

ANALISIS SIMULASI PERBANDINGAN ANTREAN FIFO DAN RED DENGAN MENGGUNAKAN NETWORK SIMULATOR 2 (NS2)

Ramadhani Ulansari
Program Studi Teknik Informatika, Fakultas Teknologi Informasi,
Universitas Respati Indonesia
Jl. Bambu Apus I No.3 Cipayung, Jakarta Timur 13890
E-mail : ramadhani.ulansari@gmail.com

ABSTRAK

Sistem antrean sangat banyak ditemui dalam dunia nyata, salah satu yang sering kita temui adalah sistem antrian pengiriman paket data pada jaringan berkabel. Antrean itu sendiri adalah suatu kejadian pada suatu garis tunggu yang memerlukan layanan dari suatu fasilitas layanan. Pada umumnya, sistem antrean dapat diklasifikasikan menjadi sistem yang berbeda-beda seperti FIFO (*First In First Out*), RED (*Random Early Detection*) dll, namun bagaimanakah cara kerja sistem antrean FIFO dan RED. Antrean manakah yang lebih bagus dalam pengiriman paket data sebuah jaringan kabel, Hal ini dapat kita analisis melalui simulasi yang membandingkan kedua antrian tersebut. Tujuan penelitian inii menganalisis perbandingan antrean FIFO dan RED menggunakan networks simulator 2 (NS2) sehingga dapat memperoleh data seperti *delay*, *packet loss*, *throughput*, dan total data yang diterima pada suatu topologi jaringan kabel. Metode software, hardware, perancangan topologi. Hasil simulasi diketahui antrean RED terlihat lebih mampu menangani antrean ketika sedang padat. Sedangkan antrean FIFO lebih unggul ketika menangani antrean berasal dari agent UDP ketika antrean tidak terlalu padat.

Kata kunci: Network Simulator, Antrean FIFO, Antrean RED, *Throughput*, *Delay*, *Jitter*.

ABSTRACT

Queue system was very much encountered in the real world, one of which we often encounter was the queue system delivery of data packets on wired networks. The queue it self was an event on a waiting line that requires the services of a service facility. Generally, queuing systems can be classified into different systems such as FIFO (First In First Out), RED (Random Early Detection) etc., but how the FIFO and RED queuing systems work. Which queue was better in sending data packets of a wired network, This we can analyze through a simulation that compares the two queues. The purpose of this study was to analyze the comparison of FIFO and RED queues using network simulator 2 (NS2) so as to obtain data such as delay, packet loss, throughput, and total data received on a cable network topology. Software method, hardware, topology design. Simulation results were known to queue RED looks more able to handle the queue while it was solid. While the FIFO queue was superior when handling the queue comes from the UDP agent when the queue was not too dense.

Keywords: Network Simulator, FIFO Queue, RED Queue, Throughput, Delay, Jitter.

PENDAHULUAN

Pada umumnya, sistem antrean dapat diklasifikasikan menjadi sistem yang berbeda-beda seperti FIFO (*First In First Out*), RED (*Random Early Detection*), namun bagaimanakah cara kerja sistem antrean FIFO dan RED dalam jaringan kabel, dan sistem antrean manakah yang lebih baik dalam pengiriman paket datanya ketika antrean sedang padat hal ini jika dijelaskan hanya melalui teori akan sulit untuk di diterima. (Ramadhani Ulan sari. 2011).

Karena hal tersebut timbul suatu gagasan untuk menganalisis perbandingan dua antrean FIFO dan RED, analisa dilakukan dengan membandingkan data yang diterima, *packet loss*, *throughput* dan *delay* dari kedua antrean. Prinsip kerja FIFO adalah Semua paket diperlakukan sama (antrean tak berprioritas). seperti menempatkannya pada sebuah antrean, lalu dilayani dengan urutan yang sama ketika paket-paket tersebut memasuki antrean. FIFO tidak mempertimbangkan prioritas paket, *bandwidth*, dan alokasi *buffer* yang diperuntukkan paket tersebut. Ketika *buffer* pada *router* sudah penuh, maka paket yang datang selanjutnya akan di buang, sehingga metode ini dinamakan juga *tail-drop*. Sedangkan dipilih antrean RED karena sedikit sekali yang membahas tentang antrean ini, padahal antrean ini juga memiliki banyak keunggulan dengan prinsip kerja mekanisme *dropper* berdasarkan ke rata-rata panjang antrean.

