

ANALISA PERBANDINGAN ALGORITMA KONTROL KEMACETAN LAPISAN TRANSPORT OSI DENGAN METODE RED, SFQ DAN REM MENGGUNAKAN NETWORK SIMULATOR NS-2

*COMPARATIVE ANALYSIS OF CONGESTION CONTROL ALGORITHM OSI
TRANSPORT LAYER WITH RED, SFQ AND REM METHOD USING NETWORK
SIMULATOR NS-2*

¹*Bayu Pratama Putra (bp8508060@gmail.com)*

²*Made Kamisutara (kamisutara@gmail.com)*

³*Nyoman Suryadipta (nyoman.surya@narotama.ac.id)*

¹*Program Studi Sistem Komputer, Fakultas Ilmu Komputer, Universitas
Narotama Surabaya, Indonesia*

²*Dosen Pembimbing I, Fakultas Ilmu Komputer, Universitas Narotama Surabaya,
Indonesia*

³*Dosen Pembimbing II, Fakultas Ilmu Komputer, Universitas Narotama
Surabaya, Indonesia*

ABSTRAK

Penelitian terhadap beberapa algoritma control kemacetan aktif telah banyak dilakukan salah satunya dalam meneliti perbandingan algoritma SFQ, RED, dan REM. Algoritma tersebut merupakan suatu manajemen antrian yang aktif mendeteksi dinamisnya jumlah paket data dalam antrian. Tugas akhir ini bertujuan untuk membuktikan analisa perbandingan delay, loss, throughput yang diterapkan oleh ketiga algoritma diatas menggunakan network simulator NS-2 yang banyak digunakan dalam dunia pendidikan untuk menganalisa algoritma dalam setiap pengaturan jaringan komunikasi data. SFQ (Stochastic Fair Queuing) merupakan sebuah metode penjadwalan antrian yang dirancang berdasarkan metode First Come First Serve yang berarti juga First In First Out. Random Early Detection (RED) merupakan suatu metode penjadwalan antrian yang mampu mendeteksi sejak dini kemungkinan antrian yang dapat menyebabkan kemacetan sehingga mengakibatkan terjadinya delay, loss, jitter hingga error pada penyusunan kembali paket data tersebut. Random Exponential Marking (REM) algoritma manajemen antrian aktif yang bertujuan untuk mendapatkan utilisasi yang tinggi dan menurunkan loss dan delay dengan cara yang sederhana dan terukur. Dalam makalah ini nantinya disimulasikan konfigurasi jaringan yang mempunyai delay yang lebih tinggi dan bandwidth yang lebih rendah pada bottleneck saluran utama.

Kata kunci: Active Queue Management, Stochastic Fair Queuing (SFQ), Random Early Detection (RED), Random Exponential Marking (REM), First Come First Serve (FCFS), Throughput, and Delay.

ABSTRACT

Research on several active congestion controls algorithms many has done one in investigating comparison algorithms of SFQ, RED, and REM. Those algorithm was a queue management active detects dynamically the amount of data packets in a queue. The mission of this last research was intended to prove comparative analysis delay, loss, throughput applied by those algorithms up using network simulator NS-2 much used in education to analyze algorithms in any portsetup data communications network. Stochastic Fairness Queuing (SFQ) is a queue scheduling method designed base on the method of first come first serves meaning also first in first out. Random Early Detection (RED) is a queue scheduling method which able to detect since early possible line that can cause the delay, loss, jitter until error on the realignment data packets. Random Exponential Marking (REM) is the queue management active algorithms aimed to get the high utilization and lowering loss and a delay with a simple manner and measured. In this paper will be simulated a network configuration has higher delay and lower bandwidth in the bottleneck main channel.

Keywords: Active Queue Management, Stochastic Fair Queuing (SFQ), Random Early Detection (RED), Random Exponential Marking (REM), First Come First Serve (FCFS), Throughput, delay, Utilization and Loss rate.

