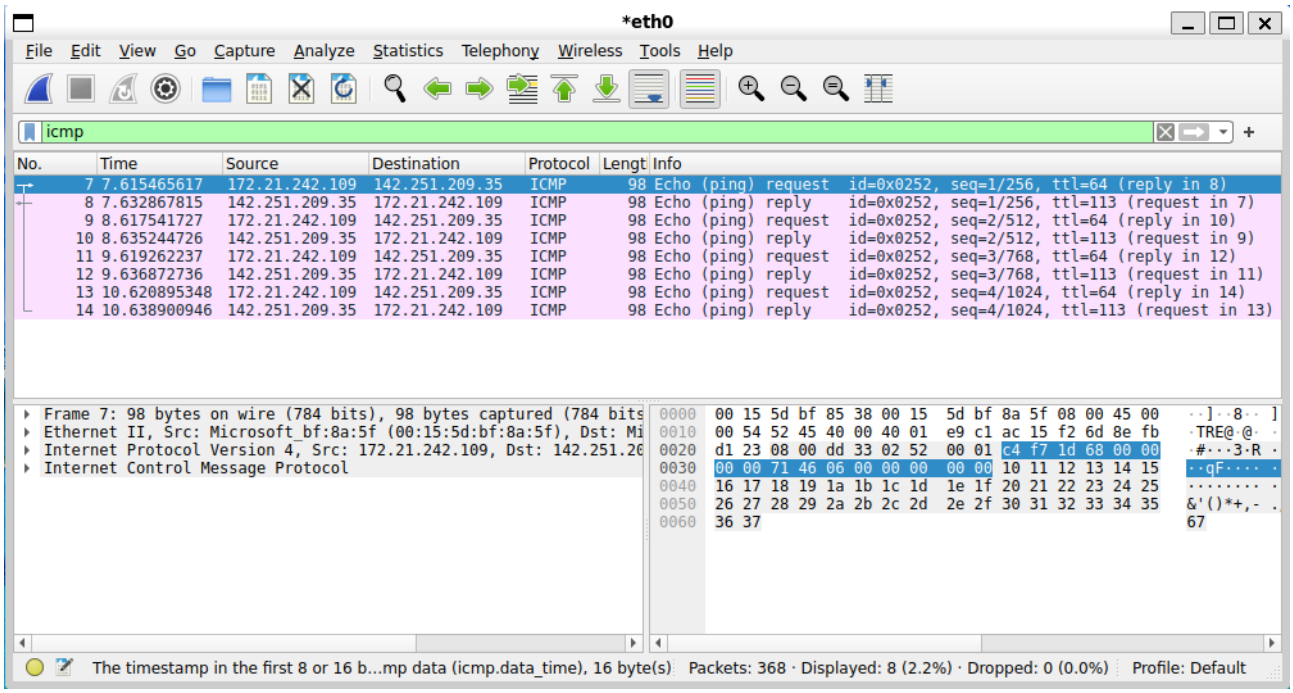
Subito dopo aver eseguito il comando ping è possibile stampare l'exit code con il comando:

$ echo "$?"

Oppure si possono combinare le due cose:

$ ping www.google.it ; echo "$?"

