

Transport Layer

1

Muhammad Zen S. Hadi, ST. MSc.

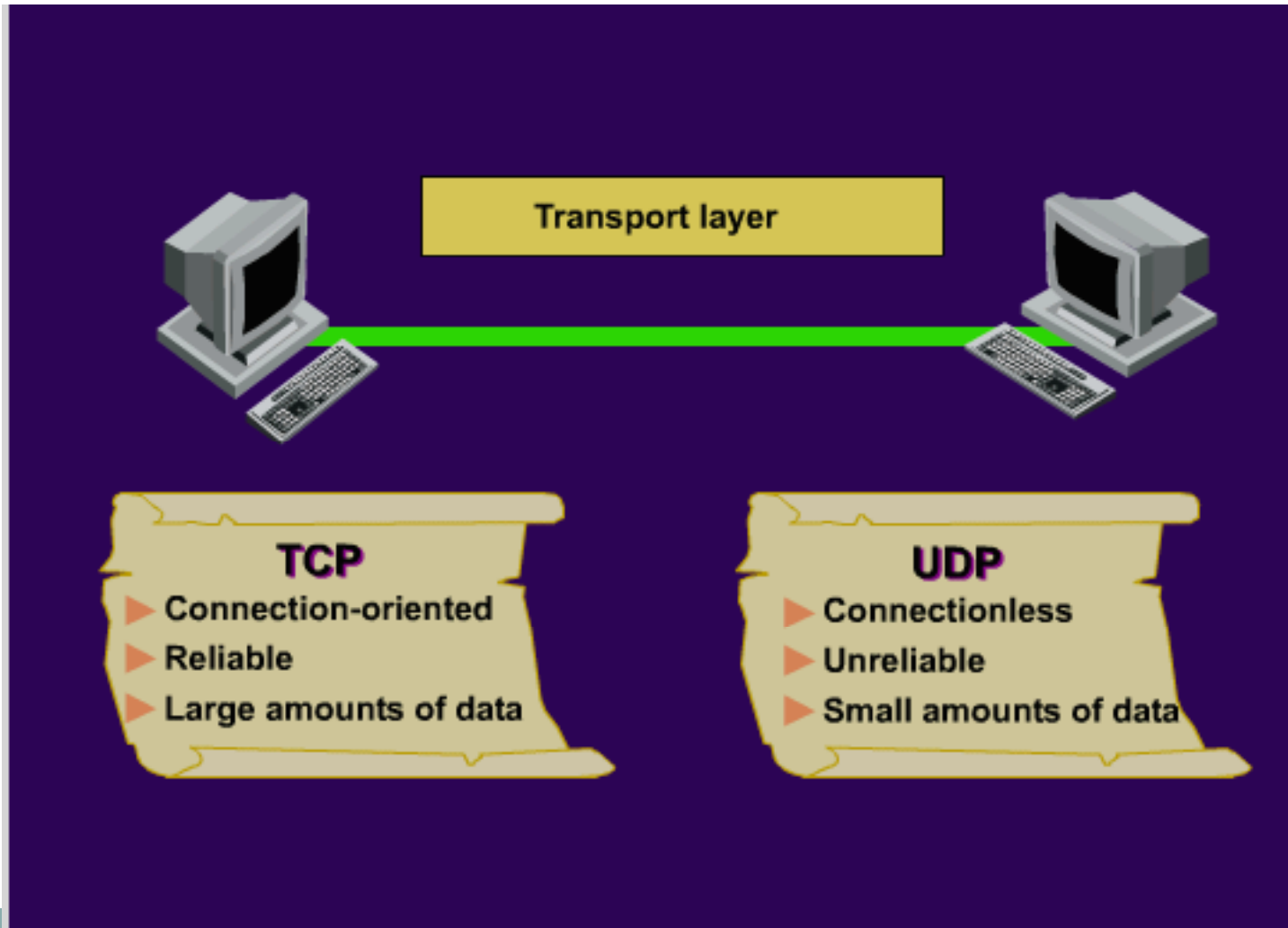
Overview



- **Layer Transport** bertugas melakukan sesi komunikasi antara komputer dalam jaringan. Menentukan bagaimana data ditransmisikan.
- **Dua Protocol Transport Layer yang dipakai :**
 - **Transmission Control Protocol (TCP)**
 - **User Datagram Protocol (UDP).**
- **User Datagram Protocol**
 - UDP merupakan **connectionless communication**, bekerja tidak menjamin data sampai ditujuan secara utuh.
 - Normalnya untuk mentransmisikan data dalam jumlah kecil pada satu waktu.
 - Reliabilitasnya/penjaminan data sampai pada penerima tergantung dari aplikasi.
- **Transmission Control Protocol (TCP)**
 - **Connection-oriented dan reliable communication** yang artinya data dijamin sampai tujuan
 - Untuk menjamin diperlukan komunikasi awal dengan penerima sebelum transfer data dilakukan
 - Membutuhkan **ack** setiap penerimaan data
 - Dipakai untuk mentransmisikan data dalam jumlah besar

Overview

3



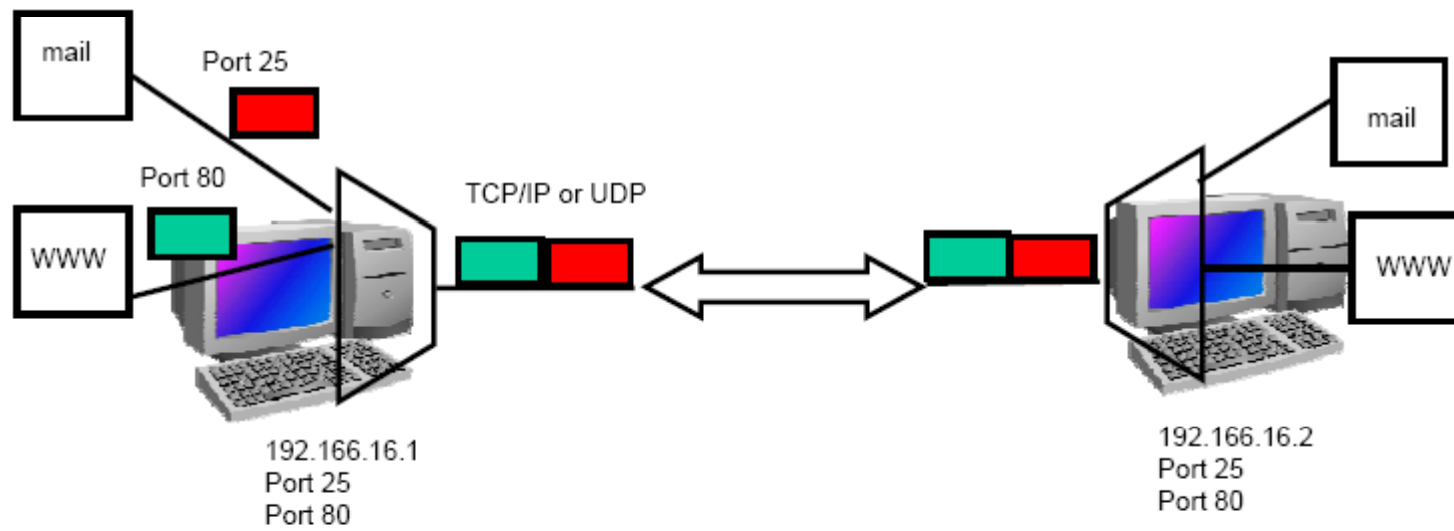
Port and Socket

4

- **Ada dua komponen yang biasa dipakai selama komunikasi pada layer transport yaitu port dan socket**
- **Port**
 - Port bisa dikatakan internal address yang disediakan untuk aplikasi tertentu pada komputer. Setiap aplikasi mempunyai port yang berbeda
 - Port bisa TCP atau UDP, tergantung pada pemakaian protocol apa pada layer transport apakah UDP atau TCP
 - Nomor Port antara 0 and 65,535.
 - Aplikasi TCP/IP biasanya menggunakan nomor port dibawah 1,024, dimana setiap aplikasi biasanya nomornya sudah pasti. Port ini biasa disebut "Well-Known Ports".
- **Socket**
 - Merupakan kombinasi dari IP address dan TCP atau UDP port.
 - Aplikasi men-generate socket ketika berkomunikasi dengan komputer lain
 - IP address menentukan tujuan komputer dan Port menentukan aplikasi yang dipakai.

Contoh aplikasi socket

5

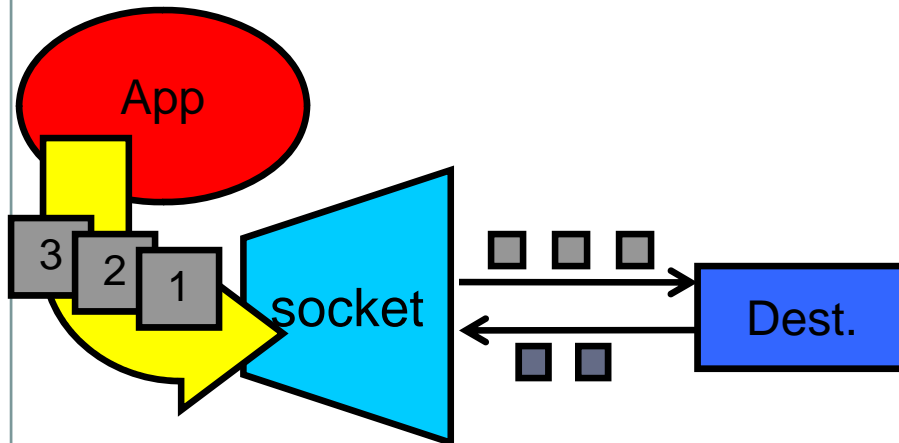


- TCP / IP and UDP are multiplexing different applications on one IP link

2 Type dasar dari Socket

- SOCK_STREAM

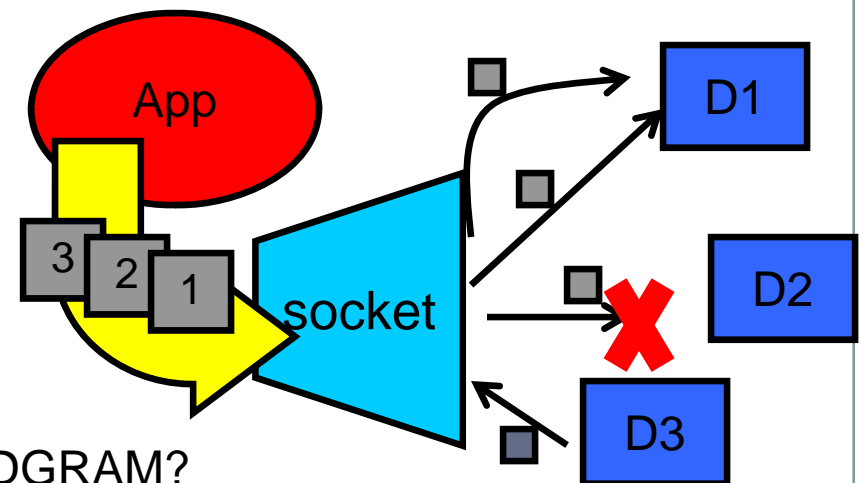
- a.k.a. TCP
- reliable delivery
- in-order guaranteed
- connection-oriented
- bidirectional



