

22 Februari 2025

Workshop Administrasi Jaringan



Nama dosen pengampu:

Dr. Ferry Astika Saputra ST, M.Sc

Dikerjakan oleh

Nama : Muhammad Zaidan Zhafiz

NRP : 3123600021

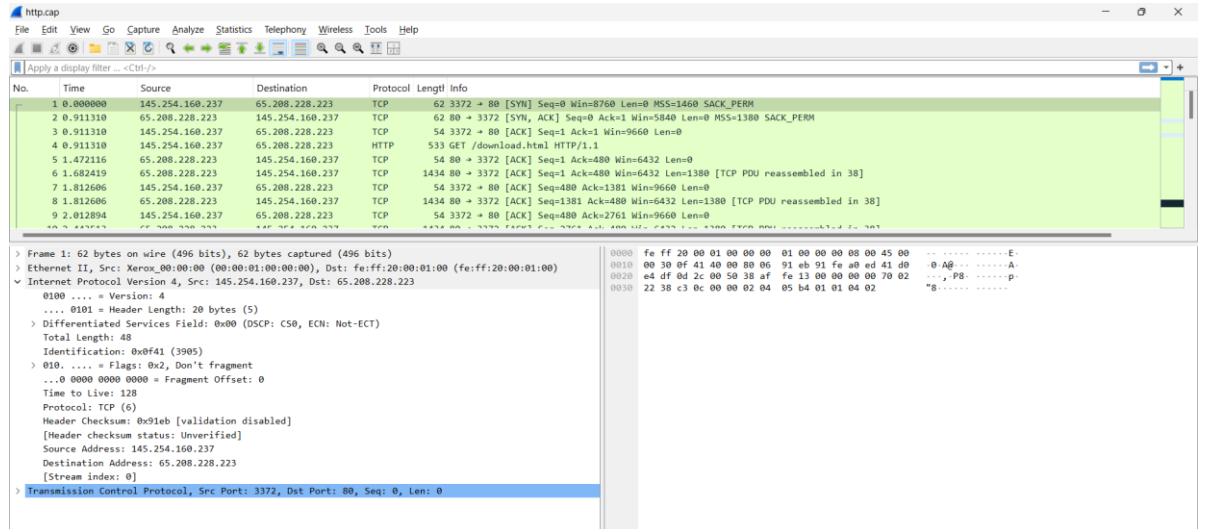
Kelas : 2 D4 IT A

Minggu 1

A. Review

Eksplor http.cap di wireshark, lakukan analisis:

a. IP server dan client

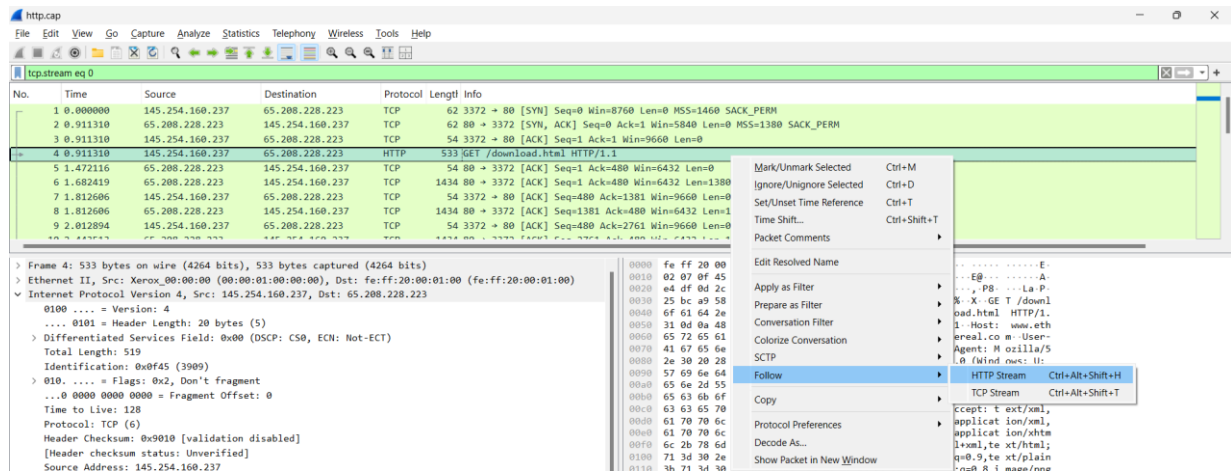


Pilih salah satu paket, kemudian buka detail dari pake tersebut. Pilih Internet Protocol Version. Terdapat informasi Source Adress dan Destination Adress.