Untuk membandingkan dua antrean tersebut selain mempelajarinya menggunakan teori dibutuhkan juga sebuah praktik dengan menggunakan simulasi. Simulasi tersebut ditujukan untuk mengetahui *delay*, *packet loss*, *throughput*, dan total data yang diterima. Hal ini dapat *diobservasi* menggunakan *network simulator 2* karena software ini memiliki banyak fasilitas seperti adanya *tool validasi* untuk menguji *validitas* permodelan. (Andi Bayu Wirawan & Eka Indarto.2004).

METODE PENELITIAN

Hal-hal yang termasuk metodologi pembuatan simulasi pada kedua antrean FIFO dan RED seperti berikut :

1. Software

Software yang digunakan dalam simulasi yaitu *software network simulator 2* (NS2) dengan linux ubuntu 9.04. Dipilih *network simulator 2* karena *software* ini memiliki banyak fasilitas seperti adanya *tool validasi* untuk menguji *validitas* permodelan, pembuatan simulasi pada *networks simulator 2* jauh lebih mudah dibandingkan menggunakan *software developer* seperti *Delphi* atau C++, *network simulator 2* bersifat *open source* di bawah GPL (*GNU Public License*) untuk itu digunakan untuk OS linux.

2. Hardware

Spesifikasi *hardware* yang digunakan untuk pembuatan simulasi ini intel Pentium *processor* (2.1 Ghz, 800 MHz, 800 MHz FSB), intel GMA 4500M, 14,0" HD LED LCD, 1 GB *memory*, 160 GB HDD.

3. Perancangan topologi

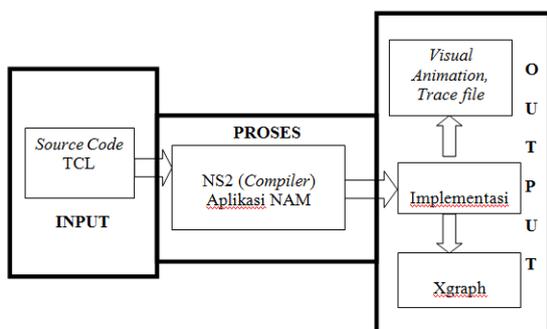
Simulasi ini dirancang menggunakan topologi star pada jaringan kabel, yang terdiri 5 buah *node*, n0, n1, n2, n3, n4 dengan hubungan komunikasi *duplex* (dua arah) terjadi pada n0-n3, n1-n3, n2-n3, dan komunikasi *simplex* (satu arah) pada n3-n4. Komunikasi *duplex* memakai *bandwidth* 10 mbps dengan *delay* 5ms, sedangkan komunikasi *simplex* memakai *bandwidth* 5mbps dengan *delay* 5ms. Dengan *agent* yang digunakan pada *node* n0 *agent* UDPO, pada *node* n1 *agent* UDP1, dan *node* n2 digunakan *agent* TCP. (Karina Auliasari, Dewi Awirat.2006).

HASIL ANALISIS DAN PEMBAHASAN

Hasil analisis dan pembahasan analisis perbandingan antrean FIFO dan RED dengan menggunakan Network Simulator 2 NS2 dijelaskan sebagai berikut :

1. Analisis Rancangan secara Blok Diagram

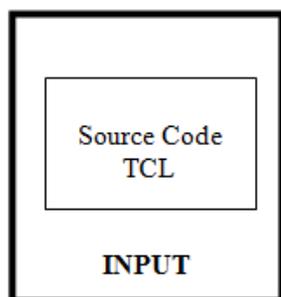
Berdasarkan proses simulasi dengan menggunakan *network simulator2* (NS2), dapat di bagi menjadi beberapa blok. Masing – masing blok tampak pada gambar berikut :



Gambar 1
Blok Diagram Proses Simulasi

Gambar 1 diatas merupakan gambar blok secara keseluruhan, dari proses simulasi. Terdiri dari blok input berupa *source code* TCL, blok proses *NS2 compiler*, dan blok output berupa *visual animation, trace file*, dan *xgraph*.

