

# **William Stallings**

# **Data and Computer**

# **Communications**

---

## **Chapter 20**

## **Transport Protocols**

# **Koneksi yang diorientasi pada tata cara pengangkutan atau transportasi**

---

- Kemungkinan terhubung
- Penetapan
- Penghentian pemeliharaan
- Keandalan
- TCP

# Keandalan dalam layanan jaringan

---

- Panjangnya pesan diasumsikan terserah
- Keandalan pengiriman diasumsikan hampir 100% oleh layanan jaringan
  - Keandalan paket dalam memilih jaringan yang menggunakan X.25
  - frame relay menggunakan LAPF control protocol
  - IEEE 802.3 menggunakan koneksi yang diorientasikan pada layanan LLC
- Layanan transportasi dari ujung ke ujung protokol antara dua sistem pada jaringan yang sama

# **Protokol Transportasi sederhana pada umumnya**

---

- Pengalamatan
- Multiplexing
- Flow Control
- Hubungan penetapan dan penghentian

# Pengalamatan

---

- Target pengguna ditetapkan oleh:
  - Identifikasi pengguna
    - biasanya host, port
      - Disebut socket di TCP
    - Port menghadirkan transportasi service (TS) tertentu pengguna
  - Identifikasi Transport entity
    - Umumnya hanya satu per host
    - Jika lebih dari satu, kemudian biasanya salah satu dari masing-masing jenis
      - menetapkan protokol transportasi (TCP, UDP)
  - Alamat host
    - Dipasang alat jaringan
    - Didalam internet,
  - Nomer jaringan

# Pencarian alamat

---

- Ada 4 metode
  - Mengetahui alamat sebelum waktu yang ditetapkan
    - Koleksi perencanaan peralatan jaringan
  - Alamat dimengerti
  - Menyebut server
  - Proses pengiriman meminta ke alamat yang diketahui

# Multiplexing

---

- Para pemakai dikerjakan pada protokol transportasi yang sama
- Pemakai dikenali dengan nomer port atau service access point (SAP)
- Kemungkinan multiplex dengan layanan network menggunakan
  - multiplexing a single virtual X.25 sirkuit to a number of transport service user
    - X.25 charges per virtual circuit connection time

# Flow Control

---

- Delay yang panjang antara kesatuan pengiriman dibandingkan dengan waktu trasmisi sebenarnya
  - Delay dalam komunikasi dari informasi flow kontrol
- Delay transmisi variabel
  - Sulit untuk digunakan dalam timeout
- Flow may be controlled because:
- Aliran data dapat dikontrol karena:
  - User penerima tidak dapat melanjutkan
  - Kesatuan pengiriman tidak dapat melanjutkan
- Hasil dari buffer memenuhi



# Mengkopi dengan Syarat Flow Control(1)

---

- Do nothing
  - Segment yang overflow dibuang
  - Kesatuan pengiriman akan mendapat ACK dan mentransmisikan kembali
    - Akan ditambahkan pada data yang masuk
- Menolak bagian selanjutnya
  - Bagian yang janggal
  - Sambungan multiplex dikontrol di jumlah aliran data

# Mengkopi dengan Syarat Flow Control(2)

---

- Menggunakan sliding window protokol tertentu
  - Lihat bab 7 untuk detail operasinya
  - Bekerja dengan baik di network yang diandalkan
    - Kegagalan menerima ACK didapat sebagai indikasi flow control
  - Tidak bekerja dengan baik di network yang tidak dapat diandalkan
    - Tidak dapat membedakan antara segmen yang hilang dengan flow control
- Menggunakan credit scheme

# Credit Scheme

---

- Kontrol yang lebih besar di jaringan yang diandalkan
- Lebih efektif di jaringan yang tidak diandalkan
- Pasangan flow control dari ACK
  - Kemungkinan ACK tanpa granting credit dan sebaliknya
- Setiap octet mempunyai nomor urut
- Setiap segmen transportasi mempunyai nomor urut, nomor permintaan dan ukuran window pada header

# **Penggunaan dari Header Fields**

---

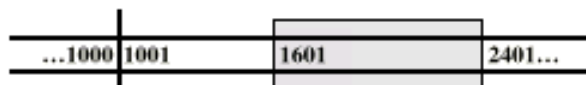
- Ketika pengiriman, seq. number is that of first octet in segment
- ACK termasuk  $AN=i$ ,  $W=j$
- Semua octets melewati  $SN=i-1$  acknowledged
  - Next expected octet is  $i$
- Meminta ijin untuk mengirim additional window dari  $W=j$  octets
  - i.e. octets through  $i+j-1$

# Credit Allocation

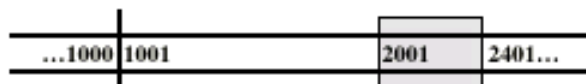
Transport Entity A



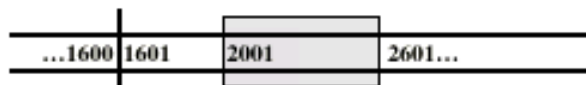
A may send 1400 octets



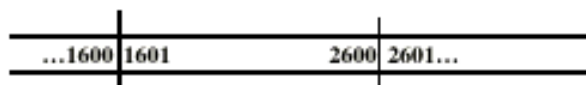
A shrinks its transmit window with each transmission



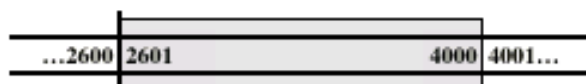
A adjusts its window with each credit



A exhausts its credit



A receives new credit



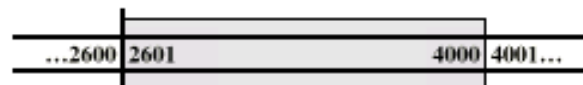
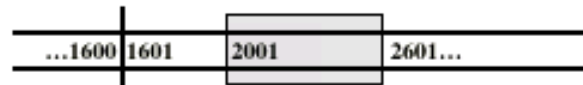
Transport Entity B



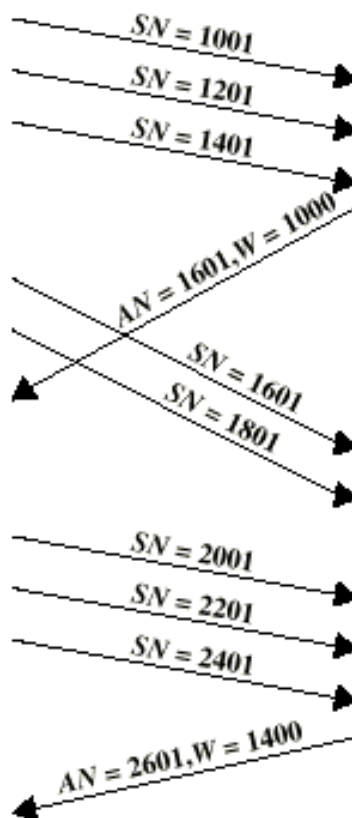
B is prepared to receive 1400 octets, beginning with 1001



