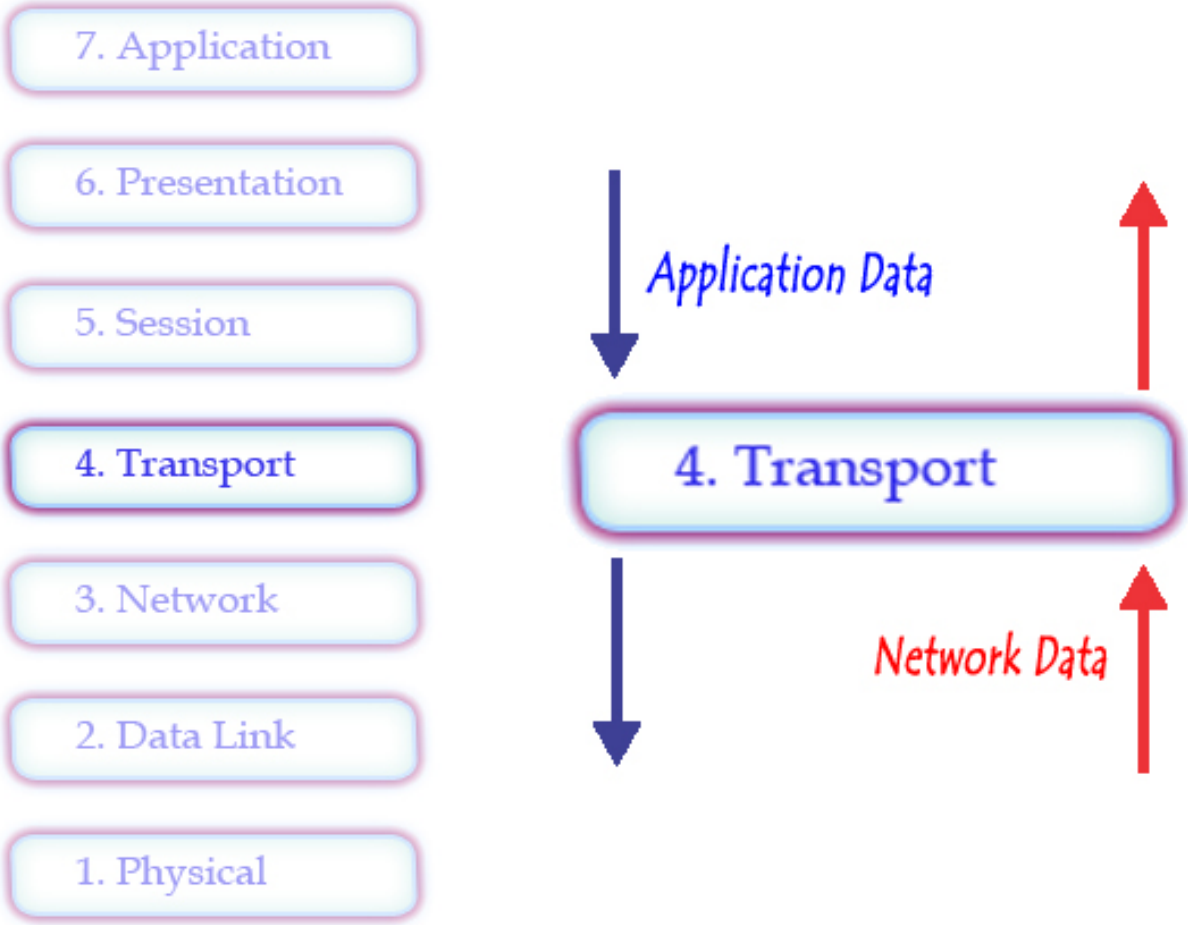
Transport Layer

Oleh : Akhmad Mukhammad

Objektif

- Menjelaskan pentingnya layer Transport.
- Mendeskripsikan peran dua protokol pada layer Transport : TCP dan UDP.
- Menjelaskan fungsi-fungis layer Transport meliputi reliability, port addressing, dan segmentation.
- Menjelaskan bagaimana TCP dan UDP menjalankan fungsi-fungsi layer Transport.
- Mengidentifikasi kapan harus menggunakan TCP atau UDP

Transport Layer



Transport Layer memindahkan data antar-aplikasi antar-device dalam network. Transport Layer menyiapkan Application Data untuk dikirim kedalam network dan menyiapkan Network Data untuk di proses oleh aplikasi.