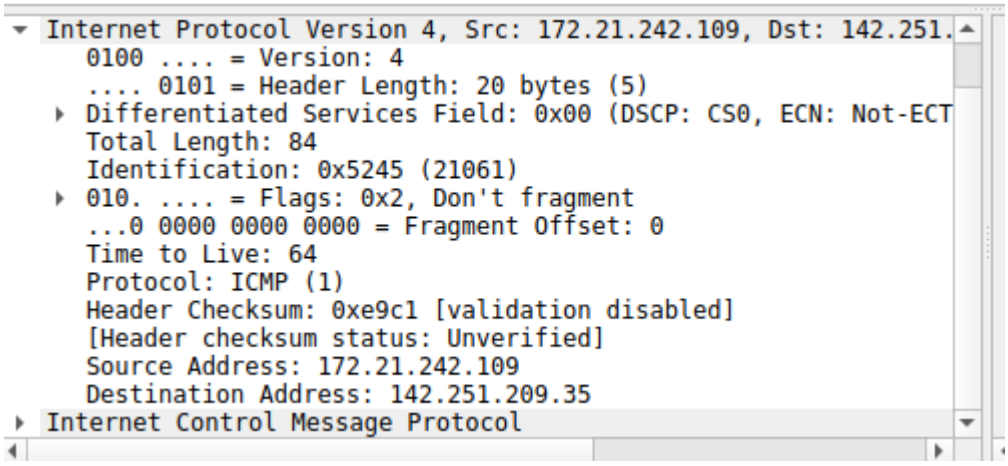
In Wireshark, trovare i pacchetti rilevanti usando il filtro "icmp".



È possibile notare nell'elenco dei pacchetti i messaggi *Echo request* e i corrispondenti *Echo reply*.

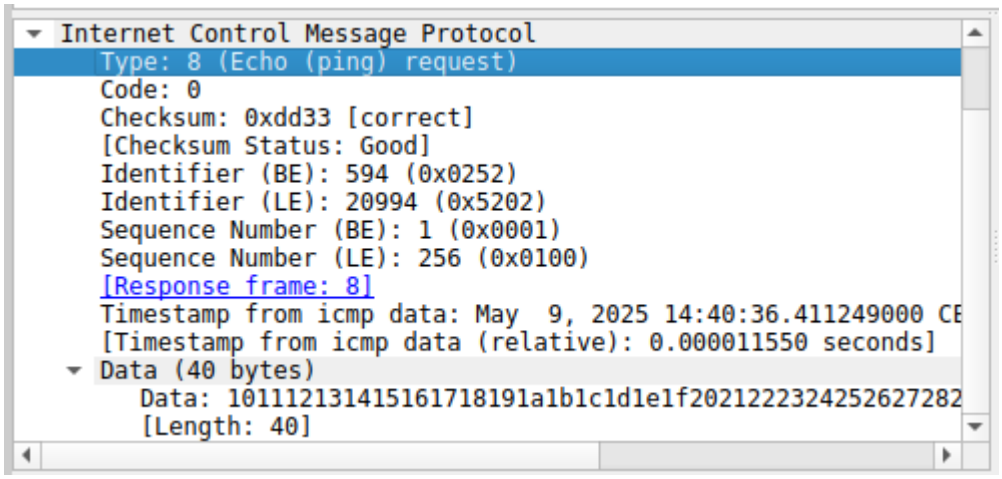
Avendo selezionato un messaggio (nello specifico una Echo request), si può notare nei dettagli del pacchetto che il messaggio ICMP è incapsulato direttamente dentro un datagramma IP.

Qui di seguito la intestazione IP:



Assicurarsi di ripassare il significato di tutti i campi. La novità rispetto a quanto visto in precedenza è il fatto che nel campo *Protocol* c'è il valore 1 a rappresentare il protocollo ICMP (invece, per esempio, dei numeri 6 per TCP e 17 per UDP).

Qui di seguito l'intestazione e i dati del messaggio ICMP.



Notate i campi *type* e *code* che classificano il messaggio come un *Echo request*. Troviamo anche la checksum che protegge l'intero messaggio.

Troviamo anche i campi *indentifier* (comune a tutti i messaggi generati da una sessione ping) e *sequence* (incrementati di uno in uno in ciascuna richiesta), interpretati come big endian e little endian.

Vien mostrato anche il *timestamp*, che è inserito nei primi 16 byte del campo di dati. Quest'ultimo può includere ulteriori dati (con un pattern noto! Pertanto, si può facilmente verificarne la correttezza nel messaggio *echo reply*) usati per rilevare problemi *data-dependent* nei link, nonché per valutare la MTU.

