

# Tugas Praktikum SKJ ke-9

Nama : Bagus Cipta Pratama

NIM : 23/516539/PA/22097

Kelas : KOMC

# Aktivitas 1 : Sniffing Transfer TCP dari File Besar yang Dikirim dari Komputer Anda ke Server Jarak Jauh

1. Berapakah nomor urut segmen TCP SYN yang digunakan untuk memulai koneksi TCP antara komputer klien dan gaia.cs.umass.edu? Catatan: pertanyaan di sini mengacu pada nomor urut "mentah" yang dibawa dalam segmen TCP itu sendiri, dan BUKAN nomor paket dalam kolom "No." yang diberikan oleh Wireshark. Ingat bahwa tidak ada yang disebut "nomor paket" dalam TCP atau UDP; namun, ada nomor urut dalam TCP, dan itulah yang kita cari di sini. Juga, perhatikan bahwa ini bukan nomor urut relatif terhadap nomor urut awal sesi TCP ini. Apa yang ada dalam segmen TCP ini yang mengidentifikasikannya sebagai segmen SYN?

```

  Transmission Control Protocol, Src Port: 53782, Dst Port: 443, Seq
    Source Port: 53782
    Destination Port: 443
    [Stream index: 5]
    [Stream Packet Number: 1]
  > [Conversation completeness: Incomplete, DATA (15)]
    [TCP Segment Len: 0]
    Sequence Number: 0      (relative sequence number)
    Sequence Number (raw): 870240630
    [Next Sequence Number: 1      (relative sequence number)]
    Acknowledgment Number: 0
    Acknowledgment number (raw): 0
    1000 .... = Header Length: 32 bytes (8)
  > Flags: 0x002 (SYN)
    Window: 64240
    [Calculated window size: 64240]
    Checksum: 0x9914 [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
  > Options: (12 bytes), Maximum segment size, No-Operation (NOP),
  > [Timestamps]
```

Perhatikan bahwa sesuai dengan gambar tersebut didapatkan sequence number yang masig bersifat mentah (raw) adalah 870240630 . Segmen TCP SYN adalah segmen pertama dalam proses three-way handshake, digunakan untuk memulai koneksi antara klien (komputer) dan server

(gaia.cs.umass.edu). Wireshark menunjukkan bahwa segmen ini memiliki SYN flag = 1, tanpa flag ACK yang diaktifkan. Hal ini menandakan bahwa segmen tersebut adalah segmen permintaan awal untuk koneksi.

2. Berapakah nomor urut segmen SYNACK yang dikirim oleh gaia.cs.umass.edu ke komputer klien sebagai tanggapan terhadap SYN? Apa yang ada dalam segmen yang mengidentifikasikannya sebagai segmen SYNACK? Berapa nilai bidang Pengakuan dalam segmen SYNACK? Bagaimana gaia.cs.umass.edu menentukan nilai ini?

```
✓ Transmission Control Protocol, Src Port: 443, Dst Port: 53782, Seq:
  Source Port: 443
  Destination Port: 53782
  [Stream index: 5]
  [Stream Packet Number: 2]
  > [Conversation completeness: Incomplete, DATA (15)]
  [TCP Segment Len: 0]
  Sequence Number: 0 (relative sequence number)
  Sequence Number (raw): 591796375
  [Next Sequence Number: 1 (relative sequence number)]
  Acknowledgment Number: 1 (relative ack number)
  Acknowledgment number (raw): 870240631
  1000 .... = Header Length: 32 bytes (8)
  > Flags: 0x012 (SYN, ACK)
```

Perhatikan bahwa dari screenshot tersebut didapatkan nilai ack adalah syn sequence number ditambah satu menunjukkan bahwa server telah menerima segmen SYN dengan sukses. Segmen SYNACK diidentifikasi dengan flag SYN dan ACK diaktifkan, yang menunjukkan kombinasi permintaan dan pengakuan koneksi.

