

# MODUL 2

# WIRESHARK

## TUJUAN PEMBELAJARAN:

1. Mengenalkan pada mahasiswa tentang konsep pengiriman data dengan TCP
2. Mengenalkan pada mahasiswa tentang konsep pengiriman data dengan UDP

## DASAR TEORI

### Protokol TCP

Di dalam penggunaan internet dan secara general jaringan TCP/IP, pengomunikasian setiap aplikasi dengan menggunakan protokol pendukung. Protokol ini bagian di dalam layer transport (*transport layer*) pada standar OSI yaitu bagian yang memberikan efisiensi dan jaminan komunikasi *end-to-end* ([Tanenbaum, 2003](#)).

Layer transport ini terdapat 2 protokol utama yaitu protokol UDP (*User Datagram Protocol*) dan protokol TCP (*Transmission Control Protokol*). Protokol ini untuk mendukung konsep jaringan berbasis IP. Telah diketahui bahwa IP (*internet protocol*) sebagai protokol jaringan internet yang mengkomunikasikan dua titik jaringan serta secara spesifik semua aplikasi dan layanan terpengaruh port tetapi kondisi konsep jaringan IP tidak memberikan jaminan. Jaminan tersebut adalah jaminan bahwa data akan tersampaikan pada *destination* yang benar dan data tersampaikan dengan benar ([Kurose dan Ross, 2000](#), section 1.3). Model layanan IP merupakan *best effort* bagi tercapainya data antara komunikasi dua titik jaringan. TCP dan UDP ini mendukung hingga 65536 virtual port dan ini digunakan oleh semua aplikasi dalam melakukan komunikasi pertukaran data.

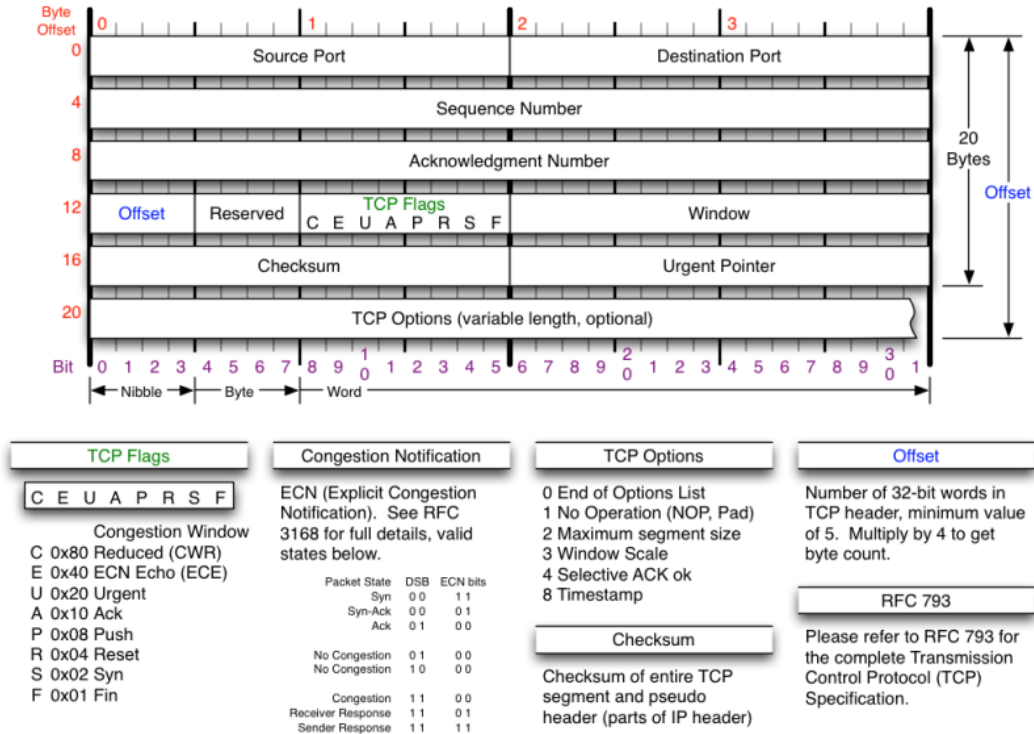
TCP adalah protokol yang dapat dipercaya dan dirancang untuk menyediakan alur data pada jaringan internet yang secara umum diketahui dengan kondisi tidak dapat dipercaya serta dirancang untuk beradaptasi dengan peralatan jaringan terhadap berbagai macam permasalahannya. Dirancangnya protokol ini untuk dapat dipercaya maka TCP bersifat *connection oriented* dalam mengirimkan data. TCP menjamin data yang terpercaya dengan menggunakan ARQ (*Automatic Repeat Request*). ARQ akan mentransmisikan secara otomatis berdasarkan informasi gagal diterimanya data ACK (*Acknowledgement*) dari penerima data. Untuk menjamin kontrol efektif terhadap hambatan maka dilakukan dengan cara mengestimasi *delay* dari transmisi *round trip time* secara akurat, sehingga dengan mempergunakan informasi balasan dari jaringan tersebut maka dapat mendeteksi sebuah kemacetan jaringan dan menyelesaikannya. Penjelasan TCP dapat ditemui pada RFC 793, 1122 dan 1323.

