
USULAN TUGAS AKHIR

1. IDENTITAS PENYUSUL

Nama : Hermawan Winata
NRP : 5109100120
Dosen Wali : Diana Purwitasari, S.Kom, M.Sc

2. JUDUL TUGAS AKHIR

Perancangan dan Implementasi Pengaturan Source Rate untuk Mengatasi Kongesti pada Transmisi Data Radar.

3. LATAR BELAKANG

Radar merupakan sistem pendeteksi objek yang menggunakan gelombang radio untuk mengetahui arah, ketinggian, dan atau kecepatan objek tersebut. Radar bekerja dengan cara memancarkan gelombang radio dan menerima pantulan balik setelah gelombang tersebut memantul dari permukaan sebuah objek ^[1].

Sebuah radar memiliki cakupan daerah pendeteksian yang terbatas. Untuk mendapatkan gambaran yang menyeluruh akan suatu wilayah yang cukup luas, misalnya negara, dibutuhkan integrasi data antara beberapa situs radar. Komunikasi data dalam skala nasional ini akan secara signifikan meningkatkan lalu lintas data ^[2] terutama ketika jumlah objek yang terdeteksi oleh radar sangat banyak.

Besarnya aliran data dan keterbatasan infrastruktur akan menyebabkan kongesti. Data hasil tangkapan radar ketika dikirimkan ke situs radar lain melalui sambungan VPN yang memiliki *bandwidth* rendah akan menurunkan *real-time-ness* dari sistem. Kongesti akan semakin parah ketika digunakan protokol TCP yang secara otomatis menangani pengiriman ulang paket data yang rusak atau hilang. Ketika terjadi pengiriman ulang terdapat kemungkinan data sampai di penerima dalam keadaan basi.

4. TUJUAN PEMBUATAN TUGAS AKHIR

Tujuan dari pembuatan tugas akhir ini adalah untuk mengatasi kongesti dengan pengaturan *source rate* dan menjaga relevansi informasi yang diterima di ujung penerima.

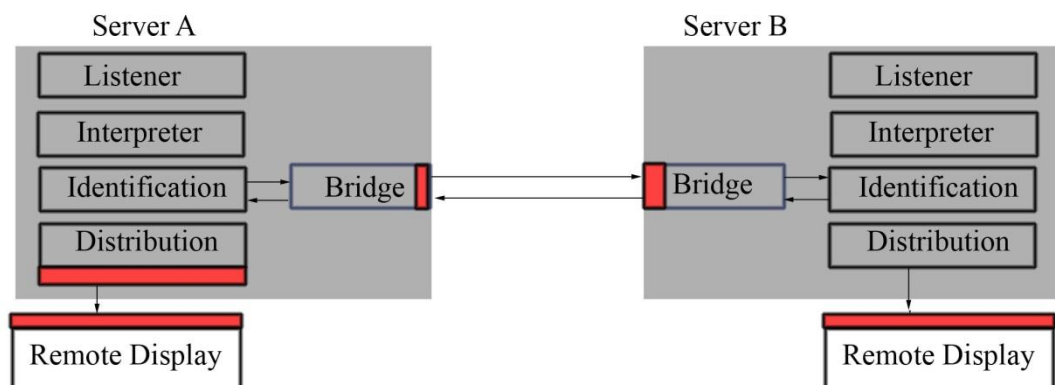
5. RUMUSAN MASALAH

Permasalahan yang diangkat dalam menyelesaikan tugas akhir ini dapat dituliskan sebagai berikut :

1. Pada saat pengiriman, paket dipilih berdasarkan relevansi informasi. Bagaimana mengkuantisasi relevansi informasi ?
2. Bagaimana menyusun antrian paket yang akan dikirim yang memadukan aspek relevansi informasi dengan penanganan Automatic Repeat reQuest(ARQ)?
3. Bagaimana mengoptimalkan kinerja pengaturan *source rate* sehingga tidak justru menambah *latency*?
4. Bagaimana penataan arsitektur perangkat lunak tersebut ?
5. Bagaimana mengimplementasi sistem di atas UDP?

6. RINGKASAN ISI TUGAS AKHIR

Dalam tugas akhir ini, untuk mengatasi kongesti pada sebuah sistem yang bersifat real-time, penulis akan membuat modul pengaturan *source rate* data radar yang dikirimkan berdasarkan prioritas tertentu. Pengiriman berdasarkan prioritas tersebut dapat didapatkan dengan menggunakan *priority queue*. Letak modul ini secara umum dapat digambarkan sesuai Gambar 1.



Gambar 1. Blok diagram sistem radar

Modul yang akan dibuat akan berada di dalam blok Bridge. Selain digunakan untuk komunikasi antarserver, modul ini juga dapat dipakai di dalam proses distribusi untuk mengirimkan paket data ke *remote display*. Modul ini akan dapat mengatur urutan antrian data yang dikirim berdasarkan kriteria prioritas tertentu, misalnya pesawat militer akan memiliki prioritas lebih tinggi daripada pesawat komersial. Selain itu, modul juga akan mampu melakukan koreksi error dengan FEC serta meminta pengiriman kembali dengan protokol ARQ jika data hilang/tidak dapat diperbaiki.

7. BATASAN MASALAH

Sejumlah permasalahan yang dibahas di tugas akhir ini akan dibatasi ruang lingkup pembahasannya, antara lain :

1. Aplikasi mengatur mekanisme pengiriman paket data radar.
2. Aplikasi dibuat dengan menggunakan bahasa C++ dan .NET framework 4.0.