3. Berapakah nomor urut segmen TCP yang berisi header pesan HTTP POST? Catatan bahwa untuk menemukan header pesan POST, Anda perlu melihat lebih dalam ke Konten Paket di bagian bawah jendela Wireshark. Cari segmen yang berisi teks ASCII "POST" di bidang DATA. Berapa banyak byte data yang terdapat dalam bidang payload (data) segmen TCP ini? Apakah semua data dalam file alice.txt yang ditransfer muat dalam satu segmen ini?

```

v Transmission Control Protocol, Src Port: 53787, Dst Port: 80, Seq: 93789, Ack: 1, Len: 59259
  Source Port: 53787
  Destination Port: 80
  [Stream index: 18]
  [Stream Packet Number: 39]
  > [Conversation completeness: Incomplete, DATA (15)]
  [TCP Segment Len: 59259]
  Sequence Number: 93789 (relative sequence number)
  Sequence Number (raw): 1128991188
  [Next Sequence Number: 153048 (relative sequence number)]
  Acknowledgment Number: 1 (relative ack number)
  Acknowledgment number (raw): 1532744956
  0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x018 (PSH, ACK)
  Window: 258
  [Calculated window size: 66048]
  [Window size scaling factor: 256]
  Checksum: 0x3776 [unverified]
  [Checksum Status: Unverified]
  Urgent Pointer: 0

```

Nomor urut segmen TCP yang membawa header POST adalah 1128991188, seperti terlihat pada Sequence Number (raw). Segmen ini memiliki panjang data 59259 byte (TCP Segment Len) dan membawa payload besar, termasuk header POST. Flag PSH menunjukkan bahwa data harus segera diproses oleh lapisan aplikasi, sedangkan ACK menandakan pengakuan atas data yang diterima sebelumnya. Segmen ini berperan penting dalam proses transfer file dengan metode POST.

#### 4. Berapa panjang (header dan payload) dari segmen yang berisi header pesan POST?

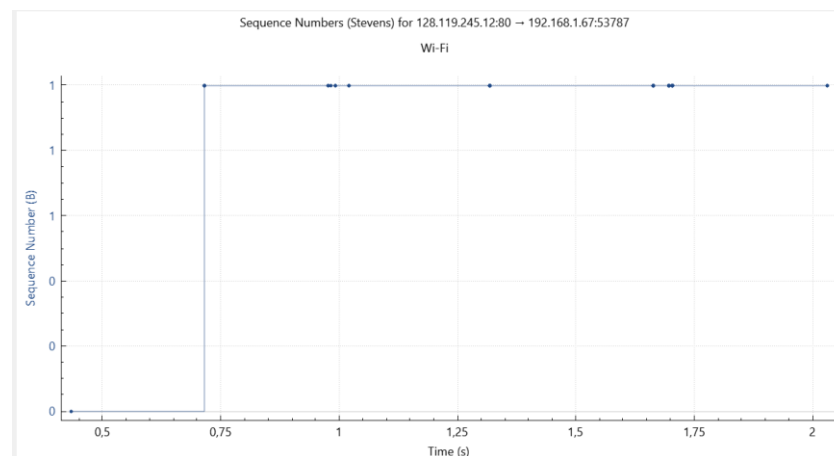
```

v Transmission Control Protocol, Src Port: 53787, Dst Port: 80, Seq: 93789, Ack: 1, Len: 59259
  Source Port: 53787
  Destination Port: 80
  [Stream index: 18]
  [Stream Packet Number: 39]
  > [Conversation completeness: Incomplete, DATA (15)]
  [TCP Segment Len: 59259]
  Sequence Number: 93789 (relative sequence number)
  Sequence Number (raw): 1128991188
  [Next Sequence Number: 153048 (relative sequence number)]
  Acknowledgment Number: 1 (relative ack number)
  Acknowledgment number (raw): 1532744956
  0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x018 (PSH, ACK)
  Window: 258
  [Calculated window size: 66048]
  [Window size scaling factor: 256]
  Checksum: 0x3776 [unverified]
  [Checksum Status: Unverified]
  Urgent Pointer: 0
  v [Timestamps]
    [Time since first frame in this TCP stream: 1.317532000 seconds]
    [Time since previous frame in this TCP stream: 0.000129000 seconds]
  v [SEQ/ACK analysis]
    [iRTT: 0.433356000 seconds]
    [Bytes in flight: 60669]
    [Bytes sent since last PSH flag: 59259]
  TCP payload (59259 bytes)
  TCP segment data (59259 bytes)

