

Jaringan Komputer

Pertemuan 4

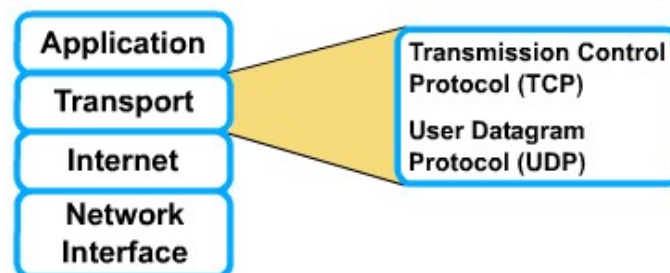


Prodi Informatika

1

Pendahuluan

- Protokol pada Transport Layer TCP/IP terdiri atas : TCP dan UDP.

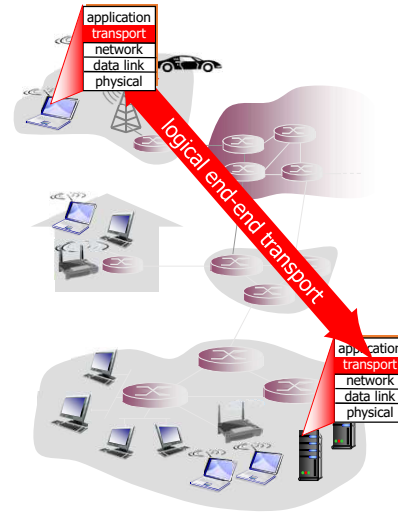


Prodi Informatika

2

Transport services and protocols

- provide *logical communication* between app processes running on different hosts
- transport protocols run in end systems
 - send side: breaks app messages into *segments*, passes to network layer
 - rcv side: reassembles segments into messages, passes to app layer
- more than one transport protocol available to apps
 - Internet: TCP and UDP



Prodi Informatika

3

Transport vs. network layer

- *network layer:* logical communication between hosts
- *transport layer:* logical communication between processes
 - relies on, enhances, network layer services

household analogy:

12 kids in Ann's house sending letters to 12 kids in Bill's house:

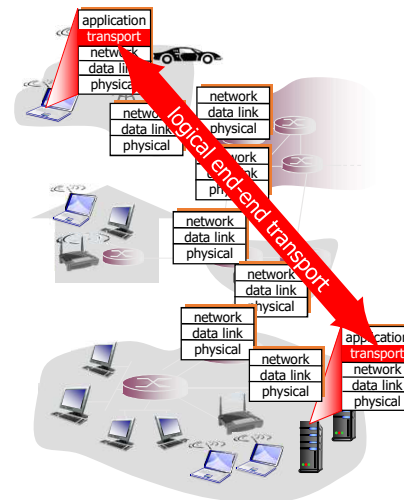
- hosts = houses
- processes = kids
- app messages = letters in envelopes
- transport protocol = Ann and Bill who demux to in-house siblings
- network-layer protocol = postal service

Prodi Informatika

4

Internet transport-layer protocols

- reliable, in-order delivery (TCP)
 - congestion control
 - flow control
 - connection setup
- unreliable, unordered delivery: UDP
 - no-frills extension of “best-effort” IP
- services not available:
 - delay guarantees
 - bandwidth guarantees



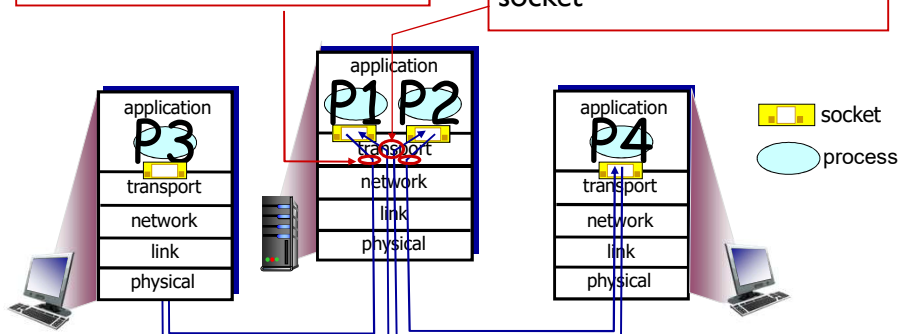
Prodi Informatika

5

Multiplexing/demultiplexing

multiplexing at sender:
handle data from multiple sockets, add transport header (later used for demultiplexing)

demultiplexing at receiver:
use header info to deliver received segments to correct socket

