

Tugas Besar 01

IF3130 Jaringan Komputer

Sliding Window

Dipersiapkan oleh:
Asisten Laboratorium Sistem Terdistribusi

Didukung oleh:



Start Date: 27 September 2017

End Date: 18 Oktober 2017

I. Background

Bunga Citra Lestari - Cinta Pertama (Sunny)

Sunny, sunny,
Jantungku berdetak tiap ku ingat padamu
Sunny, sunny,
Mengapa ada yang kurang saat kau tak ada
Sunny, sunny,
Melihatmu menyentuhmu itu yang kumau

Kau tak sempat tanyakan aku
Cintakah aku padamu..

Chorus:
Tiap kali,
Aku berlutut
Aku berdoa
Suatu saat
Kau bisa cinta padaku
Tiap kali
Aku memanggil
Didalam hati
Mana sunny,
Mana sunny ku,
Mana sunny ku,

Sunny, sunny,
Apakabarmu kabarku baik baik saja
Sunny, sunny,
Begitu banyak cerita tak habis tentangmu
Sunny, sunny,
Salamku untukmu dari hati yang terdalam

Kau tak sempat tanyakan aku
Cintakah aku padamu..

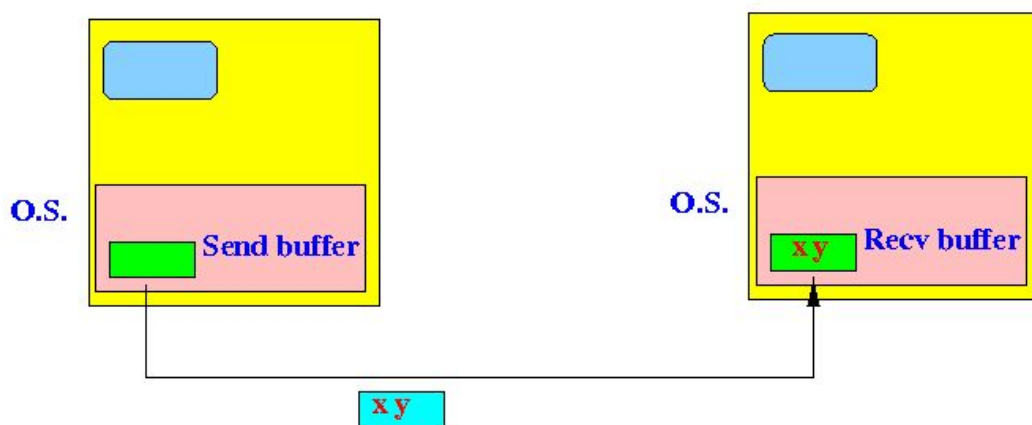
Chorus 2x

II. Deskripsi Tugas

Ketika *system call* **send()**, **sendto()**, atau **sendmsg()** dipanggil untuk mengirimkan data melalui network socket, data tidak langsung dikirim melalui network. Data akan dipindahkan dari aplikasi menuju *send buffer* yang disediakan oleh sistem operasi. Sistem operasi kemudian akan mengirimkan data yang sudah terdapat pada *send buffer* tersebut melalui network interface. Pada protokol TCP, data akan dikirim dalam rangkaian *TCP segment*.

Saat data sampai pada *receiver*, *receiver* akan mengirimkan ACK sebagai konfirmasi bahwa data sudah berhasil diterima. Penerimaan ACK akan membuat data yang berada pada *send buffer* dihapus oleh sistem operasi. Apabila data yang akan dikirimkan oleh **send()**, **sendto()** atau **sendmsg()** tidak dapat dimasukkan seluruhnya pada *send buffer*, aplikasi akan melakukan penyalinan data secara bertahap dan fungsi akan melakukan blocking hingga *send buffer* dikosongkan kembali oleh sistem operasi dan data berikutnya dapat dipindahkan.