1.1 Analisa Blok Inputan



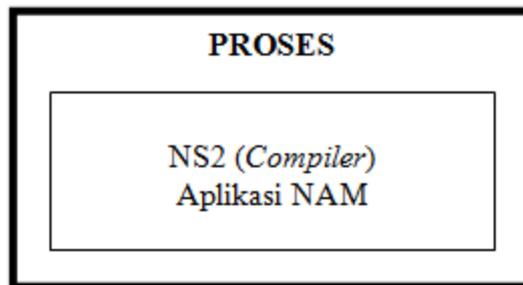
Gambar 2
Blok Input

Inputan pada proses simulasi ini adalah *source code* bahasa pemrograman TCL (*Tool command language*). *Source code* ini berisikan skenario simulasi, berupa rancangan bentuk topologi *star* pada jaringan kabel dengan menggunakan lima buah *node* yaitu n0, n1, n2, n3, n4. Dengan 3buah *node* sebagai *node* sumber, 1 *node* sebagai 1 *router*, dan 1 *node* lagi sebagai *sink*. Terjadi hubungan komunikasi *duplex* pada *node* n0-n3, n1-n3,

n2-n3 menggunakan *bandwidth* 10mbps, dan *delay* 5 ms. Dan komunikasi *simplex* terjadi pada *node* n3-n4 dengan menggunakan *bandwidth* 5mbps, dan *delay* 5 ms.

Agent yang digunakan pada ketiga *node* sumber adalah *agent* UDP0, UDP1, dan TCP. Untuk *node* n0, *agent* yang digunakan sebagai *agent* pengirim UDP0. Dan *agent* null sebagai *agent* penerima. Aplikasi yang berjalan diatas *transport agent* tersebut adalah aplikasi trafik CBR (*Constant Bit Rate*). Aplikasi ini penghasil paket dengan nilai bit konstan. Pada *node* n1 digunakan *agent* pengirim UDP1 dengan *agent* null sebagai penerima, dengan aplikasi yang berjalan diatasnya sama seperti *node* n0. Dan *node* sumber n2 digunakan *agent* pengirim TCP dengan *agent* penerima *TCPSink*. Aplikasi yang berjalan diatas *transport agent* tersebut adalah aplikasi trafik dengan fungsi FTP. Penghasil data yang akan dikirim oleh TCP.

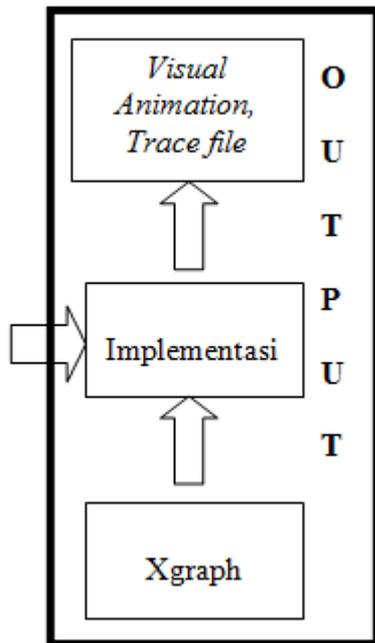
1.2 Analisa Blok Proses



Gambar 3 Blok Proses

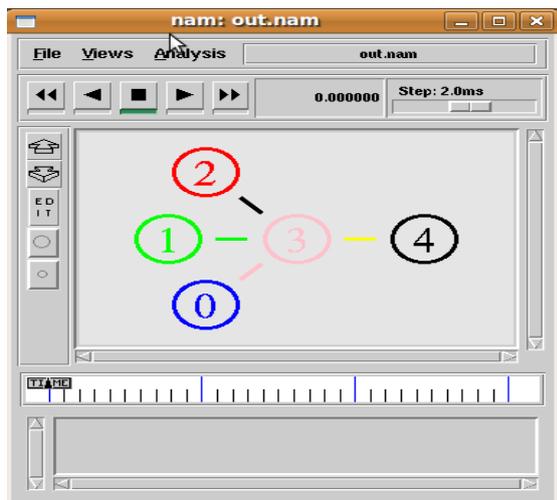
Blok proses dari simulasi ini adalah NS2 (*Compiler*) berupa aplikasi NAM. Dimana ketika eksekusi dilakukan, NS2 (*Compiler*) akan membaca skrip simulasi berupa *source code* bahasa TCL. Kemudian akan mengeluarkan output sesuai keluaran *source code* tersebut.