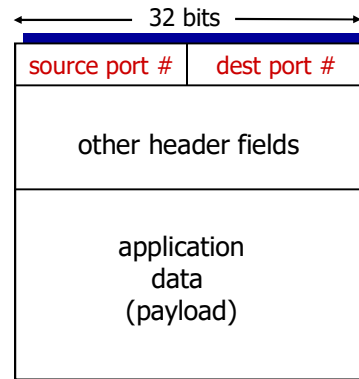


Prodi Informatika

6

How demultiplexing works

- host receives IP datagrams
 - each datagram has source IP address, destination IP address
 - each datagram carries one transport-layer segment
 - each segment has source, destination port number
- host uses *IP addresses & port numbers* to direct segment to appropriate socket



TCP/UDP segment format

Prodi Informatika

7

Connectionless demultiplexing

- recall:* created socket has host-local port #:

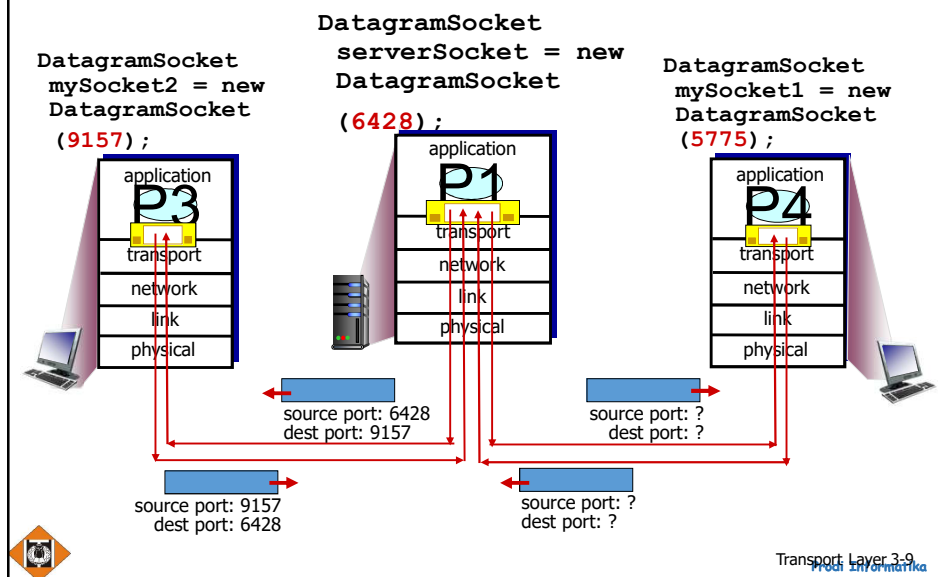

```
DatagramSocket mySocket1
= new DatagramSocket(12534);
```
- recall:* when creating datagram to send into UDP socket, must specify
 - destination IP address
 - destination port #
- when host receives UDP segment:
 - checks destination port # in segment
 - directs UDP segment to socket with that port #

IP datagrams with *same dest. port #*, but different source IP addresses and/or source port numbers will be directed to *same socket* at dest