Transport Layer -> Peran

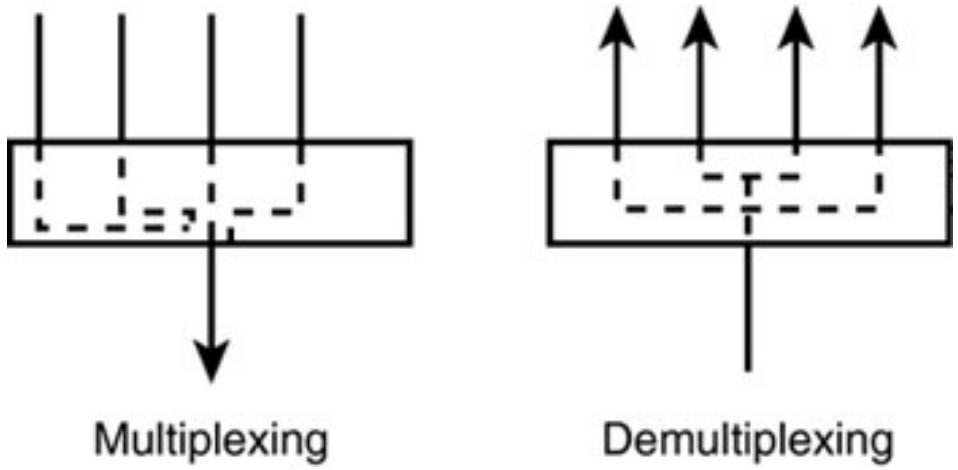
- Mengatur komunikasi end-to-end
  - Setiap host bisa saja memiliki lebih dari 1 aplikasi yang memanfaatkan network untuk proses komunikasi. Setiap aplikasi tersebut bisa saja berkomunikasi dengan satu atau lebih aplikasi pada host lain.
- Segmenting data
  - Pada umumnya network memiliki batas ukuran data dalam satu PDU.
  - Layer transport bertanggung jawab untuk melakukan segmentasi data yang diterima dari layer atas (layer application).
  - Setiap pecahan data hasil segmentasi akan di enkapsulasi dengan header yang menunjukkan komunikasi mana yang sedang menggunakan data tersebut.
  - Segmentasi data juga memungkinkan banyak komunikasi berbeda dapat menggunakan network secara bersamaan.

Transport Layer -> Peran

- Reassembling data
  - Pada sisi penerima, transport layer memanfaatkan informasi yang ada pada header untuk menyusun ulang segmen-segmen data menjadi data yang utuh sebelum diberikan ke layer atas (application).
- Identifikasi aplikasi
  - Agar data dapat disampaikan pada aplikasi yang tepat, layer transport harus mengidentifikasi target aplikasi yang dituju.
  - Untuk itu layer transport memberikan identifier kepada aplikasi yang disebut dengan port number.
  - Gabungan IP address dan port number kita kenal sebagai socket, Misalnya socket 167.205.34.1.80 menunjukkan aplikasi pada port 80 dalam komputer dengan IP address 167.205.34.1.

Transport Layer -> Peran

- Multiplexing / Demultiplexing
  - Multiplexing : membundel input dari beberapa sumber menjadi satu output tunggal.
  - Demultiplexing : menerima input dari satu sumber dan mengirimkan pada beberapa output.
  - Hal ini memungkinkan layer bawah untuk memproses data tanpa memperhatikan aplikasi mana yang menginisiasi data tersebut.



Transport Layer -> Peran

- Reliable Delivery
  - Banyak hal yang bisa menyebabkan data korup atau hilang dalam proses pengiriman, transport layer dapat memastikan penerima mendapatkan data tersebut dengan mengirim ulang data yang hilang.
- Sequencing
  - Banyaknya rute untuk mencapai tujuan dapat menyebabkan data diterima tidak berurutan, transport layer dapat menyusun ulang data secara benar dengan adanya penomoran dan sequencing.
- Flow control
  - Memori komputer atau bandwidth network tidak tak terbatas, transport layer bisa meminta aplikasi pengirim untuk mengurangi kecepatan pengiriman data. Hal ini dapat mengurangi hilangnya data dan proses pengiriman ulang.

Transport Layer -> TCP dan UDP

- Beberapa aplikasi memerlukan requirement pengiriman data yang berbeda, karena itulah dibuat protokol-protokol transport yang berbeda untuk memenuhi requirement tersebut.
- 2 protokol paling terkenal adalah TCP dan UDP.

TCP	UDP
Reliable	Unreliable, cepat, dan Low Overhead
Connection-oriented	Connectionless
Acknowledgement	Tanpa Acknowledgement
Mengirim ulang data yang hilang	Tidak ada pengiriman ulang
Sequencing.	Tidak ada sequencing, data diberikan ke layer atas sesuai dengan datangnya data.
PDU disebut Segment	PDU disebut Datagram
Overhead 20 bytes	Overhead 8 bytes.
Web, email, file transfer	Video streaming, VoIP

Transport Layer -> Port Addressing

- IANA selain berwenang atas alokasi IP address juga berwenang untuk meng-assign port number untuk aplikasi-aplikasi network.
- Well-known Ports
  - Antara 0 – 1023
  - Disediakan untuk aplikasi dan servis yang sudah umum digunakan.
- Registered Ports
  - Antara 1024 – 49151
  - Disediakan untuk aplikasi/servis yang tidak umum
  - Bisa juga digunakan secara dinamis sebagai source port di sisi client.
- Dynamic Ports
  - Antara 49152 – 65535
  - Digunakan secara dinamis sebagai source port di sisi client.