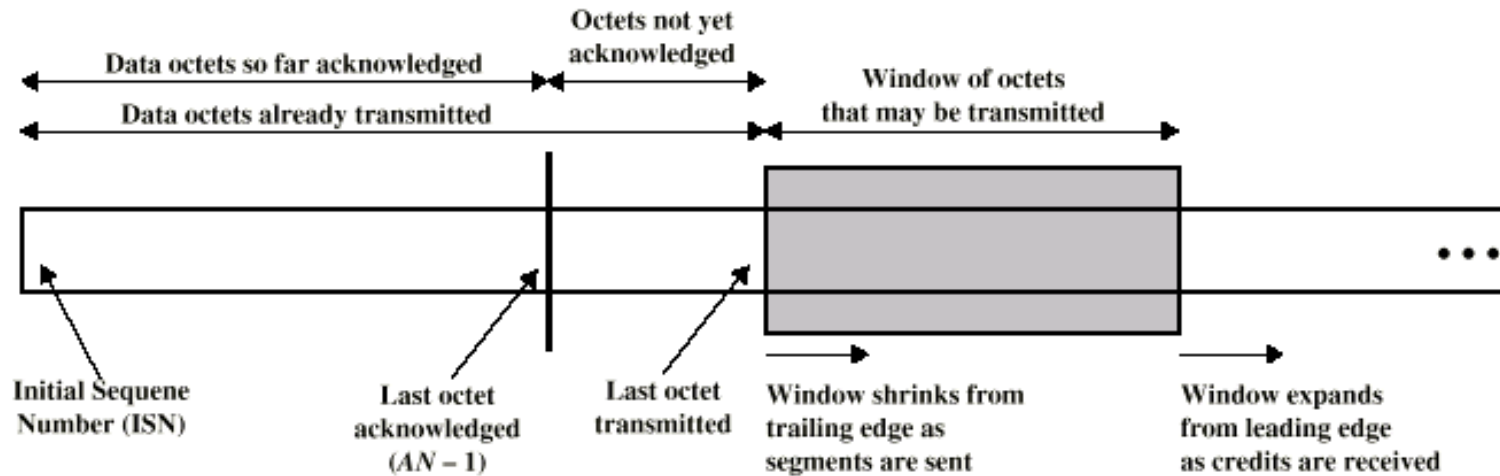
B acknowledges 3 segments (600 octets), but is only prepared to receive 200 additional octets beyond the original budget (i.e., B will accept octets 1601 through 2600)



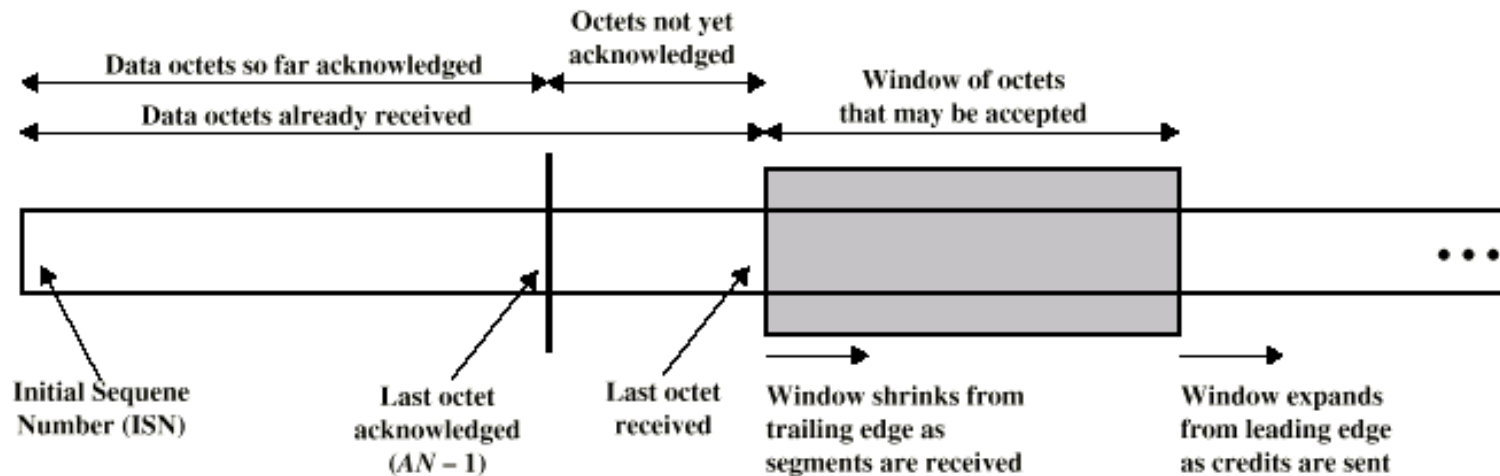
B acknowledges 5 segments (1000 octets) and restores the original amount of credit



# Pengiriman dan Penerimaan Sebenarnya



(a) Send sequence space



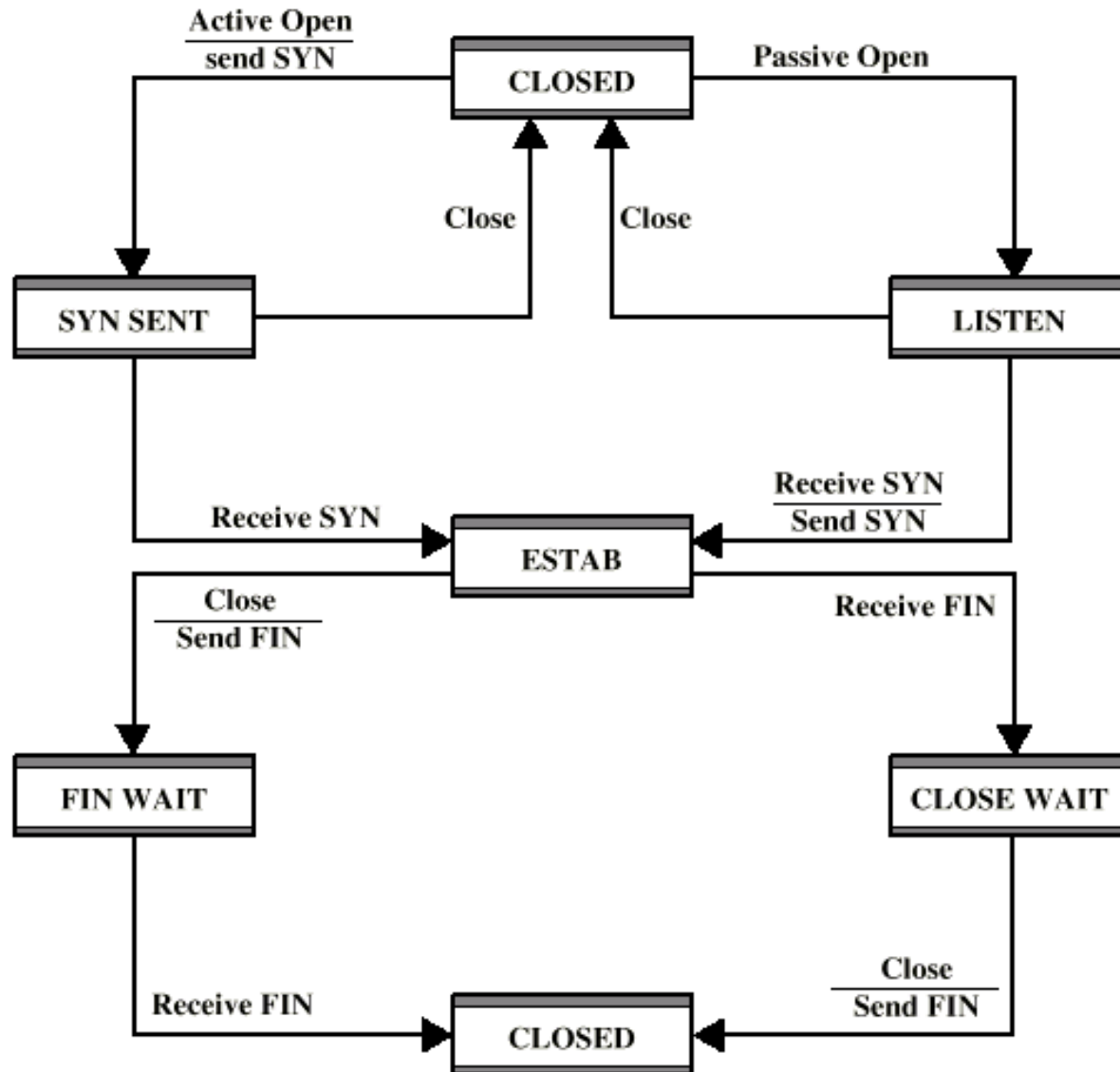
(b) Receive sequence space

# **Establishment dan Termination**

---

- Mengijinkan semua yang ada sampai akhir
- Negosiasi dari parameter pilihan
- Alokasi triger dari kesatuan pengangkutan sumber
- Dengan persetujuan bersama

