Nama : Maisy Rahmawati

NPM : 1806147035

Kelas : Jarkomdat - A

**Log Minggu 4**

**Materi : Transport Layer Service - UDP / TCP**

Materi yang sudah dipahami:

* Transport service and protocols menyediakan logical communication antar application processes running pada host-host yang berbeda. Transport protocols akan running di dalam end systems. Di dalam end sistem, terdapat dua side, yaitu
* Send side: di dalam send side, akan terjadi proses break up application message ke dalam segment-segment, dan akan masuk ke dalam network layer.
* Receive side: di dalam receive side, akan terjadi proses reassembles segment-segment menjadi sebuah message, lalu akan masuk ke dalam application layer.

Banyak sekali transport protocols yang tersedia dalam aplikasi. Contohnya adalah UDP dan TCP.

* Perbedaan antara transport layer dan network layer terletak pada siapa yang dihubungkan. Transport layer adalah logical communication antar processes. Sementara network layer merupakan logical communication antar hosts.
* Ada dua jenis internet transport layer protocols, yaitu TCP dan UDP. Perbedaan antara TCP dan UDP:
* TCP bersifat reliabel, in-order byte stream delivery dimana pesan akan dikirimkan dalam urutan yang sama dengan saat mereka dikirim. Selain itu, pada TCP juga terdapat congestion control. flow control, dan connection setup. Adanya flow control pada TCP membuat sender tidak akan membebani receiver. Pada TCP, terdapat point-to-point dari satu sender dan satu receiver. Berbeda dengan UDP, pada TCP terdapat connection-oriented (connection management) yaitu adanya handshaking inits sender state dan receiver state sebelum data mengalami exchange, di mana sender dan receiver setuju untuk melakukan connection established (setiap sender atau receiver mengetahui bahwa yang lainnya bersedia untuk menghubungkan koneksi). Di dalam TCP connection, terdapat pipelined yaitu TCP congestion and flow control yang mengatur window size. TCP bisa kita sebut bersifat bidirectional data flow dalam connection yang sama.
* UDP (User Datagram Protocols) bersifat un-reliable and unordered delivery. UDP “no frills” and “bare bones” internet transport protocols. Sebagai best effort dalam service, UDP akan memperbolehkan adanya loss data dan delivered out-of-order to application. Pada UDP, koneksinya cenderung connectionless, di mana tidak ada handshaking antara UDP sender dan UDP receiver. Selain itu, setiap UDP segments ditangani secara independen dari orang lain. Biasanya UDP digunakan dalam streaming multimedia apps karena adanya loss tolerant dan rate sensitive. Alasan kenapa adanya protocols UDP karena pada UDP boleh terdapat delay karena no connection establishment, UDP bersifat simple dimana no connection state at sender and receiver, bersifat small header size, dan tidak ada congestion control karena UDP dapat blast secepat yang diinginkan.
* TCP sebagai reliable data transfer dapat membuat rdt service di atas IP’s unreliable service, yaitu dengan pipelined segments, cumulative acks, dan single retransmission timer. Retransmission timer di-trigger oleh timeout events dan duplicate acks.
* TCP merupakan fast retransmit, dimana periode waktu tunggu seringkali relatif lama. TCP akan mendeteksi lost segments melalui duplicate ACKs, dimana sender biasanya mengirim banyak segments back-to-back. Jika segments mengalami lost, maka kemungkinan akan ada banyak duplicate ACKs.

Sebagai catatan, misalkan jika sender menerima 3 ACKs dari data yang sama (triple duplicate ACKs), kirim ulang segmen yang unacked dengan smallest seq#

* Di dalam flow control TCP, receiver “advertises” free buffer space dengan memasukkan rwnd value ke dalam TCP header dari receiver-to-sender segments. RcvBuffer size akan di set melalui socket option (by default 4096 bytes). Sender akan membatasi jumlah data yang unacked ke rwnd value dari receiver.
* Saat closing-connection pada TCP, client dan server masing-masing melakukan close connection pada side mereka, dimana akan dikirim TCP segment dengan FIN bit sebesar 1. Selanjutnya, dilakukan respons FIN yang diterima dengan ACK, dimana saat FIN diterima maka ACK dapat dikombinasikan dengan FIN yang dimiliki. FIN yang secara serentak dipertukarkan juga dapat dihandle.

Materi yang belum dan kurang dipahami:

* Household analogy
* Multiplexing/demultiplexing
* Internet checksum
* TCP seq.number ACK’s.