Prodi Informatika

8

Connectionless demux: example



9

Connection-oriented demux

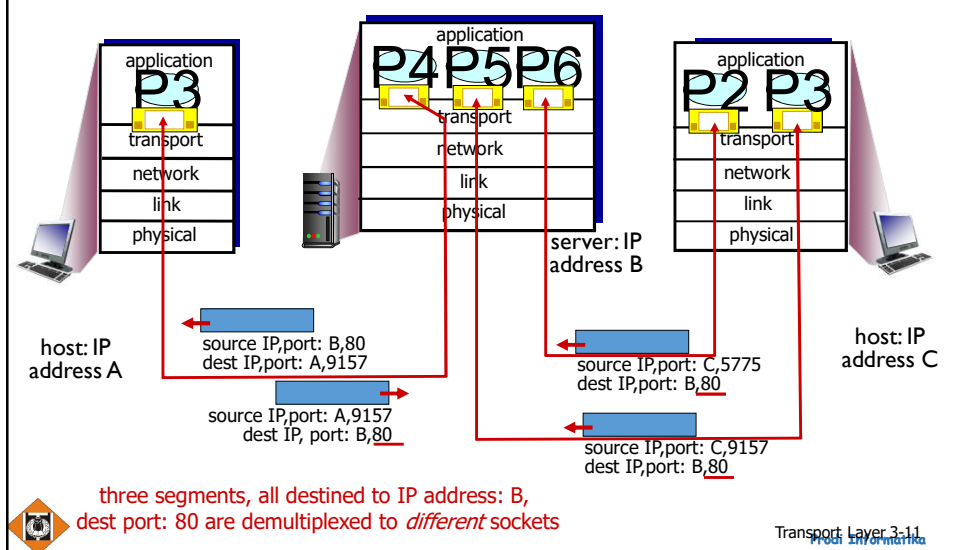
- TCP socket identified by 4-tuple:
 - source IP address
 - source port number
 - dest IP address
 - dest port number
- demux: receiver uses all four values to direct segment to appropriate socket
- server host may support many simultaneous TCP sockets:
 - each socket identified by its own 4-tuple
- web servers have different sockets for each connecting client
 - non-persistent HTTP will have different socket for each request



Transport Layer 3-10

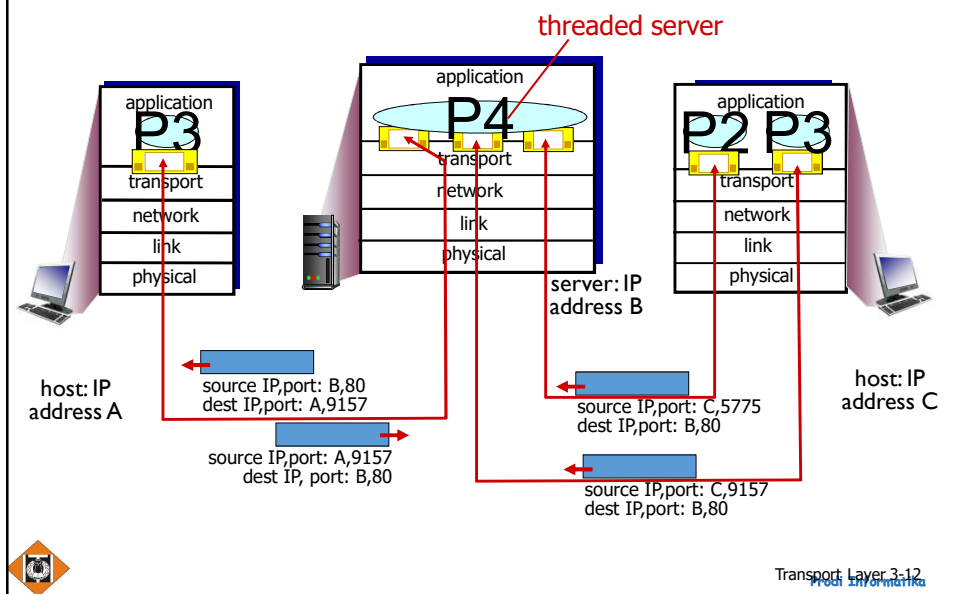
10

Connection-oriented demux: example



11

Connection-oriented demux: example



12

Overview

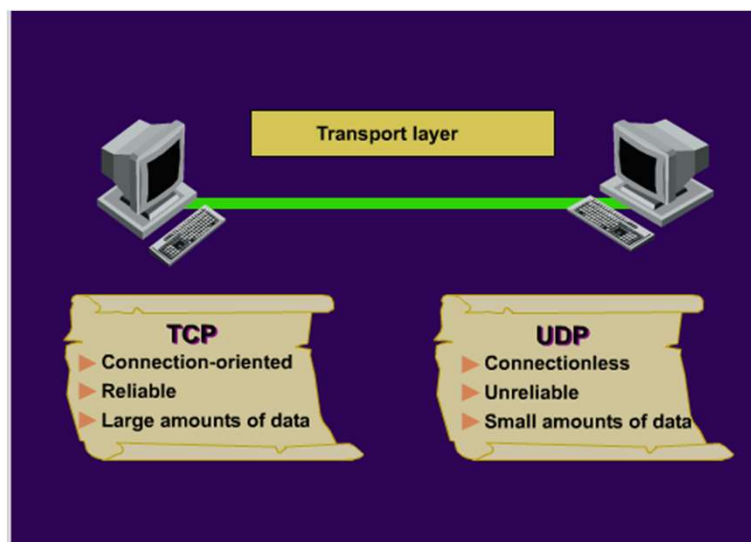
1. Layer Transport bertugas melakukan sesi komunikasi antara komputer dalam jaringan. Menentukan bagaimana data ditransmisikan.
2. Dua Protocol Transport Layer yang dipakai :
 - User Datagram Protocol
 - UDP merupakan connectionless communication, bekerja tidak menjamin data sampai ditujuan secara utuh.
 - 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



Prodi Informatika

13

Overview



Prodi Informatika

14

Port dan Socket

- 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.