1.3 Analisa Blok Output



Gambar 4 Blok Output

Setelah *compiler* membaca skrip berupa *source code* bahasa TCL, kemudian output akan ditampilkan berupa *visual animation* (NAM), *trace file* dan *Xgraph*. contoh *output visual animation*, seperti gambar dibawah ini.



Gambar 5 Visual Animation

2. Analis Kinerja Sistem

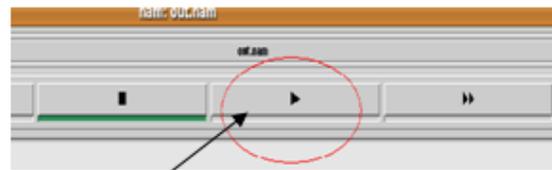
Analisis kinerja sistem pada simulasi akan di jabarkan menjadi dua seperti dibawah ini :

2.1 Analisa Berdasarkan Proses Visual Animation

perbandingan antrian FIFO (*First In Firt Out*) dan RED (*Random Early Detection*), pada suatu topologi *star* jaringan kabel berdasarkan pengoprasian *visual animation*, simulasi dilakukan selama 2 detik. Menggunakan 3 *node* sebagai *node* sumber, satu *node* sebagai *router* dan satu *node* lagi sebagai *sink*. *Node* sumber n0 mengirim paket data pada waktu 0.1 detik, *node* n1 pada waktu 0.5 detik dan *node* n2 pada waktu 0.6 detik.

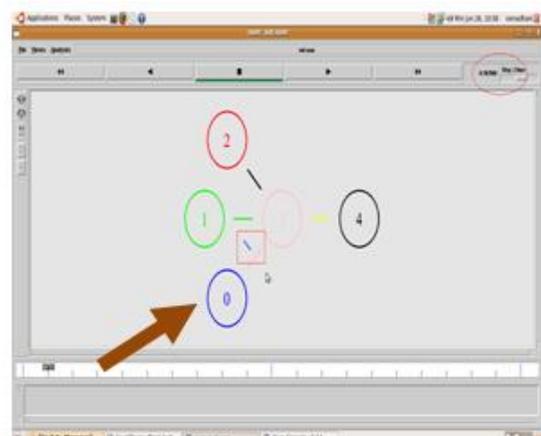
2.2 Proses Simulasi

Simulasi dijalankan dengan meng-klik *button play forward* seperti terlihat pada gambar 6.



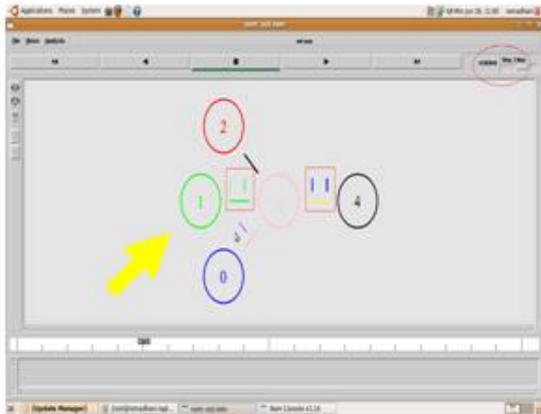
Gambar 6 Button Play Forward

Pada saat *button forward* diklik, simulasi akan terlihat seperti pada gambar dibawah ini :



Gambar 7 Pengiriman Paket Data Awal dari *node* n0

Keadaan *visual animation* pada saat *play forward* telah di klik, terlihat pada gambar 7 Panah berwarna coklat menunjukan *node* sumber n0 mulai mengirimkan paket data menuju *router* pada waktu 0.1 detik simulasi.



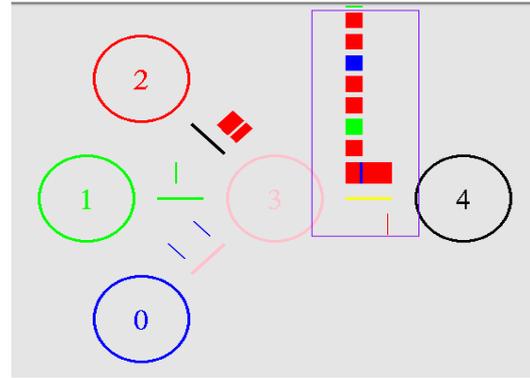
Gambar 8 Pengiriman Paket Data dari *node* n1.