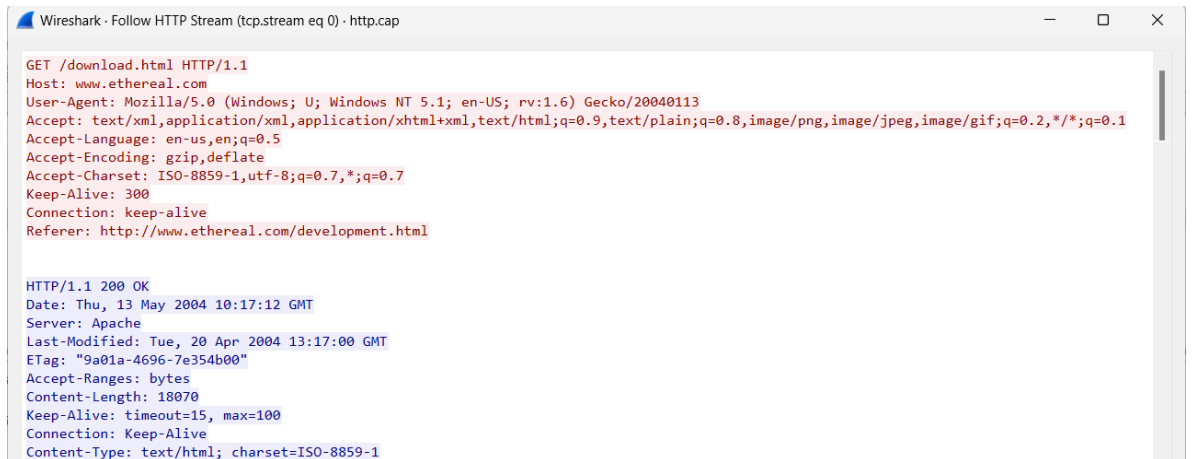
IP server : 65.208.228.223

IP Client : 145.254.160.237

b. Versi HTTP

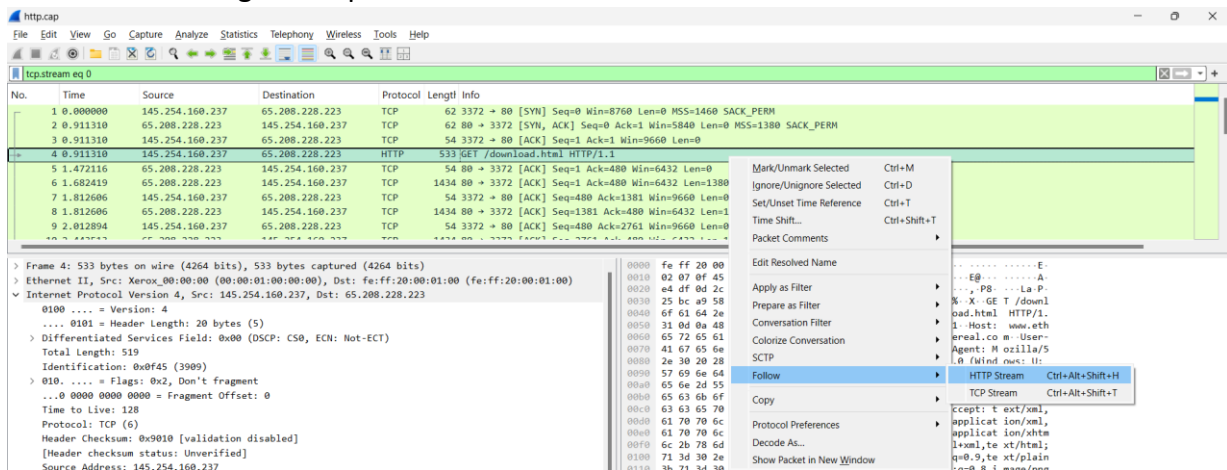


Pilih salah satu paket, kemudian pilih opsi follow, kemudian pilih HTTP Stream



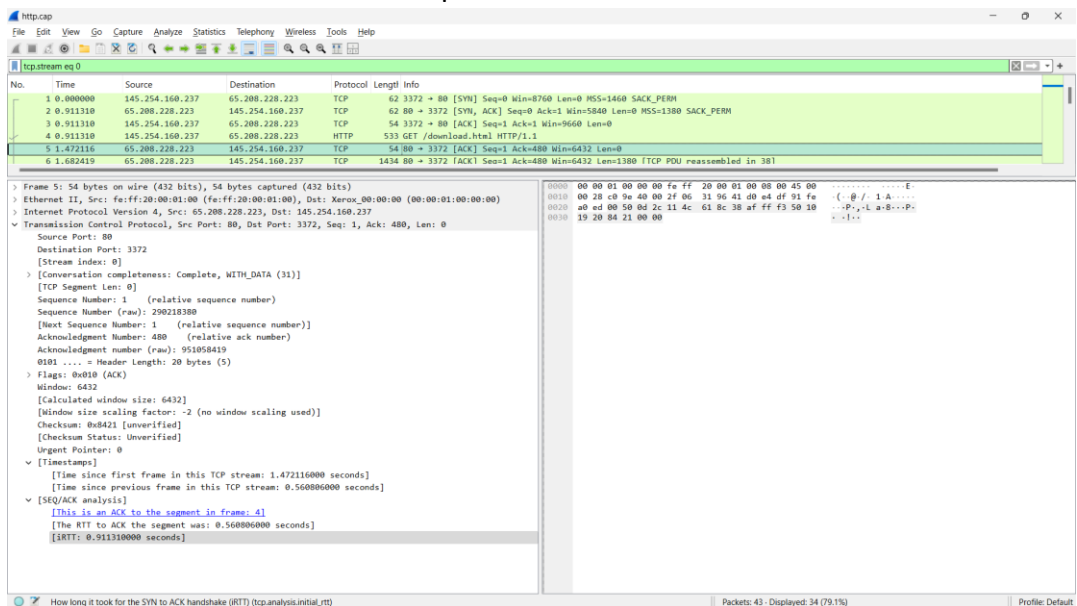
Dapat dilihat dari informasi yang diberikan bahwa:
 HTTP Version : 1.1

c. Waktu client mengirim request