6

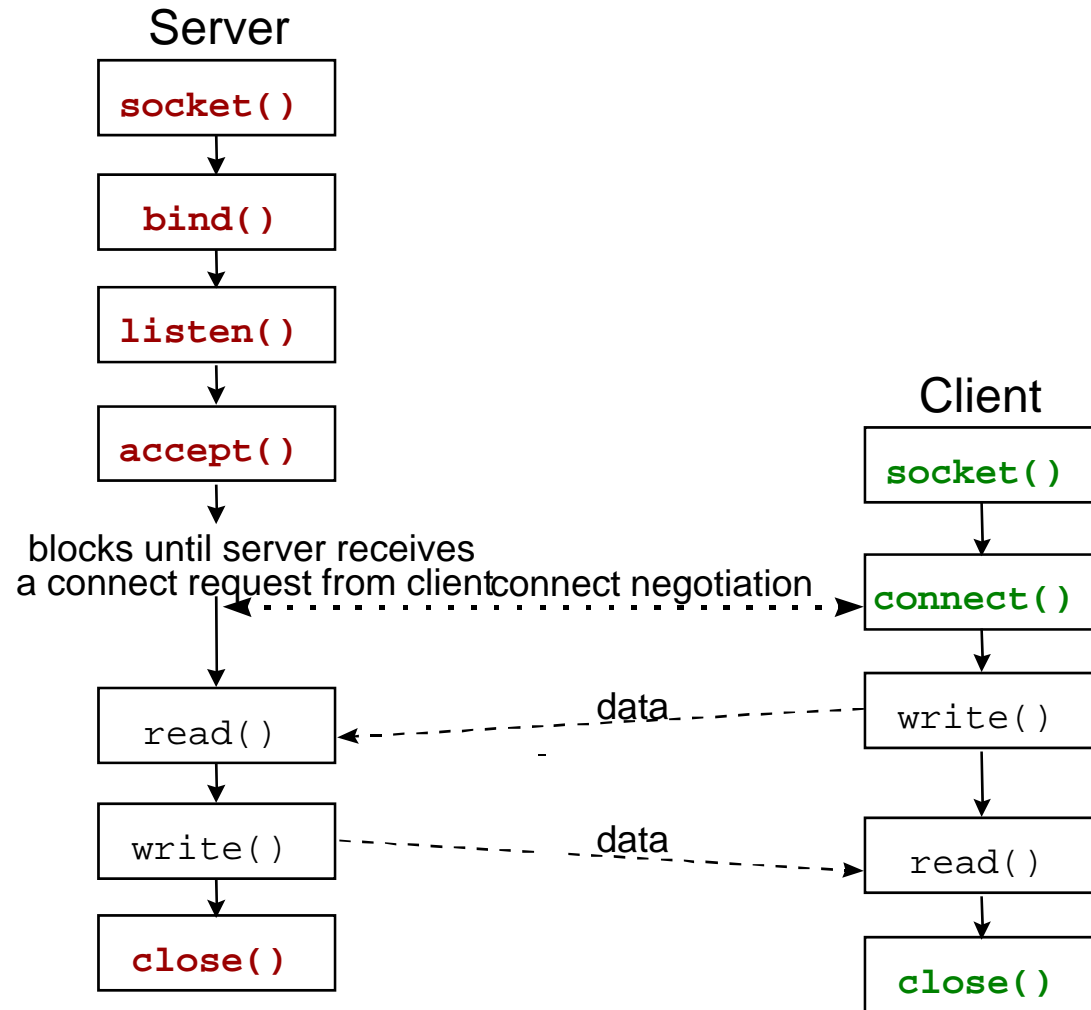
- SOCK_DGRAM

- a.k.a. UDP
- unreliable delivery
- no order guarantees
- no notion of “connection” – app indicates dest. for each packet
- can send or receive

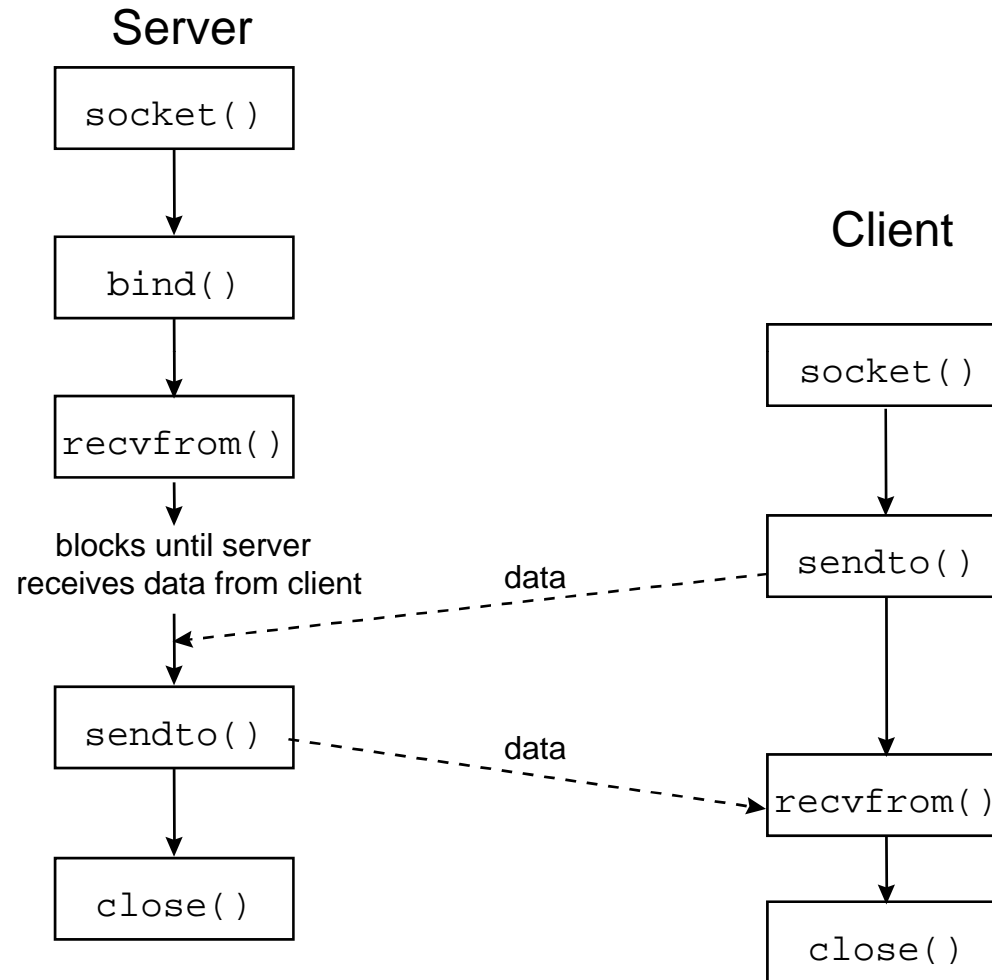


Q: why have type SOCK_DGRAM?

TCP socket calls



UDP socket calls



TCP Overview

9

- TCP merupakan protocol connection-oriented, yang artinya data hanya bisa ditransmisikan setelah ada proses negosiasi terlebih dahulu antara pengirim dan penerima
- Negosiasi diantaranya berupa : Berapa data yang bisa dikirim dalam satu waktu, nomor urut yang dipakai setiap pengiriman data dll.
- TCP biasanya merupakan komunikasi full duplex, yang artinya setiap host yang berkomunikasi mempunyai dua chanel logical untuk mengirim dan menerima message
- TCP menyediakan transmisi data yang reliable, dengan cara.
 - Setiap paket data diberi sequence number, jika tidak harus retransmite data
 - Receiver akan membuang jika terjadi duplikasi data, dan resequences packets jika kedatangan tidak urut