Transport Layer -> Port Addressing

Beberapa aplikasi, seperti DNS dan SNMP, memanfaatkan kedua protokol TCP dan UDP.

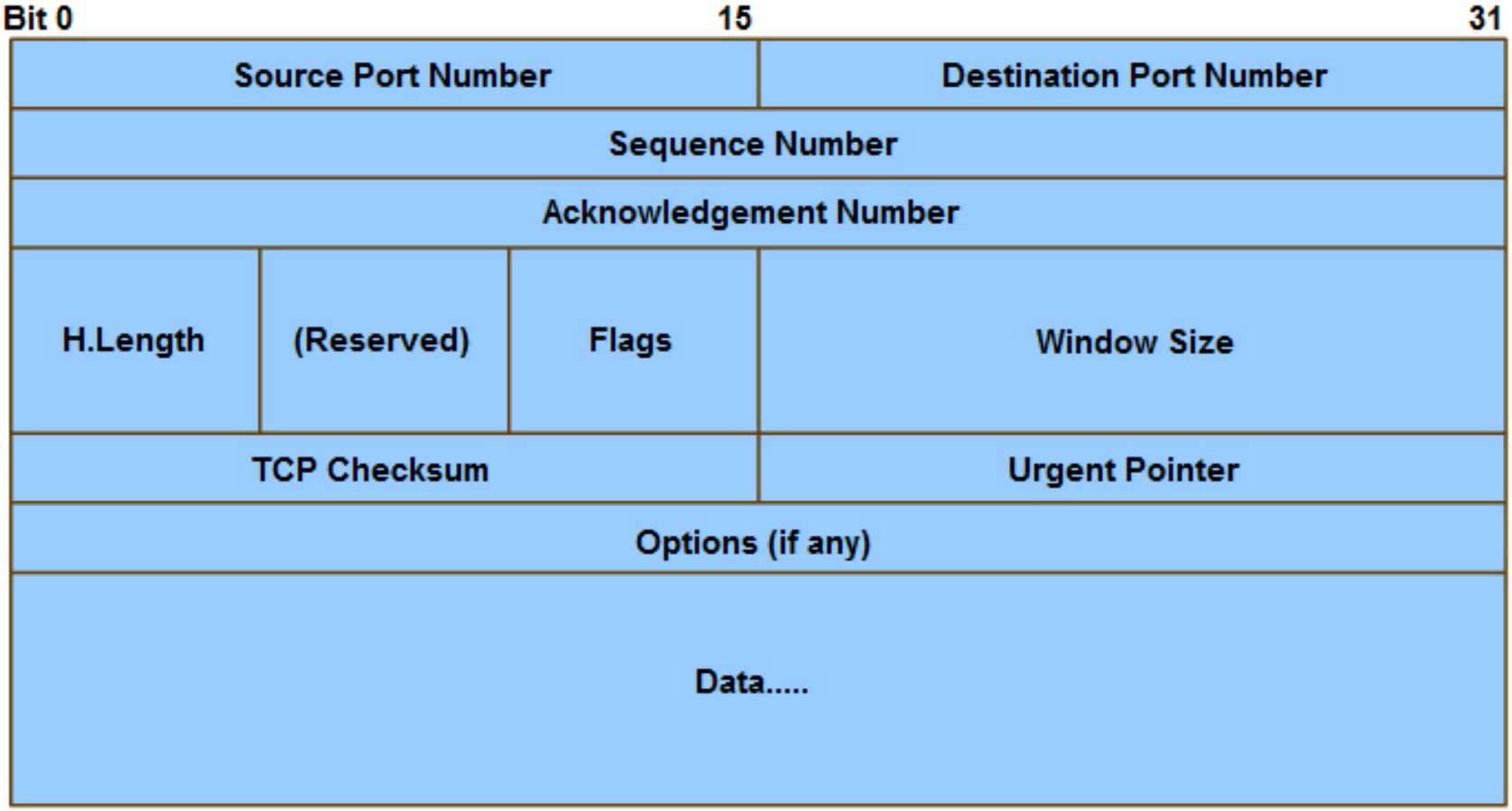
	TCP	UDP
Well-Known Ports	ftp(21), telnet(23), smtp(25), http(80), POP3(110), IRC(194), https(443), DNS(53), SNMP(161)	Tftp (69), RIP(520), DNS(53), SNMP(161)
Registered Ports	MSN(1863), Alternate http (8080), MS SQL(1433)	RADIUS(1812), RTP/voice and video transport protocol (5004), SIP (VoIP), MS SQL(1433)
Dynamic Ports		

- Cek list assignment port number di link http://www.iana.org/assignments/port-numbers
- Manfaatkan perintah netstat untuk mengetahui koneksi yang ada pada host.