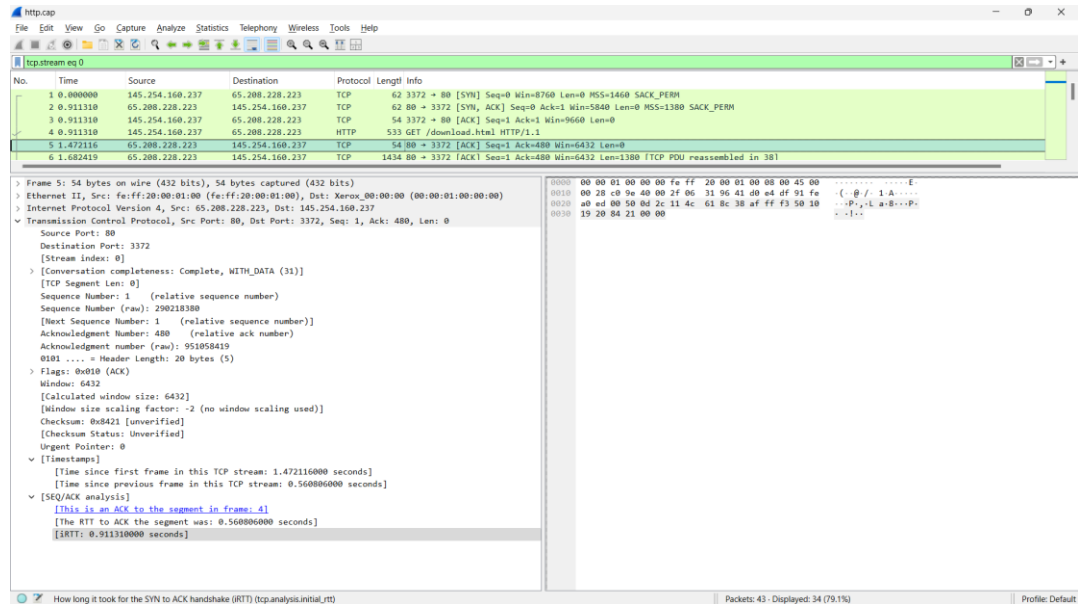
Pilih paket client yang mengirim HTTP request, kemudian lihat kolom time
 Waktu client mengirim request : 0.911310

d. Waktu server menerima HTTP request dari client



Cari acknowledge dari HTTP request client, dapat dilihat bahwa itu terdapat di paket 5, kemudian kita lihat kolom time nya
Waktu response server : 1.472116