# Diagram Bagian Koneksi



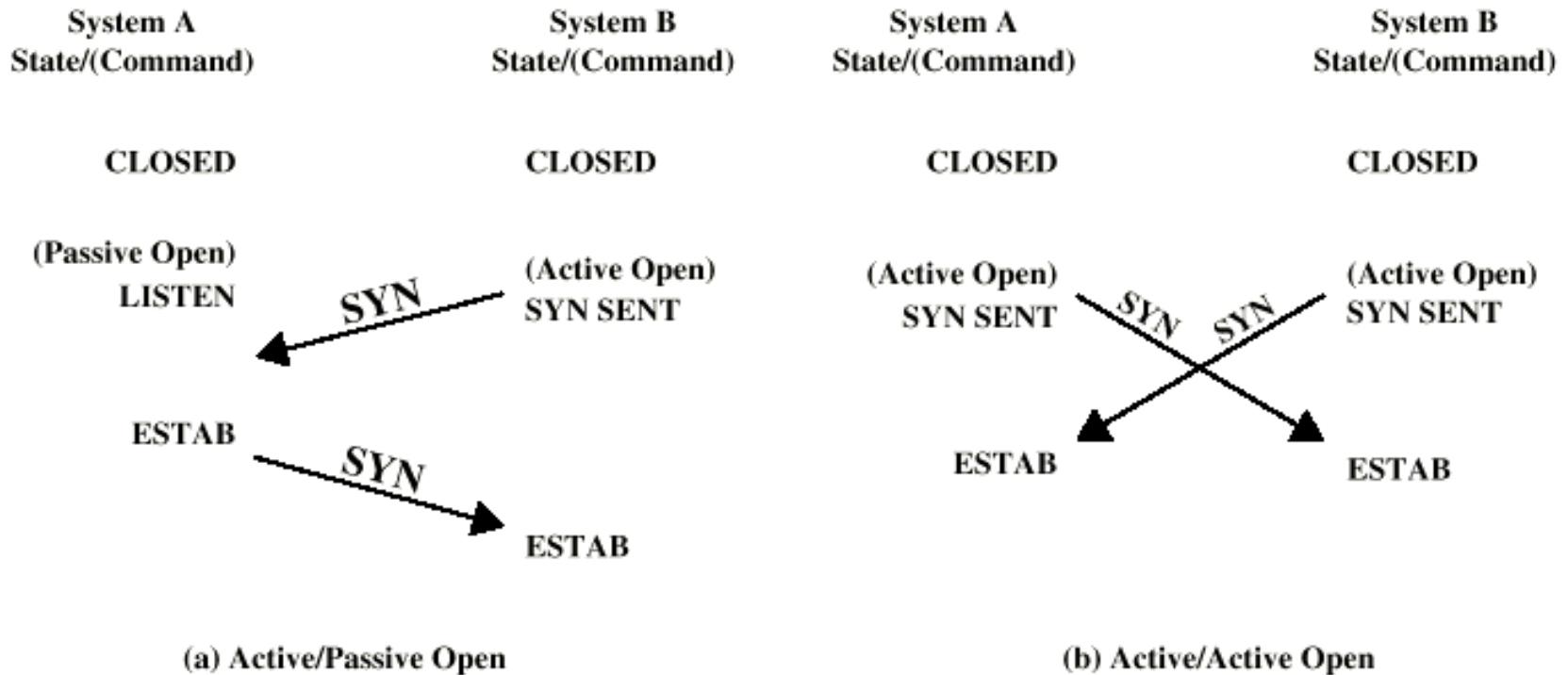
Legend:

Event  
Action

State



# Connection Establishment



# Not Listening

---

- Menolak dengan RST (Reset)
- Permintaan antrian sampai hasilnya match
- Sinyal TS memberitahu user atas permintaan yang tertunda
  - May replace passive open with accept

# Termination

---

- Salah satu atau kedua sisi
- Dengan persetujuan bersama
- Pemberhentian yang tiba-tiba
- Atau pemberhentian yang lemah
  - Close wait state must accept incoming data until FIN received

# Side Initiating Termination

---

- Pengguna TS menutup request
- Transport entity mengirim FIN, meminta termination
- Tempat koneksi di FIN WAIT state
  - Melanjutkan untuk menerima data dan mengirim data ke user
  - Tidak mengirim data lagi
- Ketika menerima FIN, memberitahukan user dan menutup koneksi

# **Side Not Initiating Termination**

---

- FIN received
- Inform TS user Place connection in CLOSE WAIT state
  - Continue to accept data from TS user and transmit it
- TS user issues CLOSE primitive
- Transport entity sends FIN
- Menutup koneksi
- Semua data ditransmisikandari kedua sisi
- Kedua sisi setuju untuk diakhiri

# Unreliable Network Service

---

- E.g.
  - internet menggunakan IP,
  - frame relay menggunakan LAPF
  - IEEE 802.3 menggunakan unacknowledged connectionless LLC
- Bagian – bagiannya bisa hilang
- Bagian – bagian yang tiba bisa sangat banyak/melebihi batas

# Problems

---

- Ordered Delivery
- Retransmission strategy
- Duplication detection
- Flow control
- Connection establishment
- Connection termination
- Crash recovery

# Ordered Delivery

---

- Segments boleh tiba out of order
- Number segments sequentially
- TCP numbers each octet sequentially
- Segments are numbered by the first octet number in the segment



# Retransmission Strategy

---

- Segment rusak saat pemindahan
- Segment gagal tiba
- Transmitter tidak mengetahui kegagalan
- Receiver must acknowledge successful receipt
- Menggunakan pengakuan kumulatif
- Time out yang menantikan ACK triggers re-transmission

# Timer Value

---

- Fixed timer
  - Based on understanding of network behavior
  - Tidak bisa menyesuaikan untuk mengubah kondisi jaringan
  - Too small leads to unnecessary re-transmissions
  - Too large and response to lost segments is slow
  - Should be a bit longer than round trip time
- Adaptive scheme
  - May not ACK immediately
  - Can not distinguish between ACK of original segment and re-transmitted segment
  - Conditions may change suddenly

# Duplication Detection

---

- jika ACK hilang, bagiannya akan dikirimkan kembali
- Receiver harus mengenali salinan/duplikasinya
- Duplicate sebelumnya diterima untuk menutup koneksi
  - Receiver menggap ACK hilang dan Asks adalah salinannya
  - Sender jangan bingung dengan banyaknya Asks
  - Sequence number space large enough to not cycle within maximum life of segment
- Duplicate diterima setelah menutup koneksi