Disusul pada waktu 0.5 detik simulasi pengiriman paket data berasal dari *node* sumber n1 yang menuju *router*. Hal ini terlihat pada gambar 8.



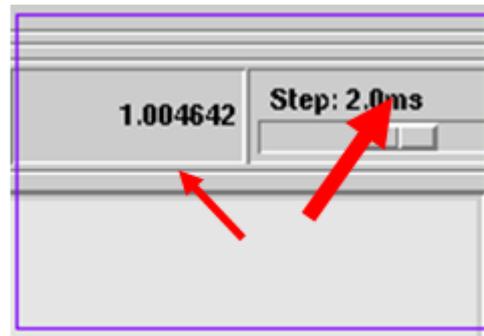
Gambar 9 Pengiriman Paket Data Dari *Node* n2.

Kemudian pada waktu 0.9 detik simulasi disusul pengiriman paket data berasal dari *node* sumber n2. Hal ini terlihat pada gambar 9 diatas.



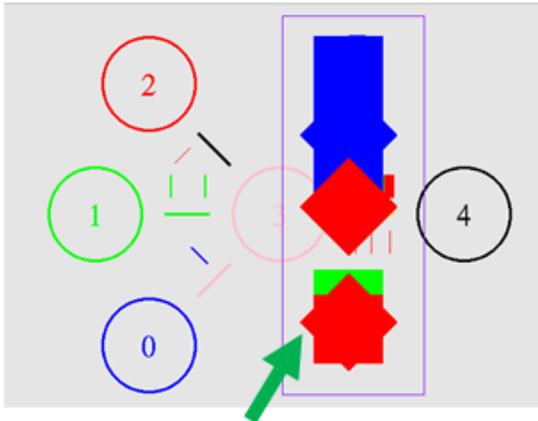
Gambar 10 Semua Paket Menuju *Sink*

Bentuk kotak-kotak kecil berwarna biru, hijau dan merah pada gambar 10 menunjukkan paket data dari ketiga *node* sumber n0, n1, dan n2 pada suatu *bottleneck link*. Kotak kecil berwarna biru merupakan paket data yang dikirim oleh *agent* UDP0, kotak kecil warna hijau oleh *agent* UDP1, dan warna merah oleh *agent* TCP.



Gambar 11 Waktu Semua Paket Data Telah Menuju *Sink*.

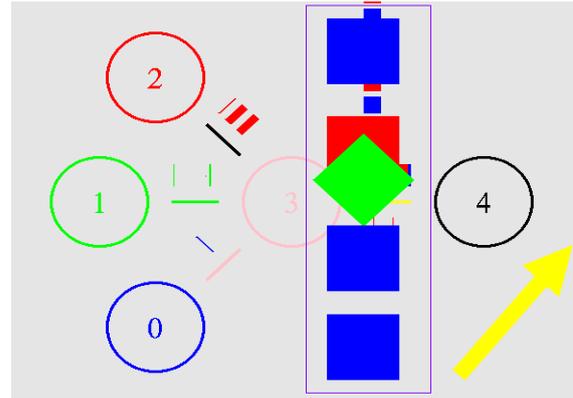
Gambar diatas menunjukkan pada waktu 1.004642 detik simulasi, semua *node* sumber telah mengirimkan paket data, dan paket data tersebut telah berada pada *bottleneck link* yaitu pada *node* n3 menuju *node* n4



Gambar 12
Packet Loss pada Antrean FIFO

Bentuk kotak-kotak yang ditunjukkan oleh panah berwarna hijau, berukuran besar pada gambar diatas, adalah *packet loss* pada antrean FIFO. Pada gambar 12 terlihat *packet loss* pada antrean FIFO lebih banyak terjadi pada paket data yang dikirim oleh *node* n0, disusul *node* n2 dan terakhir *node* n1.

