

ANALISIS DAN SIMULASI PERFORMANSI KENDALI KONGESTI DENGAN SKEMA PGMCC

Rineta Inelia¹, Sofia Naning Hertiana², Ida Wahidah³

¹Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom

Abstrak

Perkembangan protokol saat ini sangat membantu dalam komunikasi jaringan khususnya jaringan komputer yang juga menyebabkan semakin banyaknya aplikasi-aplikasi baru yang dapat dinikmati dan semakin menaikkan trafik. Jaringan komputer, khususnya LAN diharapkan mampu memberikan layanan dengan kecepatan tinggi.

Kontrol kongesti pada jaringan internet dengan pertumbuhan trafik yang naik secara eksponensial menjadi salah satu usaha penting untuk memberikan layanan yang baik bagi para penggunanya. Kongesti menyebabkan paket hilang (packet loss) yang tinggi, throughput yang rendah dan delay yang tinggi. Oleh karena itu dikembangkan suatu skema kendali kongesti baru yang lebih beragam. Salah satunya adalah PGMCC (Pragmatic General Multicast Congestion Control) dengan pengiriman yang bersifat multicast.

Hanya dibandingkan PGMCC dan TCP dengan jumlah user 50, 100, 150, 200, 250 dan 300. Juga menambahkan background traffic dengan data rate sebesar 800 Kbps dan 1200 Kbps. Dari sisi packet loss, skema PGMCC ini memiliki kualitas yang baik, yaitu dengan mengurangi terjadinya packet loss. Pada kondisi jaringan yang padat maupun yang biasa, terjadi packet loss yang jumlahnya lebih kecil jika dibandingkan dengan TCP. Namun PGMCC memiliki kekurangan yaitu delay yang besar dan throughput yang rendah apabila dibandingkan dengan performansi TCP yang memiliki delay yang rendah dan throughput yang kecil nilainya.

Kata Kunci : PGMCC, Multicast, Kendali Kongesti

Abstract

The development of the current protocol is very helpful in communication networks, especially computer networks also involving to the increasing number of new applications that can be enjoyed and the increasing traffic. Computer networks, particularly the LAN are expected to provide services with high speed.

Congestion control in computer networks with traffic growth which goes up exponentially become one of the important effort to provide good service for users. Congestion causing packet loss are high, low throughput and high delay. Therefore developed a new congestion control scheme that is more diverse. One of them is PGMCC (Pragmatic General Multicast Congestion Control) with a delivery that is multicast.

Only compared PGMCC and TCP with the number of users 50, 100, 150, 200, 250 and 300. Also add background traffic with a data rate of 800 Kbps and 1200 Kbps. In terms of packet loss, this PGMCC schemes have good quality, by reducing the occurrence of packet loss. In a dense network conditions as well as usual, there was packet loss that is smaller in number when compared with TCP. However PGMCC lacking the large delay and low throughput when compared with the performance of TCP that has low delay and throughput of little value.

Keywords : PGMCC, Multicast, Congestion Control

BAB II

DASAR TEORI

2.1 Protocol Transport

2.1.1 *Pragmatic General Multicast (PGM)*

Dalam suatu jaringan komputer, terjadi sebuah proses komunikasi antar entiti atau perangkat-perangkat yang ada pada jaringan tersebut. Untuk berkomunikasi, mengirim dan menerima antara dua entiti dibutuhkan pengertian diantara kedua belah pihak. Pengertian inilah yang dikatakan sebagai protokol. Jadi protokol adalah himpunan aturan-aturan yang mengatur komunikasi data. Protokol mendefinisikan apa yang dikomunikasikan, bagaimana dan kapan terjadinya komunikasi.

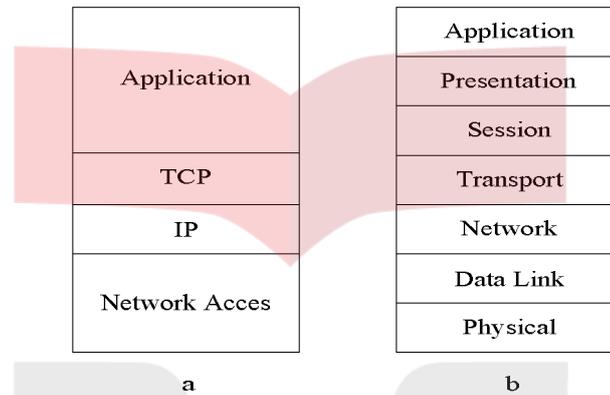
Salah satu protokol transport yang dapat digunakan adalah PGM (*Pragmatic General Multicast*). *Pragmatic General Multicast (PGM)* adalah protokol transport *multicast* yang handal. PGM menyediakan urutan paket ke beberapa penerima sekaligus, sehingga cocok untuk aplikasi seperti pengiriman *file* ke banyak penerima [5]. Protokol *multicast* ini menambahkan kemampuan pada penerima untuk mendeteksi paket data yang hilang dan mampu mengambil tindakan korektif (mirip secara prinsip dengan TCP).