Transport Layer -> TCP

- Reliability TCP dijalankan dengan membangun komunikasi connection-oriented sebelum transaksi data.
- Selain itu juga dengan acknowledgement, pengirim tahu bahwa data telah sampai ke tujuan jika menerima ack dari penerima.
- Jika tidak ada ack yang diterima maka pengirim berasumsi bahwa data yang dikirim tadi tidak sampai ke tujuan dan akan melakukan transmisi ulang data tersebut.
- Semua dukungan reliability ini diimplementasikan dalam field-field yang dimiliki oleh segmen TCP.
- Salah satu overhead TCP adalah timbulnya traffik network untuk proses ACK dan retransmision .

Transport Layer -> TCP -> Header



Field-field yang ada pada header TCP memungkinkan terjadinya komunikasi reliable dan communication-oriented.

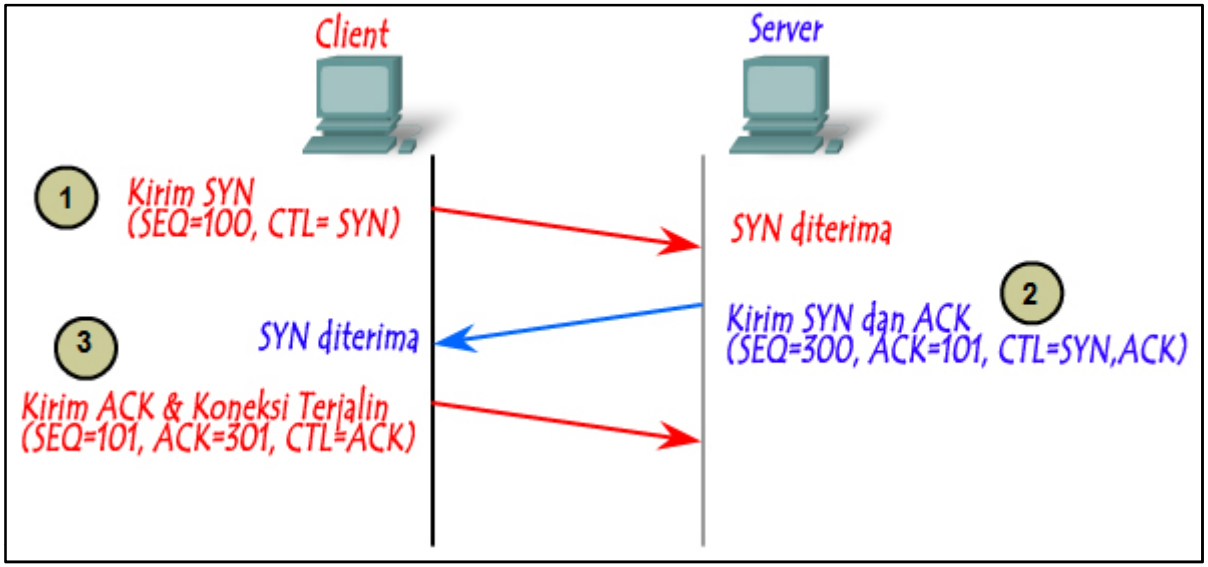
Transport Layer -> TCP -> Header

- Source Port Number
  - Port number pada device yang menginisiasi koneksi TCP
  - Biasanya bernilai random diatas 1023.
- Destination Port Number
  - Port number yang mengidentifikasikan protokol layer atas / aplikasi yang berjalan pada device tujuan.
- Sequence Number
  - Nomor urut segmen
- Acknowledgment Number
  - Nomor octet (byte) selanjutnya yang ditunggu oleh penerima.
- Window Size
  - Menunjukkan berapa banyak byte yang bisa dikirimkan sebelum menunggu datangnya acknowledgment dari penerima.

Transport Layer -> TCP -> Client Server

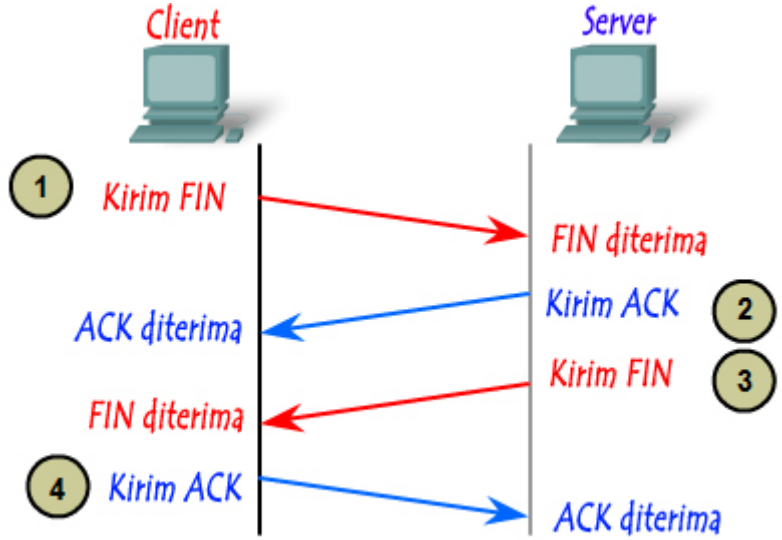
- Setiap servis akan di assign (default/manual) dengan sebuah port number.
- Dua atau lebih aplikasi servis tidak boleh menggunakan port yang sama.
- Ketika sebuah port telah di assign ke sebuah aplikasi server, maka port itu disebut open pada sisi server.
- Salah satu metode untuk meningkatkan security adalah dengan membatasi akses hanya pada port yang digunakan oleh servis saja.

