

Spesifikasi Tugas Besar 1
IF3130 - Jaringan Komputer
"Zero Percent Loss"
Reliable Connection with UDP Socket

Dipersiapkan oleh :
Asisten Lab Sistem Terdistribusi

Didukung Oleh :



Waktu Mulai :

Jumat, 20 September 2019, 14.30.00 WIB

Waktu Akhir :

Kamis, 26 September 2019, 23.59.59 WIB

I. Latar Belakang



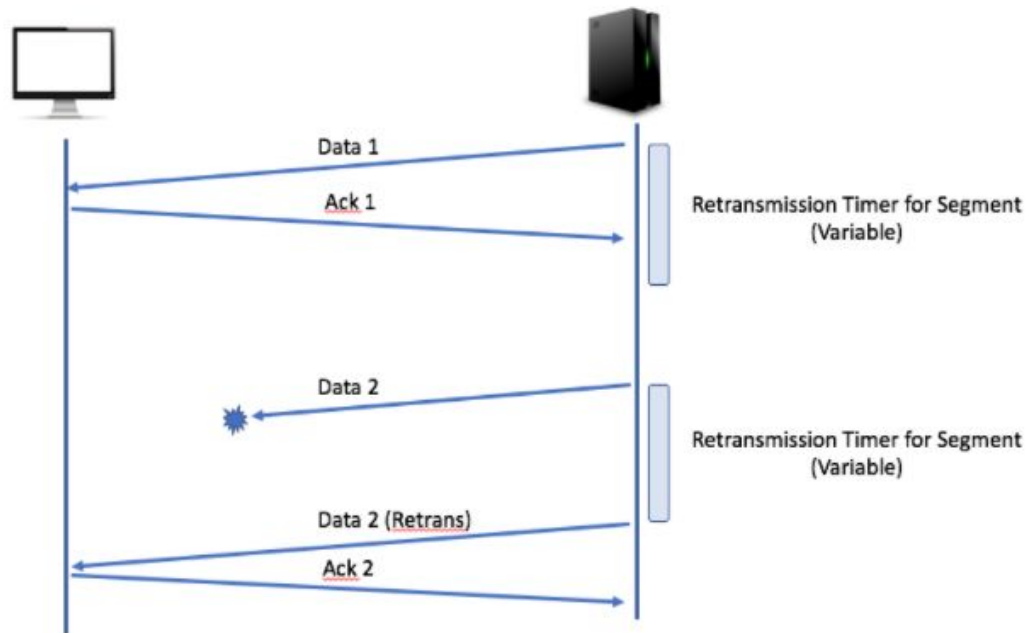
Seluruh kota di dunia tenggelam. Infrastruktur lumpuh dan tidak bisa berjalan seperti sebelumnya. Semua terjadi secara cepat, tidak ada yang menduga bahwa hal ini disebabkan oleh karena ulah sepasang remaja. Tidak ada yang percaya dan tidak ada yang akan menyalahkan mereka.

Menghadapi situasi genting ini, pemerintah darurat ingin membangun kembali infrastruktur mereka. Target utama saat ini adalah membangun ulang jaringan komputer yang dapat menghubungkan dunia dengan harapan komunikasi dengan daerah lain dapat menjadi lebih efisien. Namun sayangnya, teknologi yang manusia miliki selama 1 abad terakhir telah sirna.

Berbekal pengetahuan yang terbatas, jaringan tersebut mulai dibangun. Komputer primitif pun akhirnya terbuat beserta dengan jaringan sederhana. Namun sayangnya, jaringan tersebut sangat tidak stabil dan menyebabkan pesan menjadi hilang. Inilah saatnya anda sebagai *10x engineers* untuk membuat protokol yang tidak akan menghilangkan satu pesan pun!

II. Deskripsi Tugas

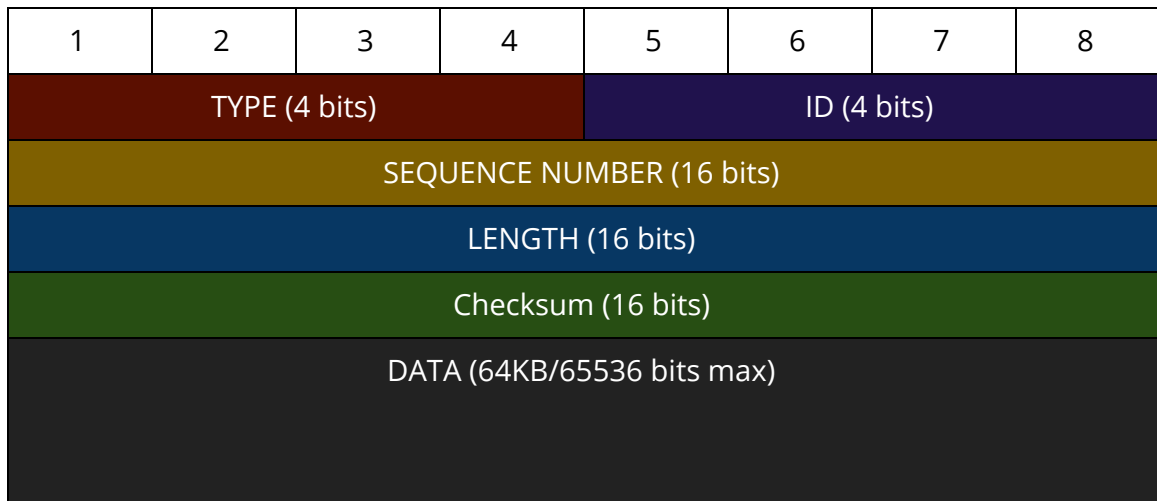
Dalam tugas besar ini, kalian akan meniru tingkah laku TCP (Transmission Control Protocol) di atas socket berbasis UDP (User Datagram Protocol). Adapun cara protokol TCP bekerja adalah sebagai berikut :



Dapat terlihat bahwa untuk setiap data yang terkirim, maka receiver akan mengirimkan ACK kepada sender tersebut. Apabila ACK tersebut tidak diterima, maka sender akan mengirimkan data yang sama kembali, hingga seluruh data diterima dengan baik.