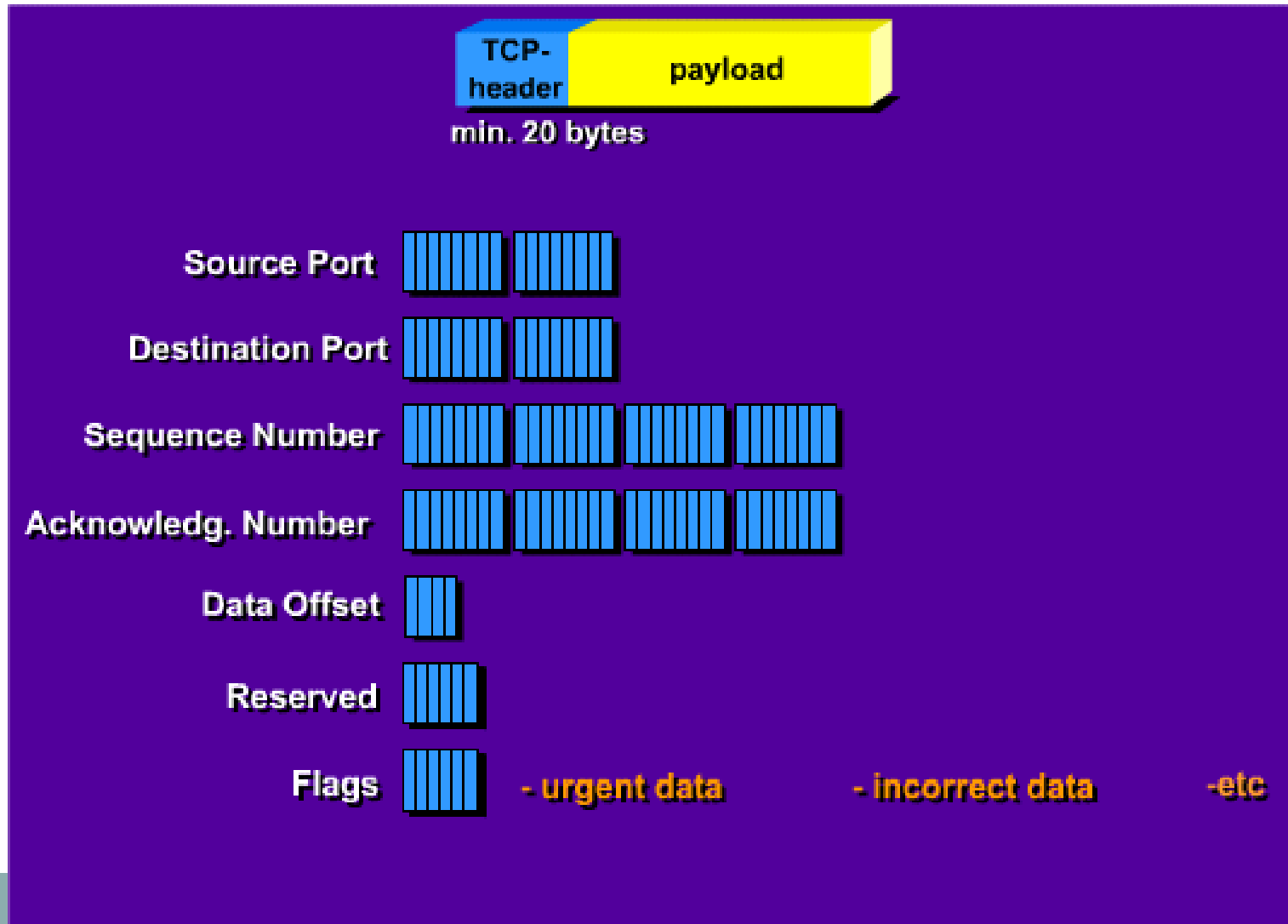
Flow Control

10

- Selain itu, TCP mensupport Flow Control untuk menghindari terlalu banyak data yang dikirim pada satu waktu dan overload pada jaringan router
- Flow Control artinya harus ada kesepakatan berapa besar data yang dikirim dalam satu waktu antara pengirim dan penerima.
- Flow Control mengindikasikan ukuran buffer penerima yang free yang bisa diisi dalam waktu tertentu

Struktur TCP

11



TCP Header

12

- TCP header panjangnya bervariasi. Panjang minimal 20 bytes. Terdiri dari 7 field : Source Port, Destination Port, Sequence Number, Ack. Number, Data Offset, Reserver dan Flag.
- 2 byte masing –masing untuk Source Port and the Destination Port. Sama seperti UDP.
- 4 byte sequence Number yang berisi nomor urut transmisi data dalam satu segment
 - Ini digunakan checking ketika semua byte telah diterima
- Acknowledgement Number terdiri dari 4 byte.
 - Berisi Sequence number berikutnya dari penerima
- Data Offset mengindikasikan awal data. Ini berhubungan dengan ukuran TCP header.
- Diikuti 6 bit reserve untuk penggunaan kedepan, diset 0.
- Flags menentukan tipe informasi pada segment.

Flag

13

Flags



Urgent		
Acknowledgement		!sending acknowledgement!
Push		!push function!
Reset		!connection reset!
Synchronization		!connection set-up!
Final		!end of the byte stream!

Tahapan Komunikasi pada TCP

14

- TCP adalah Protocol connection-oriented.
- Sebelum data ditransmisikan, koneksi yang dibuat bisa diset atau dirubah sesuai keadaan.
- Tiga tahap komunikasi
 - connection set-up
 - data transfer
 - connection release

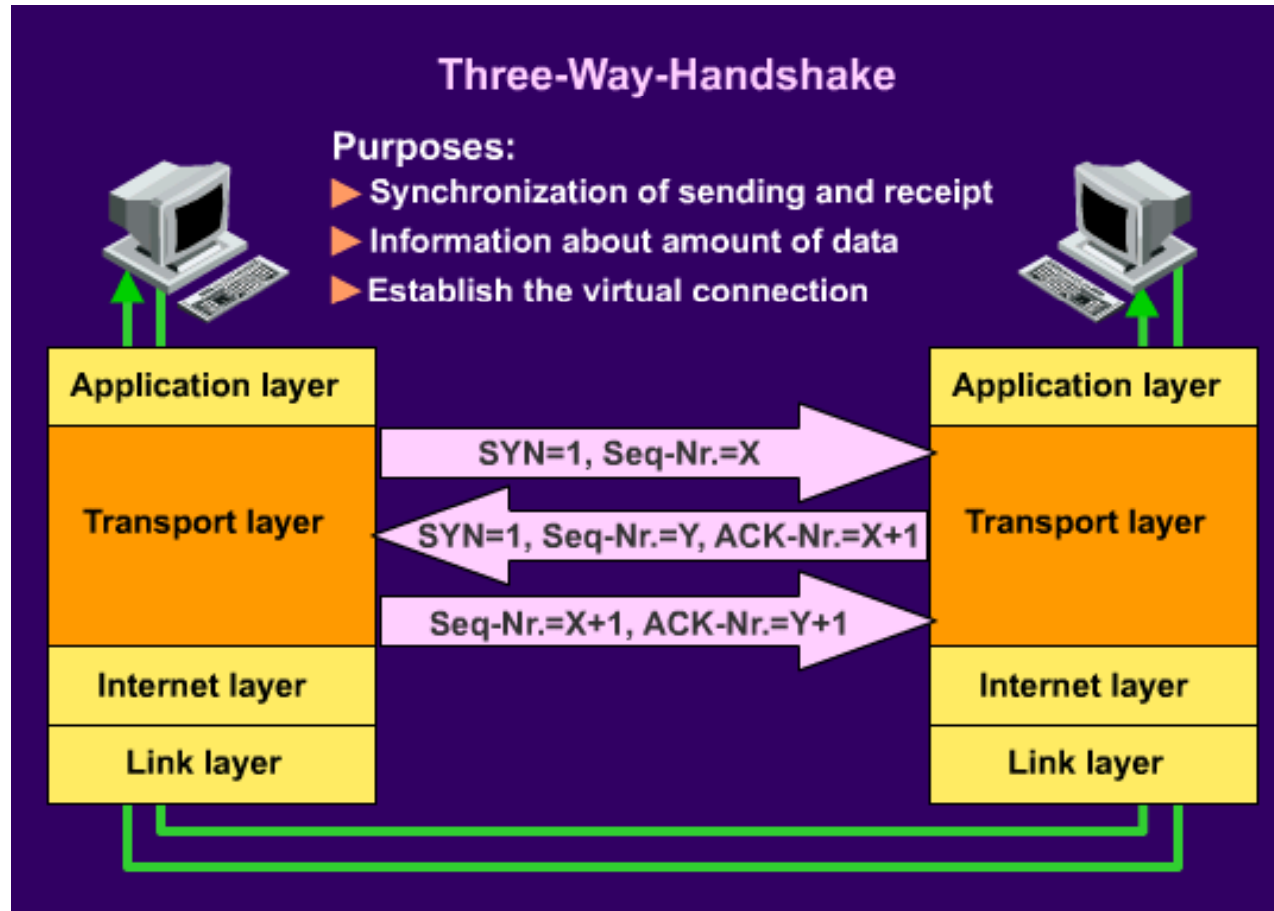
Three Way Handshake

15

- Koneksi TCP diawali oleh prosedur yang biasa disebut dengan Three-Way-Handshake. Tujuannya untuk melakukan sinkronisasi antara pengirim dan penerima. Hal yang diinformasikan selama Three Way Handshake adalah jumlah data yang bisa ditransmisikan dalam satu waktu, Sequence number yang dipakai.
- Untuk setup koneksi, host melakukan session inisialisasi dengan menset flag sinkronisasi ke 1.
- Segment juga berisi sequence number yang mengindikasikan awal byte yang ingin dikirim berikutnya. Juga berisi acknowledgement yang terdiri dari sequence number berikutnya untuk menerima data.
- Setelah Three Way Handshake dilakukan baru dianggap session established, dan koneksi dua arah siap dilaksanakan