1. PENDAHULUAN

Transport layer merupakan suatu lapisan pada OSI yang bertugas untuk memecah data menjadi beberapa packet data yang kemudian disusun kembali menjadi packet data yang utuh. Tapi sebelum pecahan packet data itu disusun kembali terjadi beberapa proses yang berfungsi untuk mengatur antrian paket data tersebut yang melalui sebuah gateway sehingga paket data tersebut tidak bertabrakan, tercampur dengan paket data lainnya hingga paket itu ditransmisikan dan disusun kembali menjadi paket data yang utuh. Dalam proses antrian tersebut jika tidak dikendalikan dengan baik akan menyebabkan kenaikan pada delay, loss, seiring dengan bertambahnya queue length. Yang dapat menurunkan kualitas layanan dari suatu jaringan yaitu throughput dan utilization.

Penelitian terhadap beberapa algoritma control kemacetan telah banyak dilakukan oleh beberapa civitas khususnya algoritma SFQ, RED, dan REM. Algoritma tersebut merupakan suatu manajemen antrian yang aktif mendeteksi dinamisnya jumlah paket data dalam antrian. Penelitian ini bertujuan untuk membuktikan analisa perbandingan delay, loss dan throughput yang diterapkan oleh ketiga algoritma diatas menggunakan network simulator NS-2.

SFQ (Stochastic Fair Queuing) merupakan sebuah metode penjadwalan antrian yang dirancang berdasarkan metode First Come First Serve yang berarti juga First In First Out dalam memecah kumpulan paket data sesuai ukuran yang ditentukan (quantum value) dan ditransmisikan ke node lain berdasarkan algoritma round robin (stochastic/random) dengan iterasi pengiriman yang disebut nilai perturb dalam satuan detik.

Random Early Detection (RED) merupakan suatu metode penjadwalan antrian yang mampu mendeteksi sejak dini kemungkinan antrian yang dapat menyebabkan kemacetan sehingga mengakibatkan terjadinya delay, loss, jitter hingga error pada penyusunan kembali paket data tersebut. Dengan menentukan batasan nilai minimum dan maksimum threshold maka tidak semua paket data dengan sengaja dibuang (dropped) tapi jika paket data melebihi limit teratas maka semua paket data dalam antrian tersebut dibuang dengan sengaja. Supaya lebih efisien, RED harus memastikan bahwa peringatan kemacetan disampaikan pada waktu yang cukup untuk menghentikan sumber transmisi tanpa menurunkan penggunaan saluran¹. RED juga harus memastikan tersedianya ruang kosong yang cukup sebagai buffer antrian agar tidak terjadi beban berlebih pada kapasitas saluran pada saat terdeteksinya kemacetan hingga saat beban pada bottleneck saluran berkurang.

Random Exponential Marking (REM) algoritma manajemen antrian aktif yang bertujuan untuk mendapatkan utilisasi yang tinggi dan menurunkan loss dan delay dengan cara yang sederhana dan terukur. Ide pertama dari REM adalah menstabilkan input rate sepanjang kapasitas saluran dan antrian sepanjang target kecil, tanpa memperhatikan jumlah pengguna

yang mengakses saluran. Ide kedua adalah menyatukan jumlah saluran yang berharga “price” (ukuran kemacetan), yaitu menjumlahkan semua price dalam semua router pada jalur dari pengguna ke end to end marking or dropping probability. Jumlah aliran yang aktif seimbang dengan jumlah perbedaan aliran pada buffer.

Masalah yang akan dibahas dalam proposal ini adalah membandingkan metode pengaturan kemacetan jaringan pada layer transport dengan menggunakan SFQ (Stochastic Fair Queuing), RED (Random Early Detection), REM (Random Exponential Marking). Dengan bantuan network simulator NS2 diharapkan dapat dibuktikan secara teoritis metode mana yang dapat dijadikan alternative untuk mengatasi kemacetan antrian paket data pada lapisan transport OSI.

Permasalahan yang harus diselesaikan pada proyek akhir ini adalah bagaimana pemrograman dari NS2 dengan topology jaringan yang telah dibuat pada jaringan yang menggunakan event TCP. Sehingga diperoleh perbandingan algoritma yang menggunakan SFQ, RED dan REM congestion control. Event TCP dipilih karena melalui TCP terdapat sinyal ACK yang merupakan sinyal notifikasi bahwa packet data telah diterima oleh penerima kemudian mengirim sinyal ACK tersebut ke pengirim bahwa packet telah diterima. Sehingga dapat dengan mudah diamati bahwa loss ataupun drop data terjadi manakala tidak ada sinyal ACK yang kembali ke pengirim.

