Transport Layer

Electronic Engineering Polytechnic Institut of Surabaya – ITS
Kampus ITS Sukolilo Surabaya 60111

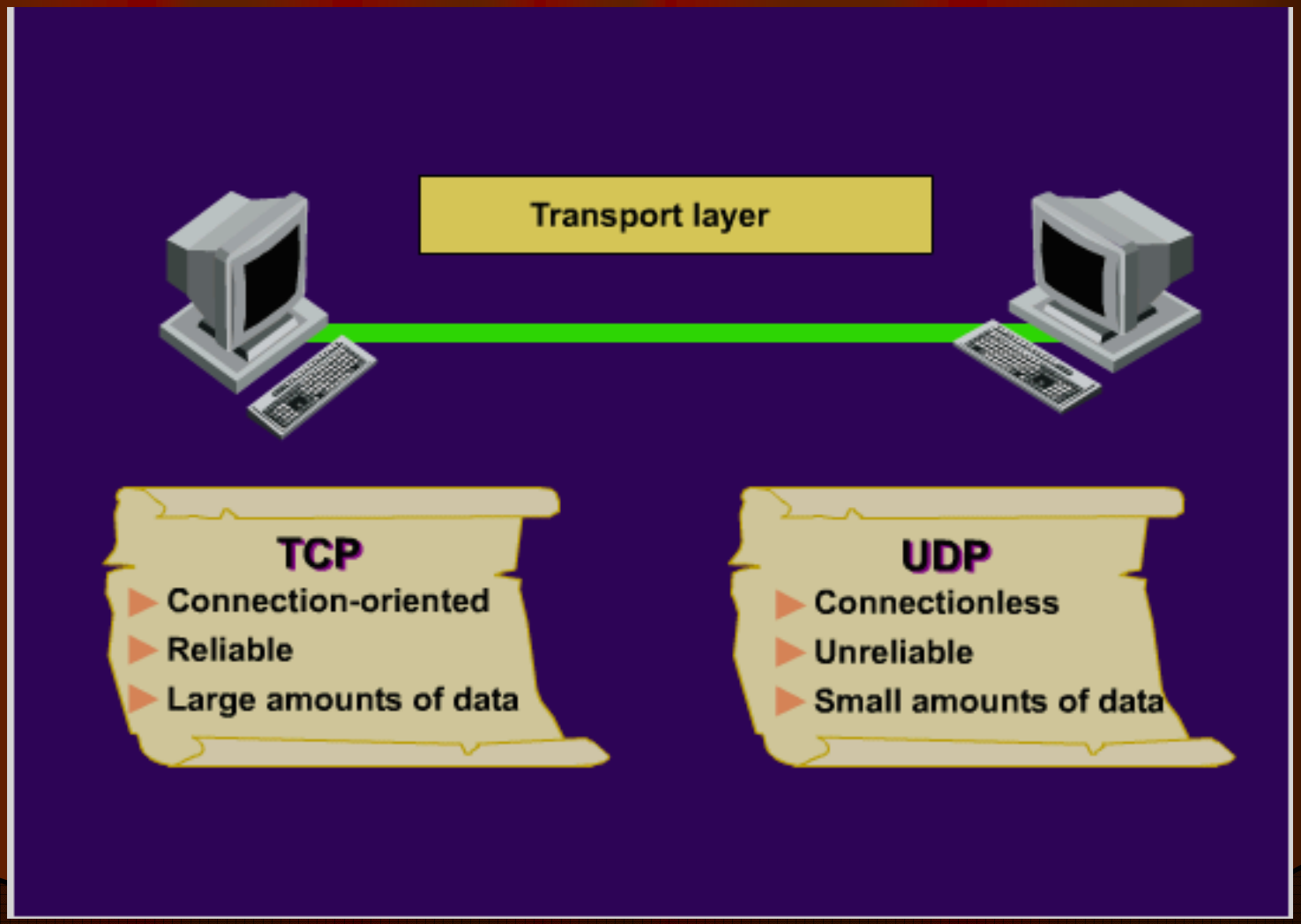
Pendahuluan

- Merupakan lapisan keempat dari model OSI.
- Memberikan layanan secara transparan terutama dalam hal error recovery dan data flow control
- Fungsinya:
  - Menerapkan mekanisme yang sesuai untuk pertukaran data antar proses dari sistem yang berbeda
  - Menjamin bahwa data yang diterima atau dikirim dari session layer dalam keadaan utuh, urut, tanpa duplikasi dan bebas dari kesalahan.
  - Mendukung penggunaan jaringan semaksimal mungkin
  - Memecah data dari lapisan session layer menjadi pesan-pesan dan memastikan pesan diterima dengan benar.

Overview

- 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


Transport layer
TCP
Connection-oriented
Reliable
Large amounts of data
UDP
Connectionless
Unreliable
Small amounts of data

Port

- Ada dua komponen yang biasa dipakai selama komunikasi pada layer transport yaitu port dan socket
- Port bisa dikatakan internal address yang disediakan untuk aplikasi tertentu pada komputer. Setiap aplikasi mempunyai port yang berbeda
- Port bisa TCP atau UDPt, tergantung pada pemakaian protocol apa pada layer transport apakah Udp atau TCP
- Nomor Port antara 0 sampai 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".

Jenis Port

- Well-known Port: yang pada awalnya berkisar antara 0 hingga 255 tapi kemudian diperlebar untuk mendukung antara 0 hingga 1023. Port number yang termasuk ke dalam well-known port, selalu merepresentasikan layanan jaringan yang sama, dan ditetapkan oleh Internet Assigned Number Authority (IANA). Beberapa di antara port-port yang berada di dalam range Well-known port masih belum ditetapkan dan direservasikan untuk digunakan oleh layanan yang bakal ada di masa depan. Well-known port didefinisikan dalam RFC 1060.
- Registered Port: Port-port yang digunakan oleh vendor-vendor komputer atau jaringan yang berbeda untuk mendukung aplikasi dan sistem operasi yang mereka buat. Registered port juga diketahui dan didaftarkan oleh IANA tapi tidak dialokasikan secara permanen, sehingga vendor lainnya dapat menggunakan port number yang sama. Range registered port berkisar dari 1024 hingga 49151 dan beberapa port di antaranya adalah Dynamically Assigned Port.
- Dynamically Assigned Port: merupakan port-port yang ditetapkan oleh sistem operasi atau aplikasi yang digunakan untuk melayani request dari pengguna sesuai dengan kebutuhan. Dynamically Assigned Port berkisar dari 1024 hingga 65536 dan dapat digunakan atau dilepaskan sesuai kebutuhan.