TCP memiliki tujuh fitur utama yaitu sebagai berikut:

1. *Connection oriented*, aplikasi meminta koneksi dan menggunakannya dalam transfer data.
2. *Point-to-point communication*, setiap koneksi TCP memiliki pasti dua titik.
3. *Reliability*, TCP menjamin bagi data yang dikirimkan dalam koneksi dapat terkirim dengan pasti tanpa ada yang hilang atau double.
4. *Full-duplex connection*, koneksi TCP memperbolehkan data untuk berkoneksi dari salah satu titik koneksi setiap saat.

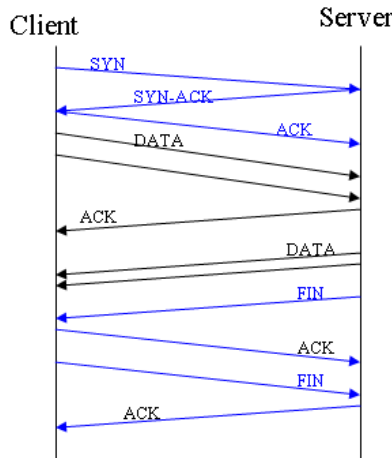
5. *Stream interface*, TCP memperbolehkan aplikasi untuk mengirimkan koneksi yang berkesinambungan.
6. *Reliable startup*, membutuhkan persetujuan dari kedua aplikasi untuk melakukan koneksi baru.
7. *Graceful shutdown*, aplikasi dapat membuka aplikasi, mengirim data dan menutup koneksi serta menjamin bahwa data sampai sebelum koneksi terputus.

Struktur header TCP dapat dilihat pada gambar berikut :



Gambar 1. Struktur header TCP

Proses komunikasi data pada protokol TCP adalah sebagai berikut :



Gambar 2. Proses pengiriman data TCP

## Protokol UDP

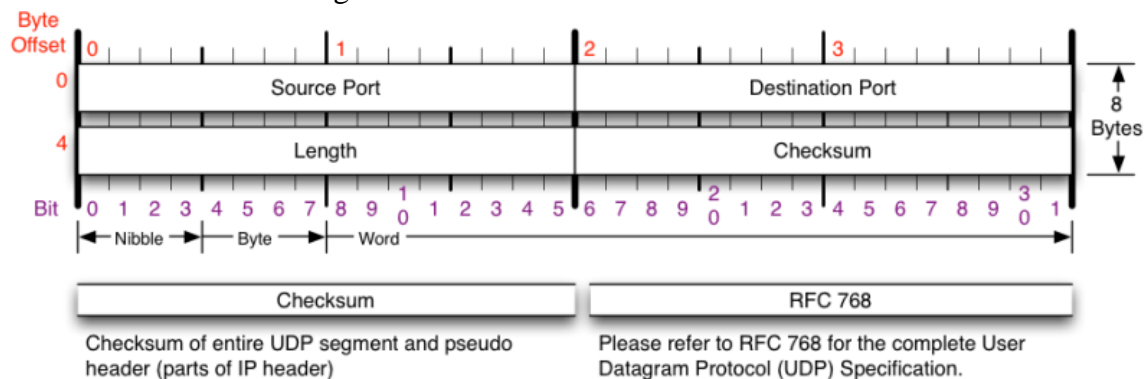
Pada section ini dijelaskan tentang protokol UDP. Memperkuat pernyataan [Tenenbaum](#) (2003), pada bagian lain blog ini yaitu protokol TCP bahwa layer transport terdapat 2 protokol utama yaitu protokol UDP (*User Datagram Protocol*) dan protokol TCP (*Transmission Control Protokol*). Protokol ini untuk mendukung konsep jaringan berbasis IP. Telah diketahui bahwa IP (*internet protocol*) sebagai protokol jaringan internet yang mengkomunikasikan dua titik jaringan serta secara spesifik semua aplikasi dan layanan terpengaruh port tetapi kondisi konsep jaringan IP tidak memberikan jaminan. Jaminan tersebut adalah jaminan bahwa data akan tersampaikan pada *destination* yang benar dan data tersampaikan dengan benar ([Kurose](#) dan [Ross](#), 2000, section 1.3).