Sementara TCP menggunakan ACK untuk mengakui kelompok paket yang dikirimkan, PGM menggunakan konsep *negative acknowledgement (NAKs)*. NAK dikirim secara *unicast* kembali ke pengirim melalui prosedur *hop by hop* setiap kali ada deteksi kehilangan data dari suatu urutan tertentu. PGM sangat bergantung pada NAKs untuk integritas, ketika NAK akan dikirim, NAK *confirmation (NCF)* dikirim melalui *multicast* untuk setiap hop. *Repaired Data (RDATA)* kemudian dikirim kembali baik dari sumber atau dari *Designated Local Repaired (DLR)*.

PGM merupakan protokol yang dikembangkan oleh IETF. PGM belum merupakan standar, tapi telah diimplementasikan pada beberapa perangkat dan *operating system*, seperti windows xp.

2.1.2 Transmission Control Protocol (TCP)

TCP merupakan *transport layer* yang terletak antara layer IP dan layer aplikasi yaitu layer 3 pada protokol TCP/IP. Saat ini sebagian besar aplikasi internet seperti www, FTP dan telnet menggunakan TCP sebagai *transport protocol* nya.



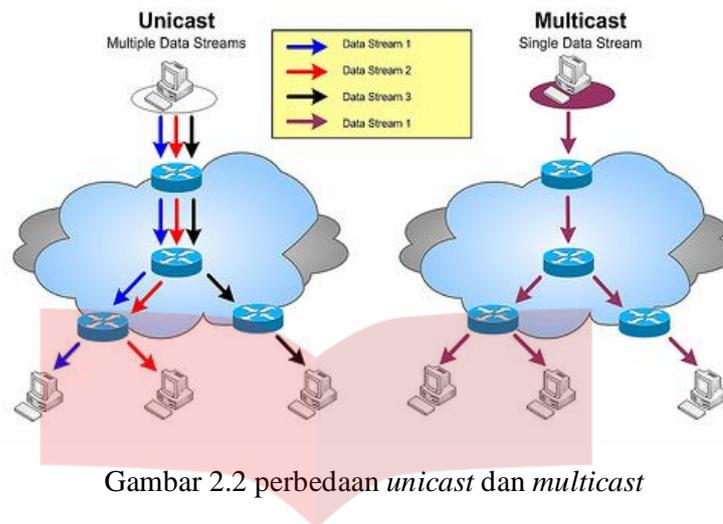
Gambar 2.1 (a) Susunan Layer pada protokol TCP/IP (b) OSI

TCP dengan sifat *connection oriented* mempunyai beberapa karakteristik yaitu *acknowledgement* terhadap data yang diterima, *retransmission* pada pengiriman data yang terdapat kesalahan, *flow control* terhadap paket yang akan dikirimkan dan *congestion avoidance* yang akan terjadi ketika adanya indikasi terjadinya kongesti yaitu oendetekdian kehilangan paket data atau *timeout ack*.

2.2 Multicast

Masalah kemacetan dapat dianggap sebagai topik dengan prioritas tinggi pada jaringan komputer. Hal ini semakin mendesak dengan berkembangnya jaringan dan aplikasi yang dijalankan diatasnya, disamping keterbatasan sumber daya jaringan untuk dibagi. Protokol *multicast* membahas cara pengiriman paket data secara efisien ke banyak penerima, misalnya untuk konferensi dan mereplikasi informasi lebih cepat dibandingkan dengan *unicast* [5].