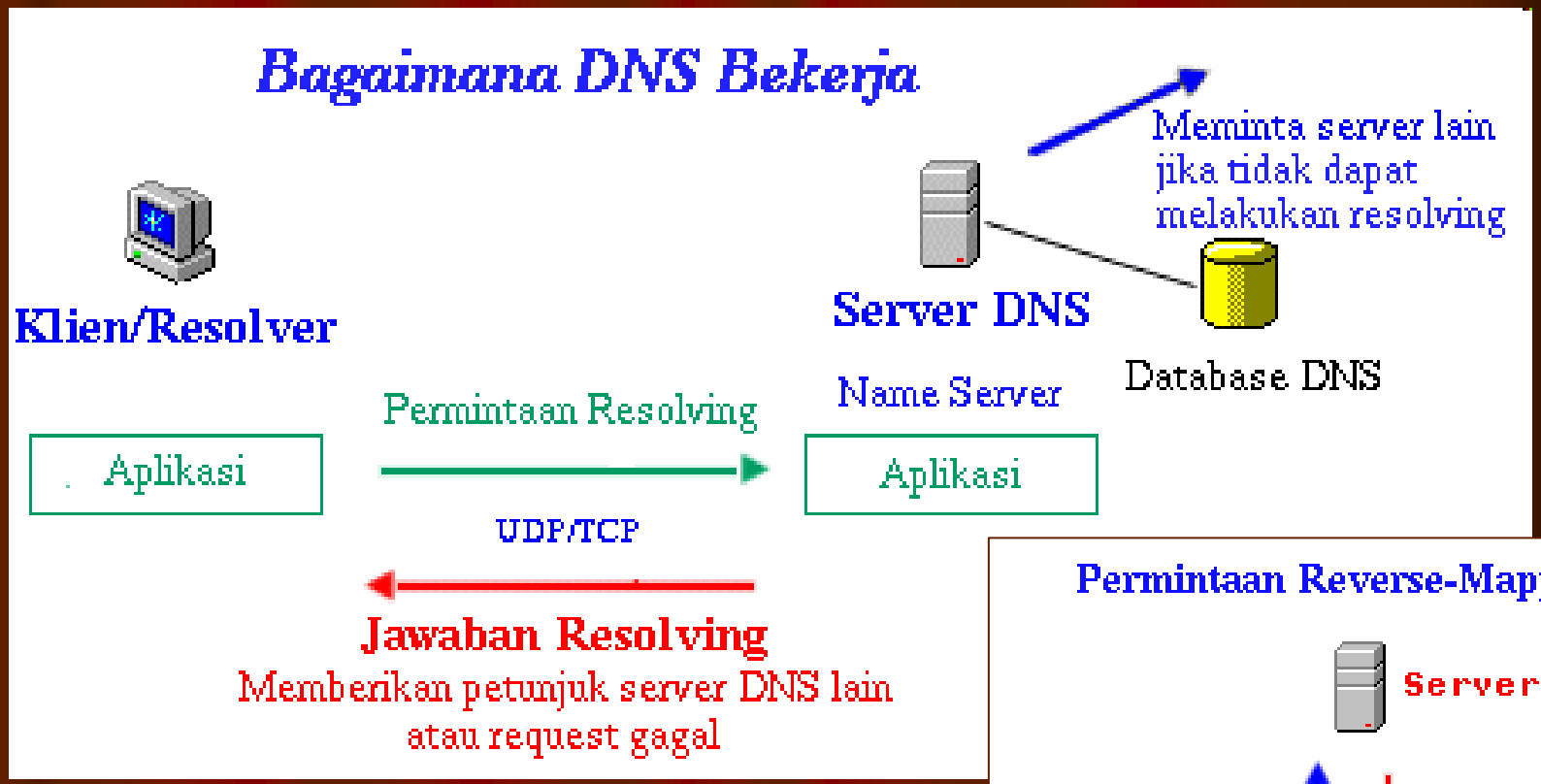
Socket

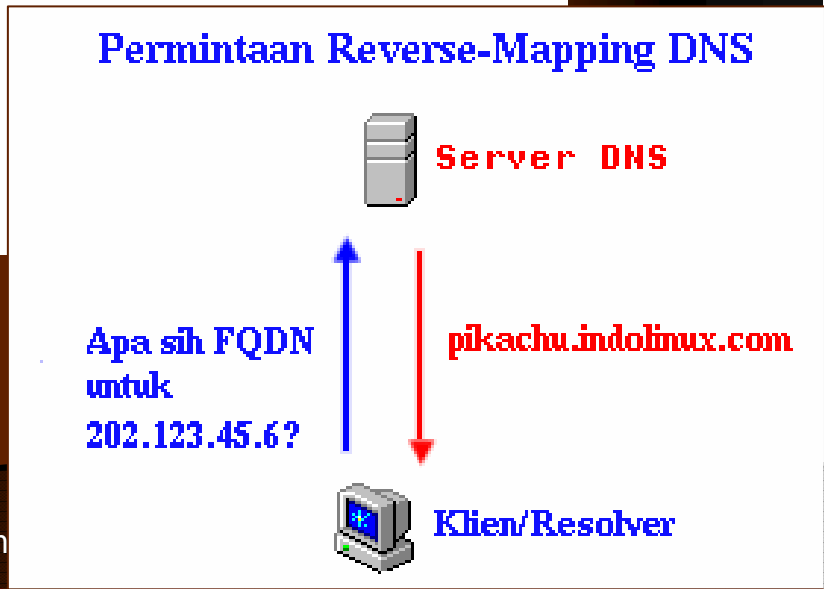
- Merupakan kombinasi dari IP address dan TCP atau UDP port.
- Aplikasi men-generate socket ketika berkomunikasi dengan komputer lain
- IP address menentukan tujuan komputer dan Port menentukan aplikasi yang dipakai.

UDP

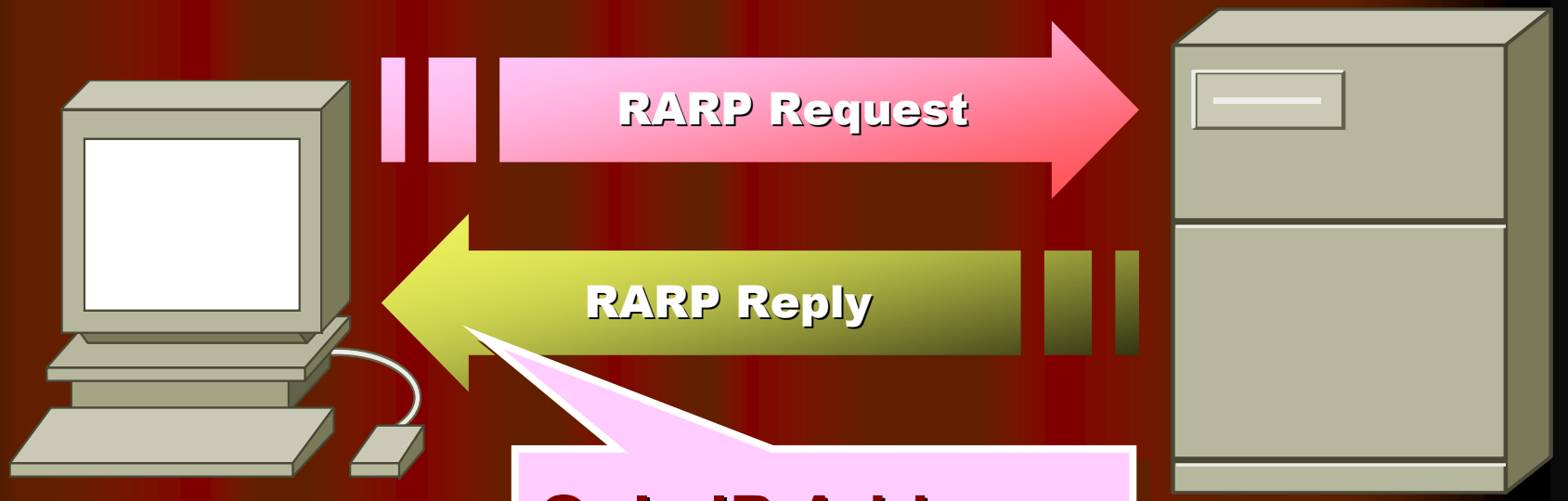
- UDP merupakan protokol connectionless, artinya tidak ada sesi komunikasi awal ketika data ditransmisikan.
- UDP merupakan unreliable protokol. Berarti pesan yang dikirim tanpa ada nomor urut dan tanpa acknowledgment dari penerima shg pengirim tidak pernah tahu apakah pesa sudah diterima penuh atau tidak. Untuk masalah ini ditangani oleh aplikasi
- Jika terjadi Lost paket data harus di-retrieve oleh layer diatasnya (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 adalah Domain Name System(DNS) dan Dynamic Host Configuration Protocol(DHCP).

