



FAKULTAS  
**ILMU  
KOMPUTER**

**Worksheet Jaringan Komunikasi Data  
(CSIM603154)– 2020-2021 Gasal**

Week : 5

Topic : TCP Reliable Data Transfer, ACK  
Generation and Congestion Control

Lecturer : 1. Ari Wibisono  
2. Muhammad Anwar Ma'sum

Name: Samuel Ludwig Ian  
NPM : 1806191471

## A. TCP Reliable Data Transfer

1. Jelaskan dengan kalimat anda sendiri apa yang dimaksud dengan reliable data transfer?

Reliable Data Transfer (RDT) adalah protokol untuk memastikan data yang dikirim sampai pada tujuan secara terurut tanpa kesalahan (no error & in order), sesuai dengan urutan paket saat data dikirimkan.

2. Jelaskan bagaimana cara kerja reliable data transfer pada TCP?

RDT bekerja dengan memanfaatkan handshake yang ada, yakni dengan berhenti sejenak untuk menunggu adanya konfirmasi yang dikirimkan dari pihak penerima yang menandakan adanya paket yang telah mereka terima. Hal ini tidak mengalami masalah pada jarak yang dekat, namun pada jarak yang jauh, mulai muncul berbagai permasalahan sehingga dilakukan pembaruan untuk protokol tersebut.

## B. TCP ACK Generation

Diketahui dua host (client dan server) berkomunikasi dengan protocol TCP (RFC 1122, RFC 2581). Host A memiliki timeout 3s dan akan melakukan fast retransmit jika terjadi timeout atau menerima 3-duplicate ACK. Waktu tempuh dari host A ke B atau sebaliknya adalah 20ms. Asumsikan bahwa kedua host telah menjalankan handshaking. Proses transmisi data dari host A dan B dengan rincian sebagai berikut:

- Saat T=100 host A mengirim segment seq=0 sebanyak 20 bytes
- Saat T=120 host A mengirim segment seq=20 sebanyak 20 bytes
- Saat T=200 host A mengirim segment seq=40 sebanyak 20 bytes tetapi hilang di perjalanan (loss)
- Saat T=250 host A mengirim segment seq=60 sebanyak 20 bytes tetapi hilang di perjalanan (loss)
- Saat T=300 host A mengirim segment seq=80 sebanyak 20 bytes tetapi hilang di perjalanan (loss)
- Saat T=400 host A mengirim segment seq=100 sebanyak 20 bytes
- Saat T=500 host A mengirim segment seq=120 sebanyak 20 bytes
- Saat T = 700 host A mengirim segment seq=60 sebanyak 20 bytes
- Saat T = 800 host A mengirim segment seq=80 sebanyak 20 bytes
- Saat T = 900 host A mengirim segment seq=140 sebanyak 20 bytes
- Saat T = 1000 host A mengirim segment seq=40 sebanyak 20 bytes
- Saat T = 1100 host A mengirim segment seq=160 sebanyak 20 bytes, tetapi ACK balasan dari host B hilang di perjalanan (loss).
- Saat T = 1200 host A mengirim segmen seq=180 sebanyak 20 bytes.