Adapun perbandingan banyak *unicast* dengan *multicast* adalah sebagai berikut:

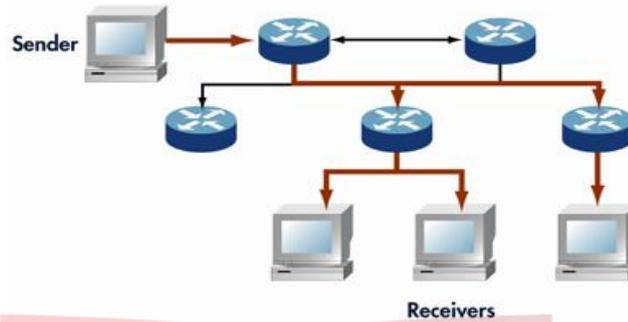


Gambar 2.2 perbedaan *unicast* dan *multicast*

Dari gambar 2.2 terdapat perbedaan pada aliran data. Pada *unicast* terdapat tiga aliran data, jadi pada pengiriman data dengan menggunakan pengalamatan secara banyak *unicast* maka pengirim akan mengirimkan aliran data sebanyak jumlah penerima. Sedangkan pada *multicast*, pengirim hanya mengirimkan satu aliran data dan bergantung dengan banyaknya jumlah penerima.

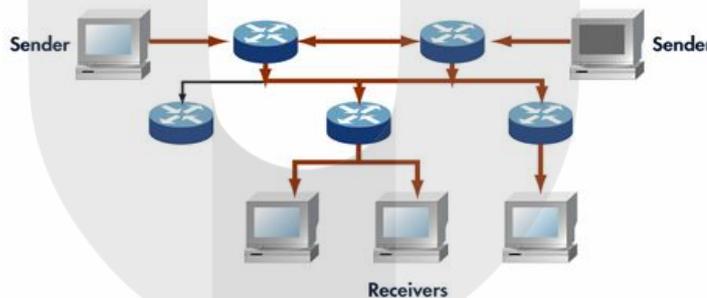
Multicast memiliki sistem pengalamatan yang berbeda dengan hubungan *point to point* yang selama ini dikenal. Pada sistem *point to point*, pengiriman data dilakukan langsung ke alamat *host* yang dituju, sedangkan pada sistem *multicast* data dikirimkan ke suatu grup *multicast*. Jika sebuah *host* ingin menjadi penerima data, *host* tersebut harus diberi sebuah alamat grup *multicast*. Sumber data akan mengirim data ke alamat grup, sehingga masing-masing *host* yang tergabung pada grup yang sama akan menerima data [1].

Multicast dapat dibagi menjadi dua tipe berdasarkan jumlah pengirimnya yaitu *multicast* satu ke banyak dan *multicast* banyak ke banyak. Adapun *multicast* satu ke banyak dapat dilihat pada gambar berikut:



Gambar 2.3 *Multicast* satu ke banyak

Dari gambar 2.3 terlihat bahwa transmisi *multicast* satu ke banyak yaitu dari sebuah *host* untuk semua *host* penerima yang dimaksud. Sender mengirimkan alamat paket *multicast* kepada multicaster group yang diteruskan ke semua penerima. *Multicast* rute ditunjukkan dengan panah merah. *Multicast* banyak ke banyak ditunjukkan pada gambar 2.4 berikut.



Gambar 2.4 *Multicast* banyak ke banyak

Transmisi *multicast* banyak ke banyak dari dua *host* pengirim ke semua *host* penerima yang dimaksud. *Host* pengirim mengirimkan alamat paket *multicast* grup yang diteruskan ke semua penerima.

2.3 Kongesti

Kongesti adalah situasi yang terjadi apabila paket-paket yang dipancarkan lewat jaringan mendekati jumlah paket maksimum yang mampu ditangani pada

jaringan. Hal ini juga terjadi ketika sejumlah paket yang ditransmisikan mengalami kebuntuan pada saat transmisi. Diperlukan kendali kongesti untuk mengatasi kejadian ini. Kongesti terjadi ketika sumber jaringan kelebihan beban, sumber akan menjadi *individual transmission link*, kelompok *buffer* dalam kondisi penuh pada sistem. Kongesti mungkin juga terjadi karena adanya gangguan. Kendali kongesti terfokus pada lalu lintas data yang masuk ke sebuah jaringan telekomunikasi, untuk menghindari kegagalan pengiriman data yaitu dengan mencoba untuk mengurangi tingkat pengiriman paket.