DNS


Bagaimana DNS Bekerja
Meminta server lain jika tidak dapat melakukan resolving
Klien/Resolver
Server DNS
Database DNS
Name Server
Permintaan Resolving
Aplikasi
Aplikasi
UDP/TCP
Jawaban Resolving
Memberikan petunjuk server DNS lain atau request gagal

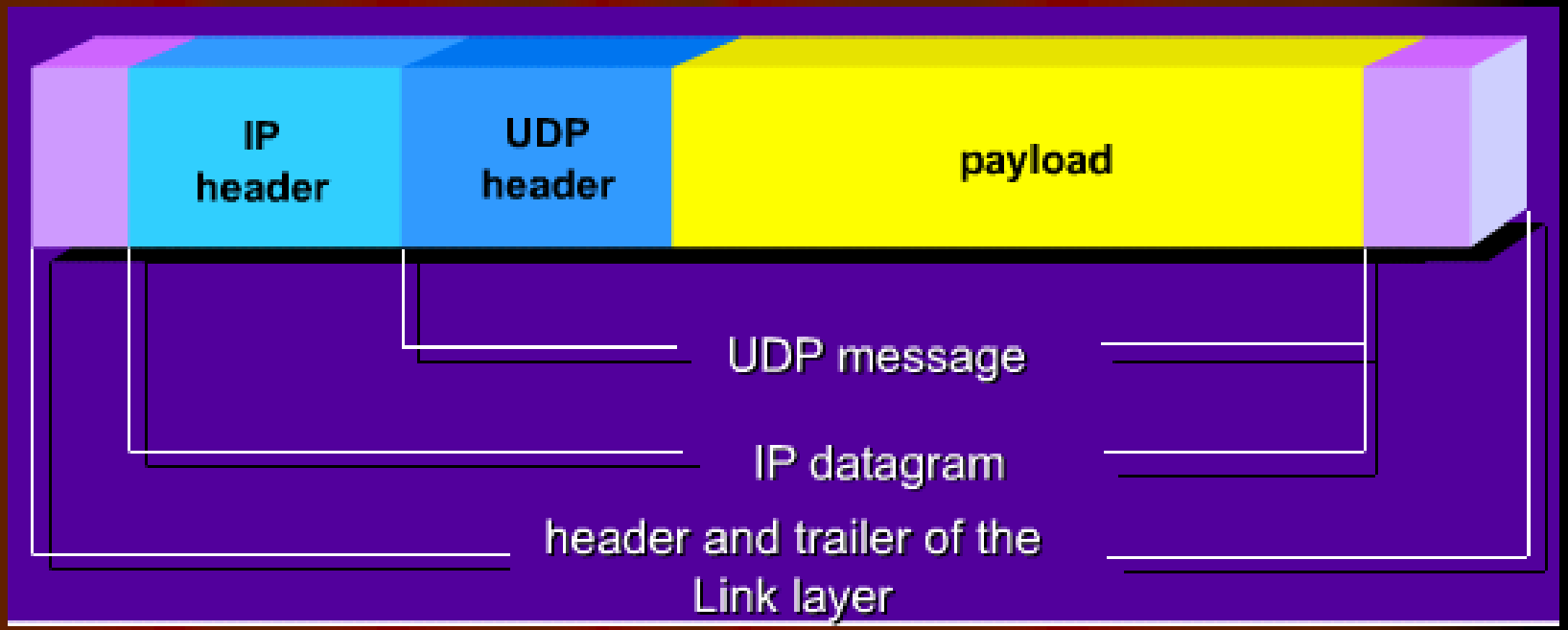

Permintaan Reverse-Mapping DNS
Server DNS
Apa sih FQDN untuk 202.123.45.6?
pikachu.indolinux.com
Klien/Resolver

System Kerja RARP



MAC HEADER	IP HEADER	RARP REQUEST MESSAGE
Destination 08-00-02-89-90-8 Source 02-60-8C-01-02-03	Destination 11111111 Source ????????	What is my IP address?

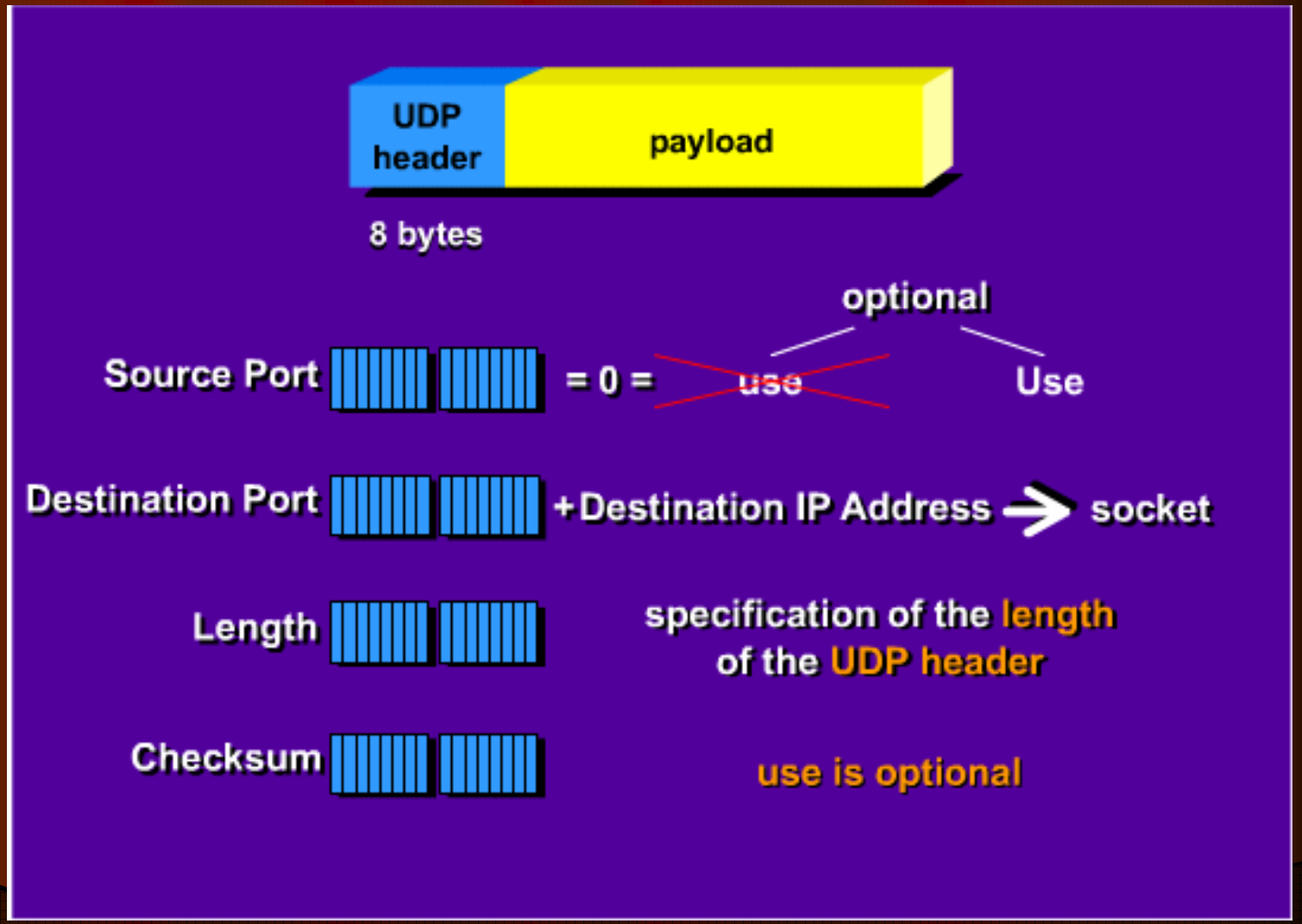
IP Datagram UDP



IP Datagram

- 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 merisi alamat penerima paket, bisa broadcast address atau multicast address.

