

Spesifikasi Tugas Besar 1

IF3130 - Jaringan Komputer

"My Luck at Gacha Games Is So Bad, I Decided to Make One Instead"

Reliable Connection with UDP Socket

Dipersiapkan oleh:
Asisten Lab Sistem Terdistribusi

Didukung oleh:



Waktu Mulai :

Kamis, 24 September 2020, 20.20 WIB

Waktu Akhir :

Kamis, 1 Oktober 2020, 23.55 WIB

I. Latar Belakang



Kamu adalah seorang mahasiswa Informatika tingkat 3 dari universitas X dari negara Y. Keseharianmu rata-rata dihabiskan dengan mengikuti kelas dan mengerjakan tugas. Untuk mengurangi kejenuhan dari rutinitas tersebut, kamu pun menemukan hiburan dalam bentuk yang tidak lain dan tidak bukan adalah game *gacha*.

Sekian tahun telah berlalu sejak kamu bermula bermain game *gacha*. Selama rentang waktu itu, kamu telah merasakan berbagai macam keindahan dan kepahitan *gacha*. Namun, entah mengapa, akhir-akhir ini yang kamu rasakan hanyalah kepahitan. Antara lain *waifu* / *husbando* yang kamu tunggu-tunggu tidak kunjung datang meskipun kamu sudah menggunakan seluruh currency gratis yang telah kamu tabung selama 2 tahun kebelakang. Lalu ada juga teman-temanmu yang pamer hasil *gacha* mereka setiap hari, termasuk *waifu* / *husbando* yang kamu incar.

Akhirnya kadar “garam” yang terkumpul melebihi batas toleransimu, menyebabkan kamu pun tidak tahan lagi.

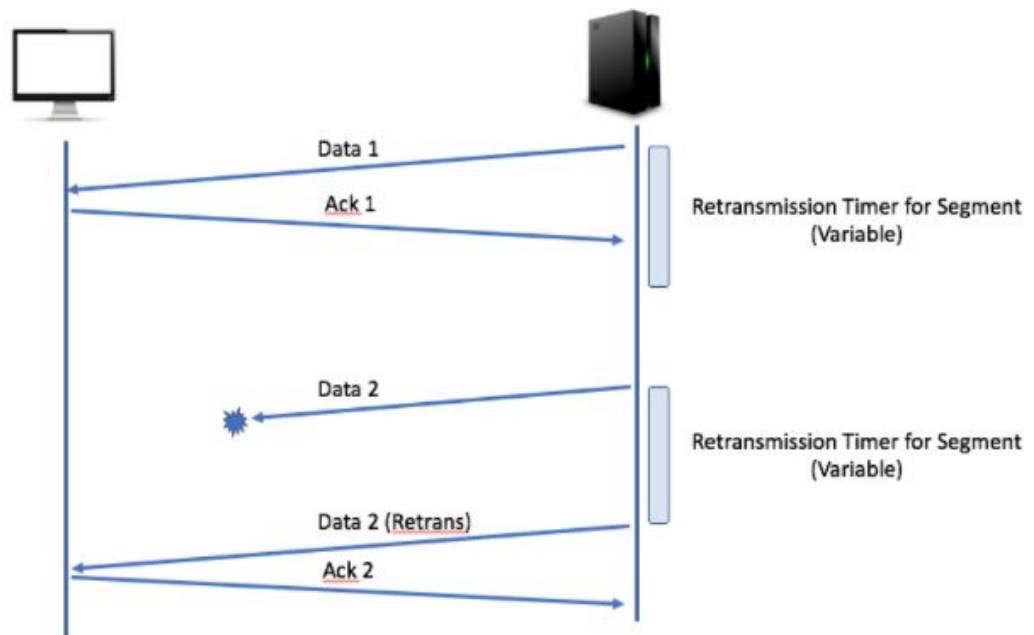
Setelah sesi introspeksi yang memakan waktu beberapa jam, matamu pun terbuka dan kamu mendapatkan sebuah pencerahan. Rasanya seakan ada seseorang yang membisikkan jawaban atas segala kesulitan yang telah kamu alami hingga saat ini. Ya, jawabannya adalah...