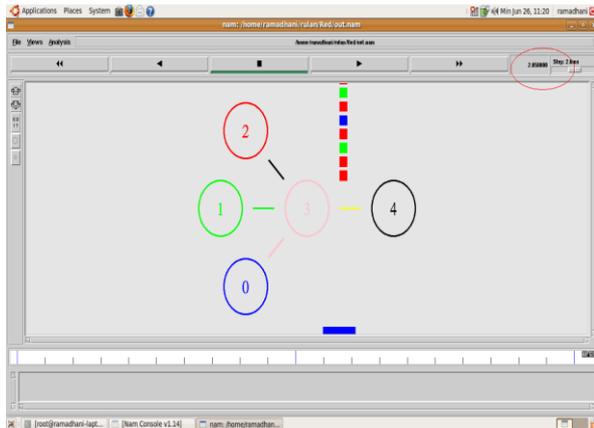
Packet loss pada antrean FIFO lebih banyak terjadi pada antrean yang dikirim oleh *agent* UDP0. Dikarenakan sistem antrean ini, hanya melayani satu pelanggan pada satu waktu. Dan pelanggan yang sudah menunggu paling lama, yang akan dilayani terlebih dahulu. Sistem antrean ini menyebabkan ketika *buffer* pada *router* penuh, maka paket data yang lebih lama menunggu yang akan terbuang. Paket data yang berasal dari *agent* UDP0, terlebih dahulu mencapai *router* dibandingkan *agent* UDP1 dan TCP. Karena waktu dimulainya pengiriman paket data pada *agent* ini 0.1 detik, sebelum *agent* lain mengirimkan paket data. Hal ini yang menyebabkan *packet loss* lebih banyak terjadi, pada paket data yang dikirim oleh *agent* UDP0. Selanjutnya pcket loss lebih banyak terjadi pada paket data yang dikirim oleh *agent* TCP. dikarenakan antrean ini tidak memiliki pengaturan terhadap trafik.



Gambar 13
Packet Loss pada Antrean RED

Pada gambar 13 terlihat *packet loss* pada antrean RED lebih banyak terjadi, pada paket data yang berasal dari *agent* UDP0. Hal ini dikarenakan, sistem pada antrean RED bekerja berdasarkan mekanisme *dropper* panjang antrean. Antrean RED ini juga sangat bergantung pada paket data yang dikirim oleh *agent* TCP. Untuk itu ketika paket data yang dikirim oleh *agent* UDP0 berada pada *router*, kemudian disusul paket data berasal dari *agent* UDP1. Paket yang berada pada antrean, yang paling belakang tersebut bisa melebihi beberapa *drop level* sehingga paket tersebut terbuang.

Berdasarkan simulasi pada *visual animation*, terjadi perbedaan *paket loss* pada kedua antrean FIFO dan RED. antrean RED memiliki jumlah *paket loss* lebih sedikit yang kirim dari *agent* TCP, dibandingkan pada antrean FIFO. Hal ini disebabkan, antrean RED mampu menghindari sinkronisasi trafik yang akan menyebabkan paket data TCP hilang dalam satu waktu. Antrean ini juga sangat bergantung pada paket data yang dikirim oleh *agent* TCP, sehingga mampu menghindari kemacetan, pada saat trafik sedang tinggi. Hanya saja selama simulasi *packet loss* pada antrean RED, yang dikirim oleh *agent* UDP lebih sering terjadi, dibandingkan antrean FIFO. Hal ini dikarenakan sistem antrean ini proses pengerjaannya berdasarkan panjang antrean, semakin panjang antrean yang melebihi *drop level*, maka antrean akan semakin sering terbuang.



Gambar 14
Akhir Simulasi

Visual animation pada gambar 14 menunjukkan proses akhir simulasi, pada waktu 2 detik.

3. Analisa Interaksi Kedua Sumber Pada Antrean FIFO Dan RED

Pada bagian ini akan dianalisa bagaimana interaksi kedua sumber menggunakan antrean FIFO dan RED pada suatu *bottleneck link*. Hasil interaksi kedua sumber seperti pada tabel-tabel berikut :

Tabel 1 Perbandingan nilai *packet loss*, *throughput*, dan *delay* dari *agent* UDP0 pada antrean FIFO dan RED.