Berbeda dengan TCP, protokol UDP adalah protokol yang bersifat *connectionless* dalam mentransmisi data dan tidak mengenal dalam pengecekan terhadap error pengiriman data. Protokol UDP pada dasarnya hanya mengandung IP dengan tambahan *header* singkat. Protokol UDP tidak melakukan sebuah proses kontrol alur data, kontrol kesalahan ataupun pengiriman ulang terhadap kesalahan sehingga hanya menyediakan *interface* ke protokol IP. UDP sangat berguna sekali pada situasi *client-server* dan penjelasan UDP lebih detail dapat ditemui pada RFC 768.

[Comer](#) (2003, section 25), UDP memiliki karakteristik yaitu sebagai berikut:

1. *End-to-end*, UDP dapat mengidentifikasi proses yang berjalan dalam computer.
2. *Connectionless*, UDP memiliki paradigma *Connectionless* tanpa membuat koneksi sebelumnya dengan tanpa adanya control.
3. *Message-oriented*, mengirimkan dan menerima data secara segmen.
4. *Best-effort*, yang utama adalah pengiriman yang terbaik.
5. *Arbitrary interaction*, UDP dapat menerima dan mengirim dari banyak proses.
6. *Operating system independent*, berdiri sendiri dalam operating system.

Struktur header UDP sebagai berikut :



Gambar 3. Struktur header UDP

## TUGAS PENDAHULUAN

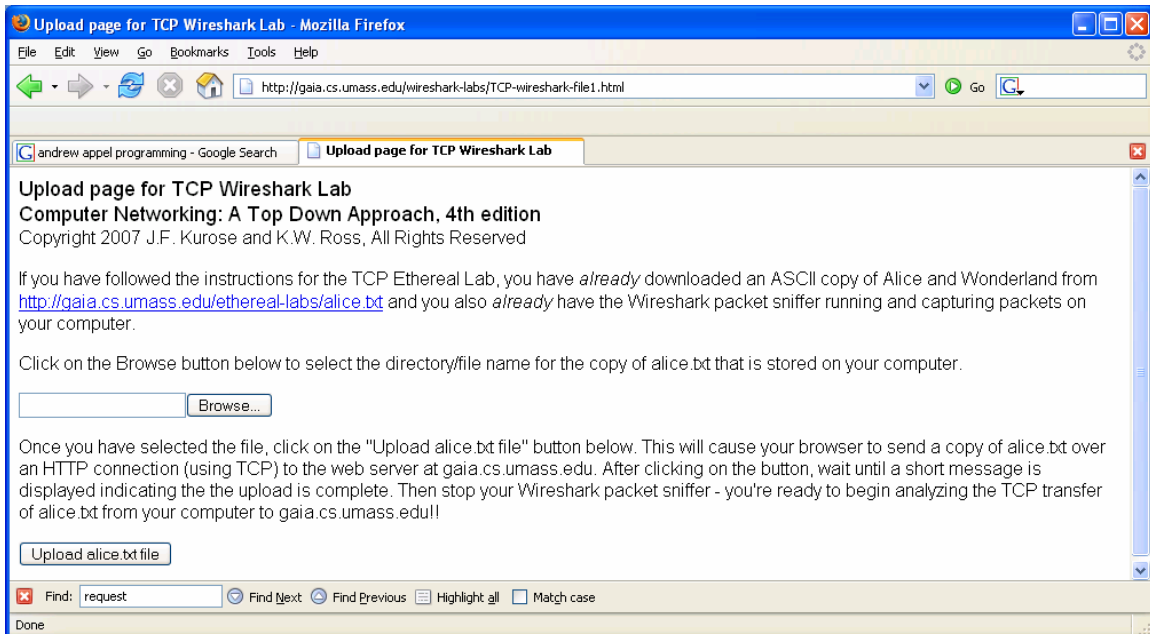
1. Beri contoh penggunaan protokol TCP dan UDP dalam aplikasi sehari-hari.