Pertama akan dibuatkan simulasi pengiriman data melauai transport layer dengan menggunakan algoritma SFQ (Stochastic Fair Queuing), Random Early Detection (RED) dan Random Early Marking (REM) dimana akan diperoleh data berupa jumlah packet data yang dikirim (tracing file), delay waktu (time delay) yang terlihat saat pengiriman data, jumlah packet hilang (loss packet) yang diterima oleh penerima, jumlah bit sebenarnya yang diterima oleh node dalam waktu tertentu (throughput).

Setelah memperoleh hasil percobaan melalui simulasi tersebut diharapkan dapat diketahui dan dibuktikan bahwa masing-masing algoritma mempunyai keunggulan dan kelemahan masing-masing. Akan tetapi dalam hal tertentu ada skala prioritas dimana parameter throughput, loss ratio dan utilization akan berdampak pada performa masing-masing algoritma tanpa mengesalkan algoritma yang lebih unggul pada parameter delay.

2. METODOLOGI PENELITIAN

Sebelum melakukan penelitian ini terlebih dahulu diperlukan landasan teori yang mendasari penelitian ini menjadi penelitian yang berbobot dan layak untuk dijadikan penelitian skripsi. Landasan teori meliputi *queue management*, *OSI Transport Layer*, *SFQ*, *RED*, dan *REM*, parameter *Quality of Service* suatu jaringan.

Kemudian dilakukan perancangan dan pembuatan simulasi untuk membuktikan fungsi *active queue management* pada suatu perangkat yang mengalami *bottleneck link*.

- Throughput

Merupakan suatu parameter performa utama yang menentukan baik buruknya kualitas layanan yang diberikan oleh suatu perangkat pengatur jaringan dalam hal ini adalah router. Throughput merupakan jumlah bit sebenarnya yang ditransmisikan oleh suatu perangkat router dalam periode waktu tertentu (second). Satuan dari throughput adalah bps (bit per second). Semakin tinggi nilai throughput dalam pengatur jaringan tersebut menunjukkan kelebihan pengatur jaringan tersebut dalam memanfaatkan bandwidth yang disediakan pada link tersebut.

- Utilization

Merupakan perbandingan dari nilai throughput dengan nilai bandwidth yang disediakan pada link tersebut. Merupakan prosentase dari throughput terhadap bandwidth. Utilization

menunjukkan seberapa optimal pengatur jaringan tersebut dalam memanfaatkan bandwidth yang disediakan pada link tersebut.

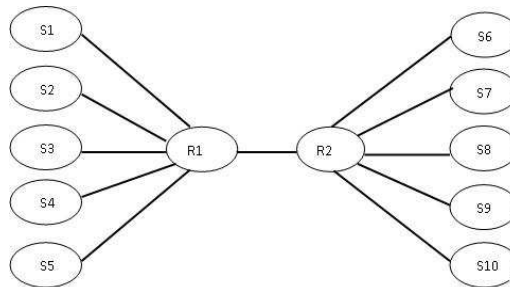
- Delay

Merupakan waktu yang diperlukan untuk mentransfer paket data dari node satu ke node lainnya. Semakin tinggi waktu yang diperlukan oleh suatu paket data untuk berpindah dari node satu ke node lainnya menunjukkan kelemahan suatu perangkat control jaringan dalam mengelola antrian paket data. Pada dasarnya semakin tinggi jumlah antrian paket data linier dengan tingginya delay, tapi jika pengatur jaringan tersebut mampu mengatur antrian yang masuk maka delay yang terjadi dalam perangkat tersebut mampu diminimalisir dengan baik.

- Loss rate

Pada jenis transmisi Transfer Control Protocol (TCP) suatu node pengirim membutuhkan sinyal ACK dari penerima yang menunjukkan bahwa paket yang dikirim telah diterima dengan baik oleh penerima. Akan tetapi jika tidak ada sinyal ACK yang dikirim balik oleh penerima ke pengirim maka paket yang dikirim tersebut dianggap hilang atau loss. Dalam proses pengiriman sinyal ACK pun juga dilakukan proses manajemen antrian di pengontrol jaringan tersebut. Oleh sebab itu semakin tinggi nilai loss rate dalam pengatur jaringan tersebut menunjukkan kelemahan perangkat tersebut dalam mengontrol antrian paket.