Transport Layer -> TCP -> 3 way Handshake



- Sebelum transaksi data via TCP, 2 host harus menjalin koneksi.
- Client menginisiasi komunikasi dengan server.
- 3 way handshake menunjukkan :
  - Ada tidaknya mesin tujuan.
  - Apakah mesin tujuan menjalankan aplikasi yang direquest pada port tujuan.
  - Client ingin menjalin komunikasi pada port tujuan.

Transport Layer -> TCP -> Termination

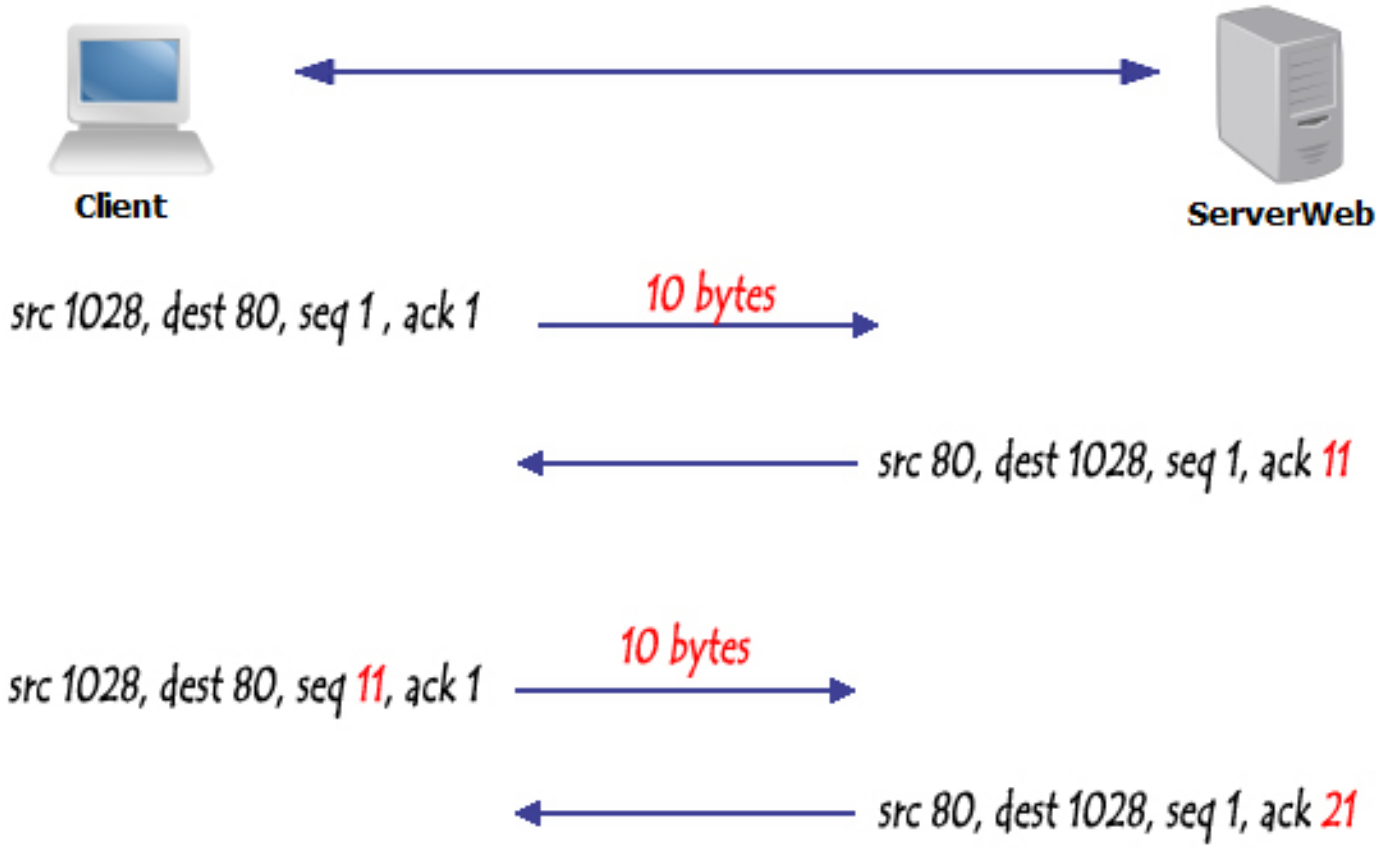


- Untuk menutup koneksi klien mengirim paket FIN (Finish).
- Dibutuhkan two-way handshake untuk menutup satu arah session.
- Ada 2 arah session dalam 1 komunikasi (client-server, server-client).
- Karenanya untuk menutup satu komunikasi client dan server dibuthkan 4 kali pertukaran data.