Kongesti merupakan permasalahan yang muncul di jaringan akibat banyak *user* yang berbagi *resource (bandwidth)* dimana jumlah permintaan akan *bandwidth* melebihi kapasitas *link* yang tersedia. Kongesti melibatkan beberapa hal berikut:

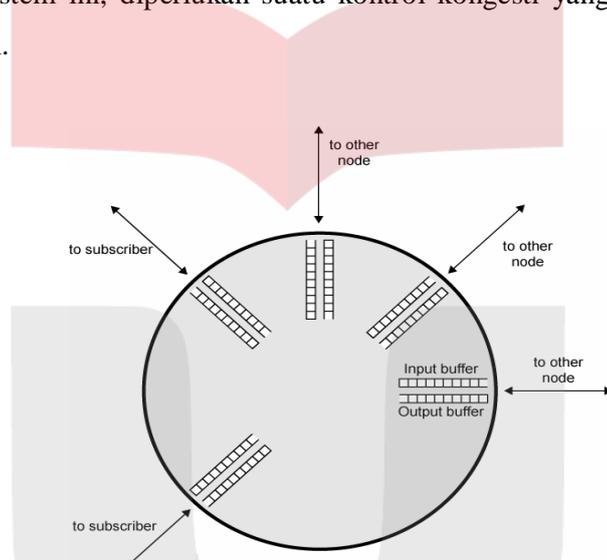
- Kemampuan jalur komunikasi data dalam mentransfer data terbatas.
- Beban jaringan, jumlah data yang dapat dikirim dapat bervariasi tergantung jumlah pengguna dan kecepatan pengiriman data.

Dalam jaringan *best effort*, variasi beban akan mengakibatkan variasi kualitas yang dirasakan oleh pengguna. Saat beban naik, dengan kemampuan jalur komunikasi data yang tetap dan terbatas, maka kualitas layanan akan menurun.

Kendali kongesti sangat penting untuk menjaga beban *radio interface* dibawah *threshold* untuk menjamin ketersediaan persyaratan untuk sebuah panggilan. Beban yang berlebihan menyebabkan kapasitas rendah, penurunan QOS, ketersediaan layanan di daerah jangkauan layanan atau jaringan berada dalam keadaan tidak stabil.

Kendali kongesti merupakan algoritma terdistribusi dalam jaringan dalam tujuan untuk berbagi sumberdaya. Terdiri dari dua komponen yaitu *link* algoritma yang dijalankan didalam jaringan oleh *router* dan *switch*. *Source* algoritma yang dijalankan oleh *host* atau sumber. *Link* algoritma mendeteksi adanya kongesti dan memberikan informasi berupa *feedback* dan *source* bereaksi atas *feedback* kongesti dari jaringan melalui pengaturan kecepatan pengiriman paket ke jaringan. *Link* algoritma yang berfungsi memberikan *feedback* dari *router*.

Kendali kongesti tidak dapat tercapai dengan menambahkan sumber-sumber dalam jaringan dalam formasi kapasitas *buffer* atau menambah kecepatan link lebih tinggi. Dengan solusi ini kemacetan masih dapat terjadi. Kongesti mungkin terjadi dalam beberapa jaringan *packet* dan jaringan Frame Relay tanpa kecuali. Jika beban trafik terus meningkat, kongesti akan menjadi semakin serius dan dapat mengakibatkan kegagalan sistem. Untuk mencegah terjadinya kegagalan sistem ini, diperlukan suatu kontrol kongesti yang dapat mengurangi beban sistem.



Gambar 2.5 Sistem antrian pada *buffer*

Model jaringan data adalah model antrian sehingga berpeluang besar terjadinya kongesti. Apabila pada jaringan menggunakan sistem antrian terbatas maka yang terjadi beberapa data mungkin hilang. Paket yang datang pada jaringan akan disimpan pada *buffer* masukan, kemudian terbentuk *routing*. Paket pada *buffer* kemudian bergerak keluar ke *buffer* keluaran. Tetapi paket-paket yang akan keluar ini mengalami antrian untuk dipancarkan keluar dengan metoda *time division multiplexing statistic*. *Buffer* akan terisi oleh paket-paket yang tiba untuk kemudian dirutekan kembali. Apabila digunakan *buffer limited*, maka beberapa paket akan dibuang.

