**Consegna del progetto**

**Progetto Wireshark**: Analisi del Protocollo TCP/IP

Descrizione: Usare Wireshark per catturare e analizzare il traffico di rete durante una sessione di trasferimento file (es. FTP o HTTP). Gli studenti dovranno esaminare i pacchetti per comprendere il funzionamento del protocollo TCP/IP.

Obiettivi: Identificare la sequenza di handshake TCP, analizzare la frammentazione dei pacchetti e il controllo di flusso, e riconoscere i campi chiave nei pacchetti.

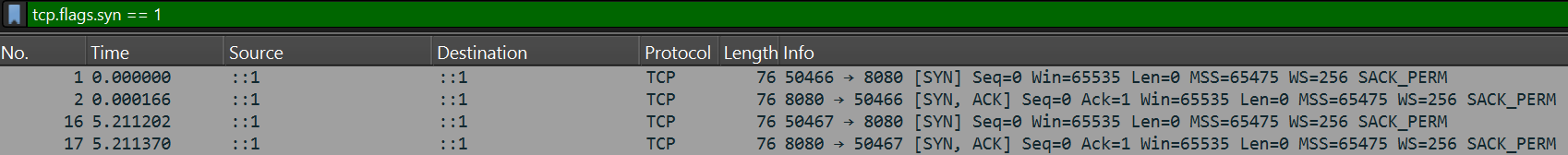
Consegne richieste: Report con screenshot delle catture di pacchetti, analisi delle metriche chiave (es. numero di pacchetti, tempi di latenza) e spiegazione di eventuali ritrasmissioni o problemi di connessione osservati.

**SETUP**:

Ho creato un server con il comando: **python3 -m http.server 8080**, in modo da poter scaricare un qualsiasi file, per analizzarlo e catturare il traffico di loopback. Nel mio caso ho creato un file “programma da scaricare.txt” e dopo aver avviato la cattura su Wireshark, ho proceduto a scaricare il file entrando tramite browser nella pagina <http://localhost:8080/>.

Ho fatto un ulteriore esempio andando a scaricare XAAMP tramite browser per andare ad analizzare la perdita di pacchetti, catturando il traffico su WI-FI.

**SEQUENZA DI HANDSHAKE:**



Filtro i pacchetti con “tcp.flags.syn == 1” per trovare la sequenza di handshake TCP(solamente SYN e SYN-ACK, poi imposto “tcp.flags.ack == 1 && tcp.flags.syn == 0”).

SYN: pacchetto iniziale inviato dal client(50466) al server(8080).

Immagine che contiene testo, schermata, software, Carattere

Descrizione generata automaticamente

Flags SYN (0x002): il flag indica che questo pacchetto avvia la connessione.

Sequence Number 0: indica che il numero iniziale di sequenza per il flusso di dati.

SYN-ACK: risposta del server che conferma di aver ricevuto SYN per l’accettazione della connessione.

Immagine che contiene testo, software, Carattere, Software multimediale

Descrizione generata automaticamente



Acknowledgment Number 1: Il server conferma di aver ricevuto il SYN del client, indicando il prossimo numero di sequenza atteso.

ACK: il client risponde con un ACK, completando l’handshake.

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Sequence number 1: client invia il proprio numero di sequenza successivo al SYN.

Acknowledgment number 1: client conferma di aver ricevuto SYN-ACK.

**TRASFERIMENTO HTTP:**

Immagine che contiene testo, schermata, Carattere, numero

Descrizione generata automaticamente

Pacchetto 4: richiesta HTTP GET dal client al server per la risorsa / (nel mio computer).

Pacchetto 8: risposta del server alla richiesta.

Pacchetto 19: Richiesta HTTP GET per andare a scaricare il file /programma da scaricate.txt dal server localhost:8080.

Immagine che contiene testo, schermata, Carattere, linea

Descrizione generata automaticamente

Tempo dall’inizio della connessione e tempo dall’ultimo pacchetto.

Pacchetto 23: il server risponde al GET del client con un pacchetto http 200 OK, inviando il file .txt.

Immagine che contiene testo, schermata, Carattere, linea

Descrizione generata automaticamente

Latenza tra richiesta e risposta: 0.003408 - 0.000584 = 0.002824 seconds (2.8 ms)

Non è stata rilevata frammentazione dei pacchetti, ritrasmissioni o problemi di connessione durante il trasferimento HTTP. Il download è avvenuto senza perdita di pacchetti, con una latenza minima di 2.8 ms e un trasferimento completo dei dati.