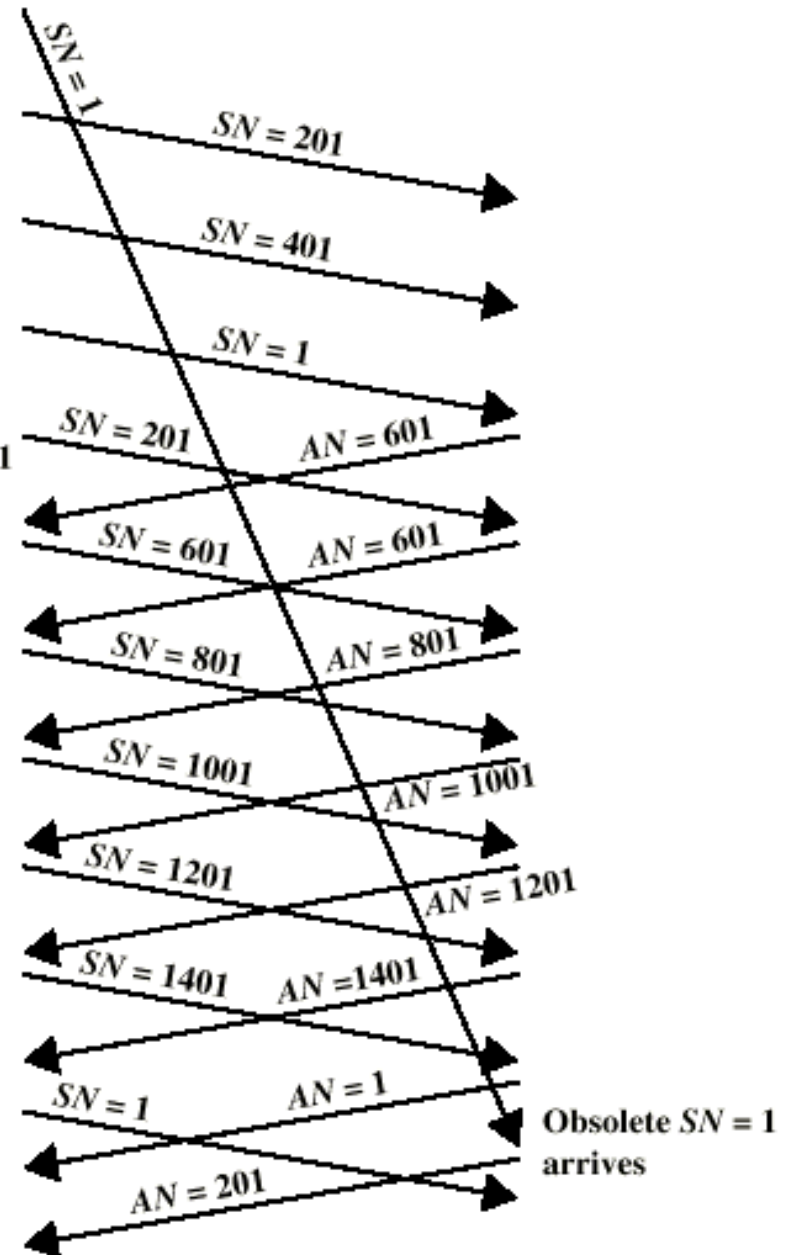
# Incorrect Duplicate Detection

Transport  
Entity A

Transport  
Entity B

A times out and  
retransmits  $SN = 1$

A times out and  
retransmits  $SN = 201$



# Flow Control

---

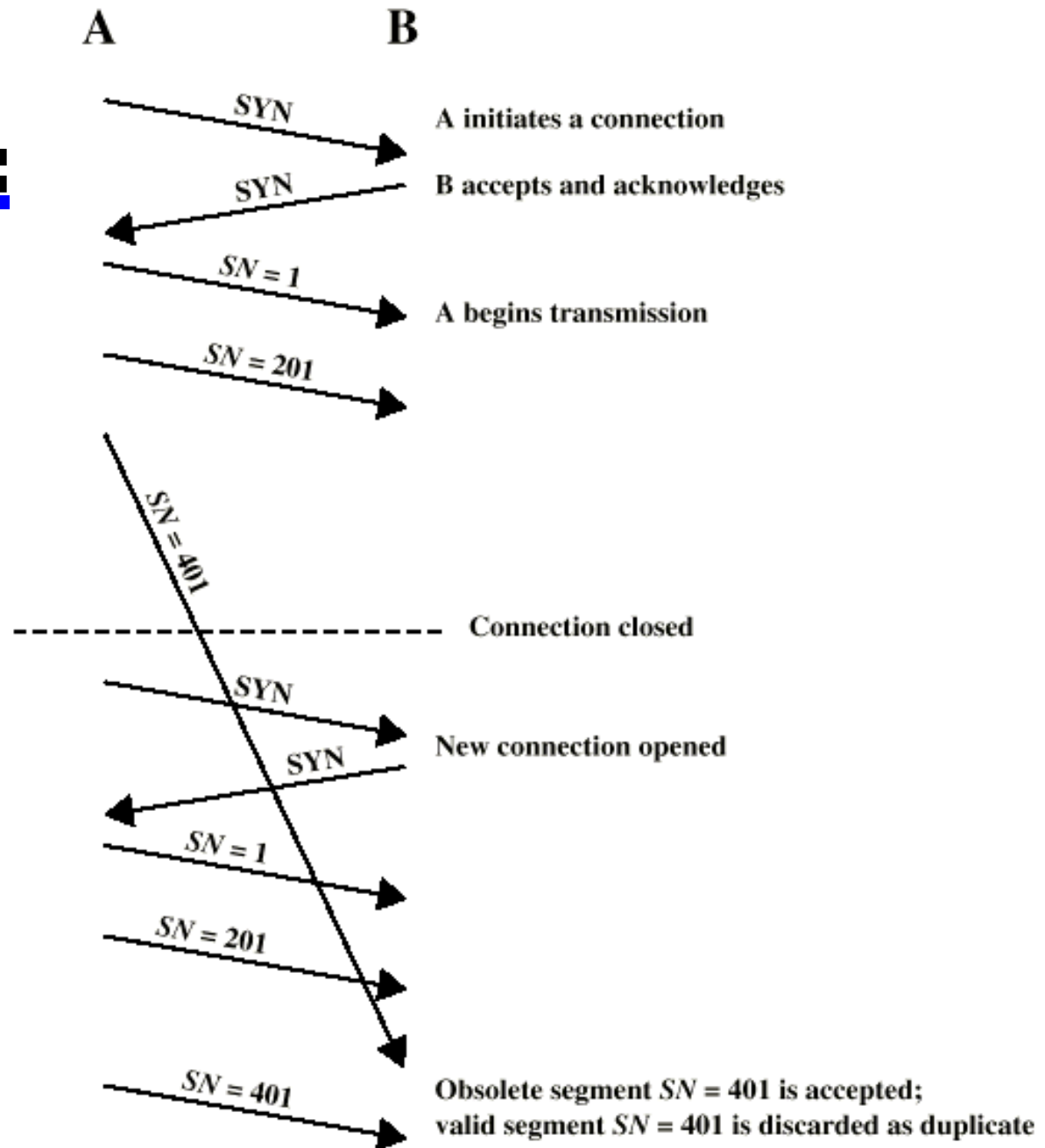
- Alokasi credit
- Masalah jika  $AN=i$ ,  $W=0$  menutup window
- kirim  $AN=i$ ,  $W=j$  untuk kembali membuka, tapi window akan hilang
- Sender menganggap window tertutup, receiver menganggap window terbuka
- Menggunakan window timer
- jika waktunya berakhir, kirim sesuatu
  - Could be re-transmission of previous segment