Notate il collegamento (aggiunto da Wireshark!!!!) al messaggio di risposta: è analogo alla richiesta ma con sorgente e destinazione IP invertite, nonché *type* e *code* diversi, mentre i dati sono gli stessi (in questo modo posso calcolare l'RTT e verificare i dati ricevuti).

Esercizio 2

L'opzione *-c* permette di indicare il numero di pacchetti da inviare.

```
$ ping -c 1 www.google.it

PING www.google.it (142.250.180.163) 56(84) bytes of data.

64 bytes from mil04s44-in-f3.1e100.net (142.250.180.163): icmp_seq=1 ttl=114 time=18.2 ms


--- www.google.it ping statistics ---

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

rtt min/avg/max/mdev = 18.240/18.240/18.240/0.000 ms
```

Esercizio 3

L'opzione *-w* permette di indicare il numero di secondi (scadenza o deadline) trascorsi il quale ping deve terminare a prescindere dal numero di pacchetti inviati o ricevuti.

```
$ ping -w 3 www.google.it

PING www.google.it (142.250.180.163) 56(84) bytes of data.

64 bytes from mil04s44-in-f3.1e100.net (142.250.180.163): icmp_seq=1 ttl=114 time=17.7 ms

64 bytes from mil04s44-in-f3.1e100.net (142.250.180.163): icmp_seq=2 ttl=114 time=17.0 ms

64 bytes from mil04s44-in-f3.1e100.net (142.250.180.163): icmp_seq=3 ttl=114 time=17.6 ms


--- www.google.it ping statistics ---

3 packets transmitted, 3 received, 0% packet loss, time 2003ms

rtt min/avg/max/mdev = 16.967/17.415/17.712/0.322 ms
```

Notate che ping ha trasmesso 3 pacchetti e ricevuto 3 risposte. Come mai? Ping invia un pacchetto al secondo, agli istanti 0, 1, 2: il quarto pacchetto sarebbe inviato all'istante 3, quando però il comando termina.

Esercizio 4

Usando l'opzione *-i* si può indicare l'attesa in secondi (possibilmente col punto per indicare un numero frazionario) tra l'invio di un pacchetto e il successivo. Per valori molto bassi (su Ubuntu 24.04.1 LTS < 2 ms) occorre avere privilegi di root.

Giocando con le opzioni *-i* e *-w*, si può fare in modo di non lasciare tempo a sufficienza affinché arrivi l'ultima risposta:

```
$ ping -i 1.495 -w 3 www.google.it

PING www.google.it (142.251.209.3) 56(84) bytes of data.

64 bytes from mil04s50-in-f3.1e100.net (142.251.209.3): icmp_seq=1 ttl=114 time=17.1 ms

64 bytes from mil04s50-in-f3.1e100.net (142.251.209.3): icmp_seq=2 ttl=114 time=17.2 ms



--- www.google.it ping statistics ---

3 packets transmitted, 2 received, 33.3333% packet loss, time 2995ms

rtt min/avg/max/mdev = 17.105/17.153/17.201/0.048 ms
```

L'esempio di sopra potrebbe non funzionare nel caso il test sia eseguito su un nodo il cui RTT verso *www.google.it* è differente (inferiore).

Stampando l'exit status con *echo $?* viene indicato 0 perché www.google.it è raggiungibile.

Indicando anche l'opzione *-c 3*, il comando terminerebbe invece con exit status non ricevendo in tempo 3 risposte: essendo il terzo messaggio inviato all'istante 2.99 s, con un RTT > 10 ms, la risposta non arriverebbe entro i 3 secondi impostati con l'opzione *-w*.

```
$ ping -i 1.495 -c 3 -w 3 www.google.it

PING www.google.it (142.251.209.35) 56(84) bytes of data.

64 bytes from mil04s51-in-f3.1e100.net (142.251.209.35): icmp_seq=1 ttl=113 time=17.7 ms

64 bytes from mil04s51-in-f3.1e100.net (142.251.209.35): icmp_seq=2 ttl=113 time=18.0 ms



--- www.google.it ping statistics ---

3 packets transmitted, 2 received, 33.3333% packet loss, time 2994ms

rtt min/avg/max/mdev = 17.699/17.848/17.997/0.149 ms



$ echo $?
```