Hal yang sama juga terjadi pada penerima pesan. Ketika sistem operasi menerima data dari *network interface*, sistem operasi akan mengecek terlebih dahulu apakah data yang diterima rusak atau tidak. Pengecekan dilakukan dengan menggunakan data checksum yang terdapat pada data yang diterima. Jika proses pengecekan berhasil, maka data disimpan dalam *recv buffer* dan sistem operasi akan mengundang aplikasi untuk melakukan pemanggilan **recv()**, **recvfrom()**, atau **recvmsg()** untuk membaca data yang terdapat pada *recv buffer*. Ketika aplikasi sudah membaca data tersebut, maka sistem operasi akan mengosongkan *recv buffer* dan siap untuk menerima pesan berikutnya dari pengirim pesan.



Gambar 1. Ilustrasi penggunaan TCP Buffer pada aplikasi

Untuk memastikan agar tidak terjadi kehilangan data antara *send buffer* dan *recv buffer*, perlu adanya error detection serta flow control. Pada TCP, digunakan algoritma sliding window untuk mengatasi hal ini. Penerima dan pengirim pesan pada algoritma sliding window memiliki sebuah 'window' pada *send/recv buffer* miliknya. Adanya window tersebut menandakan data yang dapat dikirim/diterima secara berurutan tanpa menunggu pengiriman ACK.

Pada algoritma sliding window, pengirim pesan dapat mengirimkan lebih dari satu segment sekaligus. Setiap segment yang dikirim berisi data yang ingin dikirim, sequence number dari

data yang dikirim, serta checksum dari segment tersebut. Sequence number digunakan agar penerima pesan mengetahui nomor urut dari pesan yang diterima. Checksum akan digunakan oleh receiver untuk mengecek apakah data yang diterima mengalami kerusakan.

Penerima akan mengirim ACK ketika berhasil memasukkan data yang ada ke dalam TCP buffer. Data ACK yang dikirim akan berisi sequence number dari data berikutnya yang diharapkan serta ukuran window saat ini. Ketika pengirim mendapatkan ACK, pengirim akan menggeser kepala windows ke data yang memiliki sequence number sesuai pada ACK kemudian mengirimkan. Jika penerima belum mendapat ACK hingga jangka waktu tertentu, ia akan mengirimkan kembali seluruh data yang berada pada window yang ia miliki.

Pada penerima pesan, window akan bergeser ketika ACK untuk data tersebut sudah dikirimkan. Window pada penerima mungkin mengecil jika recv buffer hampir penuh dan tidak memungkinkan untuk menerima data dari pengirim pesan. Ukuran window akan membesar kembali ketika recv buffer sudah dapat menerima sekelompok data yang lebih besar.

Penjelasan lebih lengkap dari proses yang sudah dijelaskan dapat anda lihat pada:

<http://www.mathcs.emory.edu/~cheung/Courses/455/Syllabus/7-transport/flow-control.html>

Sementara itu, simulasi dari algoritma sliding window dapat anda lihat pada: *(require flash)*

http://www2.rad.com/networks/2004/sliding_window/

Pada tugas besar ini, anda akan melakukan simulasi mengenai proses pengiriman data yang terjadi dari sebuah program ke program lainnya dengan mengikuti proses yang sudah dijelaskan. Dalam tugas besar ini, anda dapat menganalogikan:

- **network interface** sebagai pemanggilan fungsi `sendto()` dan `recvfrom()` pada sistem operasi.
- **send/recv buffer** sebagai array of byte pada aplikasi yang anda buat.
- **Systemcall send()/recv()** sebagai fungsi `send_data()` dan `recv_data()` yang akan anda implementasikan.

III. Spesifikasi Tugas

Dalam pengerjaan tugas besar ini, anda hanya diperbolehkan menggunakan bahasa pemrograman **C/C++** tanpa menggunakan library tambahan selain yang sudah disediakan oleh bahasa tersebut. Library tambahan adalah library yang penggunaannya memerlukan instalasi tambahan selain instalasi compiler. Anda hanya diperbolehkan untuk menggunakan **Socket UDP** untuk mengimplementasi tugas besar ini.

Anda akan membuat dua program berbeda, salah satu program akan membaca file pada filesystem kemudian mengirimkannya sesuai dengan proses yang sudah dijelaskan pada bab sebelumnya. Program lainnya akan menerima data file yang dikirim kemudian menuliskan data tersebut ke filesystem.