## PERCOBAAN

### A. Pengiriman data TCP

1. Buka web browser, dan arahkan ke :  
<http://gaia.cs.umass.edu/wireshark-labs/alice.txt>
2. Simpan file alice.txt pada folder yang anda tentukan.
3. Selanjutnya arahkan web browser ke :  
<http://gaia.cs.umass.edu/wireshark-labs/TCP-wireshark-file1.html>

Web tersebut untuk mengupload file alice.txt ke sisi server tersebut. **Jangan** diklik [upload](#) terlebih dahulu.



4. Jalankan wireshark untuk memulai menangkap paket data yang lewat.
5. Klik upload pada no. 2 untuk memulai proses pengiriman data dari client ke server. Setelah ter-upload, maka akan ada pesan keberhasilan proses.
6. Stop wireshark. Sehingga akan terlihat seperti berikut :

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.2.145	128.119.245.12	TCP	1250 > http [SYN] Seq=0 Len=0 MSS=1460
2	0.046402	128.119.245.12	192.168.2.145	TCP	http > 1250 [SYN, ACK] Seq=0 Ack=1 win=5840
3	0.046524	192.168.2.145	128.119.245.12	TCP	1250 > http [ACK] Seq=1 Ack=1 win=65535 [TC
4	0.046963	192.168.2.145	128.119.245.12	HTTP	POST /etherreal-labs/lab3-1-reply.htm HTTP/1
5	0.047339	192.168.2.145	128.119.245.12	HTTP	Continuation or non-HTTP traffic
6	0.128451	128.119.245.12	192.168.2.145	TCP	http > 1250 [ACK] Seq=1 Ack=514 win=6432 Le
7	0.128619	192.168.2.145	128.119.245.12	HTTP	Continuation or non-HTTP traffic
8	0.128717	192.168.2.145	128.119.245.12	HTTP	Continuation or non-HTTP traffic
9	0.214161	128.119.245.12	192.168.2.145	TCP	http > 1250 [ACK] Seq=1 Ack=1966 win=8712 L
10	0.214315	192.168.2.145	128.119.245.12	HTTP	Continuation or non-HTTP traffic
11	0.214415	192.168.2.145	128.119.245.12	HTTP	Continuation or non-HTTP traffic
12	0.298180	128.119.245.12	192.168.2.145	TCP	http > 1250 [ACK] Seq=1 Ack=3418 win=11616
13	0.298326	192.168.2.145	128.119.245.12	HTTP	Continuation or non-HTTP traffic
14	0.381927	128.119.245.12	192.168.2.145	TCP	http > 1250 [ACK] Seq=1 Ack=4870 win=14520
15	0.382241	192.168.2.145	128.119.245.12	HTTP	Continuation or non-HTTP traffic
16	0.382377	192.168.2.145	128.119.245.12	HTTP	Continuation or non-HTTP traffic
17	0.382459	192.168.2.145	128.119.245.12	HTTP	Continuation or non-HTTP traffic
18	0.421386	192.168.2.102	192.168.2.255	NBNS	Name query NB MSHOME<lb>
19	0.466467	128.119.245.12	192.168.2.145	TCP	http > 1250 [ACK] Seq=1 Ack=6322 win=17424
20	0.552453	128.119.245.12	192.168.2.145	TCP	http > 1250 [ACK] Seq=1 Ack=7774 win=20328
21	0.624375	128.119.245.12	192.168.2.145	TCP	http > 1250 [ACK] Seq=1 Ack=8957 win=23232
22	0.624707	192.168.2.145	128.119.245.12	HTTP	Continuation or non-HTTP traffic
23	0.624857	192.168.2.145	128.119.245.12	HTTP	Continuation or non-HTTP traffic
24	0.624943	192.168.2.145	128.119.245.12	HTTP	Continuation or non-HTTP traffic
25	0.708403	128.119.245.12	192.168.2.145	TCP	http > 1250 [ACK] Seq=1 Ack=10409 win=26136
26	0.794139	128.119.245.12	192.168.2.145	TCP	http > 1250 [ACK] Seq=1 Ack=11861 win=29040
27	0.866343	128.119.245.12	192.168.2.145	TCP	http > 1250 [ACK] Seq=1 Ack=13053 win=31944
28	0.868855	192.168.2.145	128.119.245.12	HTTP	Continuation or non-HTTP traffic
29	0.869431	192.168.2.145	128.119.245.12	HTTP	Continuation or non-HTTP traffic
30	0.869544	192.168.2.145	128.119.245.12	HTTP	Continuation or non-HTTP traffic
31	0.950346	128.119.245.12	192.168.2.145	TCP	http > 1250 [ACK] Seq=1 Ack=14505 win=32767
32	1.036229	128.119.245.12	192.168.2.145	TCP	http > 1250 [ACK] Seq=1 Ack=15957 win=32767
33	1.108269	128.119.245.12	192.168.2.145	TCP	http > 1250 [ACK] Seq=1 Ack=17149 win=32767