UDP Header Structure



Struktur Header UDP

- Header UDP header mempunyai panjang yang tetap yaitu 8 bytes, Terdiri dari 4 field : Son Port, Length field dam 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 dikirimi 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.

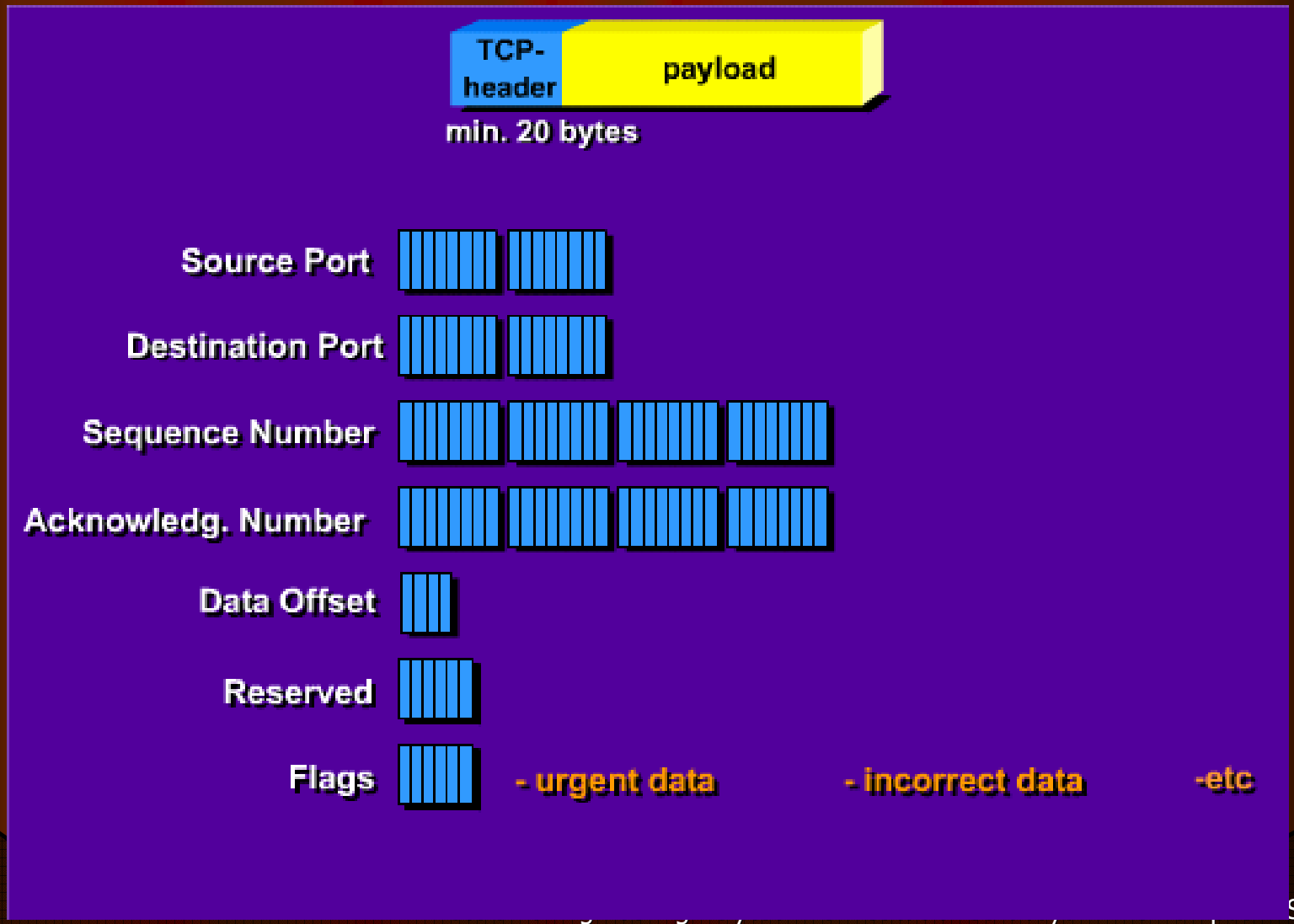
TCP Overview

- 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 fully 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, dan positive acknowledgement oleh receiver is expected, jika tidak harus retransmite data
  - Receiver akan membuang jika terjadi duplikasi data, dan resequences packets jika kedatangan tidak urut

Flow Control

- 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 berada 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