- Skenario simulasi



Gambar 1. Skenario Simulasi

Terdapat 5 buah node pengirim (S_1 - S_5) dan 5 buah node penerima (S_6 - S_{10}). Kelima node pengirim tersebut mengirim packet data dengan metode TCP melalui node R_1 dan R_2 . Besar bandwidth yang disediakan untuk masing-masing node pengirim ke R_1 adalah 100 Mbps sedangkan bandwidth yang disediakan pada node R_1R_2 adalah 10 Mbps. Kemudian bandwidth dari R_2 ke node penerima sebesar 100 Mbps. Dari data seperti jelas terlihat bahwa terjadi bottleneck jaringan dari kumpulan jalur yang besar melauai jalur yang lebih kecil.

Kemudian masing-masing node pengirim mengirimkan *packet* data sebesar 2 KB dengan kapasitas buffer sebesar 4 KB. Simulasi dilakukan dalam periode sampling 100 second. Round trip time (RTT) yang diperlukan oleh masing-masing node pengirim ke node R_1 adalah 10 ms sedangkan RTT dari R_1 ke R_2 60 ms dan dari R_2 ke masing-masing node penerima adalah 10 ms. Jadi nilai minimum end to end delay harus lebih besar dari 80 ms.

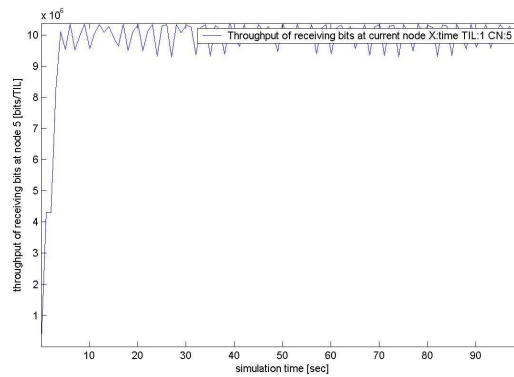
Tabel 1. Parameter Simulasi

Link	RTT (ms)	Rate (Mbps)	Protocol
S1R1	10	100	Droptail
S2R1	10	100	Droptail
S3R1	10	100	Droptail
S4R1	10	100	Droptail
S5R1	10	100	Droptail
R1R2	60	10	SFQ,RED,REM
R2S6	10	100	Droptail
R2S7	10	100	Droptail
R2S8	10	100	Droptail
R2S9	10	100	Droptail
R2S10	10	100	Droptail