Transport Layer -> TCP -> Acknowledgement

- Salah satu fitur protokol TCP adalah memastikan sampainya data ke penerima.
- Layanan TCP pada sisi penerima akan mengirimkan paket acknowledgement kepada pengirim data untuk memberi tahu bahwa data telah diterima.
- Sequence number dan acknowledgement number digunakan bersamaan untuk mengkonfirmasi diterimanya sebuah segmen data.
- Sequence number mengindikasikan jumlah byte relatif yang telah dikirim dalam satu session.
- Acknowledgement number mengindikasikan byte berikutnya yang ditunggu oleh penerima, disebut juga expectational acknowledgement.

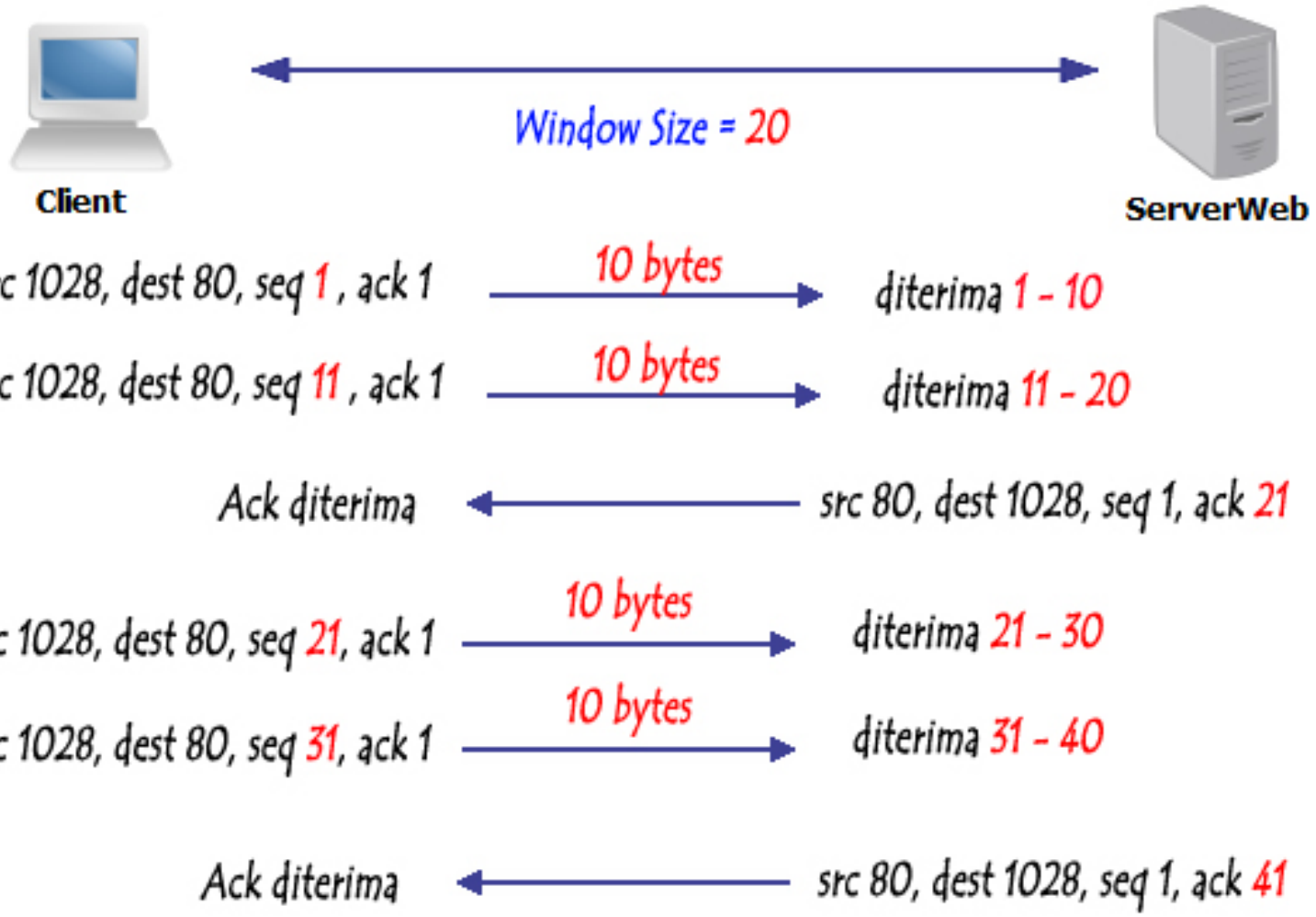
Transport Layer -> TCP -> Acknowledgement



Transport Layer -> TCP -> Flow Control

- Flow control membantu reliability proses transmisi dengan cara menyesuaikan kecepatan efektif untuk aliran data antara 2 mesin.
- Ketika pengirim (source) diberi tahu bahwa sejumlah data telah diterima, maka source dapat meningkatkan jumlah data untuk session tersebut.
- Window size adalah salah satu field header TCP yang menentukan jumlah data yang dapat dikirimkan oleh source tanpa harus menunggu adanya acknowledgement dari penerima.
- TCP akan memilih kecepatan transmisi data semaksimal mungkin yang dapat di dukung oleh network dan device dan proses retransmisi bisa dikurangi seminimal mungkin.