TCP Header

- 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


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

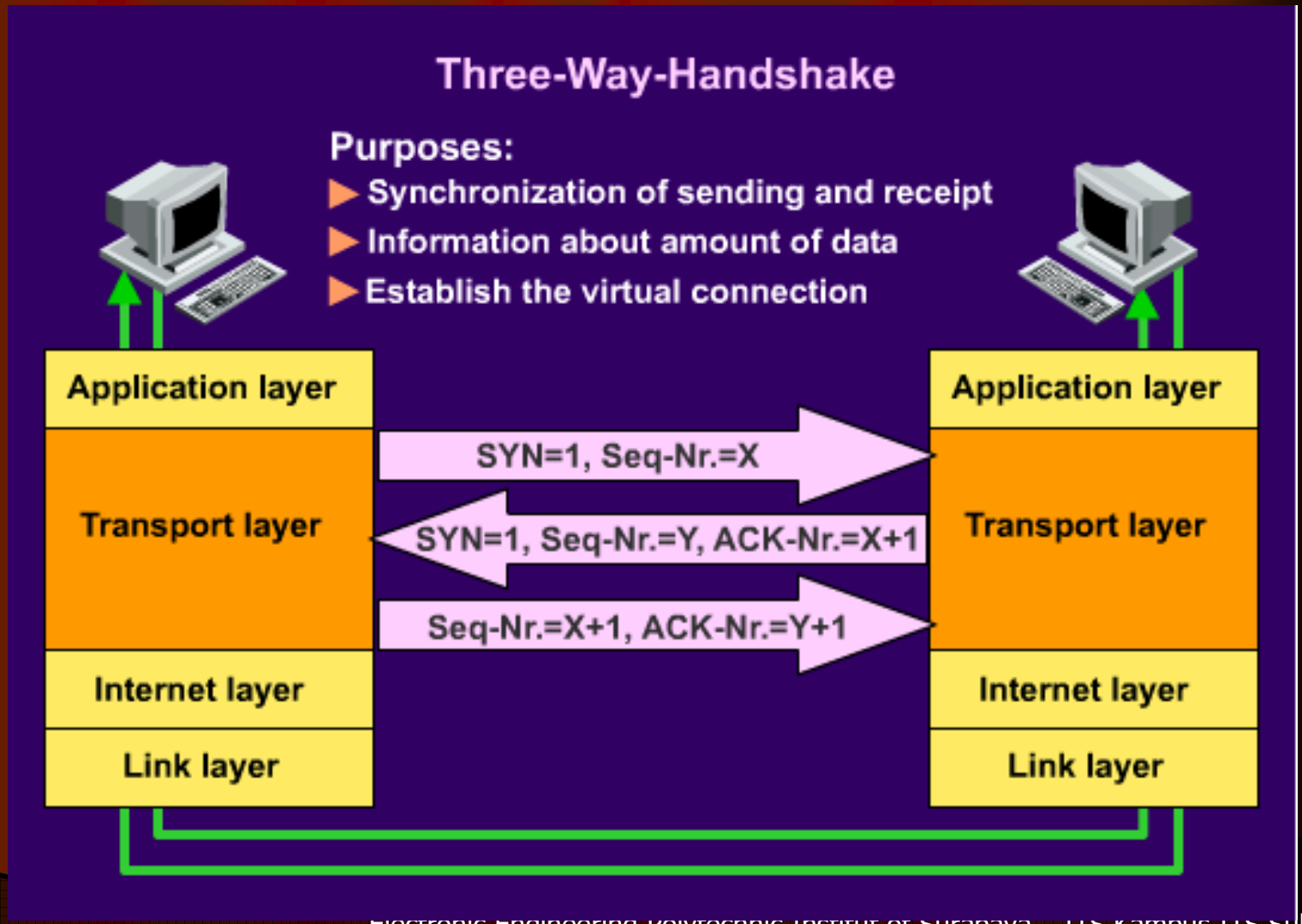
Tahapan Komunikasi pada TCP

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

Three Way Handshake

- 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 syncronisasi 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



3 Way Handshake

- Langkah-langkah:
  - Host pertama (yang ingin membuat koneksi) akan mengirimkan sebuah segmen TCP dengan flag SYN diaktifkan kepada host kedua (yang hendak diajak untuk berkomunikasi).
  - Host kedua akan meresponsnya dengan mengirimkan segmen dengan acknowledgment dan juga SYN kepada host pertama.
  - Host pertama selanjutnya akan mulai saling bertukar data dengan host kedua.