Prodi Informatika

15

UDP (User Datagram Protokol)

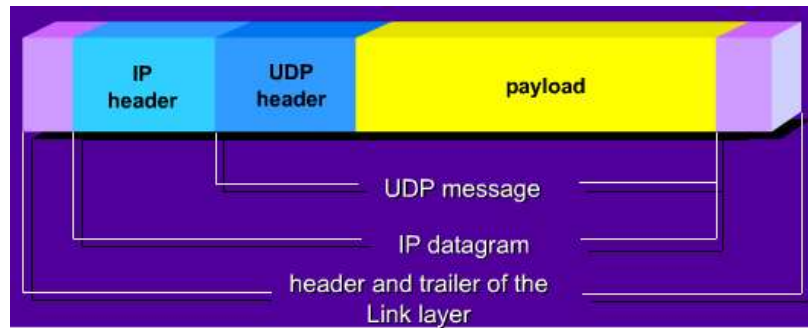
- UDP merupakan protokol connectionless, artinya tidak ada sesi komunikasi awal ketika data ditransmisikan.
- UDP merupakan unreliable protokol. Berarti pesan yang dikirim tanpa ada nomor urut dan tanpa acknowledgment dari penerima sehingga pengirim tidak pernah tahu apakah pesa sudah diterima penuh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



Prodi Informatika

16

IP Datagram UDP



Prodi Informatika

17

IP Datagram

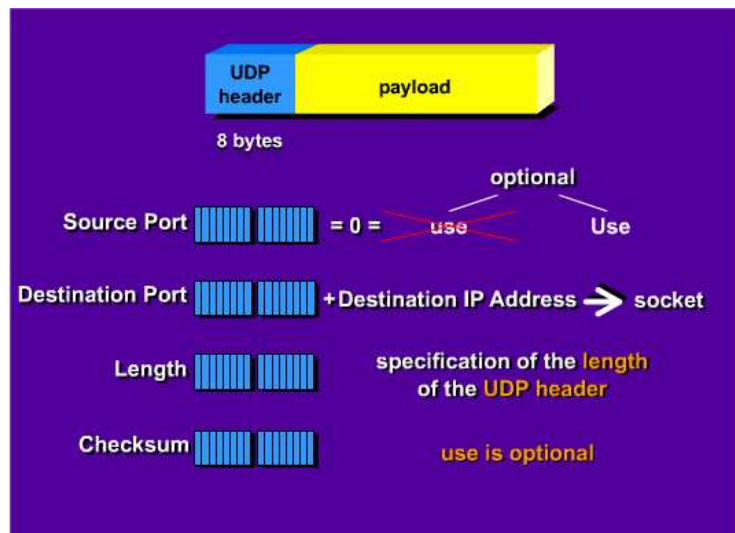
- Message UDP ditransmisikan dalam bentuk IP datagrams.
- Message UDP, terdiri dari :
 - IP Header
 - UDP header
 - Payload
- IP header terdiri dari Source IP dan Destination IP :
 - Source IP berisi IP address host yang mengirim paket
 - Destination IP berisi alamat penerima paket, bisa broadcast address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



Prodi Informatika

18

UDP Header Structure



Prodi Informatika

19

Struktur Header UDP

- Header UDP header mempunyai panjang yang tetap yaitu 8 bytes, Terdiri dari 4 field : Source Port, Destination Port 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.