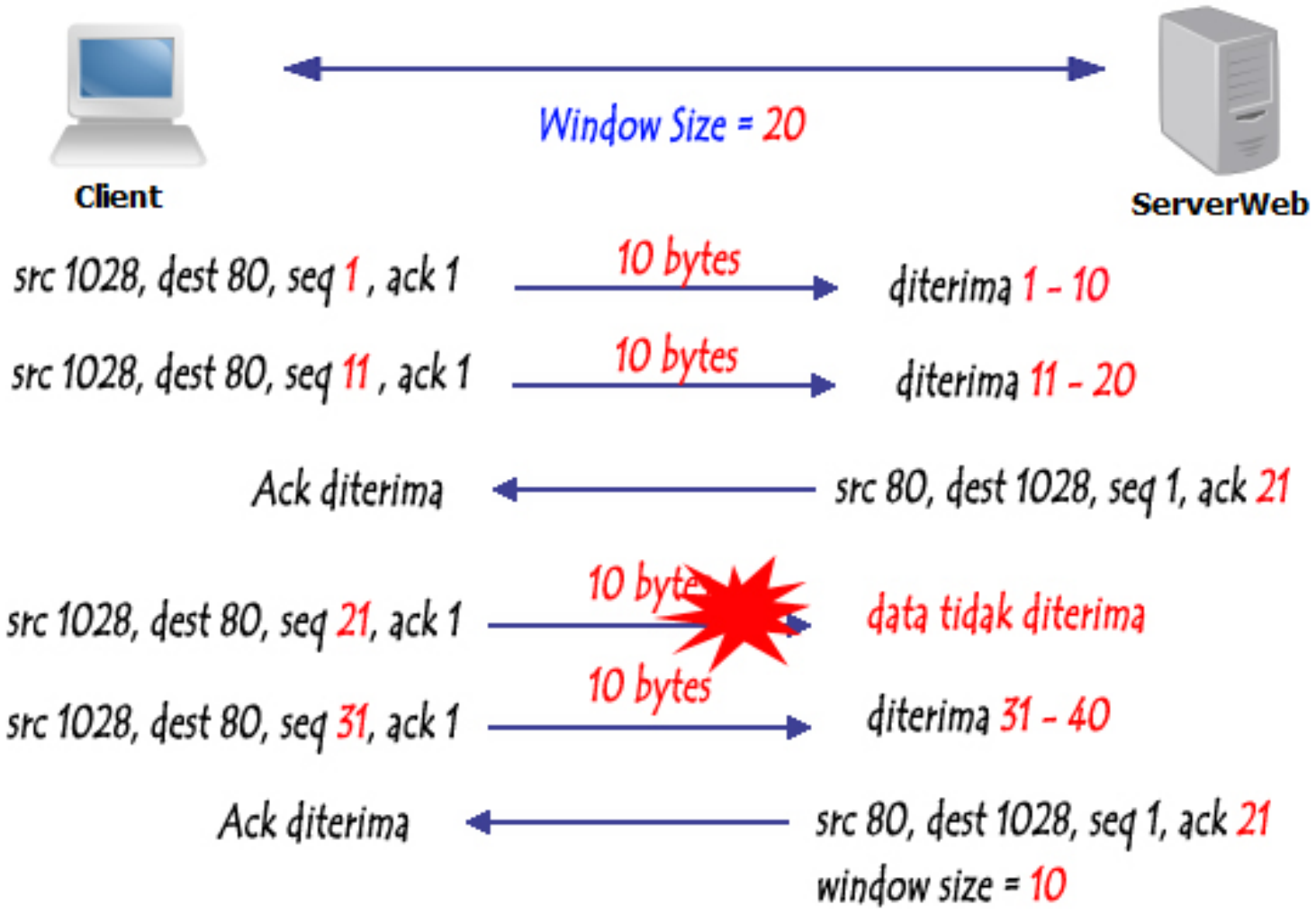
Transport Layer -> TCP -> Flow Control



Transport Layer -> TCP -> Flow Control

- Salah satu metode flow control adalah dengan menggunakan dynamic window size.
- Caranya adalah dengan mengubah-ubah nilai window size pada header TCP.
- Host penerima mengirim nilai window size yang bisa ditampung dalam satu session kepada pengirim.
- Ketika penerima ingin menurunkan kecepatan komunikasi karena terbatasnya buffer memori atau hal lain, maka dia akan mengirim nilai window size yang lebih kecil.
- Setelah beberapa kali transmisi tanpa ada data yang hilang atau buffer memori berlebih, penerima perlahan menaikkan nilai window size sehingga mengurangi jumlah acknowledgement yang harus dikirimkan.
- Nilai window size akan terus naik sampai ada data yang hilang atau alasan lain.