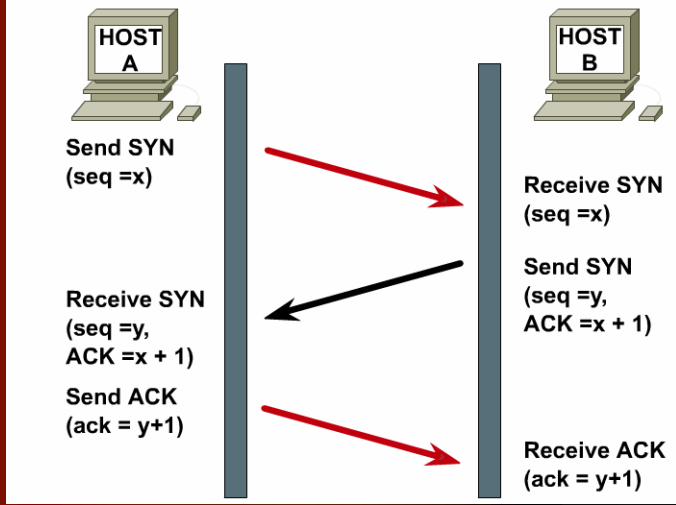
```
 Packet 1: source: 130.57.20.10  dest.:130.57.20.1
TCP: ----- TCP header -----
     TCP: Source port              = 1026
     TCP: Destination port         = 524
     TCP: Initial sequence number = 12952
     TCP: Next expected Seq number= 12953
     TCP:                 .... ..1. = SYN
     TCP: 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

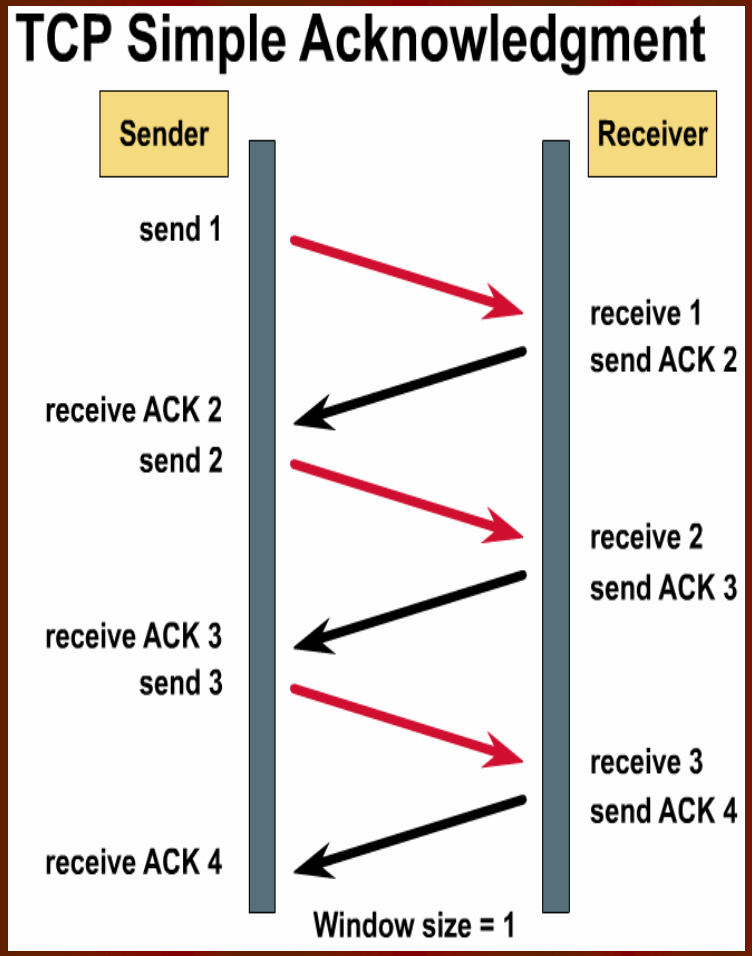


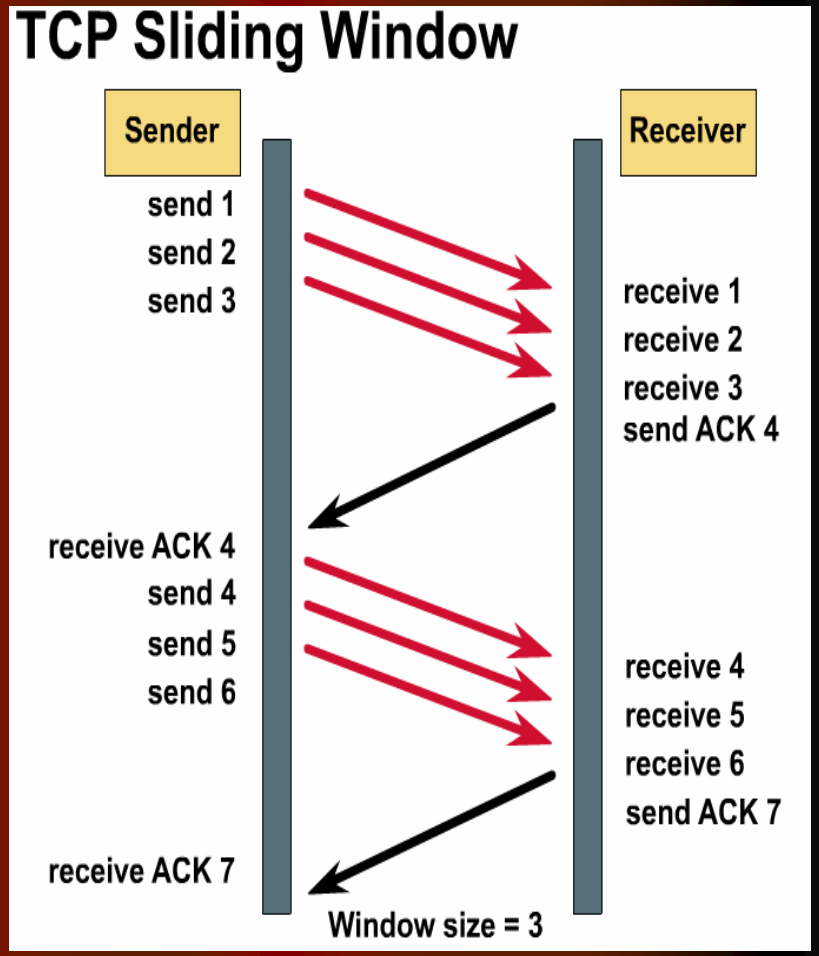
- Maksmum ukuran segment dan ukuran windows yang dinegosiasikan juga dikirim

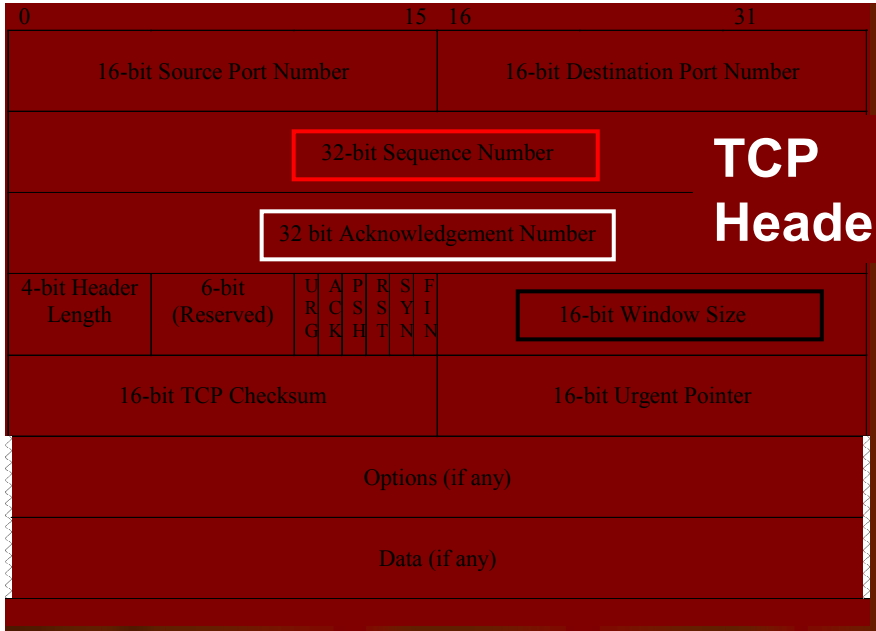
PAR

- Setelah koneksi established, ini harus dimantain sampai salah satu partner komunikasi ingin mengakhiri 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

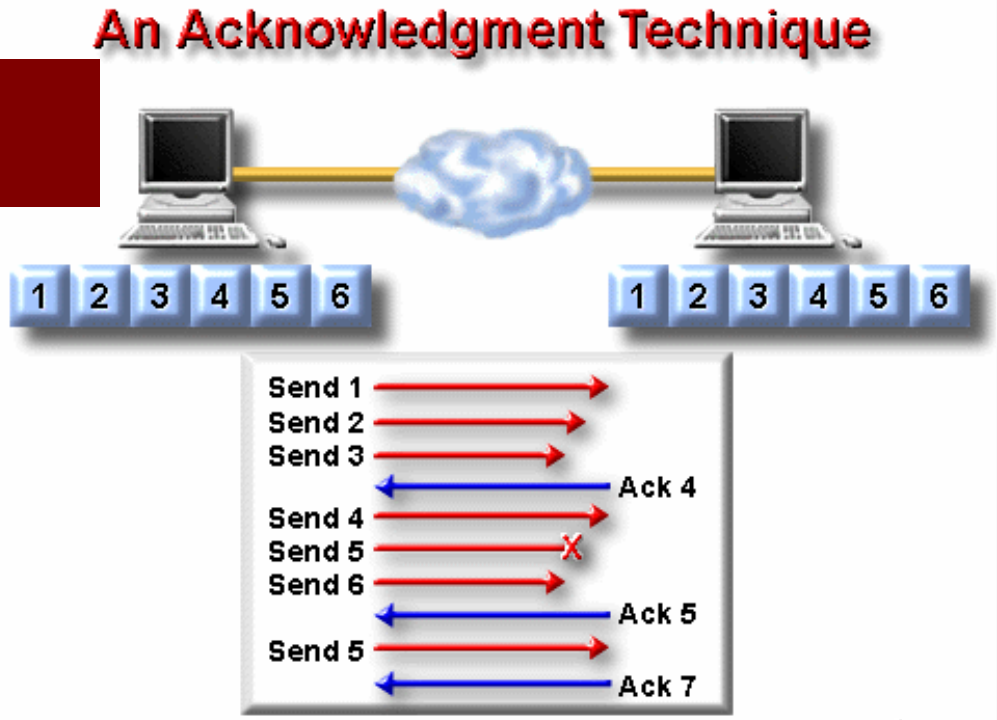
PAR


TCP Simple Acknowledgment
Sender
Receiver
send 1
receive 1
send ACK 2
receive ACK 2
send 2
receive 2
send ACK 3
receive ACK 3
send 3
receive 3
send ACK 4
receive ACK 4