Prodi Informatika

20

TCP (Transmission Control Protocol)

- 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 retransmit data
 - Receiver akan membuang jika terjadi duplikasi data, dan resequences packets jika kedatangan tidak urut



Prodi Informatika

21

Tahapan Komunikasi pada TCP

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



Prodi Informatika

22

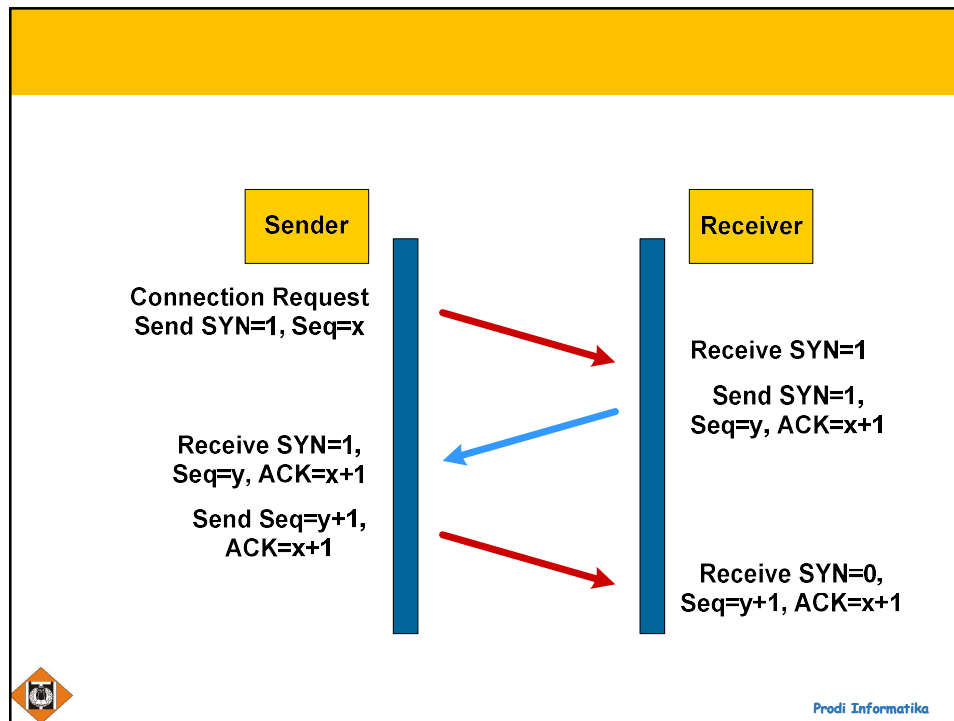
Three Way Handshake

- 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 synchronisasi 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



