

SLIDING WINDOW PROTOCOL

TUGAS BESAR I

IF3130 Jaringan Komputer

Disusun oleh:

Muhammad Nizami 13512501

Ivan Andrianto 13513039

Lucky Cahyadi 13513061



SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

PROGRAM STUDI TEKNIK INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

2015

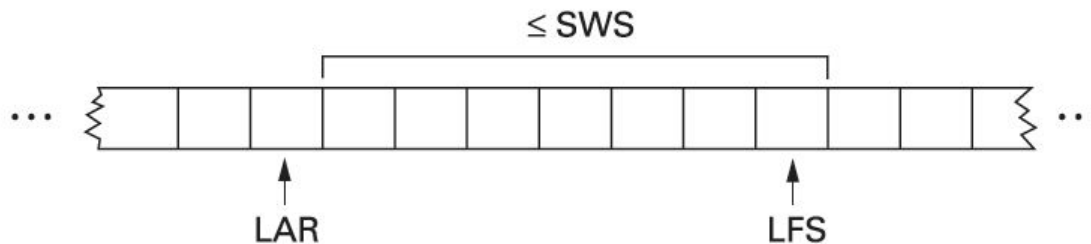
Pembahasan pertanyaan-pertanyaan

1. Bagaimana Sliding Window Protocol bekerja dalam program anda?

Pada Sender dan Receiver terdapat circular buffer dan window yang bergeser.

Sender pertama-tama meng-assign nomor pada setiap frame. Sender kemudian menyimpan 3 macam variabel, window yang bergeser, SWS (Send Window Size), yang menyimpan batas frame unacknowledged yang bisa dikirim sender, LAR (Last Acknowledgement Received) yang menyatakan nomor acknowledgement terakhir yang diterima. dan LFS (Last Frame Sent) yang menyimpan nomor frame terakhir yang dikirim. Sender selalu mengikuti aturan:

$$LFS - LAR \leq SWS$$

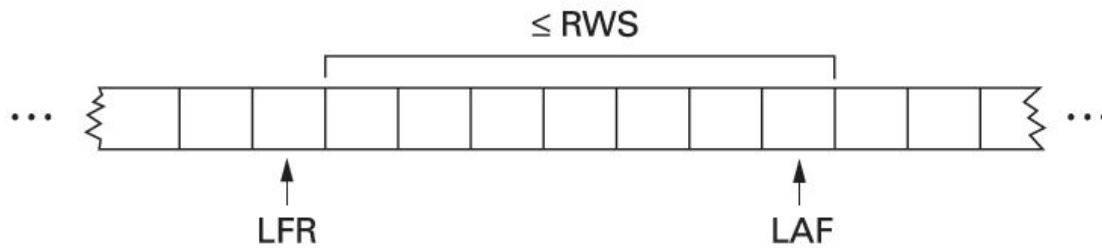


Gambar 1. Sliding Windows pada Sender

Ketika mendapat ACK, LAR akan bergeser ke kanan sehingga sender bisa mengirim lagi frame yang lain. Sender juga memiliki timer yang akan mengirim frame kembali jika ACK tidak didapat dalam waktu tertentu

Sender juga pertama-tama meng-assign nomor pada setiap frame. Receiver kemudian menyimpan 3 variabel yaitu RWS (Receive Window Size) menyatakan batas atas frame yang akan diterima oleh receiver, LFR (Last Frame Received) menyatakan frame terakhir yang diterima, LAF (Largest Acceptable Frame) menyatakan frame terbesar yang akan diterima. Receiver juga mengikuti aturan:

$$LAS - LFR \leq RWS$$



Gambar 2. Sliding Windows pada Receiver

Ketika frame dengan nomor SeqNum diterima, receiver akan mengecek jika $\text{SeqNum} \leq \text{LFR}$ atau $\text{SeqNum} \geq \text{LAF}$ maka frame yang diterima akan dibuang. jika $\text{LFR} \leq \text{SeqNum} \leq \text{LAF}$ maka frame akan diterima karena berada di dalam window receiver. Untuk mengirim ACK, digunakan sebuah variable SeqNumToAck yang menyatakan nomor terbesar yang belum di ack, sehingga nomor sebelum sampai SeqNumToAck sudah diterima. Receiver kemudian mengirim Ack dari SeqNumtoAck.

