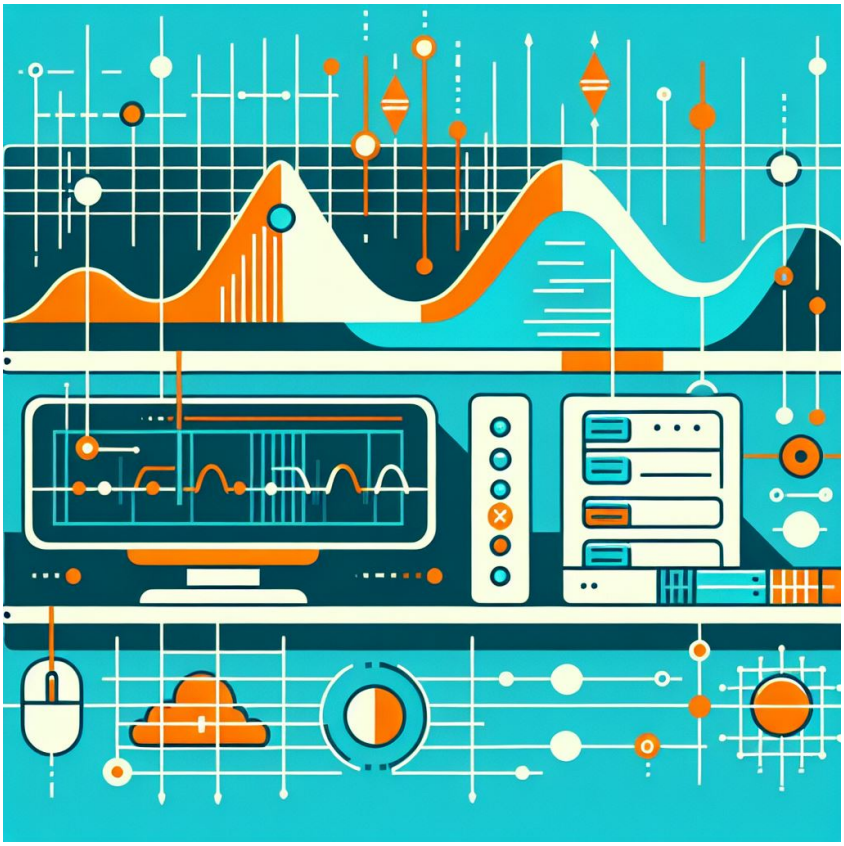
Alma Mater Studiorum · Università di Bologna

CORSO DI LAUREA IN INGEGNERIA E SCIENZE INFORMATICHE
A.A. 2024/25

Analisi del traffico ICMP e TCP con Scapy e Wireshark

Relazione per il corso di Programmazione di Reti



Presentato da:
Bartocetti Enrico, 0001115097

Indice

Capitolo 1

Introduzione

Obiettivo

Creare e inviare pacchetti ICMP e TCP con Scapy, catturarli con Wireshark e analizzare i risultati.

Requisiti minimi

- Inviare pacchetti ICMP Echo (ping) e TCP SYN verso un host
- Catturare i pacchetti in Wireshark e salvarli in .pcap
- Analizzare: IP di origine/destinazione, porte, checksum, TTL

Estensioni opzionali

- Inviare un pacchetto TCP con flag personalizzati
- Visualizzare e spiegare le differenze tra ICMP e TCP
- Generare un semplice report HTML o PDF con gli screen e analisi

Output atteso

- Script Python con Scapy
- File .pcap della cattura
- Relazione con screenshot e commenti tecnici

Capitolo 2

Pacchetti ICMP Echo

2.1 Generazione dei pacchetti con Scapy

Si vuole inviare un pacchetto `ICMP Echo Request` (ping) a un host. Per l'invio ho utilizzato il metodo `.sr` della libreria scapy, specificando il tipo `ICMP`. Riporto nel Listing 2.1 la funzione che ho scritto per poter generare il pacchetto del ping, che richiede come parametro l'indirizzo IP oppure l'hostname del destinatario. La funzione stampa anche il risultato dell'operazione.

```
1 import scapy.all as scapy
2
3 def send_ping(destination):
4     print("----- INVIO PING A", destination, " -----")
5     res = scapy.sr(scapy.IP(dst=destination)/scapy.ICMP(), timeout=4)
6     print("--- RISULTATI: ---")
7     for r in res:
8         r.show()
9     print()
```

Listing 2.1: Funzione python per la generazione di un pacchetto `ICMP Echo Request`

2.2 Analisi del traffico con Wireshark

Ho mandato il ping a `google.com`, che è stato risolto nell'indirizzo IP `216.58.204.238`. Per la cattura del traffico di rete ho utilizzato wireshark con il filtro `host 216.58.204.238` per catturare solo i pacchetti da / per l'host indicato. Nel file `wireshark/icmp_echo.pcap` è presente il risultato della cattura.
Riporto in seguito il dettaglio della richiesta e della risposta.