Kongesti ini dapat disebarkan melalui jaringan, sehingga yang terjadi adalah kongesti diseluruh jaringan. Kendali kongesti berada pada *packet jaringan switch*. Fungsi kendali kongesti adalah mengirimkan paket kendali untuk beberapa

atau semua *node* sumber sehingga dapat mengetahui apabila kongesti terjadi. Apabila terjadi kongesti, maka diperlukan lalu lintas tambahan selama kongesti.

Kendali kongesti pada dasarnya terbagi menjadi tiga bagian, yaitu *admission control*, *load control* dan *packet scheduling*. *Admission control* merupakan kumpulan tindakan yang diambil oleh jaringan pada saat pembentukan hubungan untuk menentukan suatu permintaan hubungan baru pada suatu jaringan akan ditolak atau diterima. *Load control* bertugas untuk membatasi QoS serta bertugas membatasi jaringan agar tidak terjadi *overload* dan tetap stabil. Sedangkan *packet scheduling* berfungsi untuk mengontrol *packet access*, menentukan dan memisahkan akses sumber daya radio untuk hubungan trafik *non real time*. Keadaan kongesti akan mengakibatkan kapasitas yang rendah, penurunan QoS, ketidaktersediaan layanan di daerah jangkauan layanan atau jaringan berada dalam keadaan tidak stabil.

2.3.1 TFMCC (*TCP Friendly Multicast Control Congestion*)

TFMCC merupakan salah satu metoda deteksi kongesti yang dapat digunakan. metoda ini akan sangat bermanfaat pada tipe transmisi yang berupa *multicast*. TFMCC bertujuan untuk menyediakan metode efisien untuk mentransmisi data dari satu pengirim ke banyak penerima. TCP Friendly menjadi salah satu kriteria dalam pemilihan protokol *multicast congestion control* [3]. Kriteria ini adalah salah satu pertimbangan utama dalam mendesain protokol jaringan baru dan menjadi karakteristik yang diinginkan.

TFMCC dikenal sebagai persamaan dasar mekanisme kontrol kongesti *multicast* dikenal yang memiliki keadilan terhadap aliran TCP. Algoritma ini menggunakan persamaan kontrol yang diturunkan dari persamaan kesetimbangan TCP yang berkaitan dengan throughput terhadap loss dan Round Trip Time (RTT) (skema ini disebut "base equation").

$$T_{TCP} = \frac{s}{t_{RTT} \left(\sqrt{\frac{2p}{3}} + \left(12\sqrt{\frac{3p}{8}} \right) p (1 + 32p^2) \right)}$$

..... (2.1)

Dimana: P = fungsi dari level loss pada suatu kejadian
 $t_{RTT} = RTT$
 s = ukuran paket
 T_{TCP} = Throughput TCP

Persamaan ini secara langsung mengontrol laju transmisi pengirim. Tingkat loss dan RTT adalah parameter yang menentukan target *throughput* ini. Setiap penerima menghitung target *throughput* dan menganggap itu sebagai tingkat pengiriman yang diterima dari pengirim. TFMCC menggunakan skema umpan balik yang menjamin bahwa umpan balik dari penerima menghitung laju transmisi paling lambat, selalu mencapai pengirimnya. Skema ini didasarkan pada konsep Current Limitting Receiver (CLR). Selain itu, desain TFMCC memastikan bahwa pengirim mendapatkan umpan balik dari penerima yang mengalami kondisi jaringan yang buruk tanpa kewalahan dengan adanya ledakan umpan balik dari penerima.

2.3.2 PGMCC (*Pragmatic General Multicast Congestio Control*)

PGMCC merupakan suatu skema kontrol kongesti *single rate multicast* dengan menggunakan PGM untuk mencapai kondisi jaringan yang memiliki kemampuan skalabilitas, stabilitas dan respon cepat terhadap bermacam kondisi jaringan. Masalah yang ditemui pada protokol *multicast single rate* adalah bahwa semua penerima harus menerima layanan pada *rate* yang sama, yang tidak dapat mengakomodasi sifat heterogen dari penerima dan perbedaan dari kapasitas *bandwidth* dari jaringan yang berbeda [2].