# Connection Establishment

---

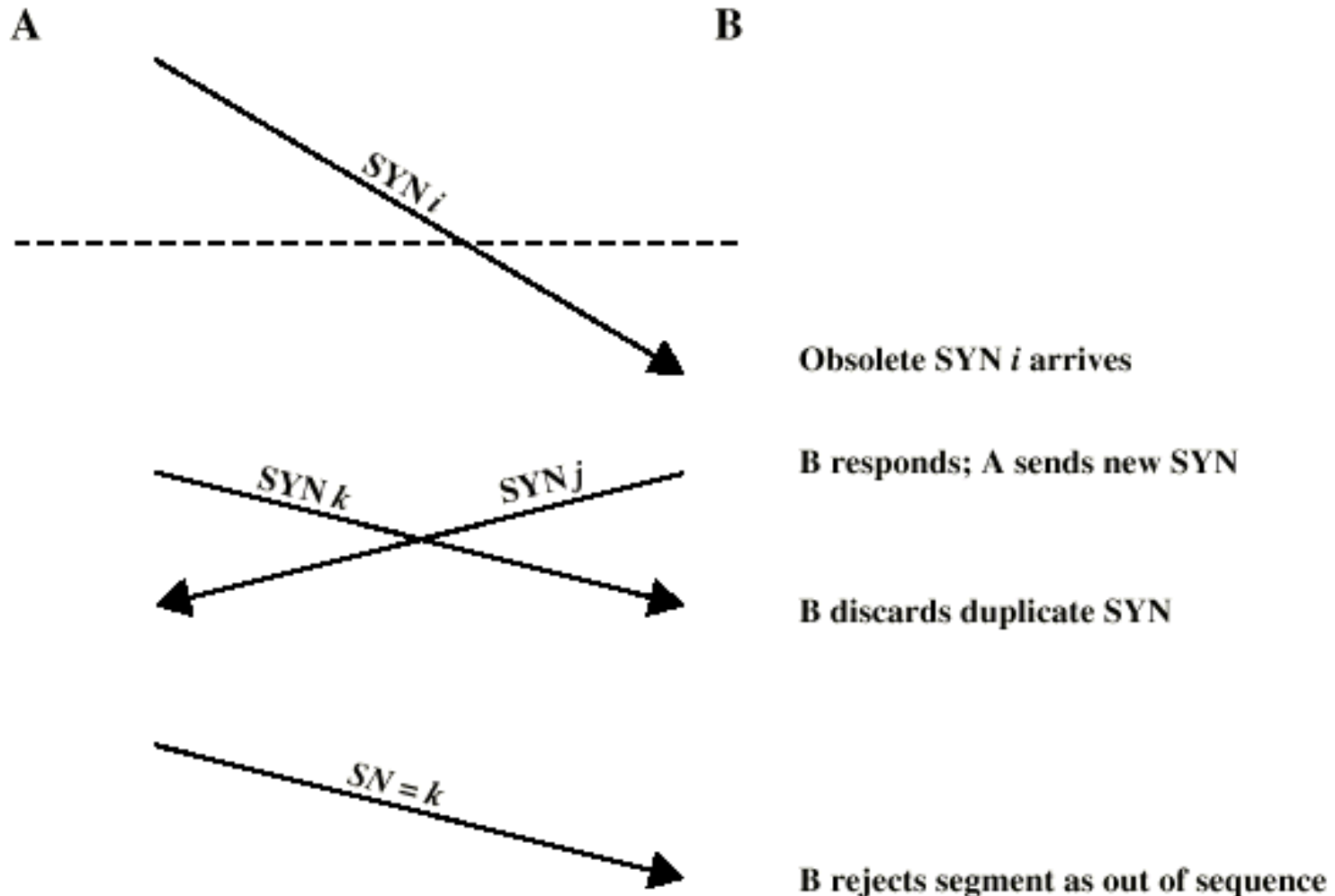
- Two way handshake
  - A mengirim SYN, B membalas dengan SYN
  - Hilangnya SYN dapat diatasi dengan re-transmission
    - Can lead to duplicate SYNs
  - Ignore duplicate SYNs once connected
- Kehilangan atau keterlambatan bagian data dapat menyebabkan masalah dalam koneksi.
  - Segment dari old connection
  - Start segment numbers are removed from previous connection
    - Use SYN i
    - Need ACK to include i
    - Three Way Handshake

# Two Way Handshake: Obsolete Data Segment



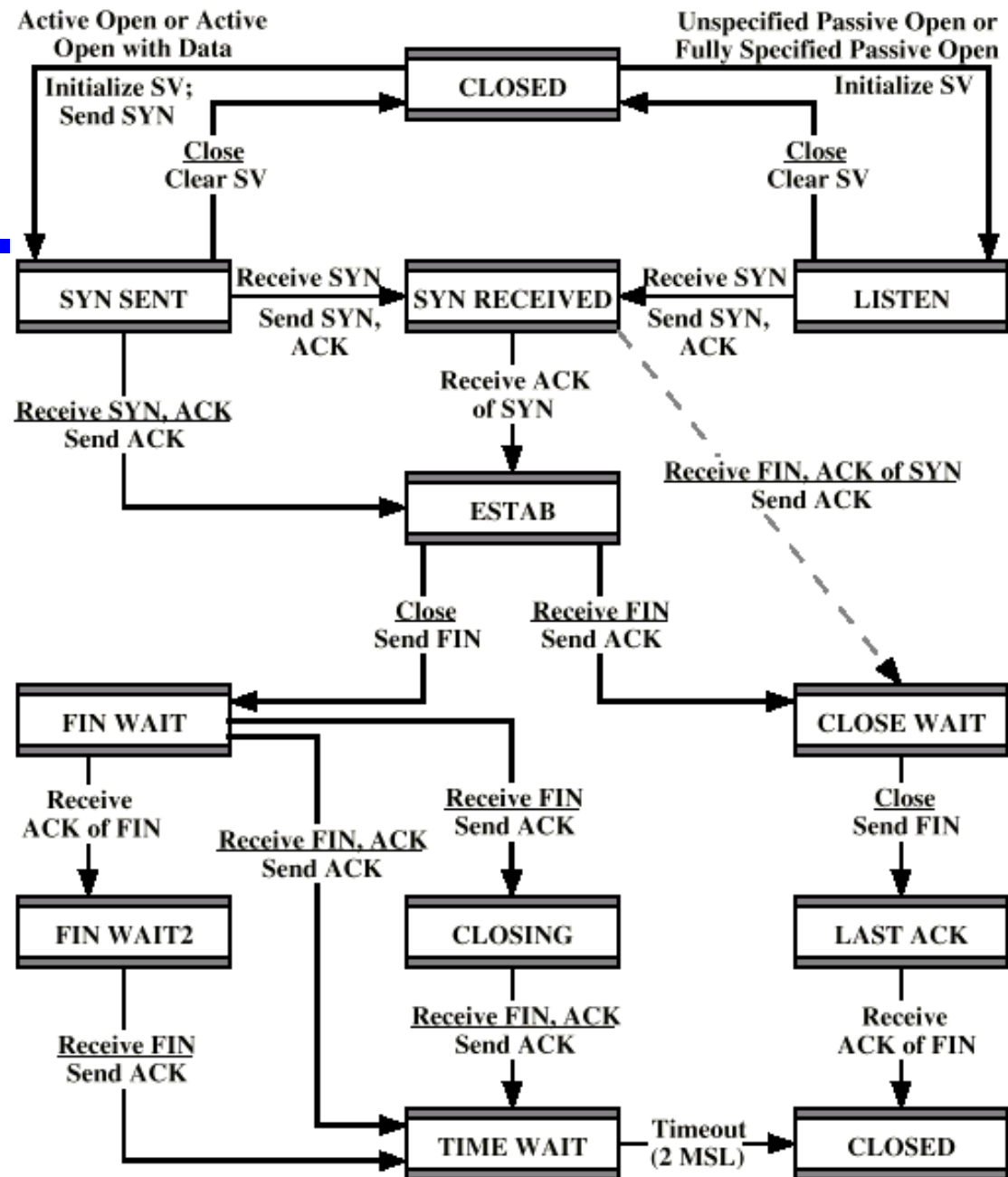
# Two Way Handshake: Obsolete SYN Segment