Prodi Informatika

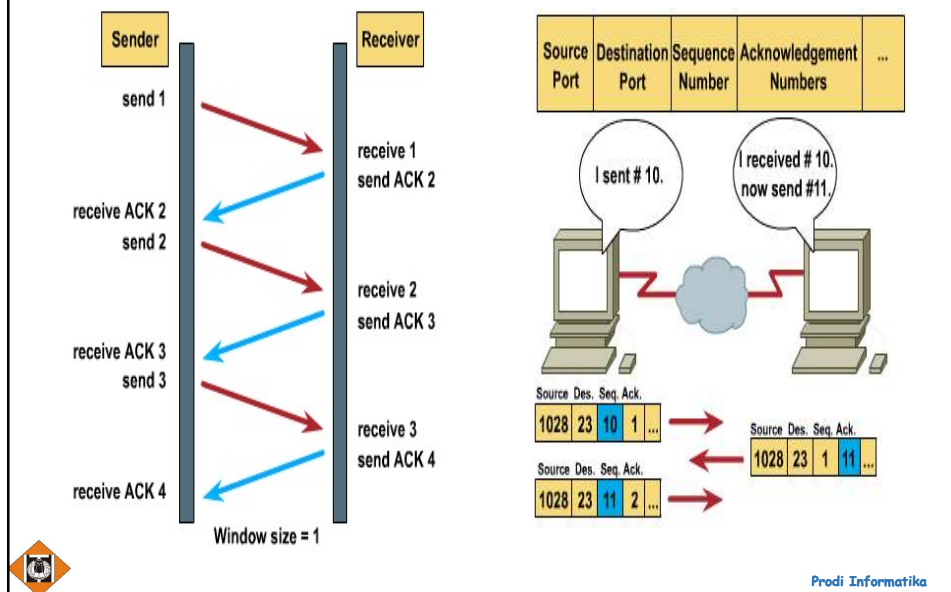
23



Prodi Informatika

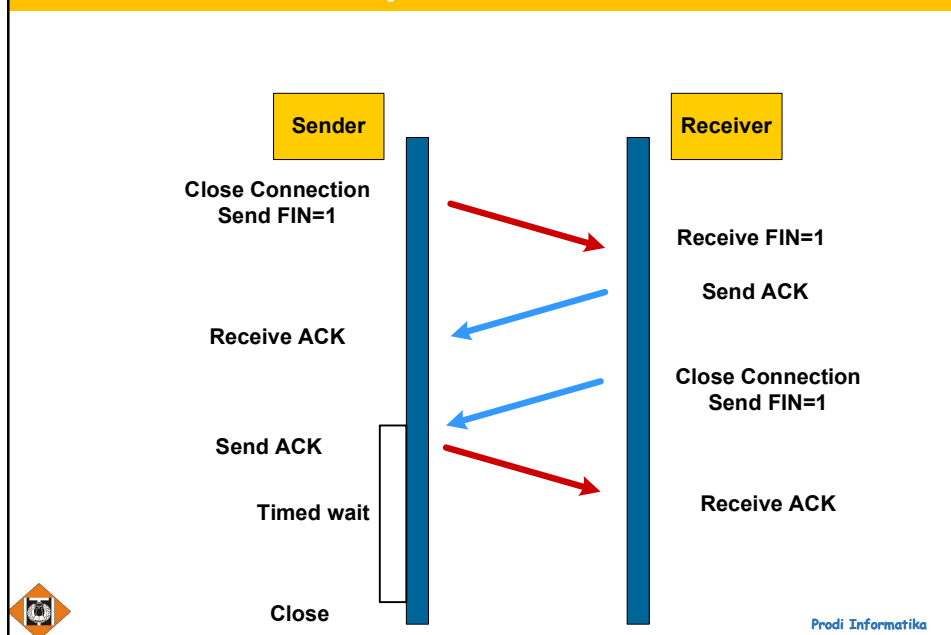
24

Fase Transaksi Pesan TCP



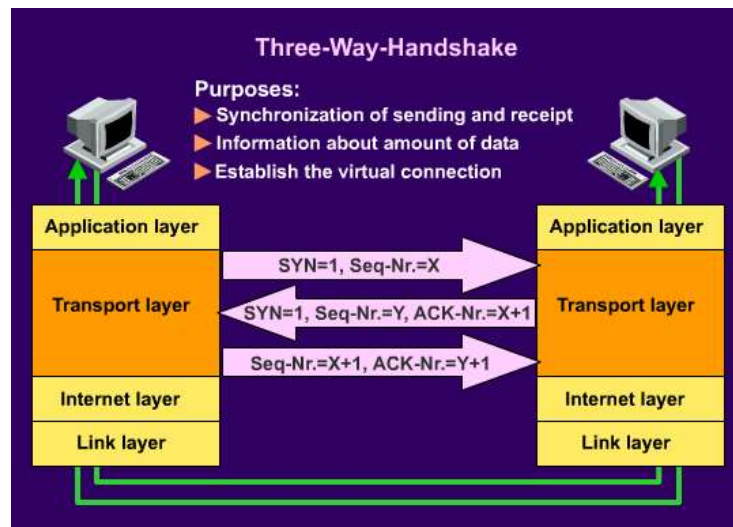
25

Fase Penutupan Koneksi TCP



26

Three Way Handshake



Prodi Informatika

27

```

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

Prodi Informatika

28

PAR (Positive Acknowledgement With Retransmission)

- Setelah koneksi establish, ini harus dijaga sampai salah satu partner komunikasi ingin mengakhir komunikasi. System Transfer Data didasarkan pada mekanisme PAR. 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

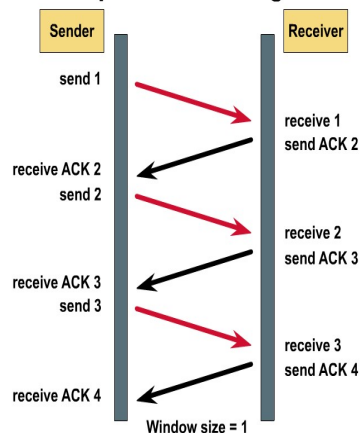


Prodi Informatika

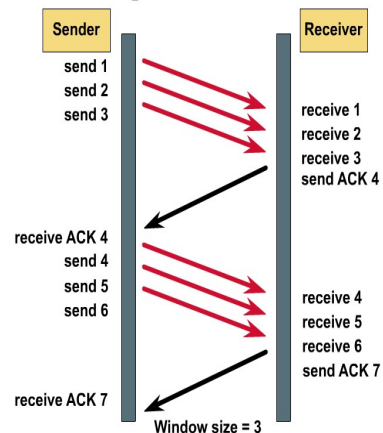
29

PAR

TCP Simple Acknowledgment



TCP Sliding Window



Prodi Informatika

30

0 15 16 31									
16-bit Source Port Number					16-bit Destination Port Number				
32-bit Sequence Number									
32-bit Acknowledgement Number									
4-bit Header Length		6-bit (Reserved)		URG		ACK		PSH	
				RST		SYN		FIN	
16-bit TCP Checksum					16-bit Window Size				
Options (if any)									
Data (if any)									

TCP Header

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, 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

31

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