```

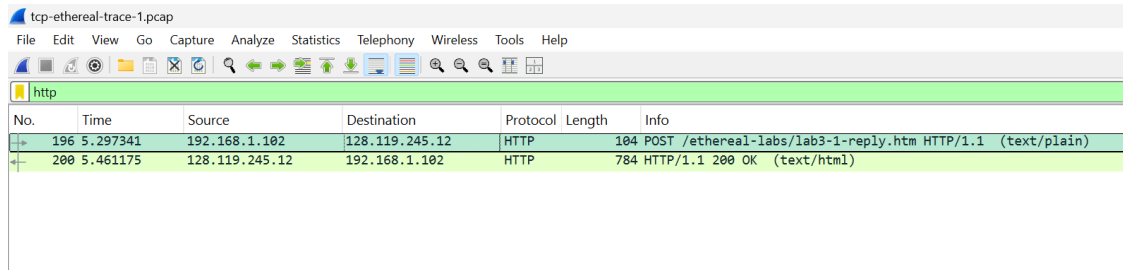
Berdasarkan output Wireshark, panjang total segmen POST adalah 59,313 bytes, yang mencakup payload data sebesar 59,259 bytes dan header TCP dengan panjang 20 bytes. Panjang payload dapat dilihat dari baris TCP payload (59259 bytes), sedangkan header TCP ditunjukkan oleh Header Length: 20 bytes (5). Selain itu, panjang total frame pada jaringan termasuk header Ethernet, IP, dan TCP juga dihitung dalam Frame Length (59313 bytes). Hal ini menunjukkan bahwa segmen tersebut berisi data POST dalam satu segmen besar, sesuai dengan efisiensi TCP dalam menangani transfer data besar. Kombinasi payload besar dengan overhead header minimal memastikan penggunaan jaringan yang optimal selama transfer data.

5. Berapa banyak segmen yang diteruskan? Untuk menjawab pertanyaan ini, lakukan hal berikut:
  - a. Pilih salah satu segmen TCP yang dikirim dari komputer Anda ke server dari Daftar Paket.
  - b. Pilih menu: Statistik → Grafik Aliran TCP → Urutan Waktu (Stevens).
  - c. Anda akan melihat plot nomor urut versus waktu. Setiap titik pada plot ini mewakili kapan segmen TCP dikirim dari PC Anda ke server.
  - d. Karena transmisi paket terjadi dalam waktu yang sangat singkat, zoom in (gulir ke atas) pada rentang waktu yang perlu dianalisis secara detail. Perhatikan bahwa sekelompok titik yang menumpuk ke atas pada waktu yang sama menunjukkan serangkaian paket yang dikirim secara berurutan oleh pengirim. Pikirkan tentang apa yang perlu Anda periksa untuk menentukan apakah ada segmen yang diteruskan.