ESEMPIO SCARICANDO XAAMP

**FRAMMENTAZIONE DEI PACCHETTI:**

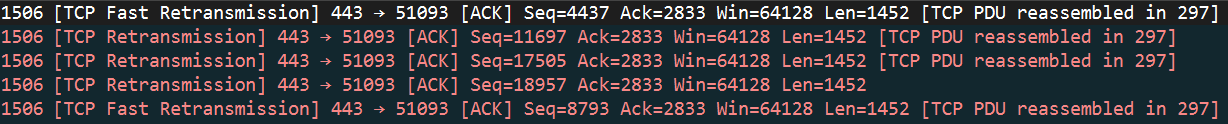
La frammentazione si verifica quando i pacchetti superano la dimensione massima dell’MTU (Maximum Transmission Unit) e vengono divisi in più frammenti.

Frammentazione pacchetti controllata con ip.flags.mf == 1 || ip.frag\_offset > 0, nulla da segnalare.

**CONTROLLO DELLE RITRASMISSIONI**

Uso “**tcp.analysis.retransmission**” per mostrare i pacchetti ritrasmessi(sono copie di pacchetti precedenti inviate per compensare perdite/timeout).

Durante il trasferimento, sono state osservate ritrasmissioni TCP tra il server (87.121.121.2) e il client (192.168.1.15).



Dimensione media dei pacchetti ritrasmessi: 1452 bytes.

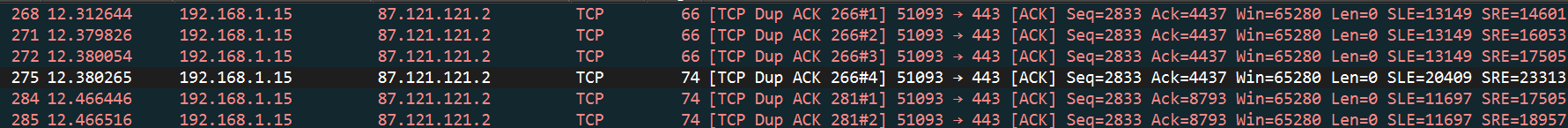
Nonostante le ritrasmissioni, il trasferimento è stato completato correttamente.

Le ritrasmissioni avvenute sono 94, tra Fast Retrasmission e Retrasmission.

Le ritrasmissioni sono causate da:

* + Perdita di pacchetti: Il destinatario non ha ricevuto il pacchetto originale.

Con **tcp.analysis.duplicate\_ack** vediamo gli ACK duplicati che segnalano al mittente che sta aspettando un pacchetto specifico(dove il Sequence Number è uguale in più pacchetti).

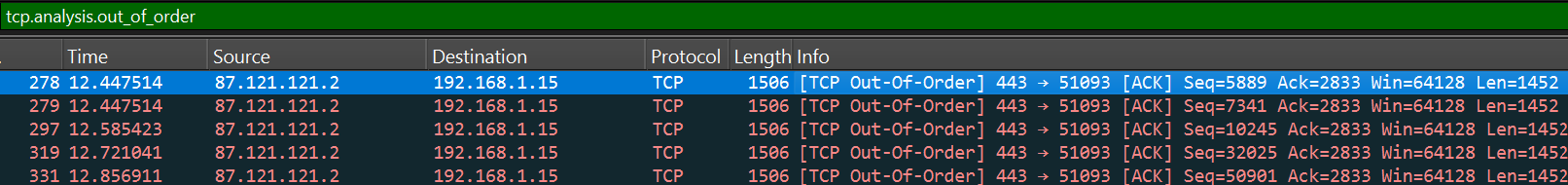


* + Timeout: Il mittente non ha ricevuto un ACK per il pacchetto entro un certo intervallo di tempo.

Immagine che contiene testo, schermata, Carattere, linea

Descrizione generata automaticamente

* + Congestione della rete: Ritardi dovuti a traffico intenso.



Il TCP Out-of-Order(trovato utilizzando tcp.analysis.out\_of\_order) ci dice che questi pacchetti indicano che i dati sono arrivati al destinatario in un ordine diverso rispetto a quello inviato dal mittente.

Immagine che contiene testo, schermata, Carattere, linea

Descrizione generata automaticamente

Questo tempo non è enorme, ma potrebbe indicare un ritardo dovuto alla congestione.

**Conclusioni riguardo al progetto**

Il progetto si è rivelato molto utile per comprendere i meccanismi del protocollo TCP/IP e mi ha permesso di analizzare la sequenza di handshake, il trasferimento dati HTTP e la gestione di errori come ritrasmissioni e frammentazioni. L’esempio HTTP ha evidenziato un trasferimento efficiente, mentre il download di XAAMP ha mostrato come il protocollo TCP gestisce errori e congestioni, garantendo comunque il completamento dei trasferimenti.

Ho approfondito il funzionamento del protocollo TCP/IP e ho osservato i meccanismi attraverso Wireshark. Sono soddisfatto del lavoro svolto, che mi ha dato nuove competenze utili per il futuro e la motivazione per continuare a esplorare il mondo delle reti.

Matricola: 0001098294

Nome: Gjovalin

Cognome: Mujaj