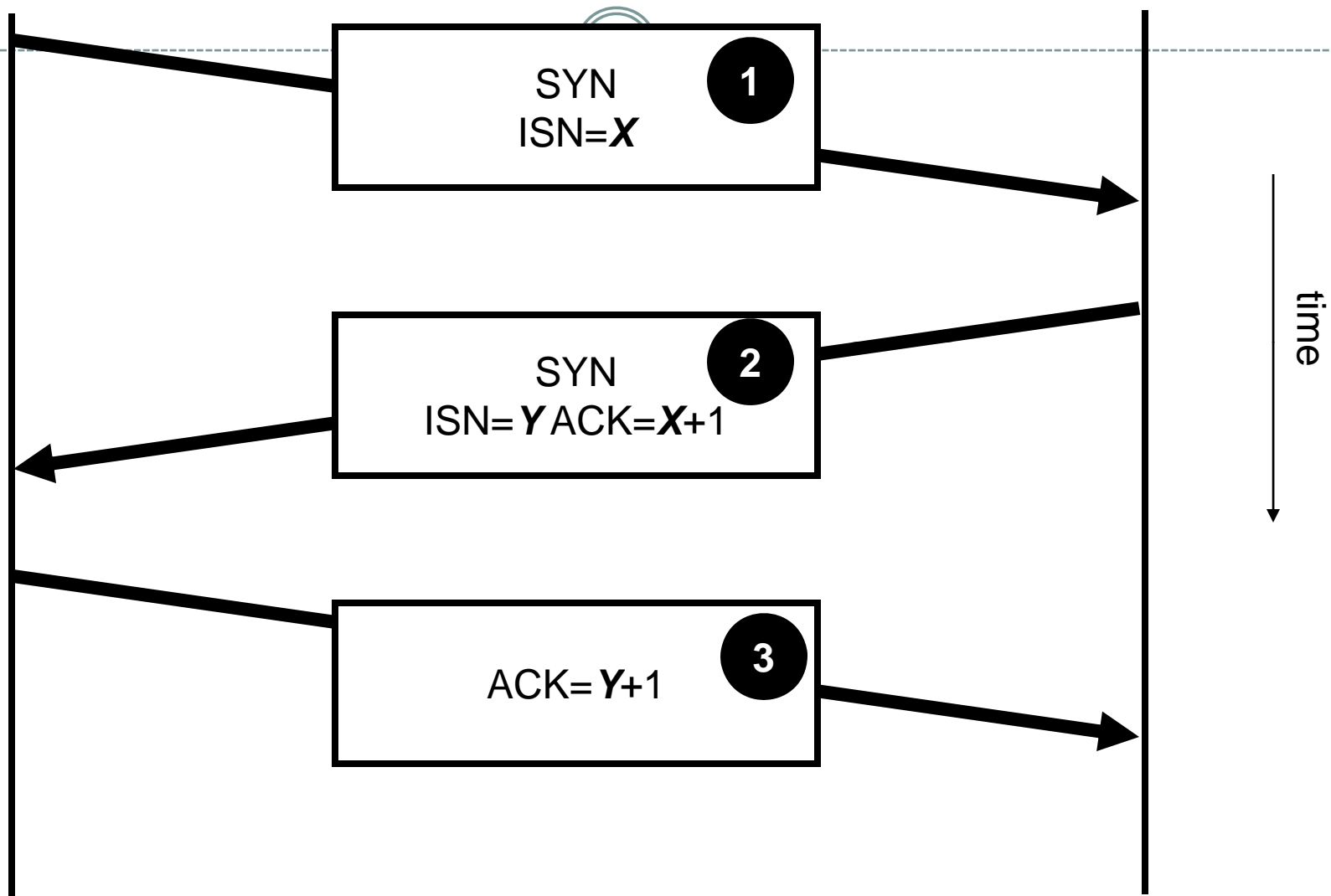
Three Way Handshake

16



Client

Server



TCP 3-way handshake

18

- 1 Client: “I want to talk, and I’m starting with byte number $X+1$ ”.
- 2 Server: “OK, I’m here and I’ll talk. My first byte will be called number $Y+1$, and I know your first byte will be number $X+1$ ”.
- 3 Client: “Got it - you start at byte number $Y+1$ ”.
- ? Bill: “Monica, I’m afraid I’ll syn and byte your ack”

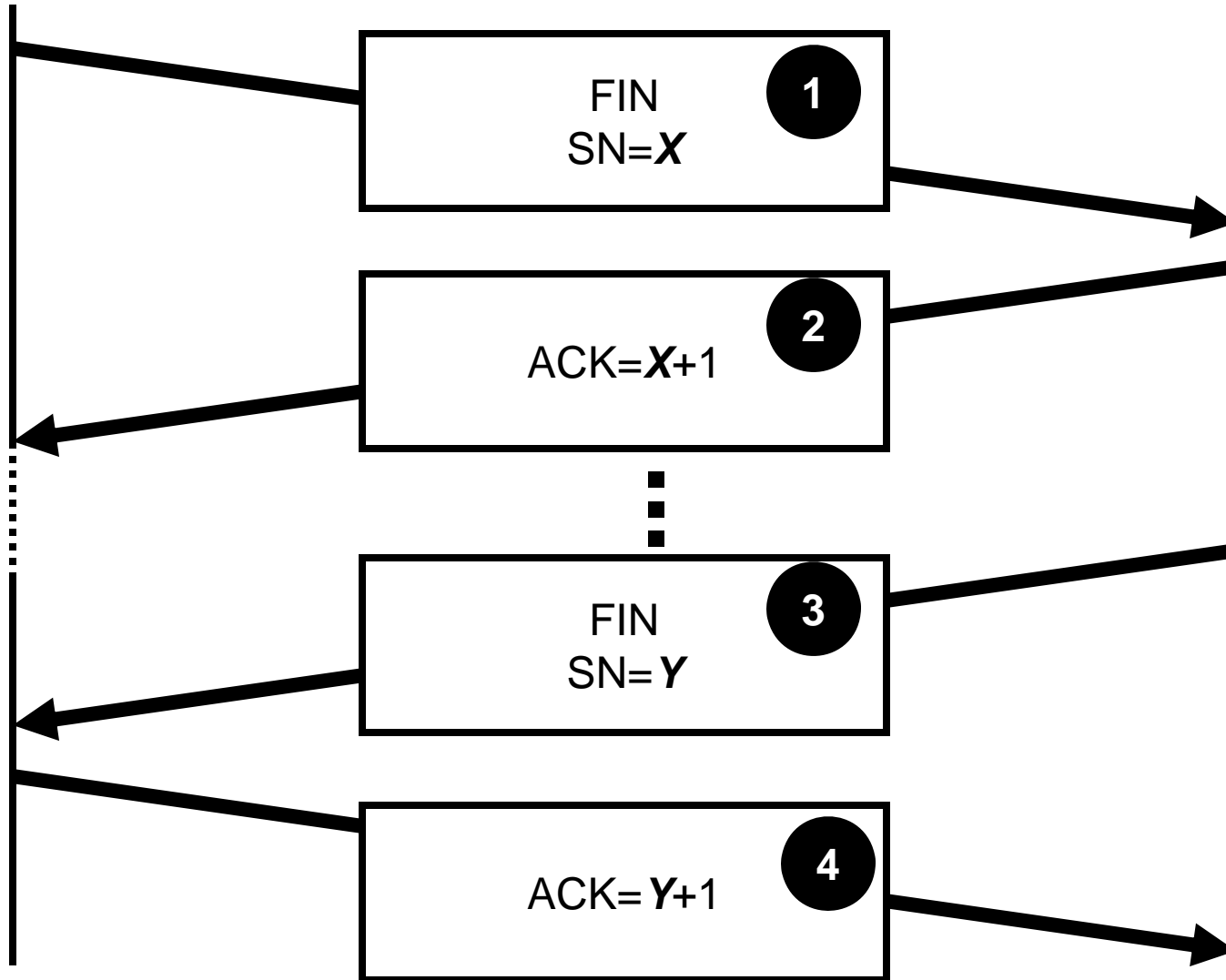
FIN

19

- **Either end of the connection can initiate termination.**
- **A FIN is sent, which means the application is done sending data.**
- **The FIN is ACK'd.**
- **The other end must now send a FIN.**
- **That FIN must be ACK'd.**

App1

App2



TCP Termination

- 1 App1: “I have no more data for you”.
- 2 App2: “OK, I understand you are done sending.”
dramatic pause...
- 3 App2: “OK - Now I’m also done sending data”.
- 4 App1: “Roger, Over and Out, Goodbye,
Astalavista Baby, Adios, It’s been real ...”
camera fades to black ...

TCP TIME_WAIT

22

- **Once a TCP connection has been terminated (the last ACK sent) there is some unfinished business:**
 - What if the ACK is lost? The last FIN will be resent and it must be ACK'd.
 - What if there are lost or duplicated segments that finally reach the destination after a long delay?
- **TCP hangs out for a while to handle these situations.**

```

Packet 1: source: 130.57.20.10  dest.:130.57.20.1
TCP: ----- TCP header -----
      TCP: Source port           = 1026
      TCP: Destination port      = 524
      TCP: Initial sequence number = 12952
      TCP: Next expected Seq number= 12953
      TCP:           .... ..1. = SYN
      TCP: Window                 = 8192
      TCP: Checksum                = 1303 (correct)
      TCP: Maximum segment size   = 1460 (TCP Option)

```

```

Packet 2: source: 130.57.20.1  dest: 130.57.20.10
TCP: ----- TCP header -----
      TCP: Source port           = 524
      TCP: Destination port      = 1026
      TCP: Initial sequence number = 2744080
      TCP: Next expected Seq number= 2744081
      TCP: Acknowledgment number  = 12953
      TCP:           .... ..1. = SYN
      TCP: Window                 = 32768
      TCP: Checksum                = D3B7 (correct)
      TCP: Maximum segment size   = 1460 (TCP Option)