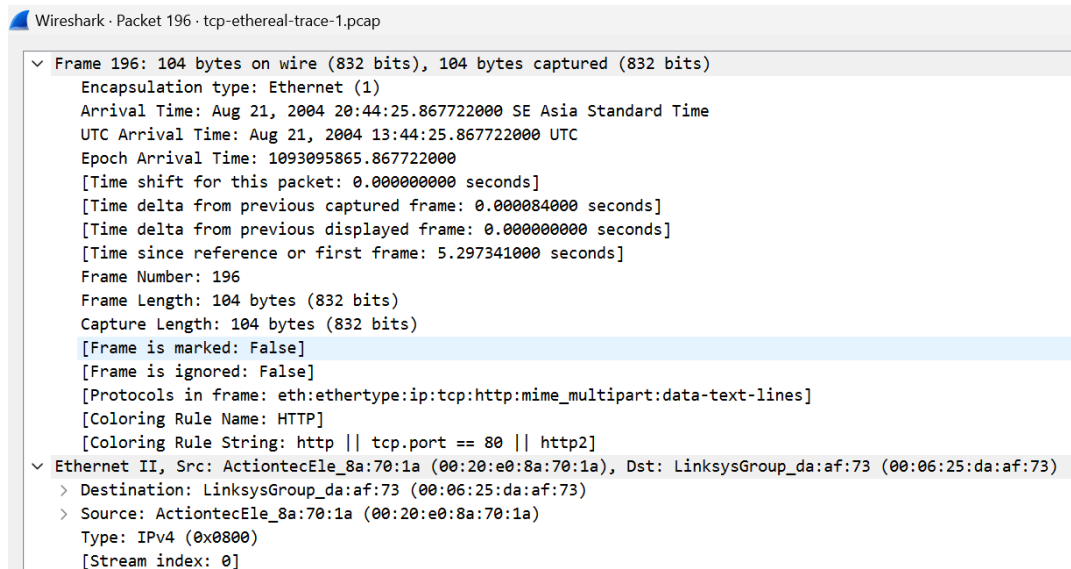
Berdasarkan grafik yang saya tangkap , Sequence Numbers (Stevens), terlihat bahwa tidak ada segmen yang diteruskan ulang (retransmission) selama proses transfer data. Hal ini ditunjukkan oleh pola grafik di mana setiap titik berada pada garis horizontal, menunjukkan bahwa nomor urut segmen meningkat secara konsisten tanpa pengulangan. Jika terjadi retransmission, grafik akan menunjukkan titik-titik vertikal yang mengindikasikan pengiriman ulang segmen dengan nomor urut yang sama pada waktu berbeda. Karena tidak ada pola seperti itu pada grafik, dapat disimpulkan bahwa semua segmen data berhasil dikirim dan diterima pada percobaan pertama tanpa gangguan. Ini mencerminkan efisiensi jaringan yang optimal selama transfer data dengan protokol TCP.

## Aktivitas 2 : Menganalisis File Jejak Wireshark dan Menghitung RTT (Round Trip Time)



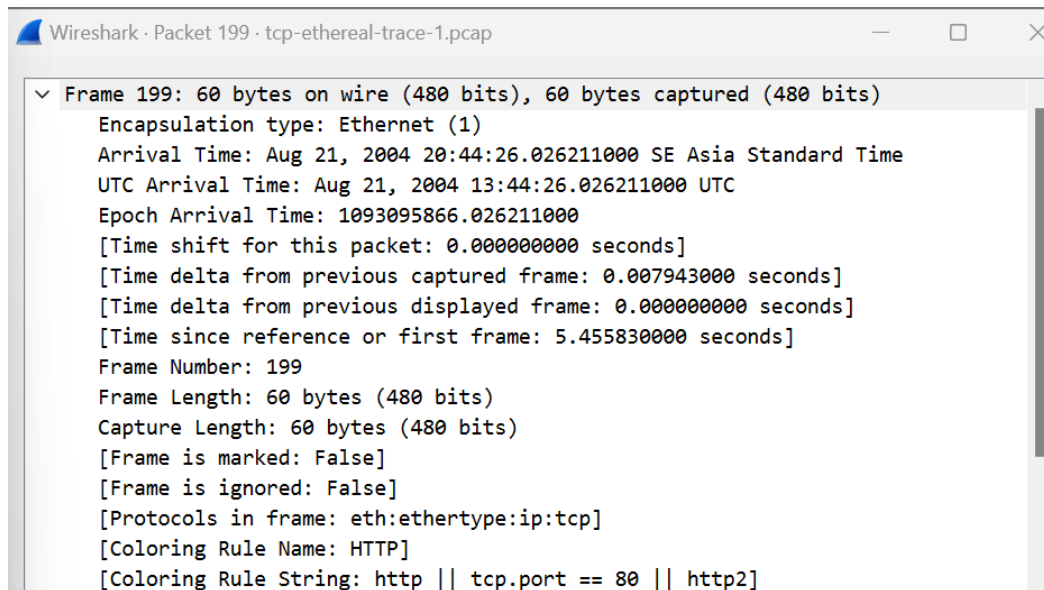
No.	Time	Source	Destination	Protocol	Length	Info
196	5.297341	192.168.1.102	128.119.245.12	HTTP	104	POST /ethereal-labs/lab3-1-reply.htm HTTP/1.1 (text/plain)
200	5.461175	128.119.245.12	192.168.1.102	HTTP	784	HTTP/1.1 200 OK (text/html)