Oleh karena itu, anda memerlukan beberapa tipe paket untuk dapat mensimulasikan hal tersebut. Berikut merupakan diagram spesifikasi paket yang akan anda implementasikan.

Diagram paket:



- **Type**
Bagian ini menandakan tipe paket yang dikirim. Value yang mungkin adalah 0x0 (DATA), 0x1 (ACK), 0x2 (FIN), dan 0x3 (FIN-ACK). Selain 0x0 dan 0x2, tipe lain tidak berisi data.
- **ID**
Bagian ini merupakan unique identifier untuk konteks data yang sedang dikirim. Hal ini memungkinkan sender untuk mengirimkan beberapa file/data secara bersamaan. Seluruh paket yang memiliki ID yang sama akan diterima receiver dan digabungkan kembali menjadi data yang utuh.
- **Sequence Number**
Sequence number menandakan urutan paket pada ID yang sama.
- **Length**
Length berisi panjang pada section DATA saja (maksimum 65536). Length merupakan unsigned integer.
- **Checksum**
Checksum merupakan hasil XOR dari seluruh paket (kecuali bagian checksum itu sendiri) yang dibagi-bagi menjadi 16 bits (2 byte). Hasilnya merupakan integer sepanjang 2 byte.
- **Data**
Data berisi keseluruhan atau potongan data yang ingin dikirim. Paket yang berisi potongan data terakhir akan dilabeli dengan type 0x2 (FIN), untuk menandakan bahwa seluruh data telah selesai dikirim. Receiver kemudian akan mengirimkan FIN-ACK untuk menandakan bahwa receiver telah

berhasil mendapatkan seluruh paket yang diinginkan. Untuk paket yang bukan merupakan paket terakhir, type yang digunakan adalah 0x0 (DATA) dan receiver akan membalas dengan 0x1 (ACK). Apabila ACK atau FIN-ACK tidak diterima oleh sender, sender akan mengirimkan paket terakhir kembali.

Penggunaan :

1. Receiver dijalankan, kemudian program akan meminta input berupa port yang akan dibind.
2. Sender dijalankan, kemudian program akan meminta address dari receiver dan PATH dari file yang user ingin kirimkan (maksimum 5 file). Langkah ini dapat diulangi beberapa kali untuk mensimulasikan multiple sender.
3. Sender akan mengirimkan file-file tersebut kepada receiver secara bersamaan. Tampilkan progress berupa print log atau progress bar (bonus).

Requirements :

1. Sender dapat mengirimkan beberapa file secara bersamaan ke sebuah receiver
2. Receiver dapat menerima beberapa file dari beberapa sender secara bersamaan
3. File yang diterima memiliki checksum yang sama dengan file yang dikirim (tidak ada paket yang hilang)
4. Sender dan receiver dapat berada di komputer yang berbeda yang berada di jaringan yang sama.
5. Bahasa yang digunakan adalah **Python 3**. Kesalahan menggunakan bahasa akan menyebabkan tugas tidak dinilai.
6. Dilarang menggunakan library external (selain bawaan Python 3).

Bonus :

1. Optimisasi memory usage dari sender dan receiver sehingga dapat mengirimkan file ukuran berapapun (penggunaan memori hampir konstan sebesar apapun data yang dikirim) tanpa mengurangi kecepatan pengiriman data secara signifikan (contoh: menulis data ke disk per paket yang dikirimkan akan menyebabkan penurunan kecepatan secara drastis).
2. Buatlah progress bar untuk menandakan persentase data yang telah dikirimkan untuk masing-masing file.

III. Pengumpulan dan Deliverables

1. Buat sebuah file zip dengan nama **IF3130-TB01-KXX-KelompokYY-NamaKelompok.zip** dengan XX adalah nomor kelas, YY adalah nomor kelompok, dan NamaKelompok adalah nama kelompok kalian. File zip ini terdiri dari 2 folder:
 - a. Folder **src** yang berisi source code yang anda buat.
 - b. Folder **doc**, berisi berkas laporan dengan nama *file* **IF3130-TB01-KXX-KelompokYY-Laporan.pdf** yang berisi:
 - i. Cover yang mencakup minimal NIM dan nama setiap anggota kelompok.
 - ii. Petunjuk penggunaan program.
 - iii. Pembagian tugas dengan dalam bentuk tabel dengan header seperti berikut:

NIM	Nama	Apa yang dikerjakan	Persentase kontribusi
-----	------	---------------------	-----------------------
 - iv. Kesulitan saat mengerjakan (jika ada) dan feedback mengenai tugas ini.
2. Teknis pengumpulan akan diberitahukan sekitar 48 jam sebelum deadline pengumpulan. Deadline terdapat pada halaman cover pada tugas ini.
Keterlambatan pengumpulan akan menyebabkan pengurangan nilai.