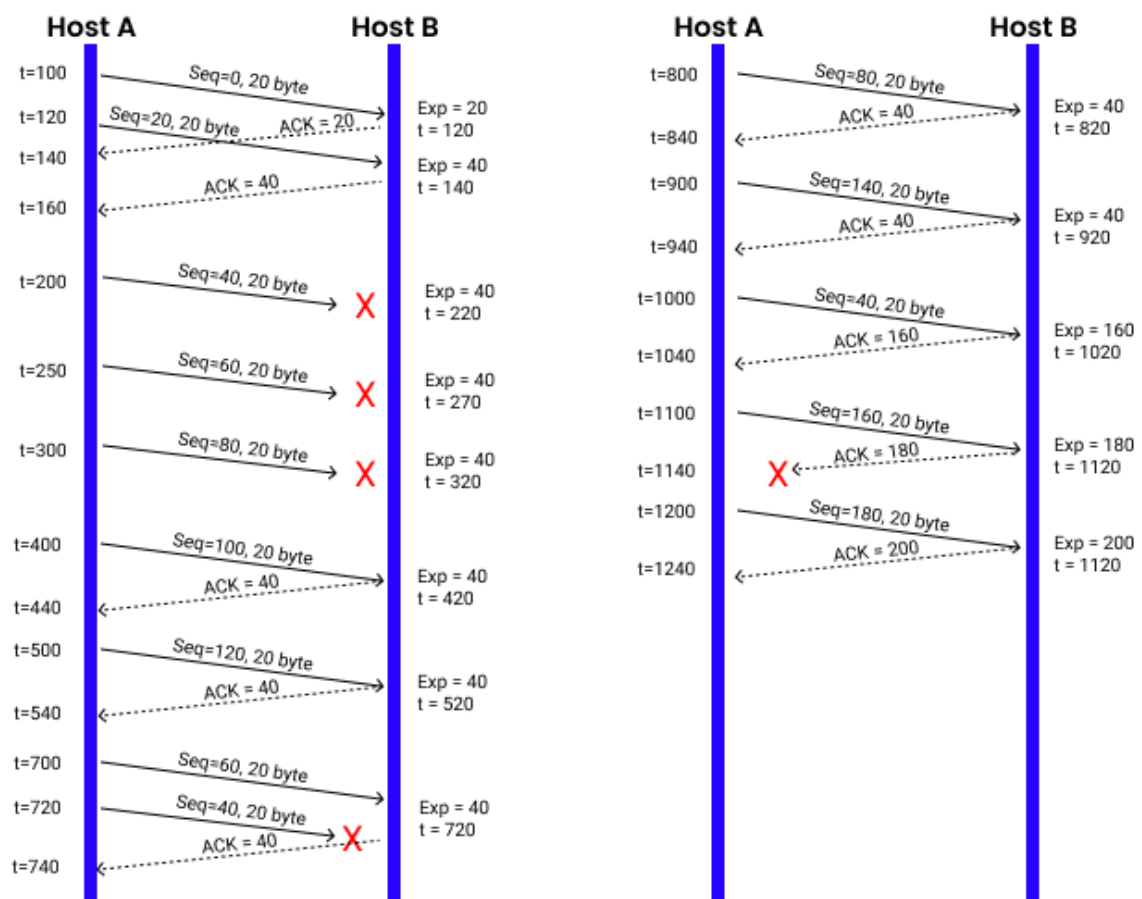
1. Isilah tabel pengiriman ACK oleh host B berikut

T(ms)	Host B Mengirimkan ACK (Ya/Tidak)	Nomor ACK (Jika Ada)	T(ms)	Host B Mengirimkan ACK (Ya/Tidak)	Nomor ACK (Jika Ada)
120	Ya	20	720	Ya	40
140	Ya	40	820	Ya	40
220	Tidak	-	920	Ya	40
270	Tidak	-	1020	Ya	160
320	Tidak	-	1120	Ya	180
420	Ya	40	1220	Ya	200
520	Ya	40			

2. Apakah Host A melakukan fast retransmit? Jika iya pada T berapa saja dan berapa nomor sequence yang dikirimkan?

Ya, pada 740ms seharusnya Host A melakukan fast retransmit dengan nomor sequence 40 (meskipun tidak tertulis pada proses transmisi data bahwa Host A mengirimkan sequence 40 pada T = 740ms dan bagaimana status penerimaan oleh Host B)

3. Gambarkan diagram interaksi antara host A dan B



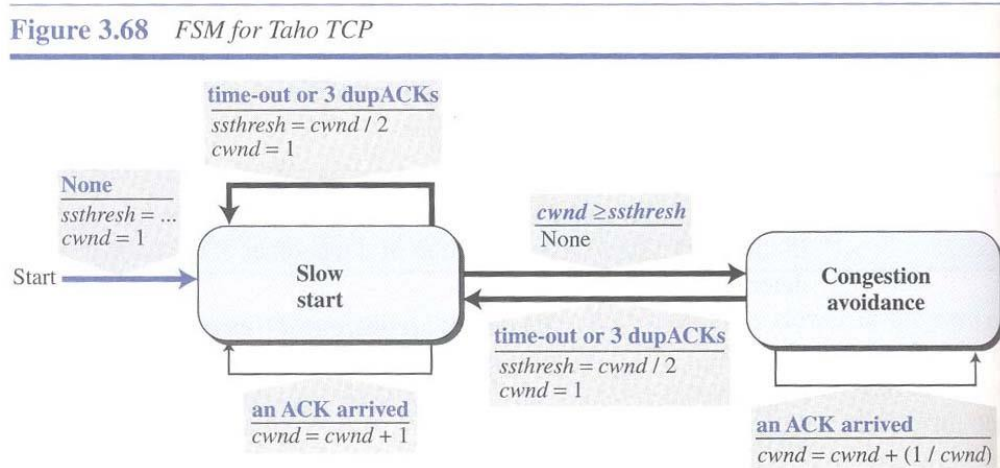
**Samuel Ludwig Ian / 1806191471 / Jarkomdat B**