```

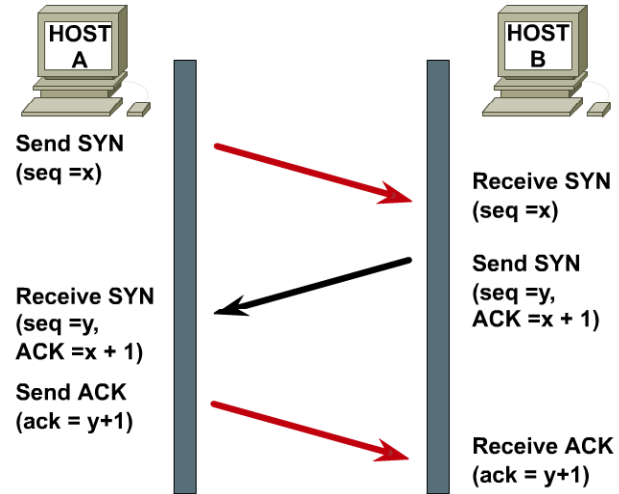
```

Packet 3: source: 130.57.20.10  dest: 130.57.20.1
TCP: ----- TCP header -----
      TCP: Source port           = 1026
      TCP: Destination port      = 524
      TCP: Sequence number        = 12953
      TCP: Next expected Seq number= 12953
      TCP: Acknowledgment number  = 2744081
      TCP:           ...1 .... = Acknowledgment
      TCP: Window                 = 8760
      TCP: Checksum                = 493D (correct)
      TCP: No TCP options

```

*

TCP Three-Way Handshake/ Open Connection



• **Maksimum ukuran segment dan ukuran windows yang dinegosiasikan juga dikirim**

PAR

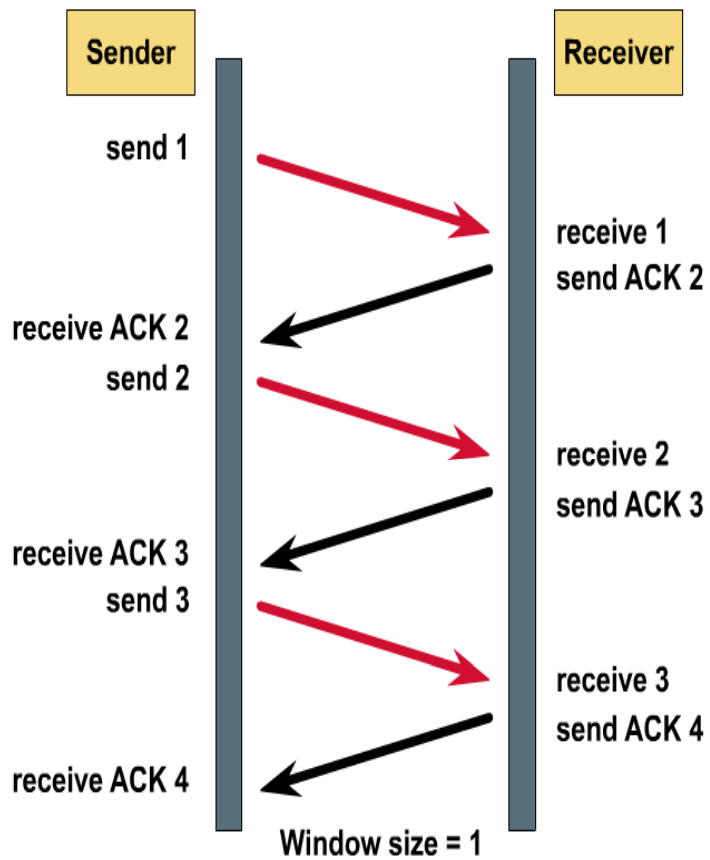
24

- Setelah koneksi established, ini harus dimantain sampai salah satu partner komunikasi ingin mengakhir komunikasi. System Transfer Data didasarkan pada mekanisme PAR - Positive Acknowledgement with Retransmission. Yang artinya bahwa untuk kebenaran data yang diterima maka penerima data harus mengirimkan acknowledgement ke pengirim.
- Untuk efisiensi, acknowledgements hanya berisi paket selanjutnya yang harus dikirim, tidak untuk setiap individu paket

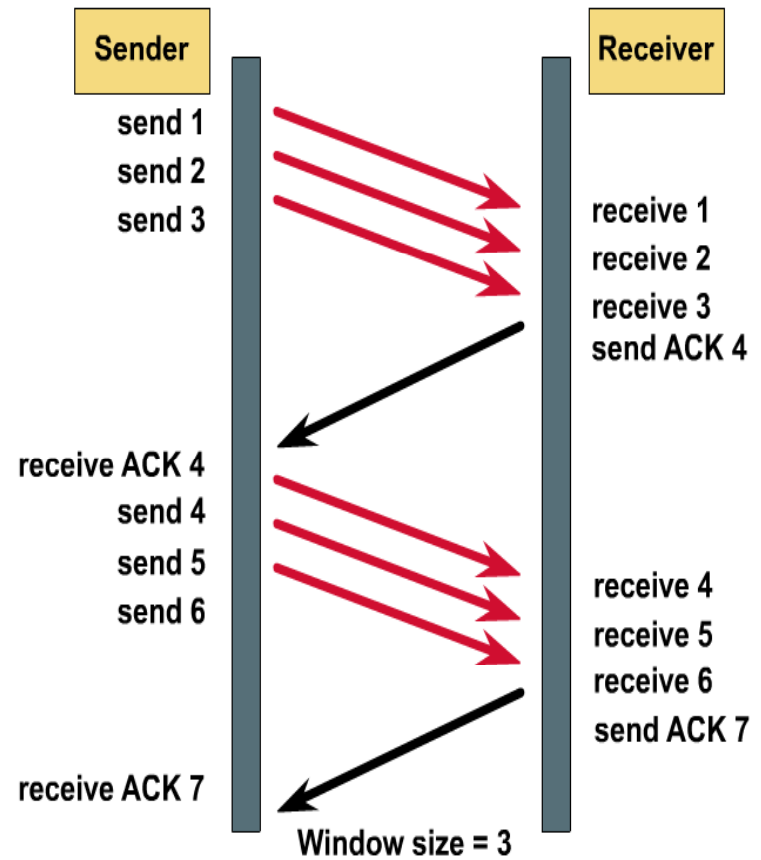
PAR

25

TCP Simple Acknowledgment



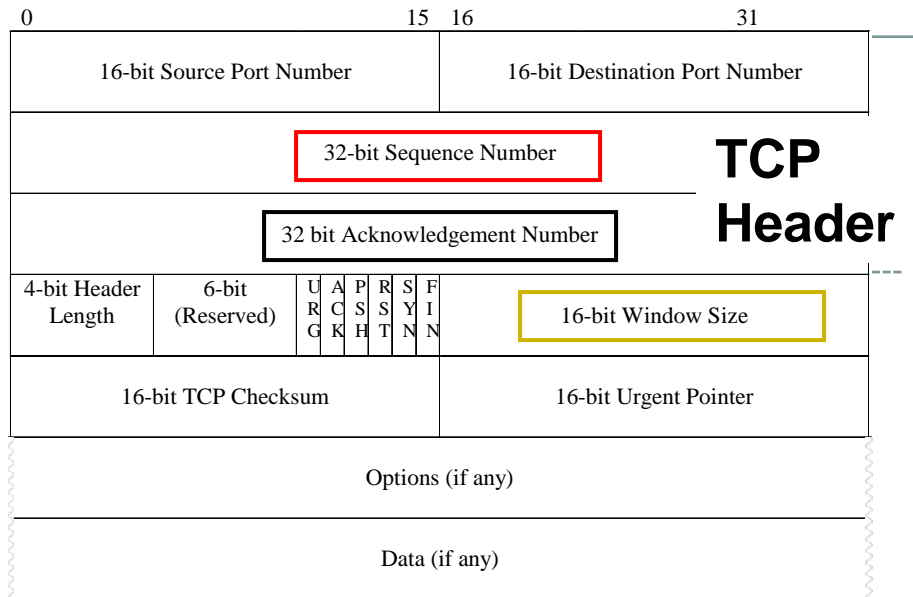
TCP Sliding Window



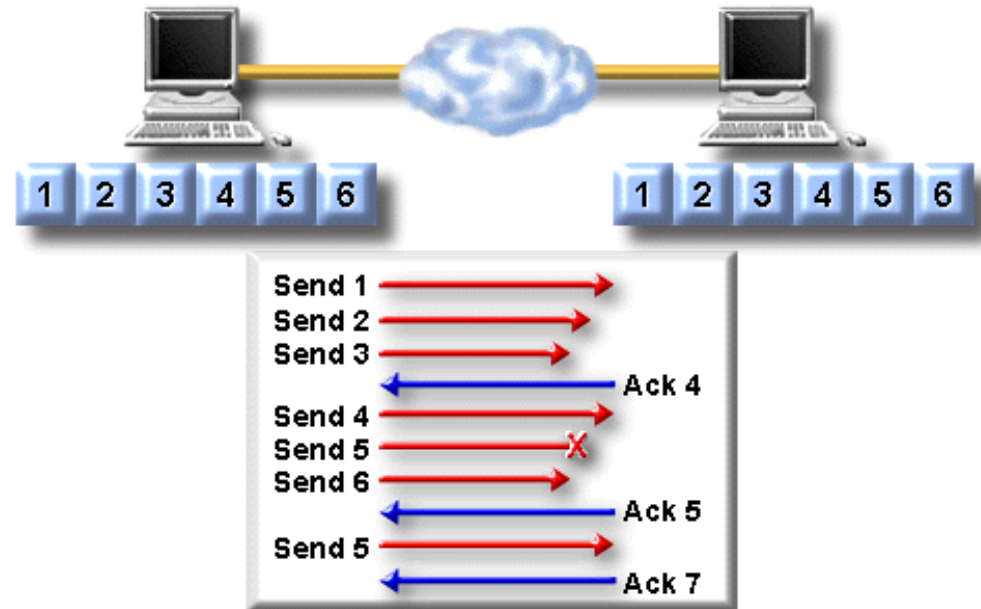
Sliding Window

26

- Untuk melakukan transmisi data penerima menyiapkan buffer, untuk mekanisme ini TCP menggunakan mekanisme sliding windows. Setiap host mempunyai akses ke dua windows: satu mengirim data dan yang lain menerima data. Ukuran windows mengindikasikan jumlah buffer yang disiapkan untuk data