“Aku akan membuat game *gacha*-ku sendiri!”

Dengan tekad barumu tersebut, kamu pun memulai perencanaan untuk membuat hal tersebut menjadi kenyataan. Untungnya, kamu adalah mahasiswa Informatika, jadi membuat aplikasi bukanlah sesuatu yang asing bagimu. Namun, kamu belum familiar dengan masalah komunikasi antar *device*. Untungnya lagi, kamu semester ini mengambil mata kuliah yang bernama Jaringan Komputer. Dengan bantuan materi dari matkul tersebut, kamu pun mencoba untuk membuat sebuah proyek kecil untuk meningkatkan pemahamanmu dengan tujuan akhir membuat game *gacha* mu menjadi kenyataan.

II. Deskripsi Tugas

Dalam tugas besar ini, anda akan mengimplementasikan protokol yang mirip seperti TCP (Transmission Control Protocol) di atas socket berbasis UDP (User Datagram Protocol). Protokol yang anda implementasikan harus dapat menjamin bahwa *packet* yang dikirimkan oleh *sender* diterima oleh *receiver* tanpa terjadi *loss*. *Sender* akan mengirimkan suatu *file* dan *file* tersebut harus diterima oleh *receiver* tanpa ada bagian yang hilang. Adapun cara kerja protokol tersebut secara sederhana digambarkan sebagai berikut.

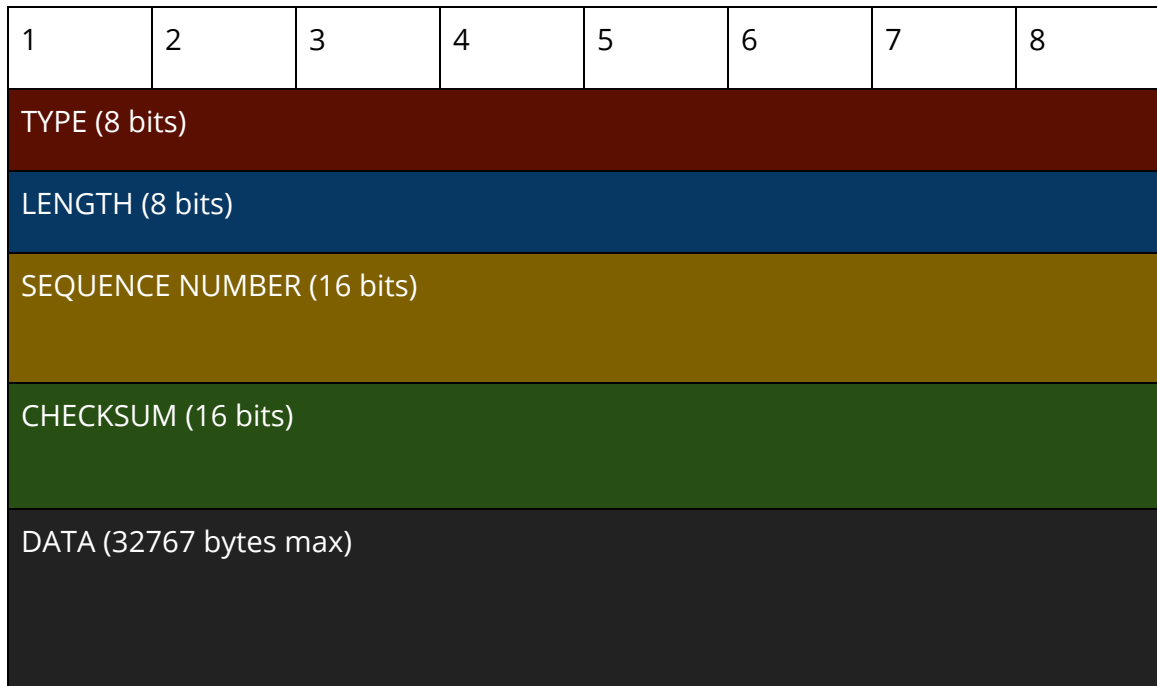


Pada protokol ini, ketika *sender* mengirimkan suatu *packet*, *receiver* akan mengirimkan respon berupa *packet* ACK yang menandakan bahwa *packet* tersebut diterima oleh *receiver*. Jika *sender* tidak menerima *packet* ACK dari *packet* data yang dikirimnya, maka *sender* akan mengirim kembali data tersebut sampai menerima *packet* ACK.