Pada tugas ini error control yang digunakan adalah Selective Repeat ARQ. Dalam Selective Repeat ARQ, sender bisa mengirim beberapa frame sekaligus tanpa harus menunggu ACK terlebih dahulu secara satu per satu. Receiver juga bisa menolak hanya 1 frame sehingga hanya frame tersebut yang harus dikirim kembali. Ukuran window sender awalnya 0 dan bisa berubah sampai ukuran maksimum yang ditentukan, sedangkan ukuran window receiver fix sebesar ukuran maksimum yang ditentukan. Receiver memiliki buffer untuk setiap nomor frame. Sender akan mengirim frame yang ada di dalam windows. Kemudian Receiver akan mengecek apakah frame tersebut berada di dalam windows atau tidak, jika iya maka receiver akan mencocokkan nomor frame tersebut dengan buffer dan mengirimkan ACK. Jika sender tidak menerima ACK dalam selang waktu yang sudah ditentukan, bisa saja karena data tidak diterima atau ACK menghilang saat dikirim ke sender, maka sender akan mengirim kembali frame yang tidak menerima ACK tersebut saja. Window pada sender akan bergeser ketika frame pada windows yang paling kiri sudah menerima ACK, sementara window pada receiver akan bergeser ketika window yang paling kiri sudah menerima frame yang dikirimkan receiver.

Implementasinya adalah sebagai berikut:

Sender

Terdapat tiga thread pada sender:

1. Thread utama mengkonstruksi frame dari input stream dan memasukkannya ke dalam buffer

2. Thread sending mengirimkan frame yang (a) belum dikirim (b) sudah dikirim namun NAK (c) sudah dikirim namun timeout. Thread ini juga memeriksa apakah bagian awal window sudah ack, dan bila sudah, window digeser
3. Thread listenack menerima ACK/NAK dan menandai pada buffer. Apabila ACK, ditandai ACK, dan apabila NAK, ditandai belum dikirim.

Receiver

Terdapat dua thread pada receiver:

1. Thread utama mengambil frame dari buffer, mengkonstruksi string dan mengeluarkannya kepada output stream
2. Thread receiving menerima frame dan memvalidasi. Bila tidak valid, buang, dan bila mungkin, kirim NAK. Bila valid dan dalam window, dimasukkan ke dalam buffer. Setelah dimasukkan dalam window, geser window bila mungkin. Bila setelah memasukkan ke dalam window, ternyata buffer penuh dan window tidak bisa digeser, tunda pengiriman ACK. Bila berhasil geser, kirim ACK.

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

Window Size sebenarnya mengikuti aturan :

$$\text{WIN_SIZE} \leq (\text{BUF_MAX} + 1) / 2$$

Alasannya bisa dilihat dari contoh berikut. Misal framenya memiliki nomor 0 1 2 3 dan window sizenya 3 yang jika mengikuti aturan window size melebihi 1 dari seharusnya , pada pengiriman awal receiver menerima 0 1 2 dengan benar, kemudian window bergeser untuk bersiap-siap menerima 3 0 1. Buffer kembali dikosongkan dan siap menerima frame baru yang akan dikirimkan. Tetapi jika ACK tidak diterima oleh sender, misal ACK untuk 0 yang pertama, maka sender akan mentransmisikan kembali 0 yang pertama tetapi diinterpret oleh receiver sebagai 0 yang baru sehingga membuat kacau urutan pengiriman frame yang sebenarnya

Petunjuk kompilasi program

`make`

Petunjuk penggunaan program

pada bin, terdapat berkas receiver dan sender

`receiver PORT < fileinput`

`sender IP PORT > fileoutput`

Pembagian kerja dalam kelompok

Muhammad Nizami: System Designer, Programmer, Tester

Lucky Cahyadi: System Analyst

Ivan Andrianto: Programmer, Tester

Pranala sumber referensi yang digunakan dalam pengerjaan tugas, dan bagian kode yang digunakan dari referensi tersebut bila ada

https://en.wikipedia.org/wiki/Cyclic_redundancy_check untuk (CRC-32Q)

www.ccs-labs.org/teaching/rn/animations/gbn_sr/

<http://en.cppreference.com/>

<http://www.cplusplus.com/reference/>

<http://www.yolinux.com/TUTORIALS/LinuxTutorialPosixThreads.html>

<http://man7.org/linux/man-pages/man7/pthreads.7.html>

http://www.microhowto.info/howto/send_a_udp_datagram_in_c.html

<https://www.abc.se/~m6695/udp.html>

Peterson, Larry L. and Bruce S. Davie. 2007. Computer Networks: A Systems Approach, Fourth Edition,. Morgan Kaufmann.

Tanenbaum, Andrew S. and David J. Wetherall. 2011. Computer Network, 5th ed. Massachusetts: Pearson.