An Acknowledgment Technique



- Paket mungkin didrop sepanjang jalan, time out atau rusak
- Jika misal 4, 5, dan 6 dikirim, tapi 5 lost, receiver hanya akan memberi ack sampai 4.
- Pengirim akan mengirim ulang paket 5 dan menunggu untuk mendengar dari penerima paket mana yang selanjutnya dikirim
- Receiver mengirim Ack 7, jadi pengirim tahu dapat memulai lagi mengirim paket ke-7 dan seterusnya

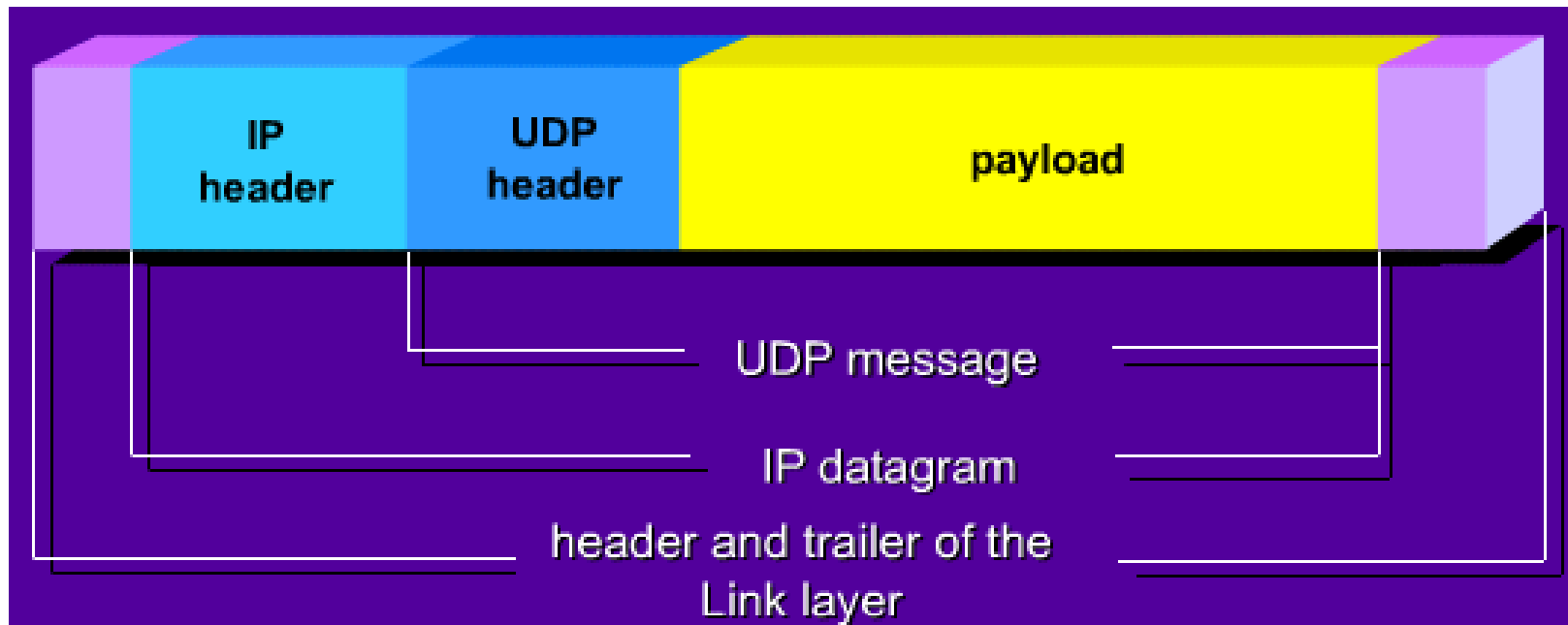
UDP

28

- UDP merupakan protokol connectionless, artinya tidak ada sesi komunikasi awal ketika data ditransmisikan.
- UDP merupakan unreliable protokol. Berarti pesan yang dikirim tanpa ada nomor urut dan tanpa acknowledgment dari penerima shg pengirim tidak pernah tahu apakah pesan sudah diterima penuh atau tidak. Untuk masalah ini ditangani oleh aplikasi
- Jika terjadi Lost paket data harus di-retrieve oleh layer di atasnya (aplikasi).
- Biasanya message UDP ditransmisikan secara regular dalam interval waktu tertentu atau setelah ditentukan batas waktu habis
- Hanya membutuhkan sedikit resource memori dan processor
- Contoh aplikasi yang menggunakan Protocol UDP Domain Name System(DNS) dan Dynamic Host Configuration Protocol(DHCP).

IP Datagram UDP

29



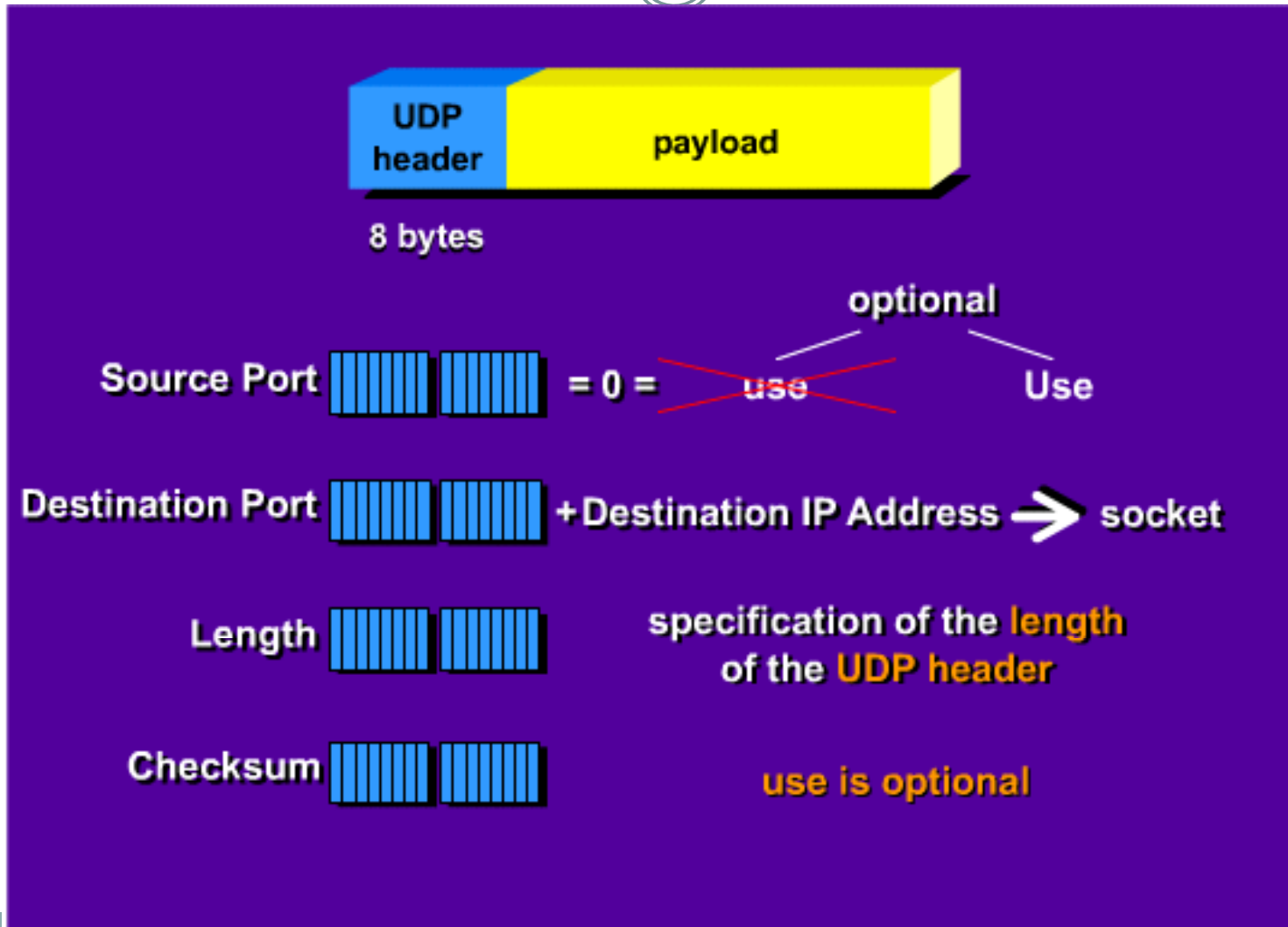
IP Datagram

30

- **Message UDP ditransmisikan dalam bentuk IP datagrams.**
- **Message UDP, terdiri dari :**
 - IP Header
 - UDP header
 - Payload
- **IP header terdiri dari Source IP dan Destination IP :**
 - Source IP berisi IP address host yang mengirim paket
 - Destination IP berisi alamat penerima paket, bisa broadcast address atau multicast address.

UDP Header Structure

31



Struktur Header UDP

32

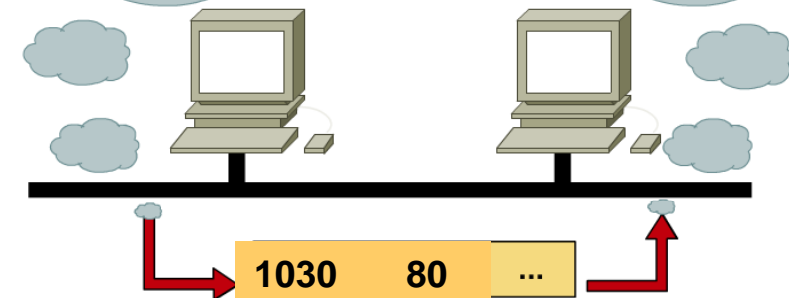
- Header UDP header mempunyai panjang yang tetap yaitu 8 bytes, Terdiri dari 4 field : Source Port, Dest., Length field dan Checksum
- Source Port terdiri dari 2 yang mengidentifikasi Port pengirim yang dipakai untuk mentransmisikan data. Source Port merupakan optional bisa diisi bisa tidak, jika tidak diisi diset 0. Misal pengirim data video yang tidak butuh reply/pengiriman balik
- Destination Port, berisi Port tujuan yang dikirim data. Gabungan Destination IP dan Destination Port membentuk Socket.
- Length field mengindikasikan panjang Header UDP.
- Checksum field, menyediakan integriti checker. Optional, jika diset 0 berarti tidak dipakai, Pengirim tidak melakukan proses perhitungan.