Transport Layer -> TCP -> Flow Control



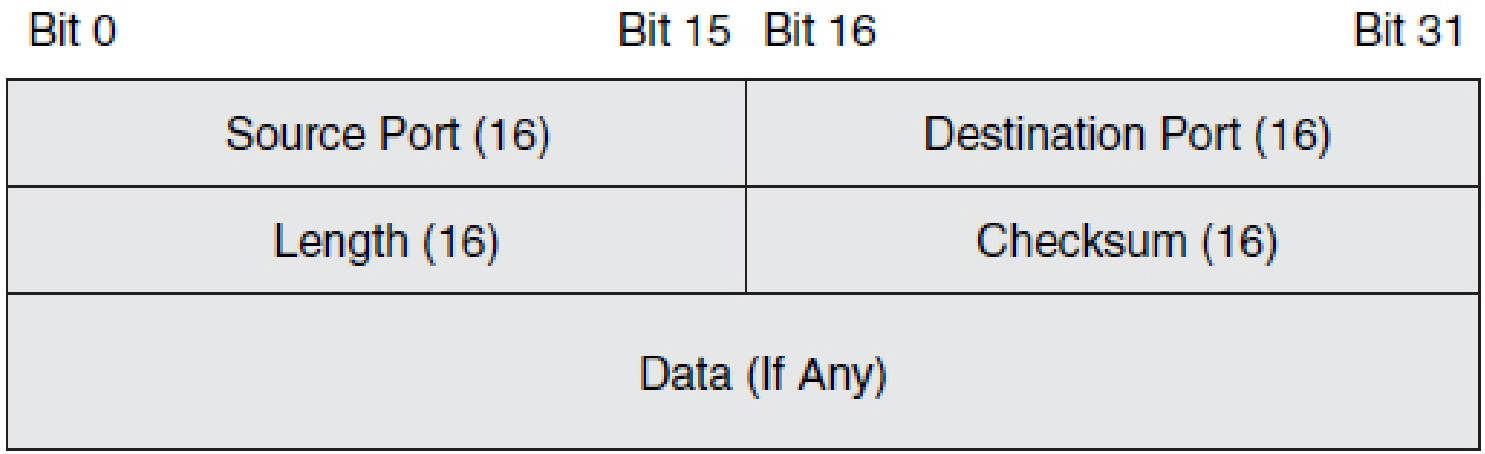
Transport Layer -> UDP

- Protokol UDP menyediakan fungsi-fungsi layer transport namun jauh lebih sederhana daripada TCP.
- Memiliki overhead yang lebih rendah daripada TCP karena tidak connection-oriented dan tidak menyediakan fitur-fitur retransmission, sequencing, dan mekanisme flow control.
- Bukan berarti UDP benar-benar "unreliable", hanya saja fungsi-fungsi yang disediakan TCP tidak ada di UDP, dan jika diperlukan harus di implementasikan pada layer lain.
- Biasanya aplikasi yang menggunakan protokol UDP adalah yang memerlukan delay serendah mungkin dan bisa mentoleransi hilangnya beberapa data.
- Contoh aplikasi : DNS, SNMP, DHCP, RIP, TFTP, online games, VOIP.

Transport Layer -> UDP

- Connection-less berarti UDP tidak menjalin koneksi sebelum mengirim data seperti yang dilakukan TCP, yang berarti data akan langsung dikirimkan begitu saja.
- PDU untuk protokol UDP biasa disebut datagram, meskipun kadang disebut juga dengan segment.
- Beberapa datagram (atau segmen untuk TCP) kadang mengambil jalur yang berbeda untuk sampai ke tujuan.
- Hal itu dapat menyebabkan datagram-datagram yang diterima dalam kondisi tidak berurutan.
- Datagram yang tidak berurutan tidak akan diurutkan seperti halnya pada TCP.
- Datagram yang hilang tidak akan dikirim ulang.

Transport Layer -> UDP -> Header



- Karena fitur yang disediakan tidak sekompleks TCP, header UDP jadi jauh lebih sederhana daripada TCP.
- Overhead juga lebih kecil karena header yang digunakan untuk enkapsulasi jadi lebih kecil.

Transport Layer -> UDP -> Client Server

- Seperti halnya aplikasi TCP, aplikasi UDP juga mendapat alokasi Well Known dan Registered port number.
- Komunikasi client/server diinisiasi oleh aplikasi client.
- Client akan memilih nomor port dynamic secara random dan menggunakannya sebagai source port.
- Karena connection-less, maka segera setelah data siap dikirimkan, UDP akan membentuk datagram dan menyerahkannya ke layer network.