8. TINJAUAN PUSTAKA

Pada bagian ini, akan dijelaskan beberapa referensi terkait mengenai pembuatan tugas akhir ini.

1. ARQ

ARQ(Automatic Repeat reQuest / Automatic Repeat Query) merupakan sebuah metode pengendalian error dalam pengiriman data yang menggunakan paket ACK dan timeout untuk mendapatkan kehandalan pengiriman data meskipun melewati medium yang kurang handal. Beberapa protokol ARQ antara lain ^[3] :

a. Stop and wait ARQ

Merupakan variasi paling sederhana dari ARQ. Protokol ini mengharuskan pengirim mengirim hanya 1 *frame* dalam satu waktu. Pengiriman *frame* data yang berikutnya dapat berlangsung jika pengirim mendapatkan sinyal ACK. Jika penerima menerima *frame* yang rusak, maka sinyal ACK tidak akan dikirim. Pengirim akan terus mengirimkan *frame* yang sama ketika *timeout* selama sinyal ACK belum diterima.

b. Go back N ARQ

Protokol ini mengharuskan pengirim mengirimkan sejumlah N *frame* data tanpa berhenti untuk menunggu setiap sinyal ACK. Ketika

terdapat *frame* yang rusak / hilang, maka pengiriman ulang akan dilakukan dimulai dari *frame* terakhir yang tidak mendapat sinyal ACK hingga *frame* ke-N.

c. Selective repeat ARQ

Protokol ini mirip dengan Go back N hanya saja ukuran jendela penerima lebih besar dari 1. Pengirim terus mengirimkan sejumlah N frame data dan penerima akan menerima setiap frame yang berada dalam jangkauan N tersebut. Penerima hanya mengirimkan ACK untuk setiap *frame* yang benar. *Frame* akan dikirim kembali ketika tidak diterima sinyal ACK untuk *frame* tersebut, sementara go back N akan mengirimkan semua *frame* dimulai dari *frame* yang rusak/hilang tersebut.

2. FEC

FEC(Forward Error Correction) merupakan teknik yang digunakan untuk mendeteksi dan memperbaiki error yang terjadi saat pengiriman data dengan cara menambahkan data yang redundan. FEC dapat diterapkan dalam situasi ketika pengiriman ulang data sangat mahal dan hampir tidak dapat dilakukan.

3. Priority Queue

Priority Queue merupakan sebuah struktur data di mana setiap elemennya memiliki sebuah nilai prioritas tertentu^[4]. Di dalam *priority queue*, elemen yang bisa diakses hanyalah elemen pertama yaitu yang memiliki prioritas tertinggi. Fungsi-fungsi yang biasa dimiliki oleh struktur data ini adalah *pop* untuk mengambil elemen pertama, *push* untuk menyisipkan sebuah elemen, dan *peek* untuk melihat isi elemen pertama. *Priority queue* biasa diimplementasikan dengan menggunakan *heap*.

4. Packet Discarding

Merupakan salah satu cara menangani kongesti yaitu dengan membuang paket ketika tidak ada ruang yang cukup untuk menyimpannya. *Packet discarding* dapat digunakan dalam *connection-less* ataupun sambungan *connection-full*.

9. METODOLOGI

Pembuatan Tugas Akhir ini dilakukan dengan mengikuti metodologi sebagai berikut :

1. Menginventarisir kebutuhan transmisi data radar.

Pada tahap ini dilakukan pengumpulan informasi struktur data yang dikirimkan dan diperlukan oleh aplikasi radar.

2. Menentukan prioritas relevansi informasi.

Pada tahap ini ditentukan kriteria pembandingan data sehingga antara 2 data dapat diketahui manakah yang memiliki prioritas lebih tinggi.

3. Merancang lapisan transport.

Pada tahap ini dilakukan perancangan protokol baru di atas UDP dengan fitur ARQ.

4. Merancang arsitektur.

Pada tahap ini dilakukan perancangan arsitektur dari perangkat lunak yang akan dibuat.

5. Implementasi dalam C++.

Pada tahap ini dilakukan implementasi sistem menggunakan bahasa C++.

6. Uji coba kinerja dengan pengujian subyektif.

Pada tahap ini akan dilakukan pengujian dengan membagikan angket untuk menilai kinerja aplikasi yang telah dibuat. Kondisi pengujian yaitu dengan volume data besar seberapa besar *bandwidth* masih bisa dibatasi sebelum data di tampilan tidak lagi dianggap sah.

10. DAFTAR PUSTAKA

- [1] Larry Gilman. (2004) "Radar" in Encyclopedia of Espionage, Intelligence, and Security. [Online]. <http://www.encyclopedia.com/doc/1G2-3403300634.html>
- [2] Markus Rupp, *Video and Multimedia Transmissions Over Cellular Networks : Analysis, Modelling, and Optimization*. Chippenham: John Wiley & Sons Ltd, 2009.
- [3] Andrew S. Tanenbaum, *Computer Networks*. Upper Saddle River, NJ: Prentice Hall, 2002.
- [4] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, *Introduction to Algorithms*, 2nd ed. Cambridge, Massachusetts: The MIT Press, 2001.

11. JADWAL KEGIATAN

Tahapan	Waktu															
	Maret 2013				April 2013				Mei 2013				Juni 2013			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Inventarisir kebutuhan transmisi data radar																
Perancangan lapisan transport																
Merancang arsitektur																
Implementasi																
Uji coba kinerja																