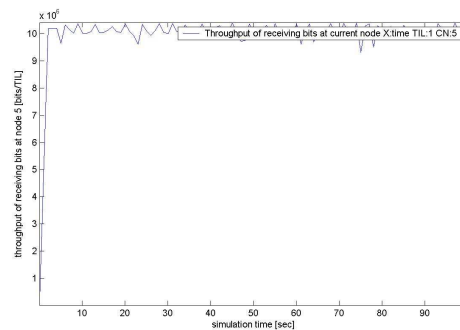
3. HASIL DAN PEMBAHASAN

Semua data dilakukan pada Noode 5 dimana terjadi bottleneck

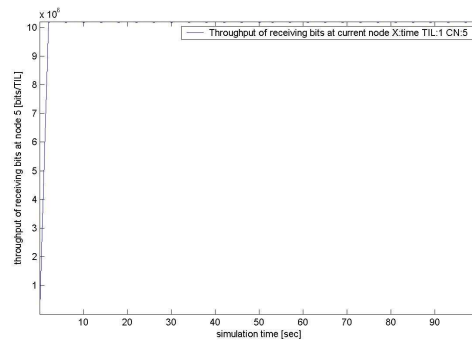
- Analisa Throughput



Gambar 2. Grafik Throughput SFQ

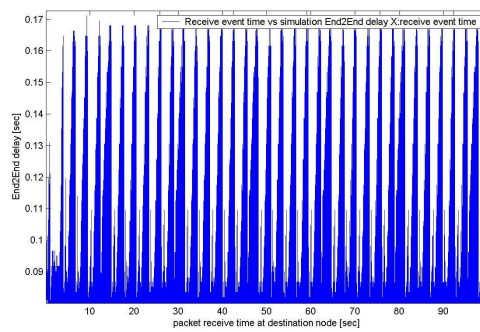


Gambar 3. Grafik Throughput RED

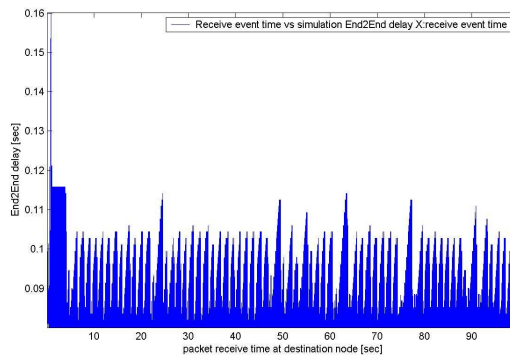


Gambar 4. Grafik Throughput REM

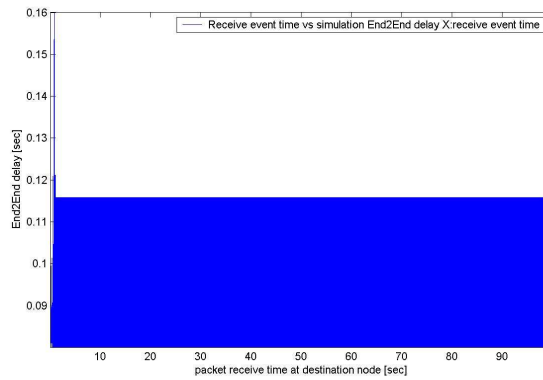
• **Analisa Delay**



Gambar 5. End2end delay SFQ



Gambar 6. End2end delay RED

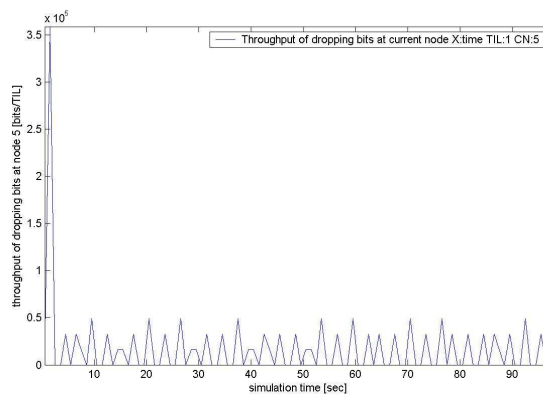


Gambar 7. End2end delay REM

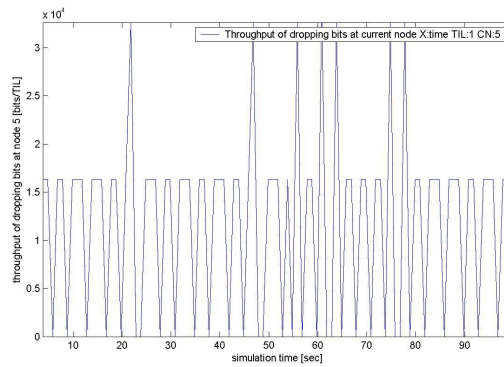
Algo	Delay Rate	Minimal Delay	Maksimal Delay
SFQ	87.1 milisecond	80.038 ms	172.758 ms
RED	87.4 milisecond	80.038 ms	160.132 ms
REM	97.9 milisecond	80.038 ms	160.132 ms