---



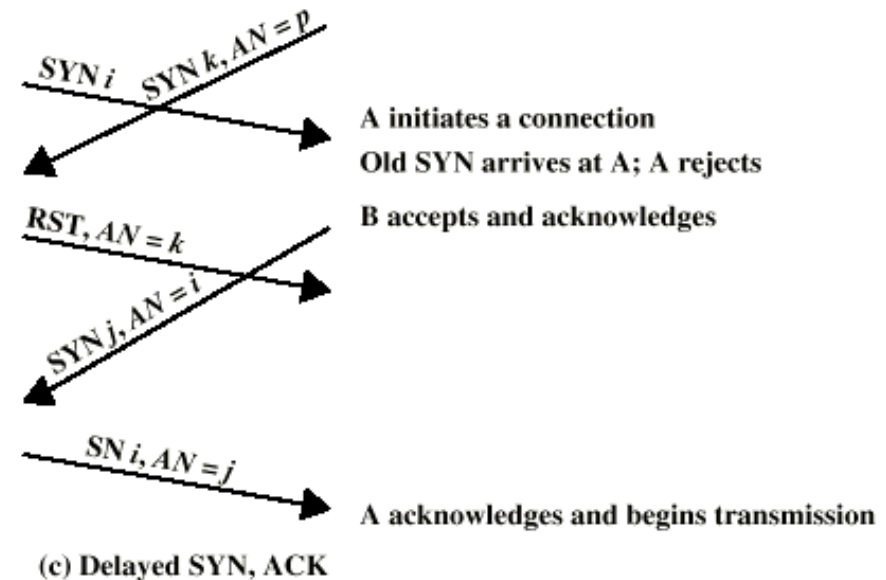
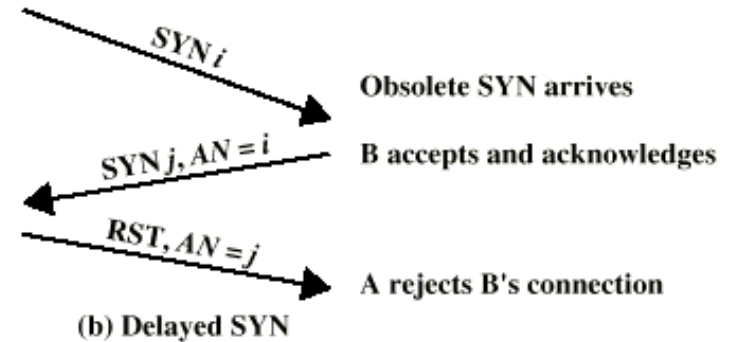
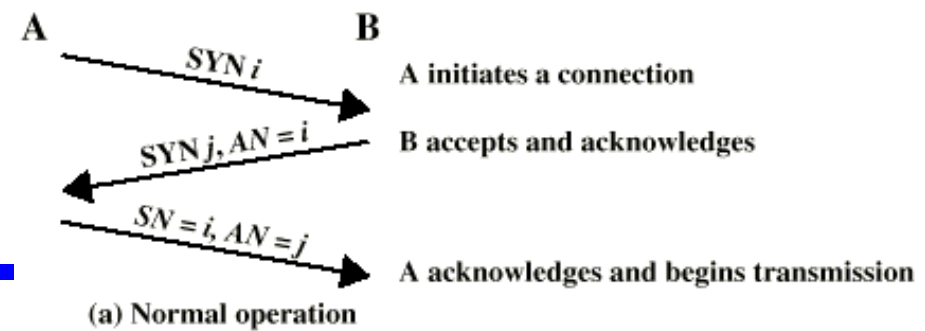


# Three Way Handshake: State Diagram



SV = state vector  
MSL = maximum segment lifetime

# Three Way Handshake: Examples



# Connection Termination

---

- Entity in CLOSE WAIT state sends last data segment, followed by FIN
- FIN tiba sebelum bagian data yang terakhir
- Receiver menerima FIN
  - Tutup koneksi
  - Kehilangan bagian data yang terakhir
- Associate mengurutkan nomor dengan FIN
- Receiver menunggu semua bagian sebelum FIN mengurutkan nomor
- Loss of segments and obsolete segments
  - Must explicitly ACK FIN

# Graceful Close

---

- Kirim FIN i dan menerima AN i
- Menerima FIN j dan mengirim AN j
- Wait twice maximum expected segment lifetime

# Failure Recovery

---

- Setelah restart semua bagian, info akan hilang
- Koneksi terbuka setengah
  - Side that did not crash still thinks it is connected
- Menutup koneksi menggunakan persistence timer
  - Wait for ACK for (time out) \* (number of retries)
  - ketika expired, tutup koneksi dan inform user
- Send RST i in response to any i segment arriving
- User harus memutuskan kapan koneksi kembali
  - Masalah masalah dengan kehilangan data

# TCP & UDP

---

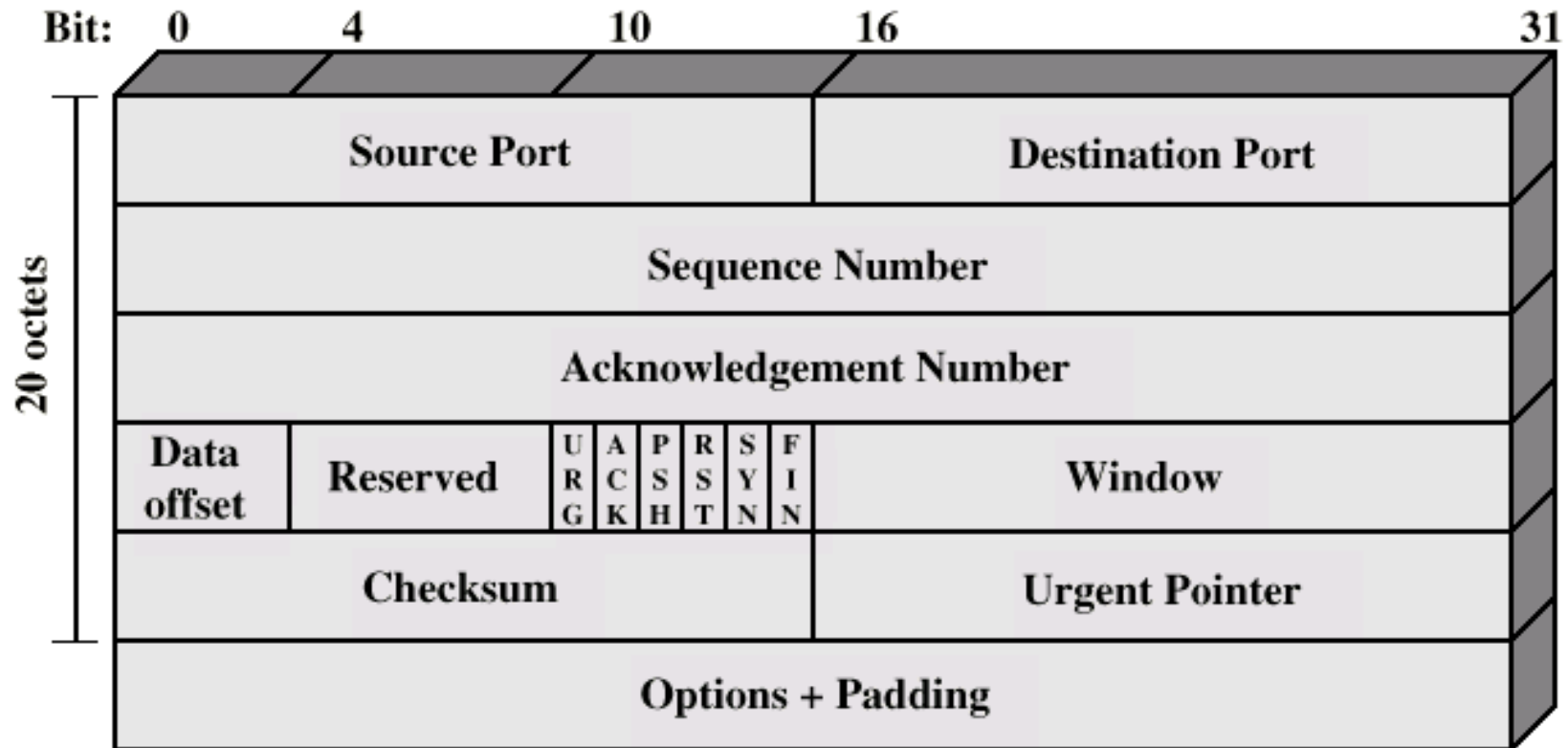
- Transmission Control Protocol
  - Connection oriented
  - RFC 793
- User Datagram Protocol (UDP)
  - Connectionless
  - RFC 768