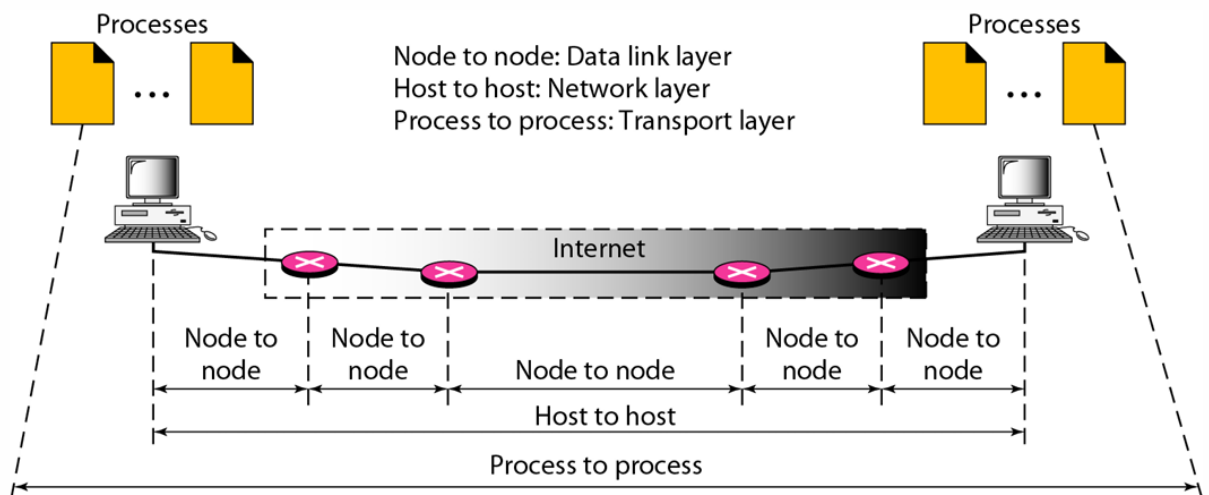
e. Waktu yang dibutuhkan untuk transfer dan response dari client ke server



Buka TCP header, kemudian lihat bagian SEQ/ACK Analysis. Setelah itu kita melihat RTT.

RTT : 0.560806000

B. Analisis Gambar



a. Node to node : Data Link Layer

- Node merujuk pada perangkat dalam jaringan, seperti komputer, switch, atau router.
- Data link layer bertanggung jawab untuk mengatur komunikasi langsung antara dua node yang terhubung dalam satu jaringan yang sama. Lapisan ini memastikan bahwa data dikirimkan dengan benar melalui media fisik, seperti kabel atau gelombang radio.

- iii. Lapisan ini menggunakan alamat MAC (Media Access Control) untuk mengidentifikasi perangkat dalam jaringan lokal.
- b. Host to Host : Network Layer
 - i. Host merujuk pada perangkat yang memiliki alamat IP, seperti komputer atau server.
 - ii. Network layer bertanggung jawab untuk mengatur komunikasi antara host yang mungkin berada di jaringan yang berbeda. Lapisan ini menggunakan alamat IP untuk melakukan routing paket data dari sumber menuju tujuan melalui berbagai jaringan.
 - iii. Protokol yang umum digunakan di lapisan ini adalah ICMP dan IGMP.
- c. Process to process : Transport Layer
 - i. Process merujuk pada aplikasi atau layanan yang berjalan di host, seperti browser web atau server email.
 - ii. Transport layer bertanggung jawab untuk mengatur komunikasi antara proses atau aplikasi yang berjalan di host yang sama atau berbeda. Lapisan ini memastikan bahwa data dikirimkan dengan andal dan dalam urutan yang benar.
 - iii. Protokol yang umum digunakan di lapisan ini adalah TCP (Transmission Control Protocol) dan UDP (User Datagram Protocol).