2.2.1 ICMP Echo Request

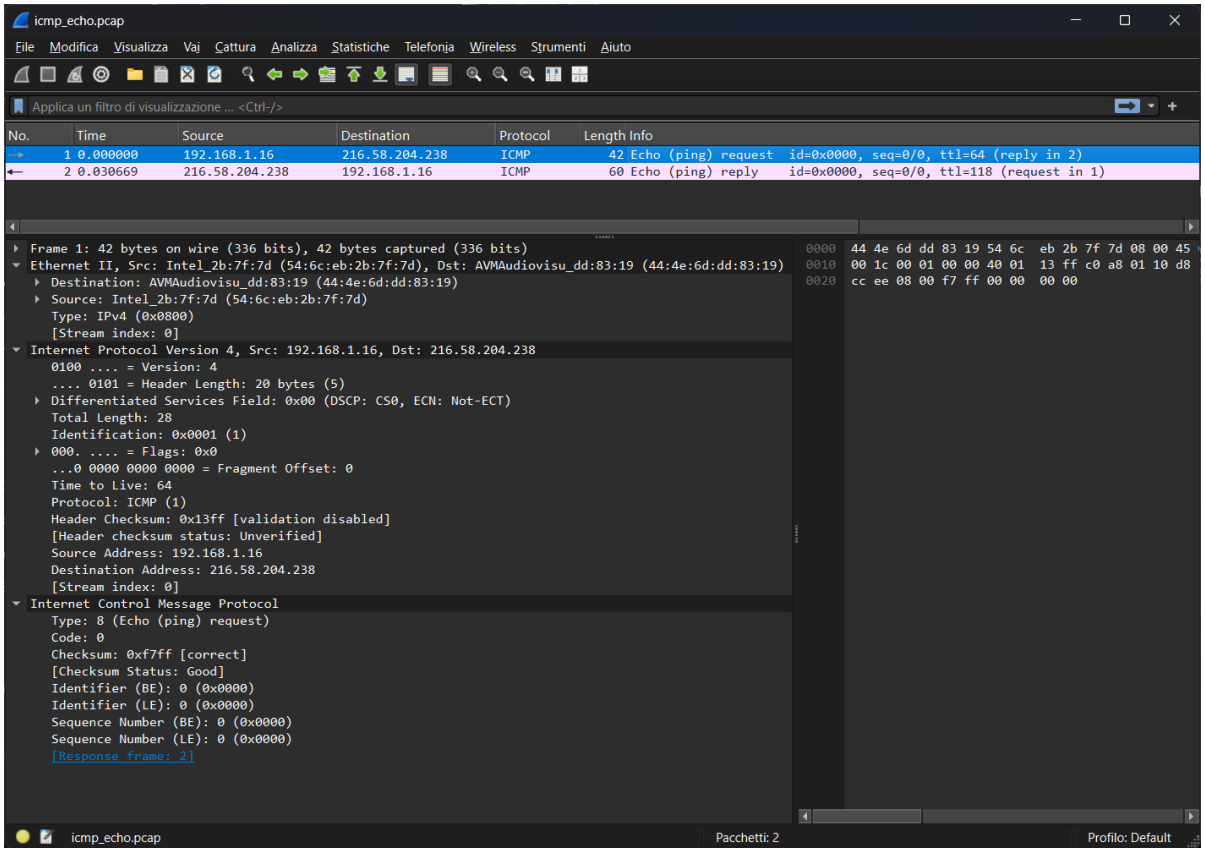


Figura 2.1: Dettaglio Wireshark dell'Echo Request

Il pacchetto parte dal mio PC connesso alla rete di casa, infatti nella PCI del livello IP troviamo il campo `Source Address: 192.168.1.16`, ovvero l'IP privato del mio PC nella mia rete. Nel campo `Destination Address: 216.58.204.238` troviamo l'IP del destinatario della richiesta, ovvero l'host verso cui ho inviato il pacchetto ping. Il campo Time To Live è settato al livello massimo (64) visto che il pacchetto è stato catturato non appena generato.
Nella parte dati del pacchetto IP viene trasportato il pacchetto ICMP: possiamo confermarlo leggendo il campo `Protocol: ICMP` nell'header del pacchetto IP. All'interno della parte riservata al protocollo ICMP viene specificato che il pacchetto è di tipo `Echo Request`.
Notiamo che non sono presenti riferimenti a nessuna porta: questo perché IP e ICMP sono protocolli dell'Internet Layer della suite TCP/IP , quindi non c'è la necessità di comunicare tramite una porta con un livello superiore (ovvero con un protocollo di trasporto).
Risulta infine evidente che sia il pacchetto IP sia quello ICMP sono dotati di un proprio checksum, che nel caso dell'IP non viene verificato mentre nell'ICMP è corretto.