### C. TCP Congestion Control

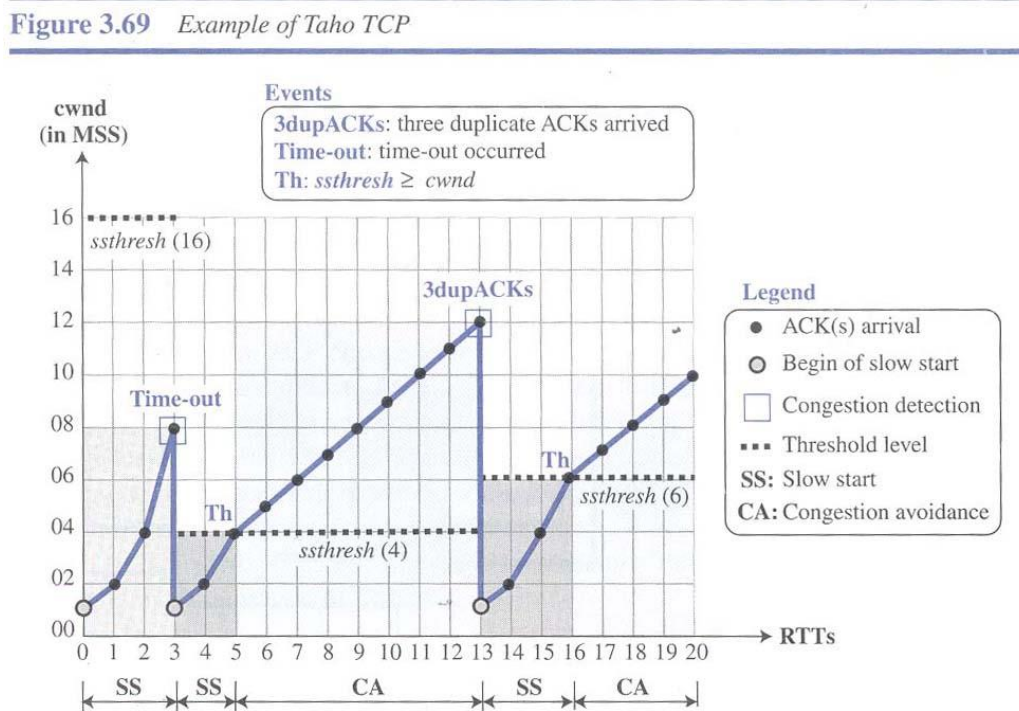
TCP Tahoe adalah salah satu tipe TCP yang menggunakan congestion control. Anda dapat mempelajari TCP Tahoe pada referensi berikut:

- Computer Networks by Andrew S. Tanenbaum
- Computer Networking: A Top-down Approach by Jim Kurose
- Penjelasan/tutorial pada berbagai links, websites, media, etc.

TCP Tahoe menjalankan fungsi congestion control dengan mekanisme seperti yang dijelaskan pada diagram finite state machine (FSM) berikut.



Gambar berikut merupakan contoh dari TCP Tahoe dalam mengatur congestion windownya.



Jelaskan dengan kalimat anda sendiri (deskripsikan) kurva di atas sesuai dengan cara kerja TCP Tahoe pada gambar sebelumnya.

Pada awalnya, slow start threshold (sssthresh) di set pada 16, dan congestion window (cwnd) di set pada 1. Setelah itu, cwnd akan bertambah terus secara eksponensial untuk setiap ACK yang diterima oleh pengirim karena bertambahnya jumlah maksimal message (masing-masing dengan panjang  $n$ ) yang dapat dikirimkan (cwnd). Kemudian, pada  $cwnd = 8$ , terjadi time-out yang menyebabkan sssthresh di set menjadi 4 (dari  $cwnd$  time-out / 2), cwnd di set ulang menjadi 1, dan sistem masuk ke dalam mode slow start lagi.

Setelah beberapa lama di dalam mode slow start, cwnd akan kembali bertambah secara eksponensial pada sistem sehingga akan melewati sssthresh. Hal ini menyebabkan sistem masuk ke dalam mode congestion avoidance, dimana cwnd akan bertambah 1 untuk setiap ACK yang diterima oleh pengirim. Pada Round Trip Time (RTT) = 13, didapatkan adanya 3 ACK yang memiliki nilai duplikat. Hal ini menandakan adanya paket yang hilang, sehingga sistem kembali masuk ke dalam slow start, sssthresh diubah kembali menjadi 6, dan cwnd diubah menjadi 1.

Setelah beberapa lama dalam slow start, sistem akan mengalami peningkatan cwnd secara eksponensial yang menyebabkan cwnd lebih besar dari sssthresh. Kemudian sistem akan masuk ke dalam mode congestion avoidance, dan seterusnya sampai selesai.