***TRANSPORT SERVICES AND PROTOCOLS***

Berada dibawah *application layer.* Tugasnya adalah menyediakan protokol (peraturan yang perlu diikuti untuk membantu agar suatu data bisa dikirimkan). Dua protokol yang umum adalah TCP dan UDP.

***TRANSPORT LAYER VS NETWORK LAYER***

*Network Layer : logical communication between hosts.* Dapat dianalogikan *network layer* adalah pengirim paket dan menjaga lalu lintas paket.

*Transport Layer* : *logical communication between processes.* Berisi aturan - aturan agar suatu paket bisa dikirimkan.

**UDP VS TCP**

TCP

* *Congestion control*
* *Flow control*
* Perlu *connection* set up terlebih dahulu

Dari 3 ciri diatas, dapat dikatakan bahwa data itu harus dipastikan terlebih dahulu.

UDP

Setiap ada paket, dibungkus, kemudian langsung dikirim. Tidak perlu membuat koneksi terlebih dahulu. Karena di UDP memerlukan kecepatan.

***MULTIPLEXING/DEMULTIPLEXING***

* *Multiplexing* sebagai pengirim

Menangani beberapa data di *socket* dan menambah *header* yang nantinya akan digunakan untuk *demultiplexing.* Disini data akan dibungkus.

* *Demultiplexing* sebagai penerima

Menggunakan *header* untuk mengirimkan data yang diterima. Di *demultiplexing* ini akan dibuka bungkusan data sebelum dikirimkan kembali.

**BAGAIMANA *DEMULTIPLEXING* BEKERJA**

1. *Host* akan mendapatkan IP datagram dan memiliki alamatnya sendiri.
2. *Host akan menggunakan* IP *address* dan *port number* untuk mengarahkan ke *socket* yang tepat.

***CONNECTIONLESS DEMULTIPLEXING***

* Membuat *socket*
* Meletakan IP *address* dan port di UDP *socket*
* Mengecek *destinationnya* benar atau tidak
* Kemudian dikirimkan ke *destination* nya

IP datagram bisa memiliki *destination port* yang sama namun IP *address* nya berbeda.

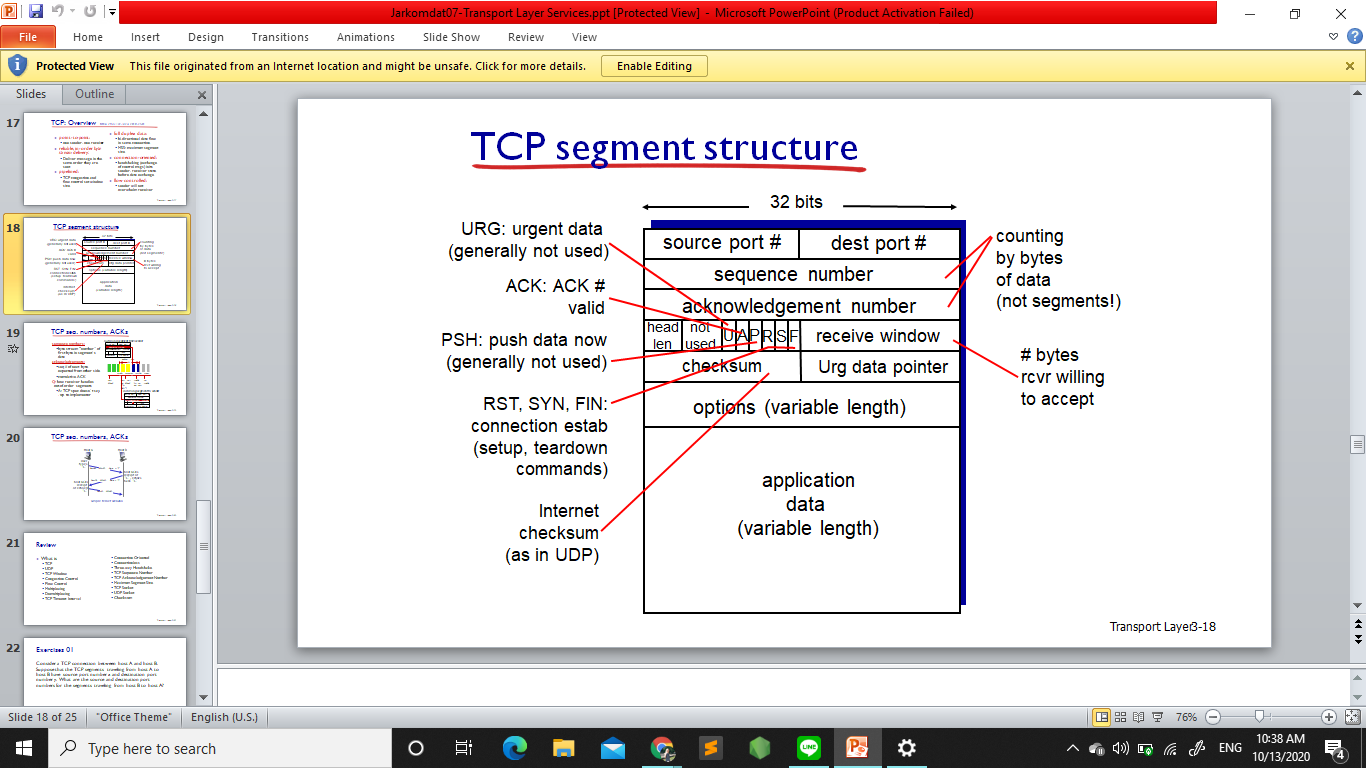
***TRANSPORT PROTOCOLS***

* TCP
  + *Reliable transport* : pada saat mengirim dan menerima proses
  + *Flow control* : tidak membebani penerima
  + *Congestion control* : mengatur kepadatan
  + *Connection - oriented* : ada *setup* antara proses *client* dan *server*

Agar TCP aman, digunakan SSL. Jadi paket - paket yang ingin dikirimkan dienkripsi. SSL berada di *application layer.*

TCP memiliki 4-*tuple* :

1. *Source* IP *address*
2. *Source port number*
3. *Destination* IP *address*
4. *Destination port number*

**

* UDP
  + *Unreliable transport*, karena ada di suatu kondisi kita membutuhkan kecepatan
  + *Connectionless,* biasa digunakan pada aplikasi multimedia. Namun ketika kita menerima *file* yang tidak lengkap bitnya, maka *file* tersebut tidak terintegrasi dan tidak bisa dibuka
  + Tidak ada inisialisasi *connection*
  + UDP *checksum :* untuk mendeteksi error atau jika ada bit yang ter-*flipped*
    - Angka di generate dari *content*
    - Kemudian ditambahkan dan dibuat *one’s complement* kemudian diletakan di UDP *checksum field*
    - Ketika sudah sampai di penerima, *checksum* tersebut di komputasi (dihitung) kembali.
    - *Checksum* ini tidak bisa menangani *error* lebih dari 2 bit