# TCP Services

---

- Tersedianya komunikasi antara pasangan suatu proses
- Macam-macam jarak dari jaringan yang tersedia dan tidak serta internet
- Dua pelabelan fasilitas
  - Data stream push
    - User TCP memerlukan transmisi semua data sampai menyentuh fleg
    - Receiver akan mengirimkan dengan cara yang sama
    - Menghindari penungguan sampai buffer penuh
  - Sinyal data yang penting
    - Indikasi datangnya data yang penting dalam aliran data
    - User memutuskan bagaimana cara menanganinya

# TCP Header





# Item yang melewati IP

---

- TCP melewati beberapa parameter sampai menuju IP
  - Lebih utama
  - Normal delay/low delay
  - Normal throughput/high throughput
  - Normal reliability/high reliability
  - Security

# Mekanisme TCP(1)

---

- Pembuatan koneksi
  - Three way handshake
  - Antara pasangan port
  - Satu port dapat disambungkan ke banyak tujuan

# Mekanisme TCP(2)

---

- Transfer data
  - Aliran logis dari octets
  - Nomor octet modulo  $2^{23}$
  - Flow control oleh credit alokasi dari nomor octet
  - Buffer data pada pengirim dan penerima

# Mekanisme TCP (3)

---

- Connection termination
  - Graceful close
  - TCP users issues CLOSE primitive
  - Transport entity sets FIN flag on last segment sent
  - Abrupt termination by ABORT primitive
    - Entity abandons all attempts to send or receive data
    - RST segment transmitted

# **Implementation Policy Options**

---

- Kirim
- Pengiriman
- Menerima
- Pengiriman kembali
- Mengakui

# Mengirim

---

- Jika tidak ditekan atau close, kesatuan sambungan TCP akan tepat dengan sendirinya
- Buffer data pada pengiriman buffer
- Kemungkinan mendirikan segmen per data batch
- Kemungkinan menunggu untuk kuantitas atau jumlah data

# Pengirim

---

- Jika tidak ditekan, terjadi pengiriman data dengan sendirinya
- Kemungkinan mengantarkan pada setiap segmen penerima
- Kemungkinan buffer data lebih dari satu segment

# Penerima

---

- Segments mungkin tiba out of order
- Tujuan
  - Hanya menerima segmen dalam pesanan
  - Pembuangan segmen yang melebihi pesanan
- Pada windows
  - Menerima semua segmen dengan menerima window



# Pengiriman kembali

---

- TCP mempertahankan antrian dari segmen pengiriman tetapi tidak diakui
- TCP akan mengirimkan kembali jika ACK tidak memberikan waktu
  - First only
  - Batch
  - Individual

# Acknowledgement

---

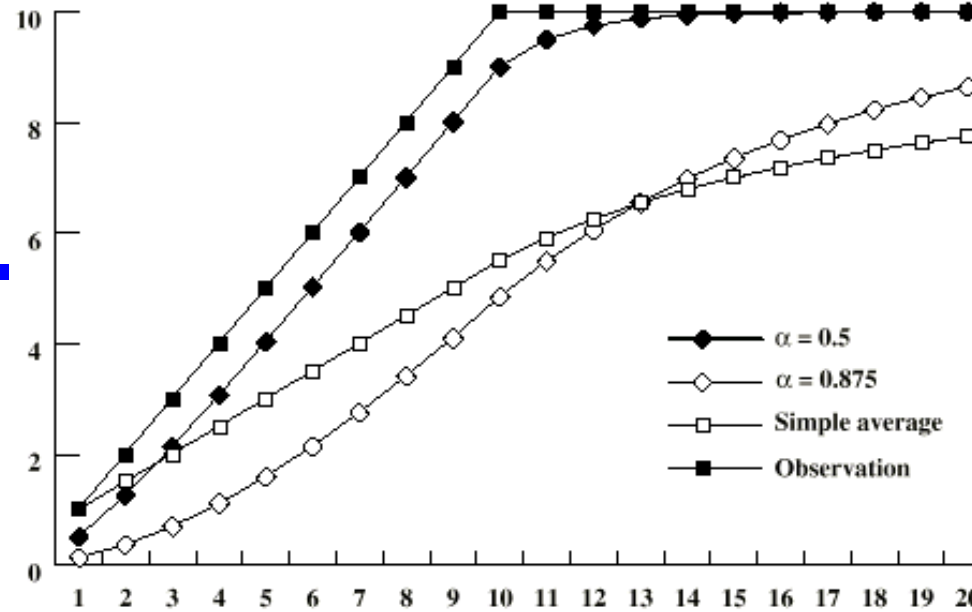
- Segera
- Kumulatif

# Congestion Control

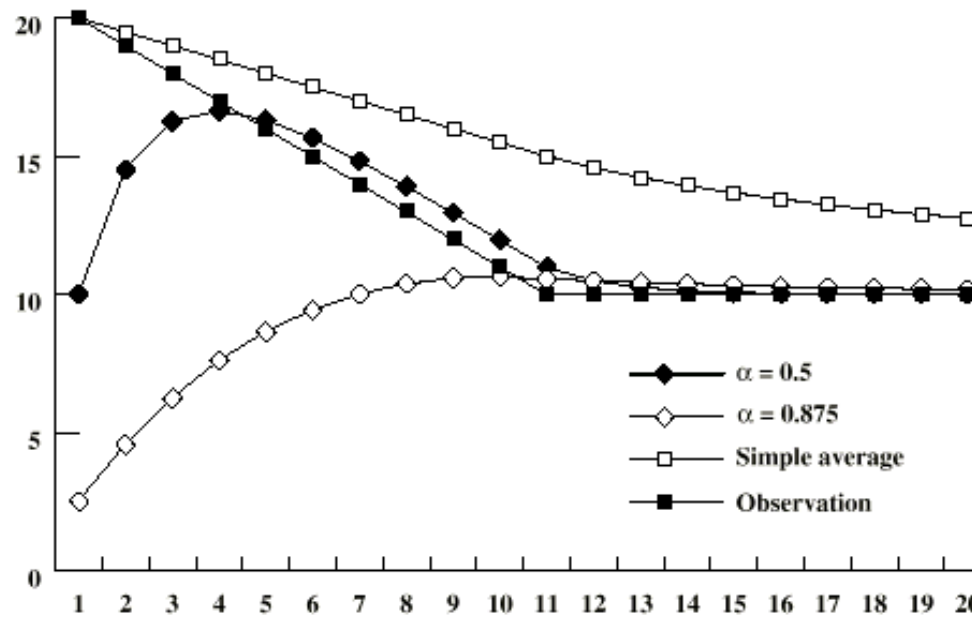
---

- RFC 1122, kebutuhan untuk pemakai internet
- Manajemen waktu pengiriman kembali
  - Perkiraan waktu perjalanan dengan mengonservasi pola dari delay
  - Pengesetan waktu lebih besar dari yang diperkirakan
  - Simple average
  - Exponential average
  - RTT Variance Estimation (Algoritma Jacobson)