PGMCC merupakan skema yang hampir sama dengan TCP di mana laporan penerima dikirim ke pengirim dalam bentuk Negative Acknowledgments (NAKs). Fungsi ini membantu pengirim untuk terus memantau status grup penerima. Penerima dengan *throughput* terburuk sesuai dengan skema kontrol perwakilan grup, Acker. Menggunakan standar *window controller* seperti TCP pada positif ACK dan dijalankan antar pengirim dan grup yang dinamakan *acker*. Skema kontrol kongesti berbasis *window* yang mirip dengan kendali kongesti TCP

dijalankan antara pengirim dan Acker, yang memiliki tanggung jawab untuk mengirim ACK positif untuk setiap paket data. Window kontrol dasar PGMCC digunakan berbeda dari kontrol kongesti TCP. Perbedaan adalah penggunaan window yang berbeda untuk menilai kehandalan/kontrol aliran, kemampuan retransmisi, menggunakan pengukuran RTT pada pengirim untuk memilih perwakilan dan ACK clocking ketika terjadi perubahan acker.

Perbedaan PGMCC adalah dengan adanya pemilihan acker yang memungkinkan acker berpindah dari satu penerima ke penerima lain. Skema ini di implementasikan pada protokol PGM. Skema PGM tergantung dari umpan balik yang dikirimkan penerima ke server agar server dapat mengatur *data rate* sesuai dengan kondisi jaringan. Acker yang dipilih merupakan penerima dengan nilai *throughput* terendah dari semua anggota pada penerima.

Kekurangan jika memilih acker dengan kondisi *throughput* terjelek adalah cepatnya pergantian acker dari satu penerima ke penerima lain yang mengakibatkan jaringan tidak stabil. Sehingga untuk mengatasi masalah tersebut adalah dengan menambahkan variable *c* dalam pemilihan acker, yaitu

$$T(X_i) < c T(X_j), 0 < c < 1 \dots\dots\dots(2.2)$$

Dimana:

$T(X_i)$ = nilai *throughput* penerima di grup selain acker

$T(X_j)$ = nilai *throughput* penerima acker saat ini

Pengirim pada PGMCC akan memonitor secara terus menerus laporan NAK dari penerima dan kemudian akan memilih acker apabila syarat terpenuhi.

Fields pada NAK adalah sebagai berikut:

- Identitas UE (recv_id)
- Estimasi *loss rate* (recv_loss)
- Paket data yang diterima (recv_lastseq)

Penerima dengan *throughput* paling jelek akan dipilih sebagai acker, yang akan mengirimkan ACK kepada pengirim. Identitas dari pengirim akan dibawa

pada masing-masing paket data. ACK terdiri dari *loss report* sama seperti NAK dan dua tambahan, yaitu:

- *Sequence number* pada paket data (*ack_seq*)
- 32 bit yang mengindikasikan status penerima pada 32 paket terakhir (*bitmask*)

Jumlah paket yang hilang di estimasikan pada masing-masing penerima. Estimasi paket hilang tersebut dikirim kembali pada pengirim *multicast* dengan NAK dan ACK. Untuk mengukur jumlah paket yang hilang tersebut, masing-masing penerima menambahkan kode paket yang diterima dengan angka 1 untuk paket yang hilang dan 0 untuk lainnya. Pengirim akan mengirim data dengan laju transmisi sesuai dengan *throughput* terendah pada penerima.

2.4 Parameter Uji perfomansi

Untuk mengontrol kongesti yang terjadi dalam suatu jaringan, diperlukan parameter-parameter dalam menguji perfomansi jaringan tersebut. Adapun beberapa parameter uji perfomansi dalam mengontrol kongesti adalah *delay*, *throughput* dan *packet loss*.

1. *Delay*

Merupakan selang interval waktu antara ketika paket dikirim dan keluaran setelah melalui proses antrian dari node sumber. Delay yang diamati pada simulasi menggunakan NS-2 ini merupakan *delay end to end* yang merupakan selang dari paket dikirimkan, diterima hingga laporan bahwa data yang dikirimkan telah diterima.

$$\text{Delay (ms)} = (\text{waktu paket diterima} - \text{waktu paket dikirim})$$

..... (2.3)

2. *Throughput*

Throughput adalah sebagai parameter banyaknya bit yang sukses dikirim dari *source* sampai ke *destination* dalam suatu selang waktu pengamatan dalam melakukan satu kali simulasi dengan satuan bit/second (bps) yang merupakan kondisi *data rate* sebenarnya dalam suatu jaringan. Persamaan umum untuk mendapatkan nilai *throughput* dari simulasi adalah :

$$\text{Throughput} = \frac{\text{jumlah paket diterima} \times 8}{\text{periode pengamatan}} \dots\dots\dots(2.4)$$

Paket diterima adalah jumlah paket yang berhasil dikirimkan dan sukses diterima pada tujuan. Periode pengamatan adalah banyaknya waktu pengamatan dalam melakukan satu kali simulasi.