1. Pada pukul berapa segmen pertama (yang berisi HTTP POST) dikirim?



Berdasarkan analisis data pada screenshot yang saya lakukan, segmen pertama yang berisi HTTP POST dikirim pada pukul 20:44:25.867722000 waktu standar Asia Tenggara (SE Asia Standard Time). Informasi ini dapat dilihat pada bagian Arrival Time dari Frame 196, yang menunjukkan waktu pasti segmen diterima oleh Wireshark selama proses capture. Waktu ini merepresentasikan titik awal pengiriman data HTTP POST pada sesi TCP yang sedang dianalisis.

2. Pada pukul berapa ACK untuk segmen pertama yang berisi data ini diterima?



Berdasarkan data yang terdapat pada screenshot yang saya lakukan, segmen ACK untuk segmen pertama diterima pada pukul 20:44:26.026211000 waktu standar Asia Tenggara (SE Asia Standard Time). Informasi ini ditemukan pada bagian Arrival Time dari Frame 199, yang menunjukkan waktu ketika segmen ACK diterima oleh Wireshark. Waktu ini mengindikasikan bahwa server telah mengirimkan acknowledgment sebagai respons terhadap segmen HTTP POST pertama.

### 3. Berapa nilai RTT untuk segmen pertama yang berisi data ini?

Perhatikan bahwa

$RTT = \text{Waktu ACK diterima} - \text{Waktu segmen dikirim}$

Berdasarkan data yang diberikan pada dua screenshot yang saya berikan sebelumnya, nilai RTT (Round Trip Time) untuk segmen pertama dapat dihitung dengan mengurangi waktu segmen pertama dikirim dengan waktu ACK diterima.

Segmen pertama dikirim pada pukul 20:44:25.867722000 (Frame 196), dan ACK untuk segmen tersebut diterima pada pukul 20:44:26.026211000 (Frame 199).

Dengan menggunakan rumus RTT, yaitu selisih waktu ACK diterima dan waktu segmen dikirim, diperoleh nilai RTT sebesar



0.158489000 detik atau 158.489 milidetik. Ini menunjukkan waktu yang diperlukan untuk mengirim segmen dan menerima respons ACK dari server.

4. Berapa nilai RTT untuk segmen TCP yang mengangkut potongan data kedua dan ACK-nya?

The screenshot shows a Wireshark capture of network traffic. The top pane displays a list of packets. Packet 9 is a TCP segment (Seq=6080, Win=1161) and Packet 10 is its corresponding ACK (Seq=4946, Win=17520). The bottom pane shows the details of Packet 9, including its Ethernet II header and Internet Protocol Version 4 header.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.1.102	128.119.245.12	TCP	62	1161 → 80 [SYN] Seq=0 Win=16384 Len=0 MSS=1460 SACK_PERM
2	0.023172	128.119.245.12	192.168.1.102	TCP	62	80 → 1161 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460 SACK_PERM
3	0.023265	192.168.1.102	128.119.245.12	TCP	54	1161 → 80 [ACK] Seq=1 Ack=1 Win=17520 Len=0
4	0.026477	192.168.1.102	128.119.245.12	TCP	619	1161 → 80 [PSH, ACK] Seq=1 Ack=1 Win=17520 Len=565 [TCP PDU reassembled in 196]
5	0.041737	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [PSH, ACK] Seq=566 Ack=1 Win=17520 Len=1460 [TCP PDU reassembled in 196]
6	0.053937	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=566 Win=6780 Len=0
7	0.054026	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=2026 Ack=1 Win=17520 Len=1460 [TCP PDU reassembled in 196]
8	0.054690	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=3486 Ack=1 Win=17520 Len=1460 [TCP PDU reassembled in 196]
9	0.077294	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=2026 Win=8760 Len=0
10	0.077405	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=4946 Ack=1 Win=17520 Len=1460 [TCP PDU reassembled in 196]
11	0.078157	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=6406 Ack=1 Win=17520 Len=1460 [TCP PDU reassembled in 196]