# Use of Exponential Averaging

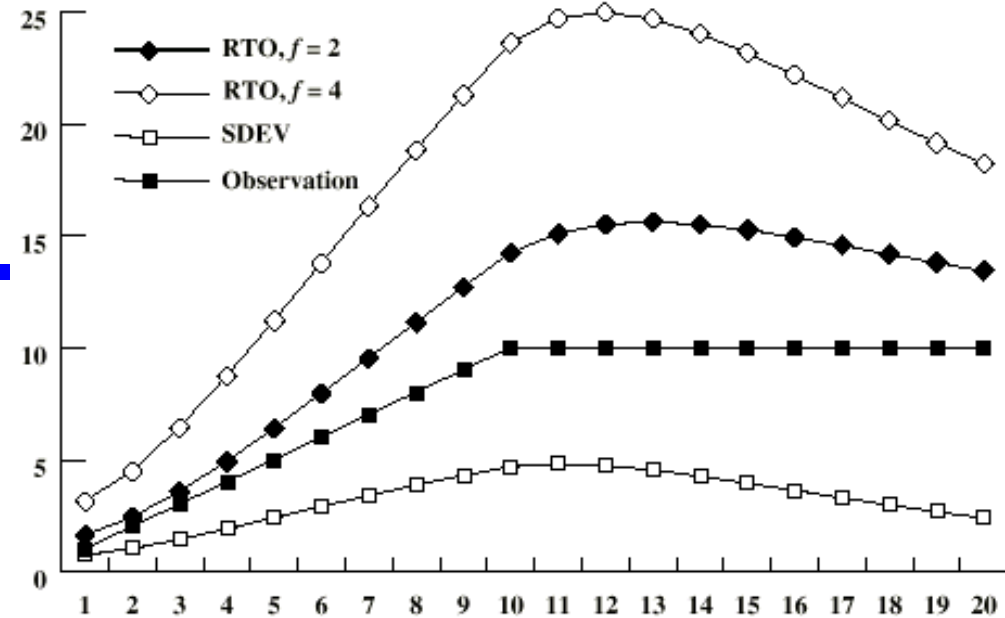


(a) Increasing function

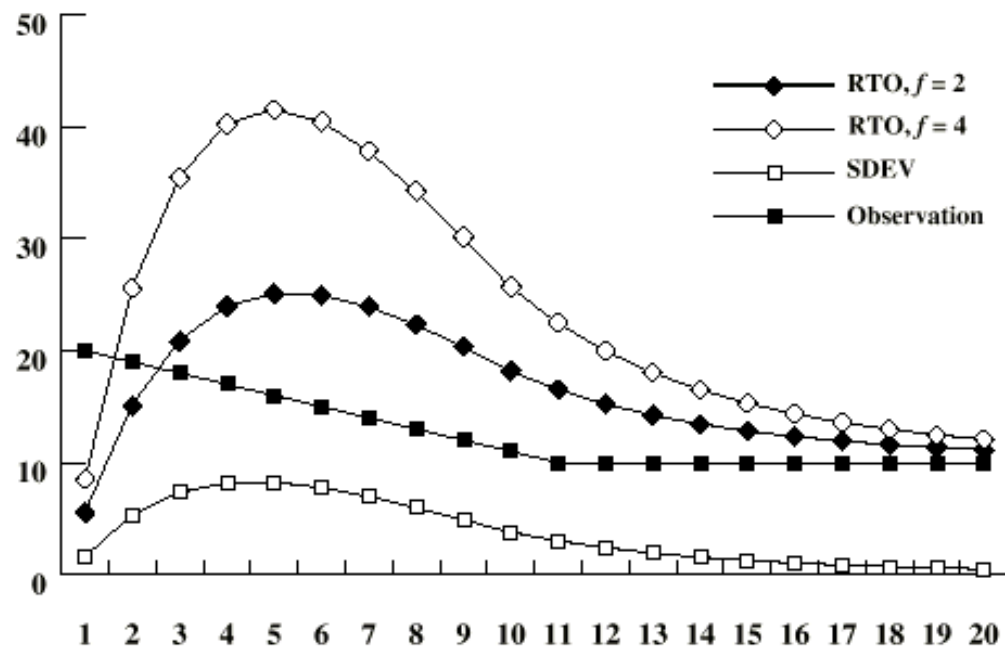


(b) Decreasing function

# Jacobson's RTO Calculation



(a) Increasing function



(b) Decreasing function

# Exponential RTO Backoff

---

- Since timeout is probably due to congestion (dropped packet or long round trip), maintaining RTO merupakan ide yang tidak baik
- RTO meningkat setiap kali suatu segment re-transmitted
- $RTO = q * RTO$
- Commonly  $q=2$ 
  - Binary exponential backoff

# Algoritma Karn

---

- Jika sebuah segmen dikirimkan kembali maka ACK mungkin akan:
  - Untuk copy pertama dari segmen
    - RTT lebih panjang dari yang diharapkan
  - Untuk copy kedua
- No way to tell
- Tidak mengatur RTT untuk re-transmitted segments
- Calculate backoff ketika re-transmission terjadi
- menggunakan backoff RTO sampai ACK tiba untuk segment yang belum re-transmitted

# Window Management

---

- Start lambat
  - $awnd = \text{MIN}[\text{credit}, cwnd]$
  - Start koneksi dengan  $cwnd=1$
  - kenaikan  $cwnd$  pada masing-masing ACK, ke beberapa max
- Dynamic window pada congestion
  - Ketika terjadi timeout
  - Set slow start threshold to half current congestion window
    - $ssthresh = cwnd/2$
  - Set  $cwnd = 1$  dan start lambat sampai  $cwnd = ssthresh$ 
    - Increasing  $cwnd$  by 1 for every ACK
  - Untuk  $cwnd \geq ssthresh$ , meningkat  $cwnd$  dengan 1 untuk setiap RTT



# UDP

---

- User datagram protocol
- RFC 768
- Tidak ada koneksi service untuk prosedur tingkatan aplikasi
  - Tidak handal
  - Kontrol pengiriman dan duplikasi tidak terjamin
- Mengurangi eksploitasi
- Manajemen jaringan (Chapter 19)

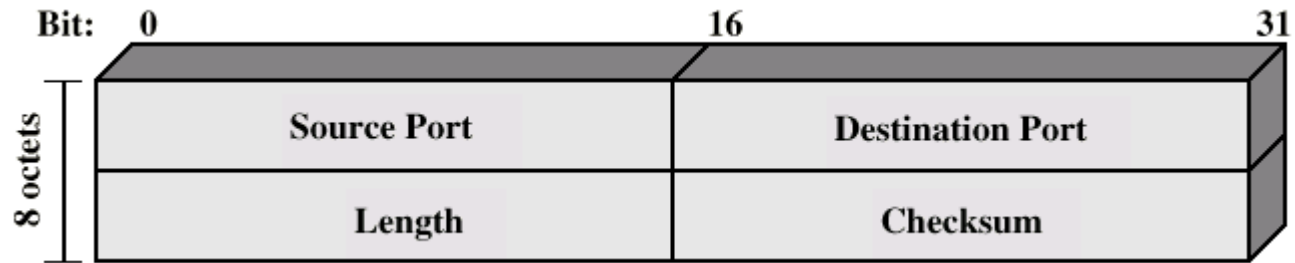
# **Menggunakan UDP**

---

- Pengumpulan data
- Pemecahan data
- Respon permintaan
- Secara langsung

# UDP Header

---



# Required Reading

---

- Stallings bab 20
- RFCs