Ketiga lapisan ini bekerja sama untuk memastikan komunikasi yang efektif dalam jaringan. Data Link Layer memastikan komunikasi lokal antara node, Network Layer memastikan data dapat dikirim melintasi jaringan yang berbeda, dan Transport Layer memastikan bahwa data sampai ke aplikasi yang tepat di host tujuan.

C. Resume

Resume tahapan TCP

- a. Connection Establishment

Sebelum dua perangkat dapat bertukar data, mereka harus membangun koneksi TCP terlebih dahulu. Proses ini disebut **3-Way Handshake** karena melibatkan tiga langkah pertukaran paket.

Langkah-langkah 3-Way Handshake:

1. SYN (Synchronize)
 - a. Klien mengirimkan paket **SYN** ke server untuk meminta pembukaan koneksi.
 - b. Paket ini berisi nomor urut acak (sequence number) yang akan digunakan klien untuk mengirim data.
 - c. Contoh : Klien mengirim **SYN** dengan **seq = X**.
2. SYN-ACK (Synchronize-Acknowledge)
 - a. Server merespon dengan paket SYN-ACK
 - b. Paket ini berisi :

- i. Nomor urut acak (sequence number) yang akan digunakan server.
 - ii. Nomor acknowledgment (ACK) yang merupakan **seq klien + 1**.
- c. Contoh : Server mengirim SYN-ACK dengan **seq = Y** dan **ack = X + 1**.

3. ACK (Acknowledge)

- a. Klien mengirim paket **ACK** ke server untuk mengonfirmasi bahwa koneksi telah terbentuk.
- b. Paket ini berisi nomor acknowledgment (ACK) yang merupakan **seq server + 1**.
- c. Contoh : Klien mengirim ACK dengan **ack = Y + 1**.

Setelah 3-Way Handshake selesai, koneksi TCP terbentuk, dan kedua perangkat siap untuk bertukar data.

b. Data transfer

Setelah koneksi TCP terbentuk, kedua perangkat dapat mulai bertukar data. TCP menjamin bahwa data dikirim dengan andal, dalam urutan yang benar, dan tanpa duplikasi.

a. Pengiriman data

- a. Klien atau server mengirim data dalam bentuk TCP
- b. Setiap segmen berisi
 - i. Sequence number untuk mengidentifikasi urutan data.
 - ii. Acknowledge Number untuk mengonfirmasi data yang telah diterima.

b. Acknowledgement

- a. Penerima mengirim ACK untuk mengonfirmasi bahwa data telah diterima
- b. Jika pengirim tidak menerima ACK dalam waktu tertentu, data akan dikirim ulang

c. Flow Control

- a. TCP menggunakan mekanisme flow control untuk memastikan bahwa pengirim tidak mengirim data terlalu banyak sehingga membanjiri penerima
- b. Dilakukan dengan menggunakan window size yang merepresentasi berapa banyak data yang dapat diterima oleh penerima

d. Deteksi error dan koneksi

- a. TCP menggunakan checksum untuk mendeteksi kesalahan dalam data
- b. Jika ada kesalahan yang terjadi, segmen data akan dikirim ulang

c. Termination

Setelah selesai bertukar data, koneksi TCP harus diakhiri atau diputus dengan benar. Proses ini kita sebut sebagai 4-Way Handshake karena melibatkan empat Langkah pertukaran paket

a. FIN(Finish)

- a. Salah satu perangkat(contoh, klien) mengirim paket **FIN** ke perangkat lain (server) untuk meminta pengakhiran koneksi.

- b. Contoh : Klien mengirim **FIN** dengan **seq = A**
- b. ACK(Acknowledge)
 - a. Server merespons dengan paket **ACK** untuk mengonfirmasi bahwa permintaan FIN telah diterima.
 - b. Contoh : Server mengirim **ACK** dengan **ack = A + 1**
- c. FIN(Finish)
 - a. Server mengirim paket **FIN** ke klien untuk meminta pengakhiran koneksi dari sisi server.
 - b. Contoh : Server mengirim **FIN** dengan **seq = B**
- d. ACK(Acknowledge)
 - a. Klien merespons dengan paket ACK untuk mengonfirmasi bahwa permintaan FIN dari server telah diterima
 - b. Contoh : Klien mengirim **ACK** dengan **ack = B + 1**

Setelah 4-Way handshake selesai, setelah itu koneksi TCP ditutup, dan sumber daya jaringan dibebaskan