Tabel 2. Delay rate SFQ, RED dan REM

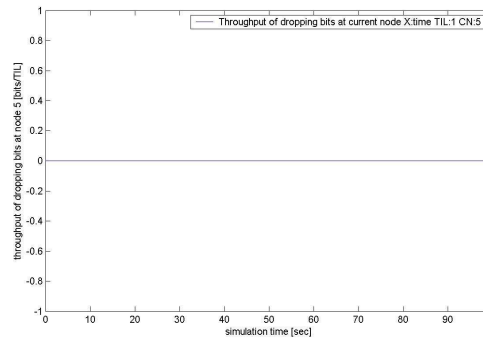
• **Analisa Loss rate**



Gambar 8. Dropping Packet SFQ Node 5



Gambar 9. Dropping Packet RED Node 5



Gambar 10. Dropping Rate Node 5 REM

4. KESIMPULAN

Quality of Service Performance	SFQ	RED	REM
Throughput	9.87	10.05	10.14
Delay Maksimum	172.758 ms	160.132 ms	160.132 ms
Delay Minimum	80.038 ms	80.038 ms	80.038 ms
Lost Packet	218	176	120
Send Packet	118521	120740	121903
Average Lost Ratio	0.184	0.146	0.098
Utilization	98.7	100.5	101.4

Tabel 3. Kesimpulan

Berdasarkan tabel 2 diatas terlihat nilai throughput pada REM lebih tinggi dibandingkan dengan SFQ dan RED, tapi pada simulasi kali ini penulis mendapatkan data bahwa algoritma SFQ mempunyai nilai terburuk dibandingkan ketiga algoritma tersebut. Dibandingkan dengan utilization ketiga algoritma AQM tersebut terlihat bahwa REM lebih optimal dalam memanfaatkan bandwidth yang disediakan oleh pengatur jaringan. Dalam hal mengelola

delay, berdasarkan tabel 3 terlihat SFQ mempunyai nilai maksimal delay yang tertinggi sedangkan jika dirata-rata berdasarkan tabel 2 nilai delay tertinggi didapatkan pada algoritma REM. Terdapat perbedaan disini karena berdasarkan gambar 5 dan 7 terlihat bahwa SFQ terjadi fluktuasi nilai delay sedangkan pada REM terjadi kondisi yang stabil dalam mengelola delay. Delay minimum adalah sama yaitu sama dengan nilai penjumlahan RTT masing-masing node pengirim ke node penerima yaitu minimum 80 ms. Berdasarkan rata-rata lost ratio SFQ paling tinggi niainya, artinya dalam mengelola panjang antrian SFQ buruk dalam mengelola paket data yang hilang atau drop. Sedangkan REM sangat baik dalam mengelola paket data yang drop yaitu paling kecil.

5. DAFTAR PUSTAKA

1. Patel, Sanjeev, P. K. Gupta, Arjun Garg, Prateek Mehrotra & Manish Chhabra. (2011). *Comparative Analysis of Congestion Control Algorithms Using ns-2*. IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 5, No 1.
2. Stevens, W. Richard. *TCP/IP Illustrated, Volume 1: The Protocols*. ISBN 0-201-63346-9
3. Stevens, W. Richard & Gary R. Wright. *TCP/IP Illustrated, Volume 2: The Implementation*. ISBN 0-201-63354-X
4. Stevens, W. Richard. *TCP/IP Illustrated, Volume 3: TCP for Transactions, HTTP, NNTP, and the UNIX Domain Protocols*. ISBN 0-201-63495-3
5. Tanenbaum, Andrew S. (2002). *Computer Networks, Fourth Edition Page 384*. New Jersey: Prentice Hall.
6. Stalling, William. (2004). *Data and Computer Communication 7th Edition. Chapter 13: Congestion Data Network*. UK: Pearson Education International.
7. <http://www.isi.edu/nsnam/ns>. *Introduction of Network simulator – ns-2*.
8. Athuraliya, Sanjeeva., Victor H. Li, Steven H. Low, Qinghe Yin. (2001). *REM: Active Queue Management*. California: IEEE Network.
9. McKeeney, P., (1990). “*Stochastic Fairness Queuing*”. Beaverton: In Proc. IEEE INFOCOM.
10. http://en.wikipedia.org/wiki/Quality_of_service. Diakses pada 10 Februari 2013 07.30 WIB
11. <http://www.ece.iupui.edu/tutorials/print.php?article=ns2>. Diakses pada 9 Februari 2013 22.15 WIB
12. Santosa, Budi ST. *Manajemen Bandwidth Internet dan Intranet*. Jakarta: Universitas Indonesia. <http://kambing.ui.ac.id>