

Tugas Besar 2
IF3130 Jaringan Komputer
Sliding Window Protocol



Dipersiapkan oleh

Kelompok Haha

13514029 - Muhammad Farhan Majid

13514047 - Bervianto Leo P

13514095 - Muhammad Az-zahid Adhitya Silparensi

PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG

2016

DAFTAR ISI

Pembahasan Pertanyaan	3
Bagaimana Sliding Windows Protocol bekerja dalam program Anda?	3
Mengapa windows size pada Selective-Repeat ARQ harus lebih kecil dari ukuran frame buffer?	3
Petunjuk Kompilasi Program	3
Petunjuk Penggunaan Program	5
Pembagian Kerja dalam Kelompok	5

1. Pembahasan Pertanyaan

1.1. Bagaimana Sliding Windows Protocol bekerja dalam program Anda?

Aplikasi kami menggunakan mekanisme flow control yang sama dengan Tugas Besar 1. Mekanisme Sliding Window Protocol dimulai dengan transmitter yang melakukan request ke receiver untuk mendapatkan window size. Setelah itu, transmitter mengirimkan frame sebanyak window size yang telah diperoleh. Jika tidak ada masalah pada frame, receiver akan mengirimkan ACK kepada transmitter. Jika terdapat masalah pada frame, receiver akan mengirimkan NAK kepada transmitter. Saat transmitter menerima ACK, transmitter akan melanjutkan pengiriman frame selanjutnya mengikuti window yang ada. Sedangkan, saat transmitter menerima NAK atau terjadi timeout, transmitter akan mengirim ulang frame tersebut.

1.2. Mengapa windows size pada Selective-Repeat ARQ harus lebih kecil dari ukuran frame buffer?

Window size adalah jumlah frame maksimum yang dapat ditransmisikan untuk sekali pengiriman, sedangkan frame buffer adalah alokasi buffer space untuk frame yang dapat diterima tanpa menunggu acknowledgment. Windows size harus lebih kecil dari ukuran frame buffer agar tidak ada paket yang hilang karena buffer space yang penuh. Karena window size lebih kecil, maka jumlah frame yang dikirim tidak akan pernah melebihi buffer space yang ada dan paket tidak akan hilang.

2. Petunjuk Kompilasi Program

Melakukan kompilasi dapat menggunakan *Makefile* untuk mempermudah. Adapun berikut ini detail melakukan kompilasi :

a. Kompilasi *Transmitter*

i. Menggunakan *Makefile*

Dalam menggunakan *Makefile* sangat mudah, lakukan perintah berikut untuk kompilasi :

`$ make transmitter`

- Catatan : Pastikan Anda pada posisi *root* dari program atau pada posisi direktori yang memiliki *Makefile*. Jangan pindahkan *Makefile*, karena akan memberikan kegagalan kompilasi. Program hasil kompilasi akan ada pada direktori *bin*.

ii. Manual

Secara manual dapat Anda lakukan dengan memasukan perintah ini :

`$ g++ transmitter.cpp -o transmitter`

- Catatan : Lakukan perintah ini pada posisi yang terdapat *transmitter*. Hasil program akan ada pada direktori yang sama dengan kode sumber.

b. Kompilasi *Receiver*

i. Menggunakan *Makefile*

Dalam menggunakan *Makefile* sangat mudah, lakukan perintah berikut untuk kompilasi :

`$ make receiver`

- Catatan : Pastikan Anda pada posisi *root* dari program atau pada posisi direktori yang memiliki *Makefile*. Jangan pindahkan *Makefile*, karena akan memberikan kegagalan kompilasi. Program hasil kompilasi akan ada pada direktori *bin*.

ii. Manual

Secara manual dapat Anda lakukan dengan memasukan perintah ini :

`$ g++ receiver.cpp -o receiver`

- Catatan : Lakukan perintah ini pada posisi yang terdapat *receiver*. Hasil program akan ada pada direktori yang sama dengan kode sumber.

c. Kompilasi Semua (*Transmitter* dan *Receiver*)

i. Menggunakan *Makefile*

Dalam menggunakan *Makefile* sangat mudah, lakukan perintah berikut untuk kompilasi :

`$ make all`

- Catatan : Pastikan Anda pada posisi *root* dari program atau pada posisi direktori yang memiliki *Makefile*. Jangan pindahkan *Makefile*, karena akan memberikan kegagalan kompilasi. Program hasil kompilasi akan ada pada direktori *bin*.

ii. Manual

Secara manual dapat Anda lakukan dengan memasukan perintah ini :

`$ g++ receiver.cpp -o receiver`

`$ g++ transmitter.cpp -o transmitter`

- Catatan : Lakukan perintah ini pada posisi yang terdapat *transmitter* dan *receiver*. Hasil program akan ada pada direktori yang sama dengan kode sumber.

3. Petunjuk Penggunaan Program

Anda dapat menggunakan program ini dengan mudah, adapun berikut untuk menjalankan program dan parameter yang harus diberikan :

a. Menjalankan *Receiver*

Pada sistem operasi Linux Anda dapat melakukan perintah berikut ini :

```
$ ./receiver [PORT]
```

Contoh :

```
$ ./receiver 20000
```

Catatan :

- *PORT* yang dimasukan ini merupakan *port* pada komputer penerima yang akan bersedia menerima koneksi.

b. Menjalankan *Transmitter*

Pada sistem operasi Linux Anda dapat melakukan perintah berikut ini :

```
$ ./transmitter [IP Tujuan atau Hostname] [PORT] [Lokasi Berkas]
```

Contoh :

```
$ ./transmitter 192.168.0.7 20000 test.txt
```

Catatan :

- *IP Tujuan* yang dimasukan ini merupakan alamat IP atau hostname yang akan dikirim sebuah berkas.
- *PORT* yang dimasukan ini merupakan *port* pada komputer tujuan yang akan bersedia menerima koneksi.
- *Lokasi Berkas* yang dimasukan ini merupakan alamat berkas yang ingin dikirim dan nama berkas yang akan dikirim.

c. Prosedur Penggunaan

Program yang dibuat saling berkaitan, oleh karena itu agar program berjalan sesuai dengan yang diharapkan adapun prosedur yang harus dilakukan sebagai berikut :

1. Jalankan *Receiver* (Lihat cara menjalankan *Receiver*)
2. Jalankan *Transmitter* (Lihat cara menjalankan *Transmitter*)

Receiver ini harus dipersiapkan untuk menerima koneksi agar *Transmitter* dapat mengirim informasi berkas pada komputer tujuan tersebut yang sudah membuka koneksi.

4. Pembagian Kerja dalam Kelompok

Adapun berikut ini pembagian kerja dalam pembuatan program ini :

- 13514029 - Muhammad Farhan Majid :
Membuat *Receiver.cpp* dan mengisi laporan.

- 13514047 - Bervianto Leo P
Membuat *Receiver.cpp* dan mengisi laporan.
- 13514095 - Muhammad Az-zahid Adhitya Silparensi
Membuat *Transmitter.cpp*, *tr_error_test.cpp*, dan mengisi laporan.

Referensi

1. <https://github.com/masphei/flow-control-udp/blob/master/Control.c>
Bagian kode yang digunakan yaitu pada pengisian prosedur *rcvchar* dan *q_get* serta cara mengimplementasi thread pada Receiver dengan sedikit perubahan.
2. http://www.diffen.com/difference/TCP_vs_UDP - Perbedaan TCP dan UDP
3. <http://stackoverflow.com/questions/5330277/what-are-examples-of-tcp-and-udp-in-real-life> diakses pada Senin, 7 November 2016