32

Sliding Window



Prodi Informatika

33

Contoh Penggunaan Transport Layer



Prodi Informatika

34

TCP/UDP Port Numbers

Source Port	Destination Port	...
1030	80	...
1031	80	...

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.c

Dest. Port = 80 Send packets to web server application

1030 80 ...

Netscape Navigator

http to www.cisco.c

Dest. Port = 80 Send packets to web server application

1031 80 ...

Netscape Navigator

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

35

Netscape Navigator

www.google.com

www.cisco.com

TCP or UDP

Source IP

Destination IP

Source Port

Destination Port

Connection State

```

C:\>netstat
Active Connections
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:1434      65.194.27.219:http      ESTABLISHED
TCP    RICK-GRAZIANI:1432      216.239.37.126:http     ESTABLISHED
TCP    RICK-GRAZIANI:1519      216.239.37.126:http     CLOSE_WAIT
TCP    RICK-GRAZIANI:1120      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      209.44.195.174: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

```

netstat command

- **Kenyataannya**, 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.

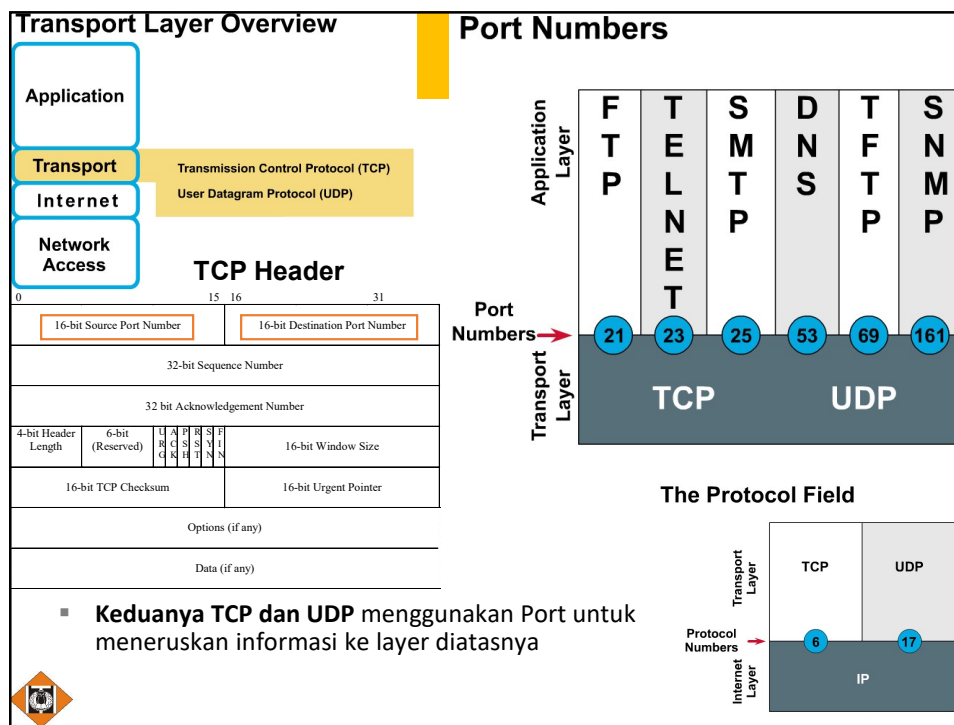
36

Hubungan antara Aplikasi, Port dan Transport Layer



Prodi Informatika

37



38