Pada tugas besar ini, ukuran dari send/recv buffer untuk masing-masing program adalah 256 byte. Ukuran window awal pada program merupakan salah satu parameter yang diberikan oleh user ketika menjalankan program. Untuk mempermudah, ukuran data yang dikirim pada sebuah segment dibatasi hanya 1 byte dan konstan selama program berjalan. Penggunaan algoritma checksum dibebaskan selama algoritma tersebut dapat menghasilkan 8 bit checksum.

Berikut merupakan format segment yang dikirim:

SOH (0x1)	Sequence number	STX (0x2)	Data	ETX (0x3)	checksum
1 byte	4 byte	1 byte	1 byte	1 byte	1 byte

Berikut adalah format dari packet ACK:

ACK (0x6)	Next Sequence Number	Advertised window size	checksum
1 byte	4 byte	1 byte	1 byte

Untuk mensimulasikan perubahan ukuran window, anda dapat menggunakan fungsi **sleep()** dengan nilai random ketika melakukan penerima memindahkan data buffer ke aplikasi sehingga pembacaan data tidak dalam kecepatan yang konstan.

Agar mempermudah penilaian yang dilakukan asisten, gunakan format eksekusi program sebagai berikut:

```
./sendfile <filename> <windowsize> <buffersize> <destination_ip>
<destination_port>
./recvfile <filename> <windowsize> <buffersize> <port>
```

IV. Spesifikasi Bonus

Program yang anda buat dapat menghasilkan sebuah log file yang menggambarkan operasi-operasi yang terjadi pada TCP buffer. Beberapa contoh operasi yang dapat dicatat dalam log adalah: penggeseran window, penerimaan data, pengiriman data, pengiriman ACK, penerimaan ACK, pembacaan buffer oleh aplikasi, serta penulisan buffer oleh aplikasi.

Sebagai contoh, berikut adalah log file yang dihasilkan oleh apache:

https://ossec-docs.readthedocs.io/en/latest/log_samples/apache/apache.html

V. Pengumpulan dan Deliverables

Berkas hasil tugas besar dikumpulkan dalam sebuah **zip** file bernama **TB1_KXX_YY_ABC.zip** dengan XX menandakan nomor kelas, YY menandakan nomor kelompok, dan ABC adalah nama kelompok anda. File zip tersebut terdiri dari:

- Direktori **src** yang berisi source code yang anda buat.

- Direktori **data** yang berisi data yang digunakan untuk testing program.
- **Makefile**. Buatlah sehingga kompilasi program dapat dilakukan hanya dengan pemanggilan command 'make' saja.
- File **README.md** yang berisi:
 - Petunjuk penggunaan program.
 - Cara kerja sliding window dalam program anda. Jelaskan juga fungsi yang terkait dengan sliding window pada program anda dan apa yang dilakukan oleh fungsi tersebut.
 - Pembagian tugas. Sampaikan dalam list pengerjaan untuk setiap mahasiswa. Sebagai contoh: XXXX mengerjakan fungsi YYYY, ZZZZ, dan YYZZ.
 - Jawaban dari pertanyaan berikut:
 1. Apa yang terjadi jika advertised window yang dikirim bernilai 0? Apa cara untuk menangani hal tersebut?
 2. Sebutkan field data yang terdapat TCP Header serta ukurannya, ilustrasikan, dan jelaskan kegunaan dari masing-masing field data tersebut!

Deadline dari pengumpulan milestone ini adalah tanggal 18 Oktober 2017 pukul 23:34:45 waktu server. Teknis pengumpulan akan diberitahukan maksimal 48 jam sebelum deadline pengumpulan. **Keterlambatan pengumpulan akan menyebabkan pengurangan nilai.**

VI. Referensi

https://en.wikipedia.org/wiki/Sliding_window_protocol

http://www2.rad.com/networks/2004/sliding_window

<http://www.mathcs.emory.edu/~cheung/Courses/455/Syllabus/7-transport/flow-control.html>