Frame 9: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0  
 Encapsulation type: Ethernet (1)  
 Arrival Time: Aug 21, 2004 20:44:20.647675000 SE Asia Standard Time  
 UTC Arrival Time: Aug 21, 2004 13:44:20.647675000 UTC  
 Epoch Arrival Time: 1093095860.647675000  
 [Time shift for this packet: 0.000000000 seconds]  
 [Time delta from previous captured frame: 0.022604000 seconds]  
 [Time delta from previous displayed frame: 0.022604000 seconds]  
 [Time since reference or first frame: 0.077294000 seconds]  
 Frame Number: 9  
 Frame Length: 60 bytes (480 bits)  
 Capture Length: 60 bytes (480 bits)  
 [Frame is marked: False]  
 [Frame is ignored: False]  
 [Protocols in frame: eth:ethertype:ip:tcp]  
 [Coloring Rule Name: HTTP]  
 [Coloring Rule String: http || tcp.port == 80 || http2]  
 Ethernet II, Src: LinksysGroup\_daf:73 (00:06:25:da:af:73), Dst: ActiontecE1a:70 (00:06:25:da:af:70)  
 Internet Protocol Version 4, Src: 128.119.245.12, Dst: 192.168.1.102  
 0100 ... = Version: 4  
 0000 ... = Header Length: 20 bytes (16)

Berdasarkan analisis screenshot yang saya lakukan, segmen TCP kedua yang mengangkut potongan data tambahan teridentifikasi sebagai Frame 9, dengan waktu pengiriman segmen sebesar 20:44:20.647675000. Segmen ACK yang relevan untuk segmen kedua ditemukan pada Frame 10, dengan waktu penerimaan ACK sebesar 20:44:20.077405000. Dengan menggunakan rumus RTT, yaitu selisih antara waktu ACK diterima dan waktu segmen dikirim, diperoleh nilai RTT untuk segmen kedua sebesar 429.730 milidetik. Perhitungan ini menunjukkan waktu perjalanan bolak-balik untuk segmen kedua hingga server memberikan respons ACK.

5. Berapa nilai RTT yang Diperkirakan setelah ACK untuk segmen data kedua diterima? Untuk menghitung RTT yang Diperkirakan setelah ACK untuk segmen kedua diterima, anggap bahwa nilai awal RTT yang Diperkirakan sama dengan RTT "aktual" yang diukur untuk segmen pertama. Kemudian, hitung menggunakan persamaan RTT yang Diperkirakan dan nilai  $\alpha = 0.125$ .

Perhatikan bahwa :

$$RTT_{\text{new}} = (1 - \alpha) * RTT_{\text{old}} + \alpha * RTT_{\text{actual}}$$

Berdasarkan perhitungan sebelumnya, nilai  $RTT_{old}$  (RTT segmen pertama) adalah 158.489 milidetik, dan  $RTT_{actual}$  (RTT segmen kedua) adalah 429.730 milidetik. Substitusi nilai ini ke dalam rumus menghasilkan perhitungan  $RTT_{new} = 192.144$  milidetik. Nilai ini merepresentasikan perkiraan RTT terkini setelah diperbarui berdasarkan data tambahan dari segmen kedua.