Untuk dapat mengimplementasikan protokol tersebut, anda perlu mendesain struktur *packet* yang dapat mengakomodasikan proses pengiriman data

sesuai dengan cara kerja protokol tersebut. Berikut ini adalah diagram sederhana dari *packet* yang dapat anda gunakan.

Diagram *packet*:



- **Type**

Bagian ini menandakan tipe dari *packet* yang dikirim. Nilai yang mungkin adalah 0x0 (DATA), 0x1 (ACK), 0x2 (FIN), dan 0x3 (FIN-ACK). Selain 0x0 dan 0x2, tipe lain tidak berisi data.

- **Sequence Number**

Bagian ini menandakan urutan *packet* pada untuk *file* yang sedang dikirim.

- **Length**

Bagian ini berisi panjang pada data yang dikirim pada *packet* tersebut (maksimal 32767). *Length* bertipe *unsigned integer*.

- **Checksum**

Checksum merupakan hasil XOR dari seluruh paket (kecuali bagian *checksum* itu sendiri) yang dibagi-bagi menjadi 16 bits (2 byte).

- **Data**

Data berisi keseluruhan atau potongan data yang sedang dikirim. *Packet* yang berisi potongan data terakhir akan dikirim dengan tipe *packet* 0x2 (FIN), yang menandakan *packet* tersebut adalah *packet* terakhir untuk *file* yang sedang dikirim. Setelah memastikan bahwa semua *packet* untuk *file* tersebut telah diterima, *receiver* mengirimkan *packet* FIN-ACK yang menandakan *receiver* telah berhasil mendapatkan seluruh *paket* yang dikirim. Untuk *packet* yang bukan merupakan paket terakhir, tipe yang digunakan adalah 0x0 (DATA) dan *receiver* akan membalas dengan 0x1 (ACK). Apabila ACK atau FIN-ACK tidak diterima oleh *sender*, *sender* akan mengirimkan kembali *packet* tersebut.

Penggunaan :

1. *Receiver* dijalankan dengan memberikan argumen berupa *port* yang akan di-*bind*.
2. *Sender* dijalankan dengan memberikan argumen berupa *path* dari file yang akan dikirim dan beberapa *address* dari *receiver* yang akan menerima *file* tersebut.
3. *Sender* akan mengirimkan *file* tersebut kepada semua *receiver* secara bersamaan. Tampilkan proses pengiriman *file* tersebut.
4. Pengiriman *file* bersifat konkuren, spesifikasi dasar tugas tidak mewajibkan pengiriman paralel. Optimisasi dapat dilakukan dengan mengerjakan bonus.

Requirements:

1. *Sender* dapat mengirimkan *file* ke beberapa *receiver*.
2. *Receiver* dapat menerima dan menyimpan *file*. *File* disimpan dengan nama ``downloaded`` pada folder ``.iout``.
3. *File* yang diterima memiliki *checksum* (secara keseluruhan) yang sama dengan *file* yang dikirim (tidak ada *packet* 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. Pastikan kode anda dapat berjalan di sistem operasi Ubuntu 20.04.
6. Buatlah dua buah *script shell* yaitu `run_receiver.sh` yang menjalankan *receiver* pada port tertentu dan `run_sender.sh` yang menjalankan *sender* untuk mengirimkan *file* yang menerima nama *file* yang akan dikirim termasuk direktori.
7. Dilarang menggunakan *library* yang bukan bawaan Python 3.
8. Untuk pengerjaan bonus, anda diwajibkan membuat video yang menunjukkan program anda menjalankan masing-masing bonus yang dikerjakan. Tunjukkan perbedaan antara penggunaan fitur bonus dan tanpa penggunaan fitur bonus. Jelaskan apa yang sedang dilakukan di video dengan suara anda.