1

Esercizio 6

L'opzione *-W* permette di indicare il timeout (in secondi, espresso come un numero decimale [nelle versioni recenti del comando: controllare la *man* page]; 0 indica invece la possibilità di attendere indefinitamente) per la ricezione delle risposte dopo l'invio dei pacchetti indicati con l'opzione *-c* e in assenza dell'opzione *-w*. Si applica solo se nel frattempo non è stata ricevuta alcuna risposta, altrimenti usa come timeout il doppio dell'RTT massimo: se inferiore all'intervallo tra l'invio dei pacchetti, usa invece quest'ultimo.

```
$ ping -W 1 -c 1 www.google.it

PING www.google.it (216.58.204.227) 56(84) bytes of data.

64 bytes from mil07s18-in-f3.1e100.net (216.58.204.227): icmp_seq=1 ttl=114 time=16.6 ms


--- www.google.it ping statistics ---

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

rtt min/avg/max/mdev = 16.630/16.630/16.630/0.000 ms
```

Con un timeout di soli 10 ms, la risposta non fa in tempo ad arrivare (almeno nel caso della macchina su cui è stato effettuato il test):

```
$ ping -W 0.01 -c 1 www.google.it

PING www.google.it (142.251.209.35) 56(84) bytes of data.


--- www.google.it ping statistics ---

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

Esercizio 7

Con l'opzione -s si può indicare la dimensione del campo del messaggio ICMP: se inferiore a 16, non può essere inserito il timestamp e il comando ping non sarà in grado di calcolare l'RTT.

```
$ ping -s 10 www.google.it
```

```
PING www.google.it (142.251.209.35) 10(38) bytes of data.
18 bytes from mil04s51-in-f3.1e100.net (142.251.209.35): icmp_seq=1 ttl=113
18 bytes from mil04s51-in-f3.1e100.net (142.251.209.35): icmp_seq=2 ttl=113
18 bytes from mil04s51-in-f3.1e100.net (142.251.209.35): icmp_seq=3 ttl=113
18 bytes from mil04s51-in-f3.1e100.net (142.251.209.35): icmp_seq=4 ttl=113
18 bytes from mil04s51-in-f3.1e100.net (142.251.209.35): icmp_seq=5 ttl=113
^C
--- www.google.it ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4007ms
```

Esercizio 8

L'opzione *-t* permette di impostare il *time-to-live* nell'intestazione IP. Non tutti i router inviano il messaggio ICMP *Time to live exceeded* (o questo può essere filtrato oppure andare perso), quindi può essere utile l'opzione -W per impostare un timeout (in questo caso 2 secondi).

Il valore di TTL necessario per contattare www.google.it dipende in generale dal nodo da cui si fa partire la richiesta e può essere differente da quello mostrato di seguito.

```
$ ping -W 2 -c 1 -t 11 www.google.it
PING www.google.it (216.58.205.35) 56(84) bytes of data.
From 172.253.73.61 icmp_seq=1 Time to live exceeded

--- www.google.it ping statistics ---
1 packets transmitted, 0 received, +1 errors, 100% packet loss, time 0ms
```

mentre

```
$ ping -W 2 -c 1 -t 12 www.google.it
PING www.google.it (216.58.205.35) 56(84) bytes of data.
464 bytes from mil07s19-in-f3.1e100.net (216.58.205.35): icmp_seq=1 ttl=113 time=17.2 ms

--- www.google.it ping statistics ---
```

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

rtt min/avg/max/mdev = 17.199/17.199/17.199/0.000 ms

In questo particolare caso, ci sono quindi 11 router tra l'host di origine e l'host www.google.it; il percorso dall'origine alla destinazione ha quindi 12 hop (da verificare con traceroute).