Antrian	Flow Type	Paket Diterima Node n4	Packet Loss %	Rata-rata Throughput (kbps)	Rata-rata Delay (ms)
FIFO	UDP0	496	12 (2,4%)	440.911	7.1129
RED	UDP0	491	27 (5,4%)	436.05	10.201

Pada tabel 1 perbandingan nilai *packet loss*, *throughput* dan *delay* tampak antrean FIFO memiliki nilai paket data yang diterima *node n4* bersumber dari *agent* UDP0, lebih kecil dibandingkan dengan antrean RED. Nilai yang diterima *node n4* pada antrean FIFO yaitu sebanyak 496 paket data, dan untuk antrean RED yaitu sebanyak 491 paket data. Hal ini dikarenakan antrean FIFO memperlakukan semua paket sama dengan menempatkannya pada sebuah antrean, lalu dilayani sesuai

urutan yang sama ketika paket-paket tersebut masuk antrean. antrean ini selalu menjaga paket data berada pada *q_referensi* sehingga jaringan tidak telalu terbebani oleh paket-paket.

Total rata-rata nilai *throughput* yang dihasilkan antrean FIFO, bersumber dari *agent* UDP0, yaitu 440.911 kbps lebih besar dibandingkan dengan antrean RED. Nilai rata-rata *throughput* antrean RED yaitu 436.05 kbps. Hal ini dikarenakan fungsi *agent* UDP0 menyediakan penyampaian paket tepat pada waktunya, namun tidak menjamin sampainya paket pada tujuan. Antrean ini juga memperlakukan semua paket sama dengan menempatkannya pada sebuah antrean, lalu dilayani sesuai urutan yang sama ketika paket-paket tersebut masuk antrean. Hal ini juga mempengaruhi nilai rata-rata *throughput*.

Pada tabel Tampak total nilai rata-rata *delay* yang dihasilkan antrean FIFO, lebih kecil dibandingkan antrean RED. Yaitu dengan nilai rata-rata 7.1129 kbps pada antrean FIFO dan 10.201 kbps pada antrean RED. Hal ini juga dikarenakan antrean FIFO melayani paket sesuai urutan yang yang sama ketika paket-paket tersebut masuk antrean. Paket yang terlebih dulu datang, akan diproses terlebih dahulu, menyebabkan antrean pada antrean FIFO tidak telalu menumpuk, dan tidak memerlukan telalu banyak *delay*.

Tabel 2 Perbandingan nilai *packet loss*, *throughput*, dan *delay* pada *agent* CBR1 kedua antrean FIFO dan RED.

Antrian	Flow Type	Paket Diterima Node n4	Packet Loss %	Rata-rata Throughput	Rata-rata Delay
FIFO	UDP1	394	0,12%	444.37	7.63527
RED	UDP1	387	0,27%	435.265	7.43381

Pada tabel 2 tampak total paket diterima *node n4*, pada antrean RED yaitu sebanyak 387 dan antrean FIFO sebanyak 394 paket data. paket data yang diterima *node n4* pada antrean RED lebih sedikit dibandingkan pada antrean FIFO. Hal ini di karenakan antrean RED, sangat

bergantung pada banyaknya aliran TCP, untuk itu ketika paket data berasal dari *agent* selain TCP, akan terjadi lebih banyak *packet loss*. Hal ini juga yang mempengaruhi, jumlah paket data yang diterima oleh node n4.

Nilai rata-rata *throughput* yang dihasilkan antrean RED berasal dari *agent* UDP1, lebih kecil dibandingkan dengan antrean FIFO. Pada antrean FIFO nilai rata-rata *throughput* 444.37 kbps dan 435.265 kbps pada antrean RED. Hal ini dikarenakan *agent* ini menyediakan penyampaian paket tepat pada waktunya, namun tidak menjamin sampainya paket pada tujuan. Sedangkan antrean yang digunakan adalah antrean RED yang sangat bergantung pada banyak aliran data dari TCP dan panjang antrean.

Pada tabel 2 tampak total nilai rata-rata *delay* pada antrean FIFO lebih besar yaitu, 7.63527 ms dibandingkan antrean RED dengan rata-rata nilai 7.43381 ms. Hal ini dikarenakan pada saat paket data dikirim oleh *agent* UDP1, terjadi kepadatan antrean yang menyebabkan timbulnya *waiting time*. Sehingga proses membutuhkan waktu lama dalam penyelesaiannya. Hal ini yang menyebabkan *delay* lebih besar.