Bonus:

1. Optimasi penggunaan memori pada *sender* dan *receiver* untuk mendukung pengiriman *file* berukuran besar (beberapa contoh permasalahan terkait yaitu penyimpanan *packet* pada memori akan sangat memakan RAM ketika *file* yang dikirimkan berukuran besar, penulisan tiap *packet* ke *storage* akan memakan waktu).
2. Melakukan *multithreading* untuk meningkatkan kecepatan pengiriman dan penerimaan *packet* (berikan opsi untuk mengaktifkan fitur ini atau tidak).
3. Menambahkan mekanisme protokol untuk mendukung pengiriman *metadata* dari *file* yang dikirimkan, paling tidak nama *file* dan ekstensi (boleh menambahkan tipe *packet* sendiri. Berikan opsi untuk mengaktifkan fitur ini atau tidak).

Tambahan:

1. Pengisian daftar kelompok dapat dilakukan melalui link ini
https://docs.google.com/spreadsheets/d/1KPOtnDHXfTFJ_8_alvINsboh76fWNFm1EAgN-MbPHP8/edit#gid=1960538947
2. Untuk pertanyaan, dapat disampaikan melalui milis maupun ditulis pada daftar pertanyaan melalui link ini
<https://docs.google.com/spreadsheets/d/19uCmsvfVRzwyxLaiMiK-CVINuRea0HnIAPUowl-2g/edit#gid=0>
3. Akan dilakukan *pretest* untuk tugas ini. Berikut hal yang perlu diperhatikan terkait *pretest*:
 - a. *Pretest* dilakukan menggunakan *platform* <https://quizizz.com/> pada tanggal 29 September 2020, pukul 09.55-10.00.
 - b. Perhatikan bahwa *pretest* hanya akan dibuka selama 5 menit.
 - c. Jika berhalangan hadir atau terdapat kemungkinan kendala terkait *pretest*, harap melaporkan ke asisten sebelum *pretest* dimulai (29 September 2020 pukul 09.55).
 - d. Pastikan tidak ada kendala internet saat *pretest* dilakukan karena laporan kendala setelah *pretest* dimulai tidak akan diterima. Jika Anda tidak bisa mengusahakan internet yang lancar pada saat *pretest*, silahkan melapor pada asisten sesuai dengan poin c.
 - e. Pastikan kalian mengerjakan *pretest* dengan menggunakan format nama: NIM Nama, misal 13517070 Aidil Rezki. Kesalahan format nama dapat menyebabkan *pretest* tidak dinilai.
 - f. Link pengerjaan *pretest* akan dikirim melalui milis paling lambat sebelum *pretest* dimulai (29 September 2020 pukul 09.55).
 - g. *Pretest* dikerjakan secara individu.
 - h. Segala tidak kecurangan selama *pretest* akan ditindaklanjuti dan dilaporkan ke dosen terkait.

III. Pengumpulan dan Deliverables

1. Tiap kelompok diwajibkan untuk *join* ke Github Classroom yang akan diumumkan nanti melalui milis.
2. Ketika akan *join*, akan ditanyakan anggota kelompok. Silahkan isi anggota kelompok sesuai kelompok anda.
3. Setelah berhasil *join*, Github akan membuatkan repo untuk kelompok anda. Silahkan kerjakan tugas ini pada repositori tersebut.
4. Untuk mencoba menjalankan *test case*, gunakan *tag* `v0.0.X` dengan X adalah *integer* bebas. Penggunaan tag dan release dapat dibaca pada referensi [berikut](#).
5. Untuk men-*submit* tugas kelompok anda, gunakan *tag* `v1.0.0`. Pastikan program anda sudah lulus *test case* terlebih dahulu.
6. Dilarang memodifikasi folder `.github`.
7. Dilarang melakukan spam untuk menjalankan *test case*. Silahkan jalankan *test case* pada *classroom* jika anda yakin terdapat perbaikan dari versi sebelumnya. Anda disarankan mencoba menjalankan *test case* pada *local machine* sebelum menjalankannya di *classroom*.