Frame 1 (62 bytes on wire, 62 bytes captured)

- Ethernet II, Src: Netgear\_61:8e:6d (00:09:5b:61:8e:6d), Dst: LinksysG\_45:90:a8 (00:0c:41:45:90:a8)
- Internet Protocol, Src: 192.168.2.145 (192.168.2.145), Dst: 128.119.245.12 (128.119.245.12)
- Transmission Control Protocol, Src Port: 1250 (1250), Dst Port: http (80), Seq: 0, Len: 0

```

0000  00 0c 41 45 90 a8 00 09 5b 61 8e 6d 08 00 45 00  ..AE.... [a.m..E.
0010  00 30 2b 6b 40 00 80 06 96 9f c0 a8 02 91 80 77  ..0+k@... ..w
0020  f5 0c 04 e2 00 50 c2 67 22 99 00 00 00 70 02  ....P.g .....p.
0030  ff ff 60 2f 00 00 02 04 05 b4 01 01 04 02      .../.....

```

File: "C:\DOCUMENT~1\PAULAW~1\LOCAL5~1\Temp\ether\XXXXa03100" 165 KB 00:00:09 P: 214 D: 214 M: 0 Drops: 0

NB: Pada keterangan diatas, terdapat informasi protokol HTTP, Continuation or non-HTTP traffic. Ini maksudnya bahwa ada banyak segmen TCP yang digunakan untuk membawa satu pesan HTTP.