Window size = 1


TCP Sliding Window
Sender
Receiver
send 1
send 2
send 3
receive 1
receive 2
receive 3
send ACK 4
receive ACK 4
send 4
send 5
send 6
receive 4
receive 5
receive 6
send ACK 7
receive ACK 7
Window size = 3



TCP Header

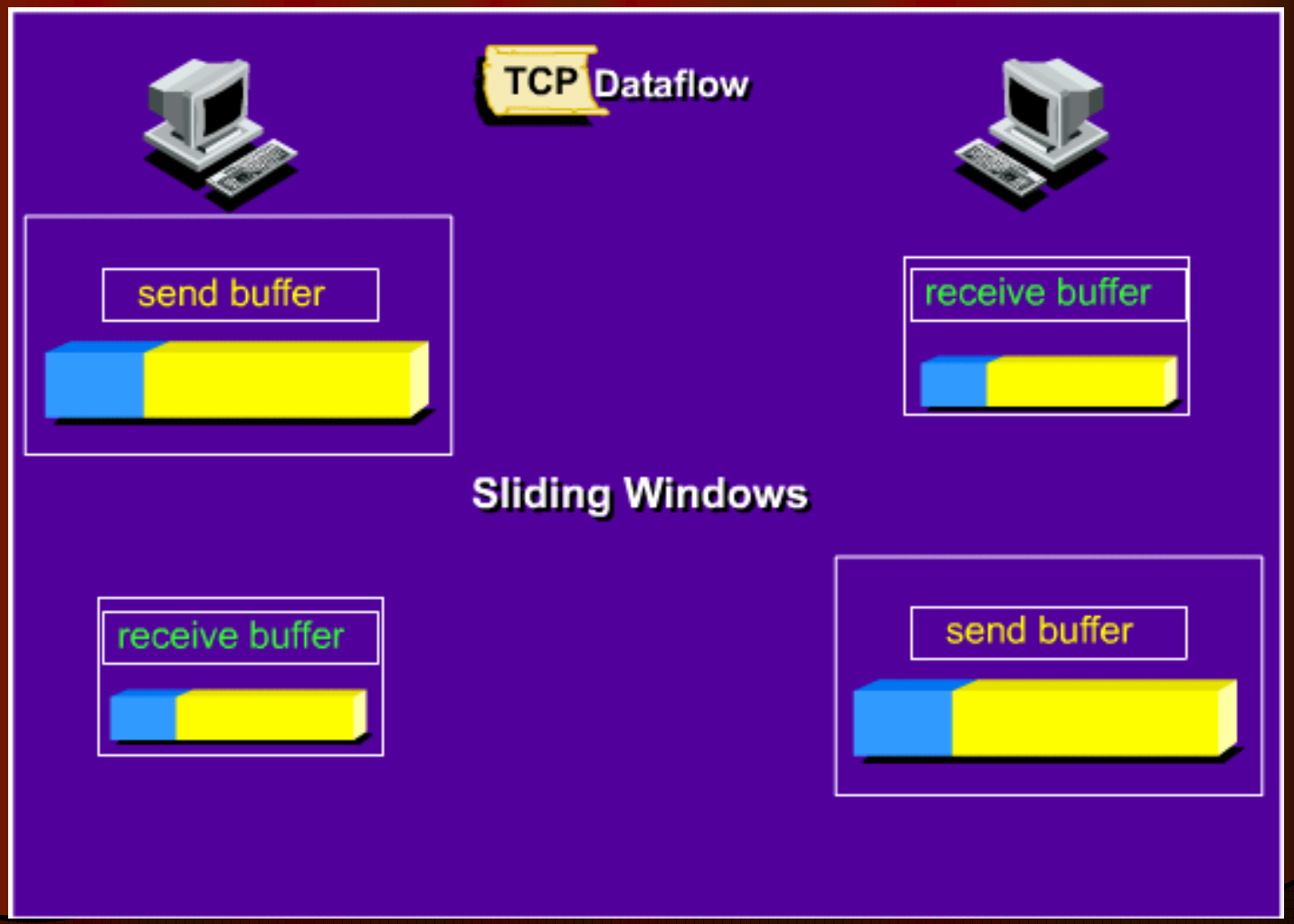


- Paket mungkin didrop sepanjang jalan, time out atau rusak
- Jika misal 4, 5, dan 6 dikirim, tapi 5 lost, receiver hanya akan memberi ack sampai 4, mengirim ack 5.
- Pengirim akan mengirim ulang paket 5 dan menunggu untuk mendengar dari penerima paket mana yang selanjutnya dikirim
- Receiver mengirim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

Sliding Window

- 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

Sliding Window

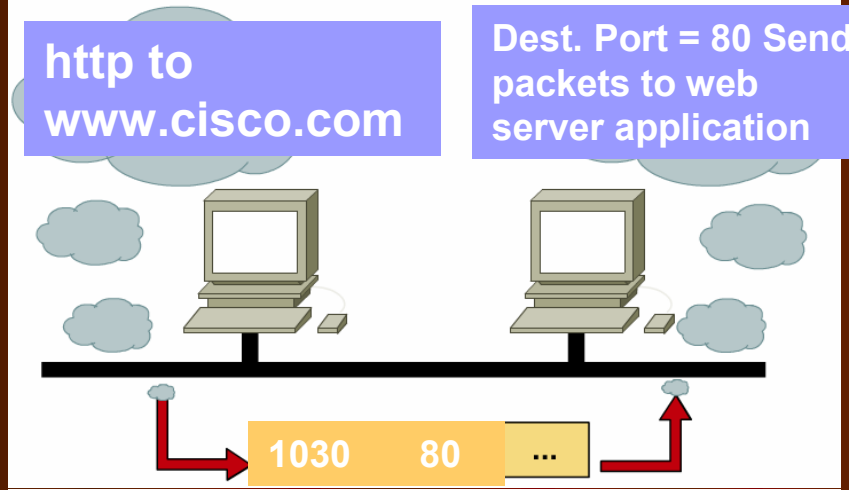


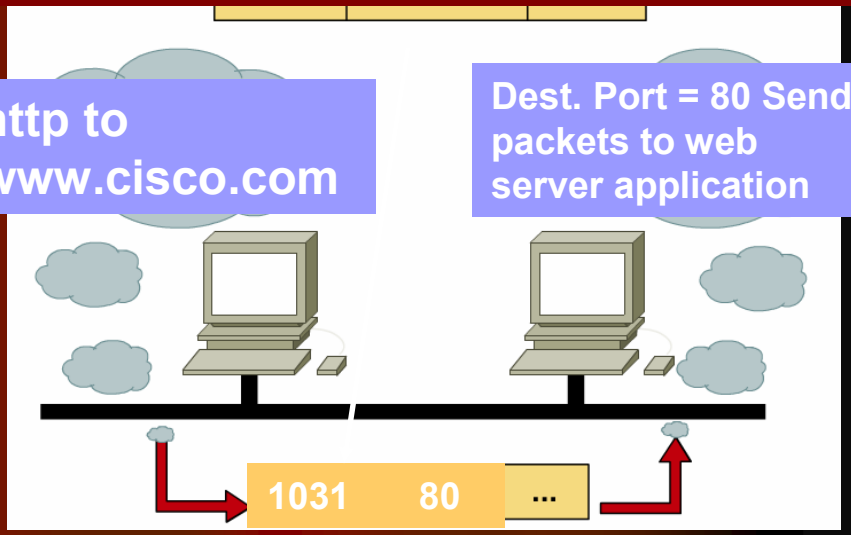
Contoh Penggunaaan Transport Layer

TCP/UDP Port Numbers

Source Port	Destination Port	...

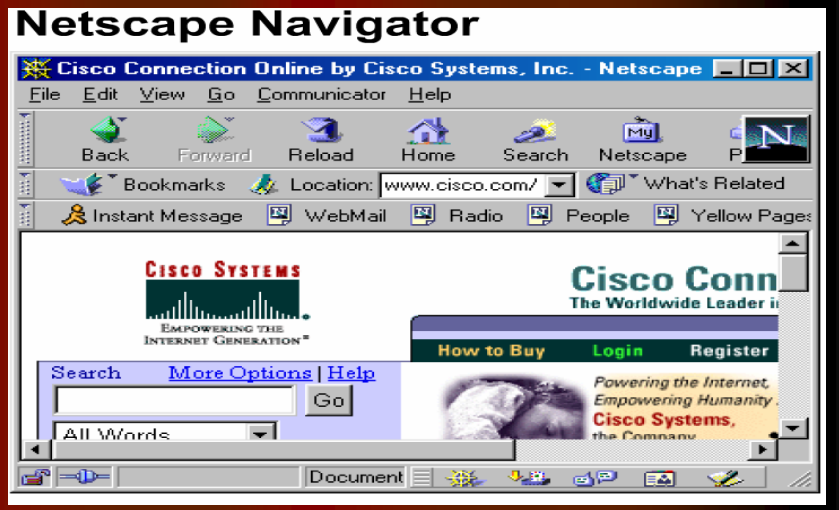