2.2.2 ICMP Echo Reply

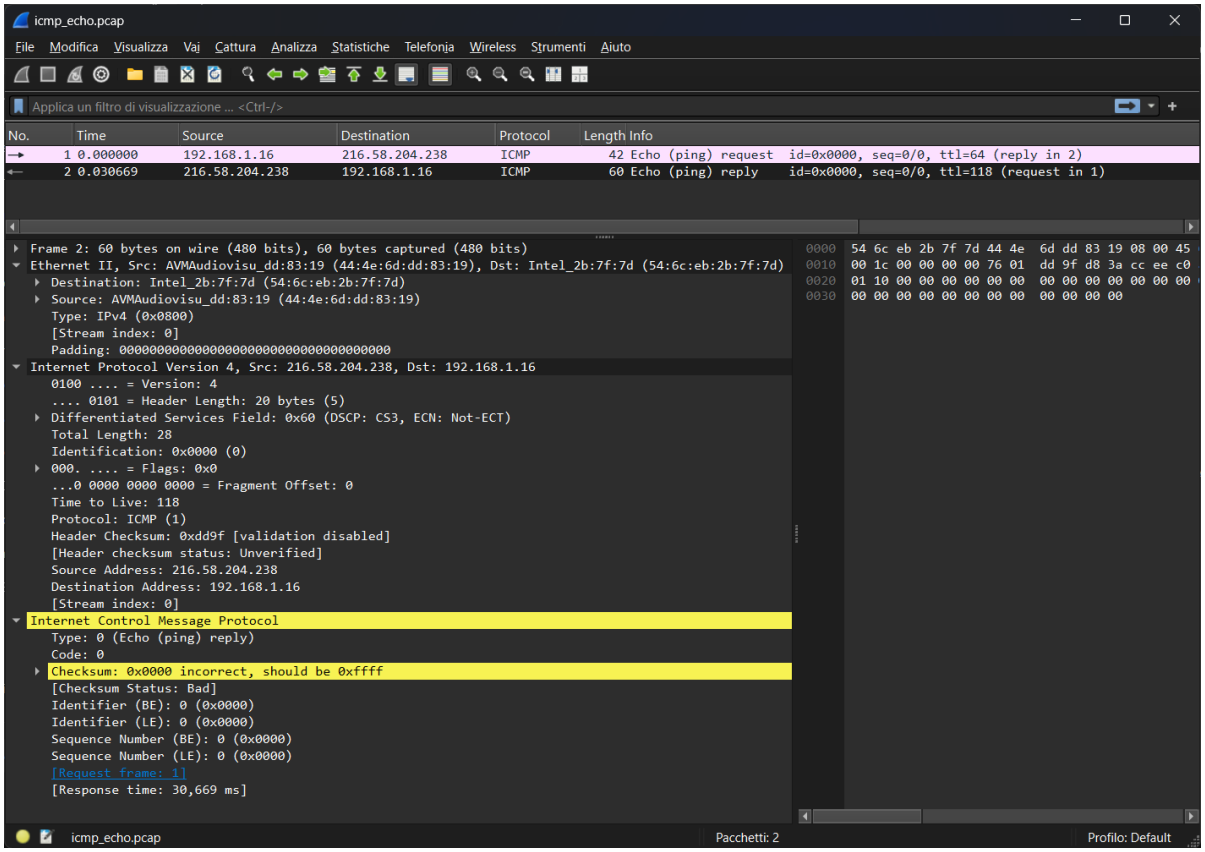


Figura 2.2: Dettaglio Wireshark dell'Echo Reply

Questa cattura presenta varie differenze rispetto a quella precedente. Quella più ovvia è l'inversione degli indirizzi IP nei campi `Destination Address: 192.168.1.16` e `Source Address: 216.58.204.238`, visto che si tratta del pacchetto di risposta (quindi generato dal destinatario del ping).
Poiché si tratta della risposta al ping iniziale, IP ci dice che il protocollo utilizzato è sempre `Protocol: ICMP`, mentre nella parte riservata al protocollo ICMP viene specificato che si tratta di una `Echo Reply`.
In questo caso il campo TTL è settato a 118: possiamo presupporre che alla generazione del pacchetto sia stato 128 (la potenza del 2 più vicina a 118), ma è stato decrementato di un'unita per ogni router che il pacchetto ha attraversato. Utilizzando il comando `tracert 216.58.204.238` ho poi verificato che per raggiungere il destinatario dal mio host sono necessari 10 salti.
Come prima il checksum dell'header IP è presente ma non verificato. Nel caso di ICMP invece, si ha che il checksum calcolato è diverso da quello riportato nel pacchetto. Ho provato a ripetere varie volte il ping, ottenendo sempre lo stesso risultato: potrebbe quindi essere che l'host destinatario sceglie di non calcolare il checksum lasciandolo a 0. Il contenuto informativo del pacchetto sembra comunque essere corretto nei campi che sono di nostri interesse.

Capitolo 3

Segmenti TCP SYN

3.1 Generazione dei segmenti

3.2 Analisi del traffico con Wireshark

Capitolo 4

Segmenti TCP con flag personalizzati

4.1 Generazione dei segmenti

4.2 Analisi del traffico con Wireshark