3. *Packet loss*

Packet loss merupakan jumlah persentase paket yang hilang dalam proses pengiriman data dari sumber trafik ke node tujuan yang dapat terjadi Karena *collision* dan *congestion* pada jaringan terhadap jumlah keseluruhan paket yang dikirim. Persamaan *packet loss* terlihat seperti gambar berikut:

$$\text{Packet loss} = \frac{\text{jumlah packet loss}}{\text{jumlah packet yang dikirim}} \times 100 \% \dots\dots\dots(2.5)$$

Jumlah *packet loss* adalah banyaknya paket yang hilang akibat kongesti. Total paket yang dikirim adalah jumlah dari *packet loss* dan yang diterima.



BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan analisa dan pembahasan hasil simulasi dari setiap skenario, dapat diambil beberapa kesimpulan masing-masing sebagai berikut:

1. Pada setiap skenario diperoleh nilai *delay* PGMCC yang lebih lebih besar dari TCP. Dengan *delay* terendah yaitu pada skenario pertama dengan kondisi jaringan yang tidak padat menggunakan user 50 diperoleh *delay* sebesar 8.56 ms pada PGMCC, sedangkan pada TCP hanya 3.73 ms. *Delay* terbesar pada PGMCC yaitu pada skenario III dengan jumlah user 300 sebesar 18.364 ms yang masih dalam ukuran wajar menurut standar ITU-T.
2. Pada masing-masing skenario diperoleh nilai *throughput* PGMCC yang lebih kecil jika dibandingkan dengan TCP. Nilai *throughput* PGMCC terbesar adalah pada skenario I pada kondisi jaringan yang tidak terlalu padat, yaitu sebesar 448.109 Kbps yang nilainya masih lebih besar dibandingkan dengan TCP yaitu 394.533 Kbps pada jumlah user 100 yaitu *throughput* maksimal sebelum terjadi kongesti.
3. PGMCC cocok dalam hal mengatasi *packet loss*, yang terlihat dari masing-masing skenario terdapat *packet loss* yang nilainya cukup kecil yaitu hanya 5.8% dengan kondisi jaringan terburuk. Bila dibandingkan dengan TCP yang mengalami *packet loss* yang cukup besar pada skenario III dan pada jumlah user 300 yaitu sebesar 16.924%. persentase *packet loss* yang dialami TCP sudah melewati batas standar maksimal *packet loss* ITU-T.
4. Kendali kongesti PGMCC memiliki perfomansi lebih baik dalam mengatasi *packet loss*, tetapi membutuhkan *delay* yang lebih besar sehingga menghasilkan *throughput* yang lebih kecil dibandingkan dengan TCP.

5.2 Saran-saran

Beberapa saran yang bisa disampaikan sebagai tindak lanjut dari penelitian yang telah dilakukan adalah sebagai berikut:

1. Dapat dilakukan perbandingan perfomansi kendali kongesti PGMCC dengan TFMCC yang memiliki sistem kerja yang hampir sama.
2. Dapat dilakukan penelitian dengan menggunakan simulator lain, sehingga dapat dibandingkan hasilnya untuk masing-masing media.



DAFTAR PUSTAKA

- [1] L. Rizzo, "A PGM Host Implementation for Free BSD",
<http://www.iet.unipi.it/~Luigi/pgm.html>
- [2] Luigi Rizzo, "PGMCC: a TCP-Friendly Single Rate Multicast Congestion Control Scheme"
- [3] Mark Handley, Sall Floyd, "Strawman Specification For TCP Friendly (Reliable) Multicast Congestion Control (TFMCC)"
- [4] Todd Montgomery, "A Loss Tolerant Rate Controller For reliable Multicast", NASA IV&V Technical Report NASA-IVV-97-011, Aug.1997
- [5] Tony Speakman Et Al, "PGM Reliable Transport Protocol Spesification", Internet draft, draft-speakman-pgm-spec-04.txt
- [6] Wirawan Bayu Andi, Indarto Eka, *Mudah Membangun Simulasi dengan Network Simulator-2*, ANDI, Yogyakarta.