Pada setiap session http antara client dan server yang sama, mempunyai destination port yang sama, tapi berbeda Source Port (unik), untuk mengidentifikasi setiap session sehingga pengembalian permintaan masuk ke sesi yang benar





Netscape Navigator

Netscape Navigator

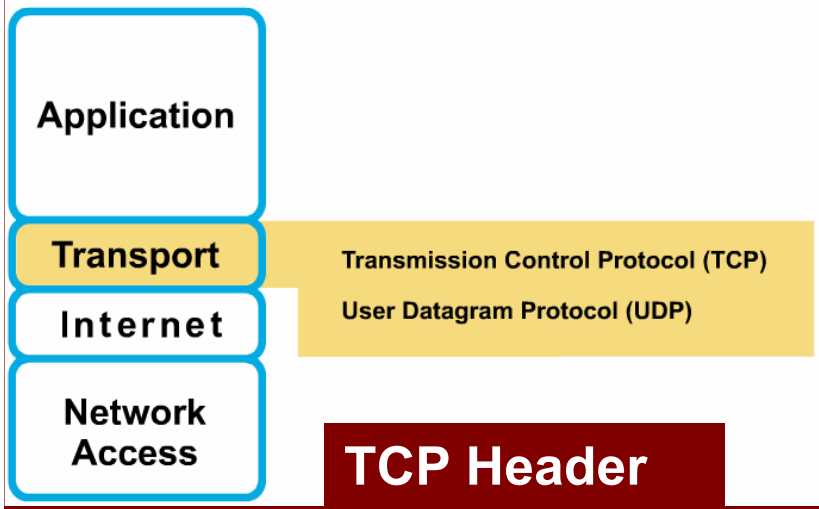


Ini menunjukkan contoh dua browser windows dengan URL yang sama.
TCP/IP menggunakan source port numbers untuk pengembalian informasi

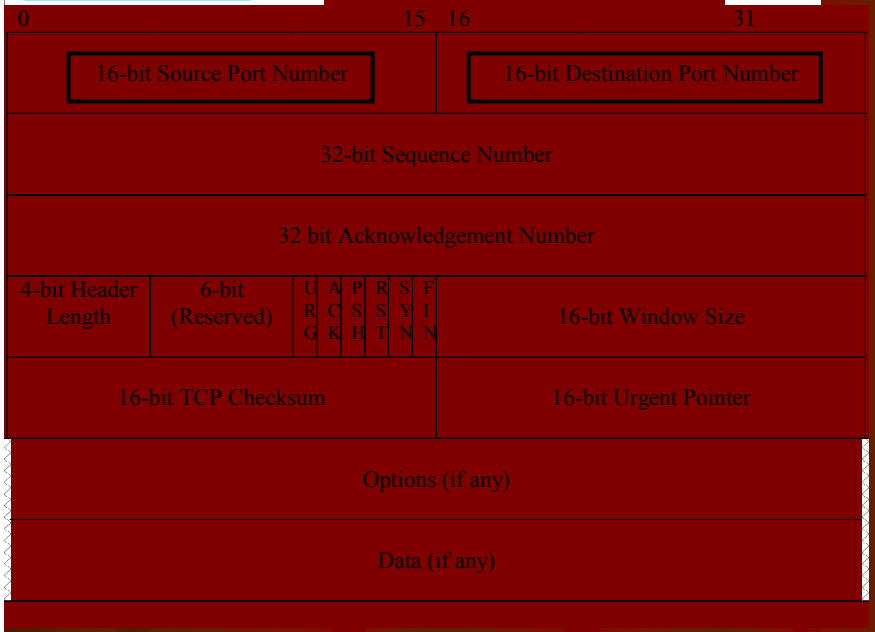
- **Aktualnya**, ketika kita membuka sebuah halaman html, maka session TCP dibangun.
- Jika membuka beberapa halaman web maka multiple koneksi TCP dibentuk, setiap koneksi di client akan membentuk port yang berbeda-beda, dengan port tujuan sama.

Hubungan antara Aplikasi, Port dan Transport Layer

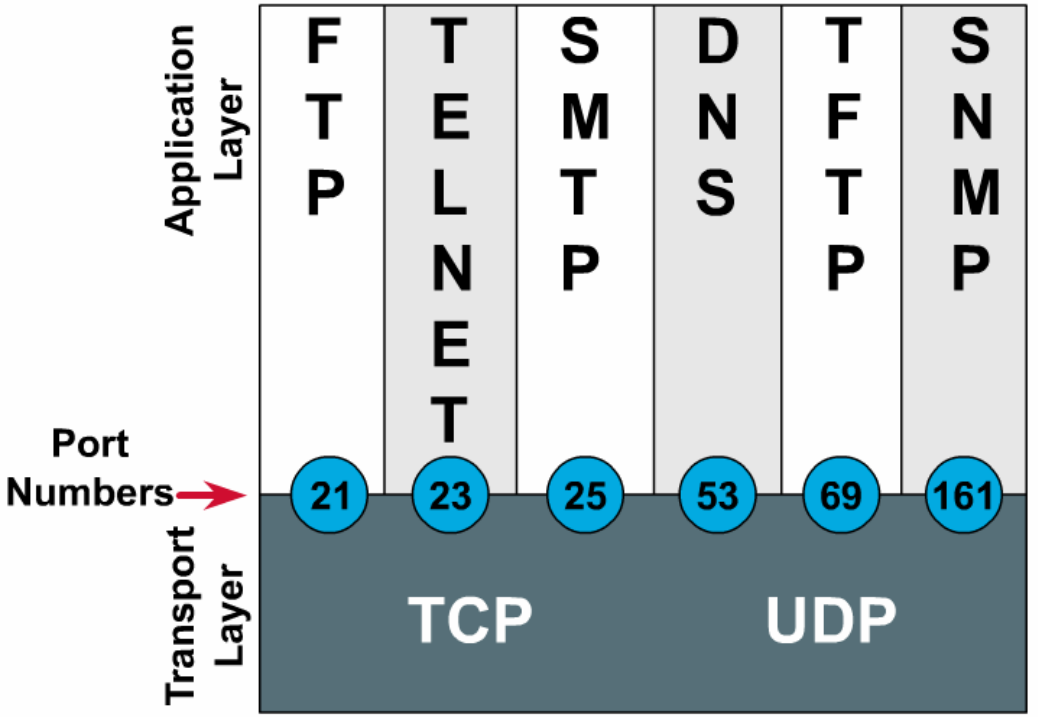
Transport Layer Overview



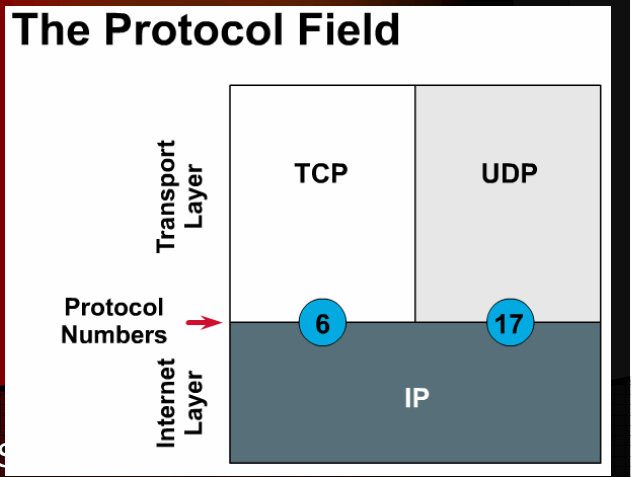
TCP Header



Port Numbers



The Protocol Field



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