TCP/UDP Port Numbers

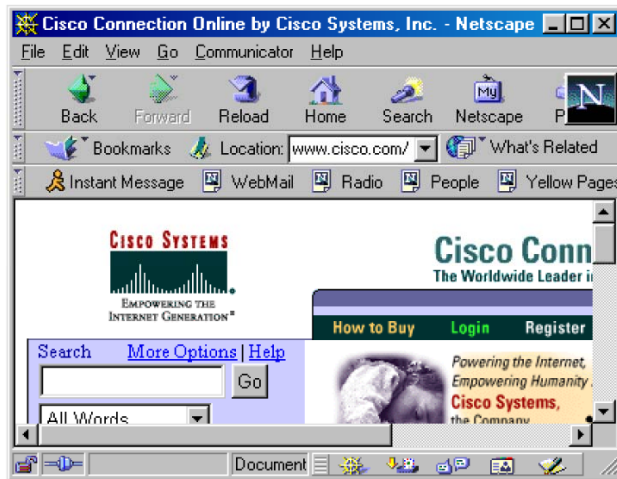
Source Port	Destination Port	...

Pada setiap session http antara client dan server yang sama, mempunyai Destination port yang sama, tapi berbeda Source Port (unik) untuk mengidentifikasi setiap session sehingga pengembalian permintaan masuk ke sesi yang benar

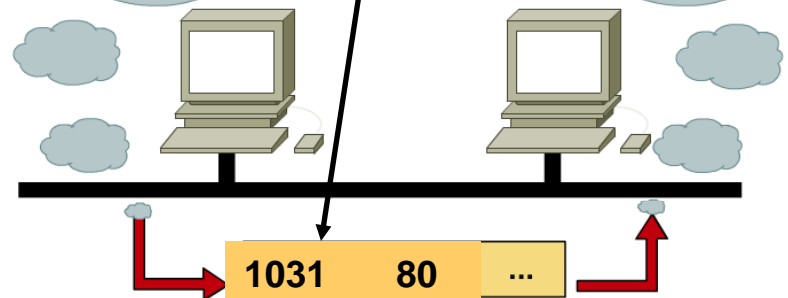
http to www.cisco.com
Dest. Port = 80 Send packets to web server application



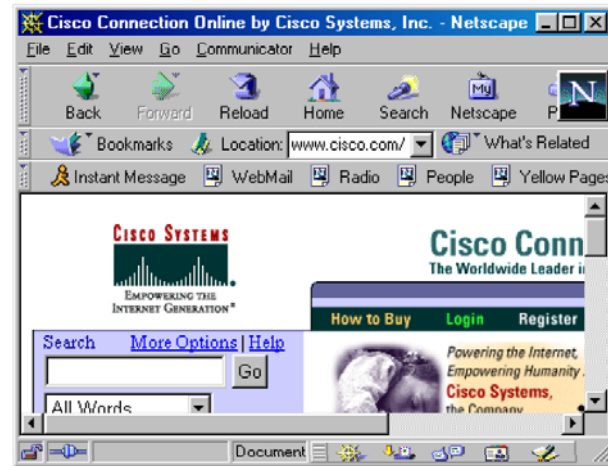
Netscape Navigator



http to www.cisco.com
Dest. Port = 80 Send packets to web server application



Netscape Navigator



Ini menunjukkan contoh dua browser windows dengan URL yang sama. TCP/IP menggunakan source port numbers untuk pengembalian informasi

Netscape Navigator

The image shows a Netscape Navigator browser window displaying the Cisco Systems website. Overlaid on this is a Windows Command Prompt window showing the output of the `netstat` command. Red arrows point from labels to specific columns in the netstat output: 'TCP or UDP' points to the 'Proto' column, 'Source IP' points to the 'Local Address' column, 'Destination IP' points to the 'Foreign Address' column, 'Source Port' points to the source port number, and 'Destination Port' points to the destination port number. A yellow box highlights several rows of connections to various IP addresses and ports. A red box highlights connections to `www.cisco.com`. A white box at the bottom of the Command Prompt contains the text 'netstat command'.

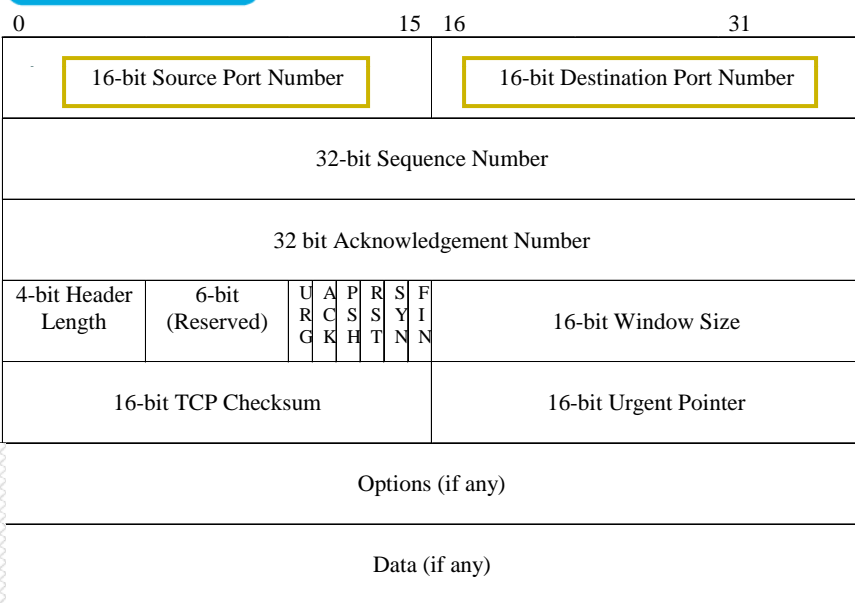
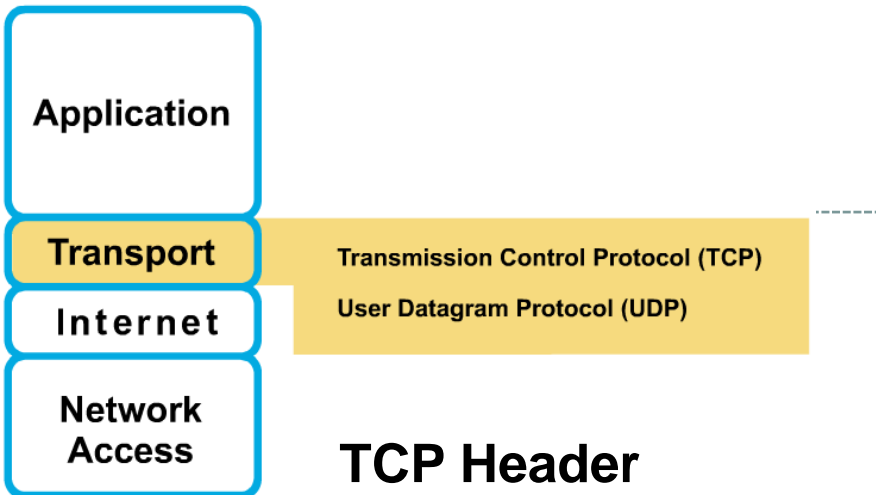
Proto	Local Address	Foreign Address	State
TCP	RICK-GRAZIANI:1430	RICK-GRAZIANI.cabrillo-lab.net:1431	ESTABLISHED
TCP	RICK-GRAZIANI:1431	RICK-GRAZIANI.cabrillo-lab.net:1430	ESTABLISHED
TCP	RICK-GRAZIANI:1414	65.104.127.219:http	ESTABLISHED
TCP	RICK-GRAZIANI:1432	boris.cabrillo.cabrillo-lab.net:imap	ESTABLISHED
TCP	RICK-GRAZIANI:1919	216.239.37.126:http	CLOSE_WAIT
TCP	RICK-GRAZIANI:1920	216.239.37.126:http	CLOSE_WAIT
TCP	RICK-GRAZIANI:1937	216.239.37.126:http	CLOSE_WAIT
TCP	RICK-GRAZIANI:1938	216.239.37.126:http	CLOSE_WAIT
TCP	RICK-GRAZIANI:1975	208.44.195.174:http	TIME_WAIT
TCP	RICK-GRAZIANI:1977	www.cisco.com:http	ESTABLISHED
TCP	RICK-GRAZIANI:1978	www.cisco.com:http	ESTABLISHED
TCP	RICK-GRAZIANI:1979	www.cisco.com:http	ESTABLISHED
TCP	RICK-GRAZIANI:1980	www.cisco.com:http	ESTABLISHED
TCP	RICK-GRAZIANI:1985	cco-sj-1.cisco.com:http	ESTABLISHED
TCP	RICK-GRAZIANI:1986	63.209.4.18:http	TIME_WAIT