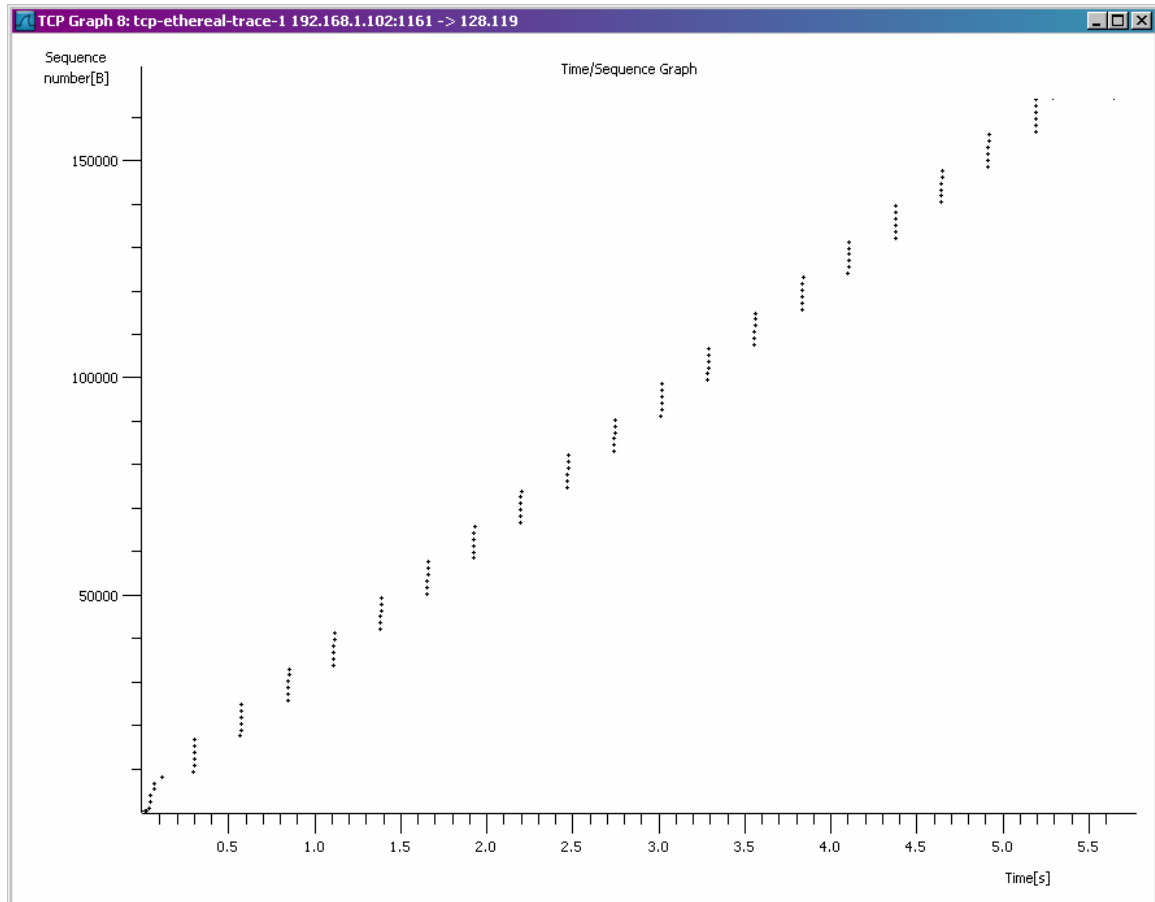
7. Filterlah wireshark diatas dengan mengetikkan "tcp" pada bagian filter.
8. Berapa no IP dan port number TCP yang digunakan oleh client dan server ?

## B. Dasar Protokol TCP

9. Berapa sequence number dari TCP SYN segmen yang digunakan untuk memulai koneksi TCP antara client dan server ?
10. Berapa sequence number dari segmen SYNACK yang dikirim oleh server ke client ? Apa nilai dari ACK dari segmen SYNACK tersebut ?
11. Carilah 6 segmen pertama dalam koneksi TCP setelah terjadinya 3-way handshake ? Amati perbedaan dari tiap segmen TCP dikirim sampai ACK diterima ? Berapa nilai RTT dari masing-masing 6 segmen yg pertama tersebut ? Berapa panjang (byte) dari masing-masing 6 segmen TCP yang pertama ?
12. Apakah ada segmen yang dikirim ulang ? Bagaimana anda mengeceknya ?

### C. TCP Congestion Control

13. Dari wireshark, pilih menu : Statistics -> TCP Stream Graph -> Time-Sequence-Graph(Stevens), maka anda akan mendapatkan plot grafik seperti berikut :



14. Dari data diatas, dimanakah congestion avoidance terjadi ?

### D. Pengiriman Data dengan UDP

1. Untuk proses pengiriman data dengan UDP akan dijelaskan pada waktu praktikum.

### LAPORAN RESMI

1. Berikan kesimpulan hasil praktikum yang anda lakukan.