www.google.com

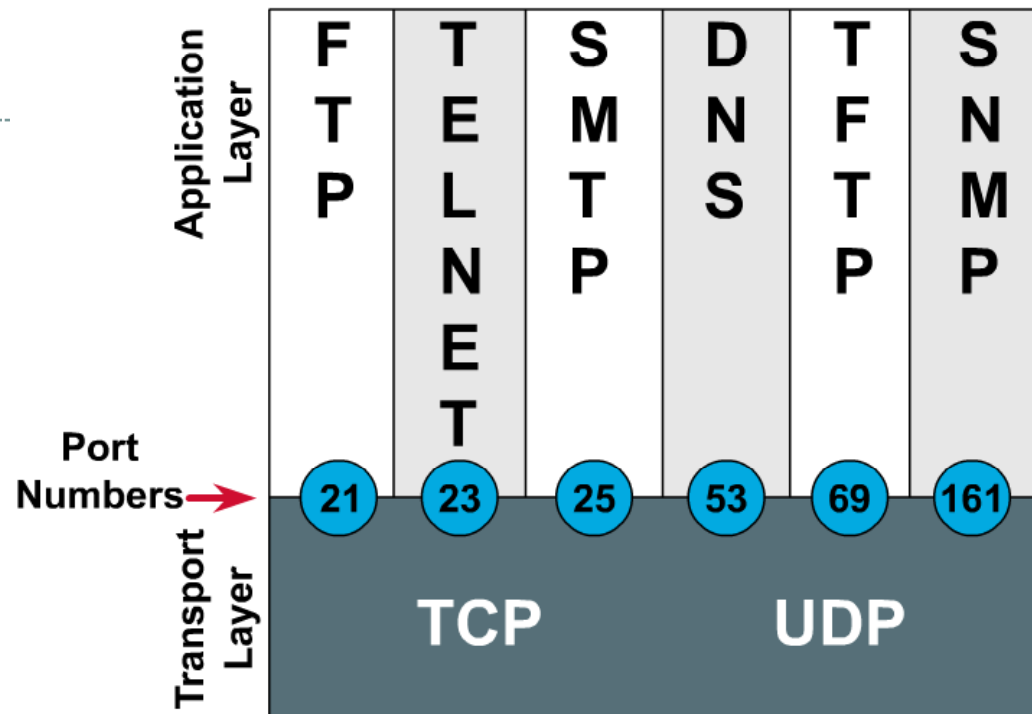
www.cisco.com

- **Note: Aktualnya**, ketika kita membuka sebuah halaman html, there are usually several TCP sessions created, not just one.
- Example of multiple TCP connections for a single http session.

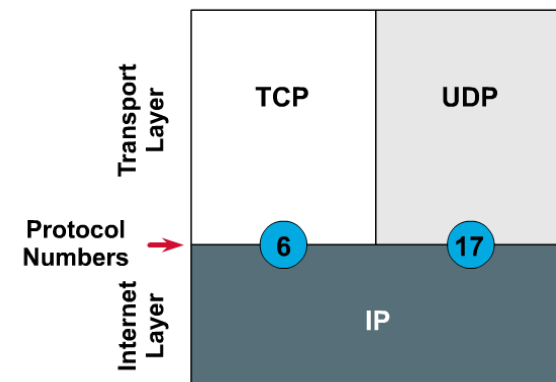
Transport Layer Overview



Port Numbers



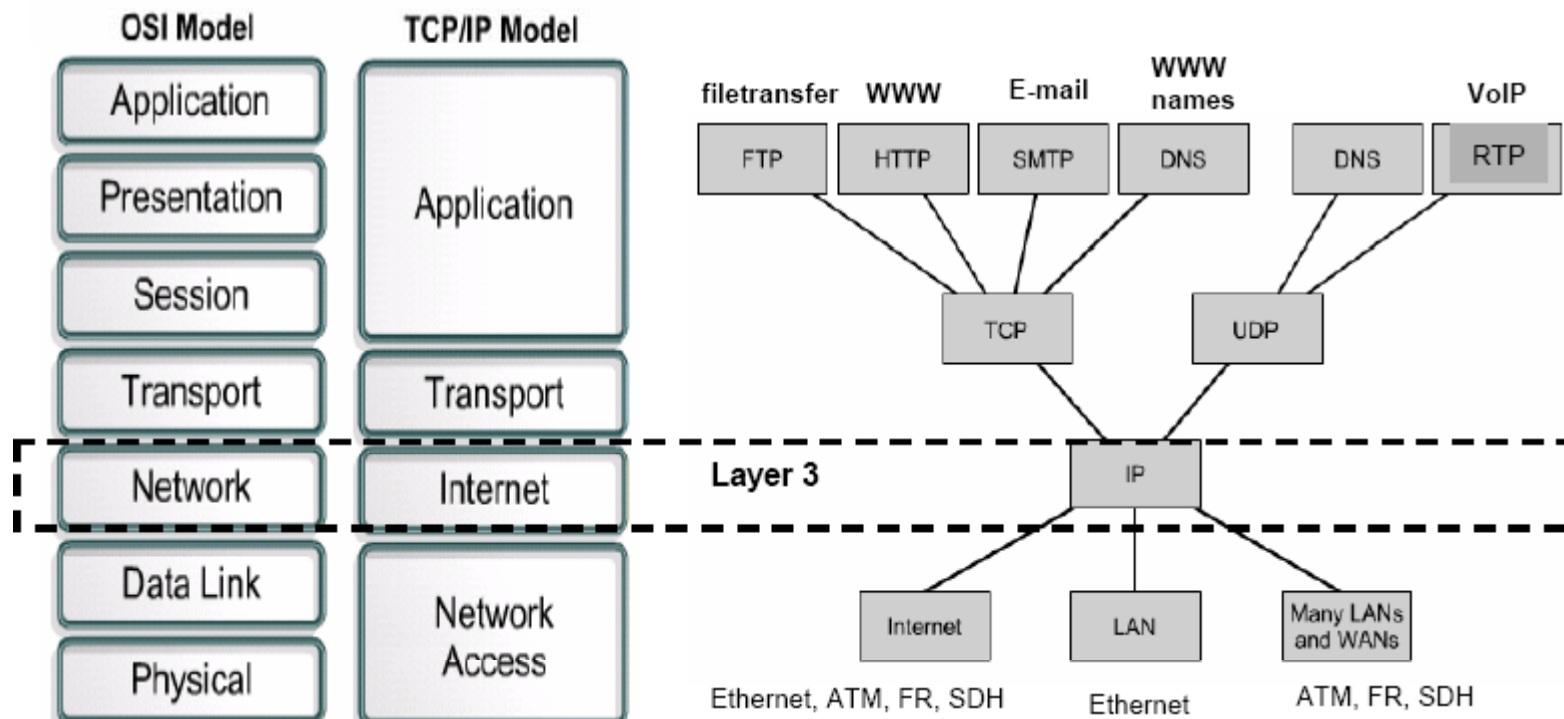
The Protocol Field



- **Keduanya TCP dan UDP** menggunakan Port untuk meneruskan informasi ke layer di atasnya

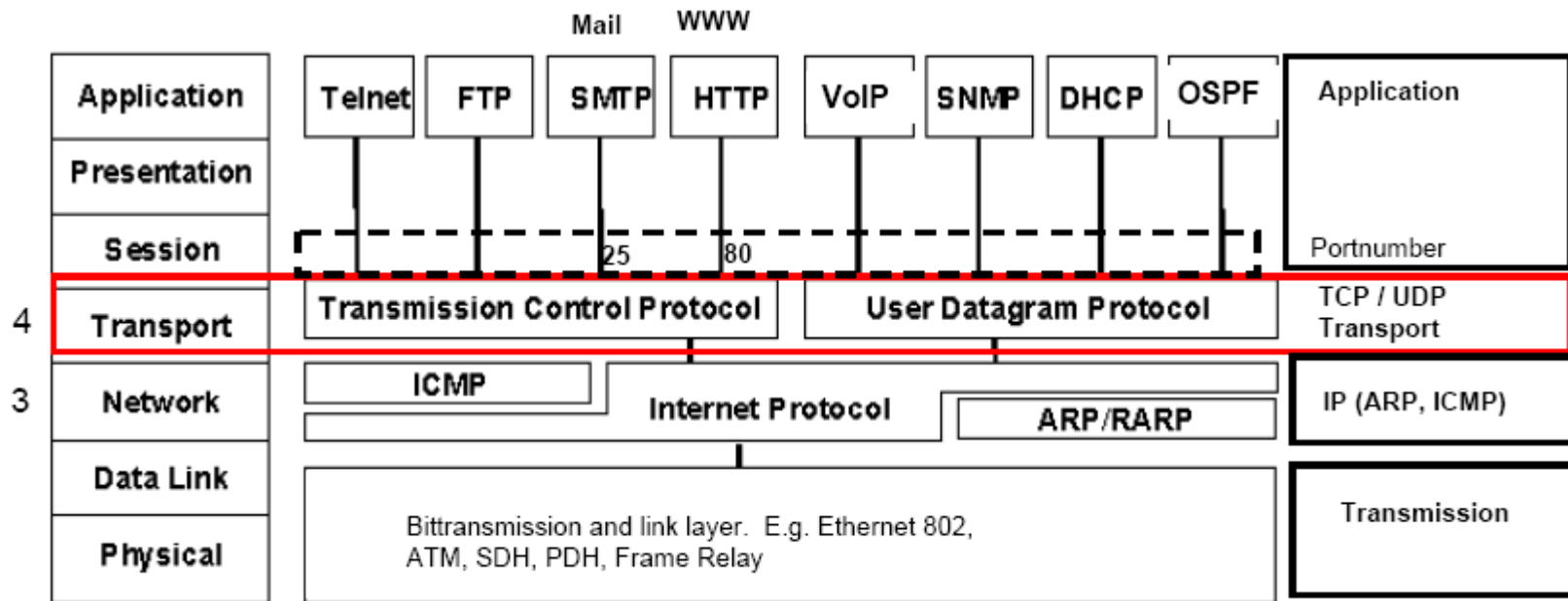
CONTOH APLIKASI TCP & UDP

36



CONTOH APLIKASI TCP DAN UDP

37



Hmmmmm. TCP or UDP ?



- **Electronic commerce?**
- **Video server?**
- **File transfer?**
- **Email ?**
- **Chat groups?**
- **Robotic surgery controlled remotely over a network?**