Tabel 3 Perbandingan nilai packet loss, throughput, dan delay pada agent TCP kedua antrean FIFO dan RED.

Antrian	Flow Type	Paket Diterima	Packet Loss %	Rata-rata Throughput (Kbps)	Rata-rata Delay (Ms)
FIFO	TCP	405	0.12%	3074.54	10.328
RED	TCP	412	0,27%	3127.4	10.201

Total nilai data yang diterima *node* n4, berasal dari *agent* TCP pada antrean RED. Lebih besar dibandingkan pada antrean FIFO. Dengan total nilai 405 paket data untuk antrean FIFO, dan 412 paket data untuk antrean RED. Hal ini dikarenakan, antrean RED Mampu mendeteksi sumber *congestion* sebelum terjadi. Sistem antrean ini juga sangat bergantung pada banyaknya aliran TCP. Sehingga mampu menghindari sinkronisasi trafik dimana paket berasal dari TCP hilang dalam satu waktu.

Total nilai rata-rata *throughput*, pada antrean RED yaitu 3127.4 kbps dan 3074.54 kbps. Nilai *throughput* untuk antrean RED ini, lebih besar dibandingkan antrean FIFO. Hal ini dikarenakan antrean RED mampu melakukan *dropping* terhadap paket yang terjadi *congestion*, mampu menghindari kemacetan pada saat trafik sedang tinggi dan antrean ini amat bergantung pada aliran yang berasal dari *agent* TCP. Hal ini juga menyebabkan paket yang berada di jaringan lebih sedikit, sehingga mempengaruhi jumlah *throughput* yang dihasilkan pada antrean tersebut.

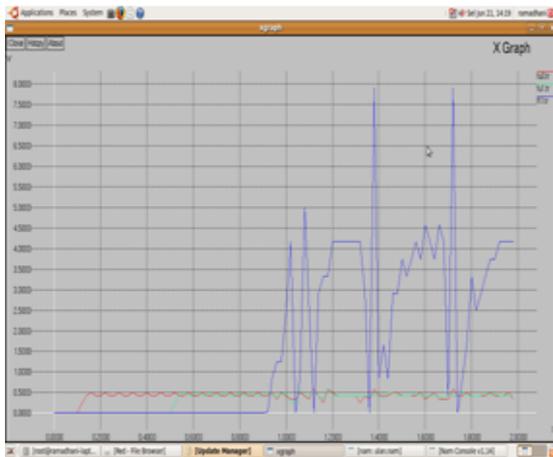
Nilai total rata-rata delay pada antrean RED yaitu 10.201 ms, sedangkan antrean FIFO 10.328 ms. Nilai *delay* yang dihasilkan antrean RED, lebih kecil dibandingkan antrean FIFO. Hal ini dikarenakan antrean RED mampu menghindari kemacetan pada saat trafik sedang tinggi.

4. Analisa Berdasarkan Xgraph (Throughput) Kedua Antrean

Pada bagian ini akan dianalisa bentuk *throughput* pada antrean FIFO berdasarkan gambar *xgraph* dibawah ini :



Gambar 15 *throughput* antrean FIFO



Gambar 16
throughput antrean RED

Gambar 15 dan 16 menampilkan *throughput* pada antrean FIFO dan RED. Garis berwarna merah merupakan *agent* UDP0 dengan aplikasi trafik yang berjalan di atasnya adalah CBR (*Constant Bit Rate*). tampak pada gambar *throughput* mulai meningkat pada waktu 0.1s. Hal ini terjadi dikarenakan *Agent* UDP0 mengirimkan paket data. Kemudian disusul garis berwarna hijau, pada waktu 0.5 detik dengan *agent* UDP1 dengan aplikasi CBR. Dan garis berwarna merah dengan *agent* TCP meningkat pada waktu 0.9 detik simulasi.

Pada gambar xgraph kedua antrean terlihat *fluktuasi* tak berarti pada kedua *agent* UDP0 dan UDP1. Hal ini terlihat pada garis merah dan hijau, gambar 3.15 dan 3.16 yang berkisar pada nilai 85000 Mbps. hal ini dikarenakan aplikasi yang digunakan kedua *agent* ini adalah CBR menghasilkan nilai paket dengan nilai bit konstan.

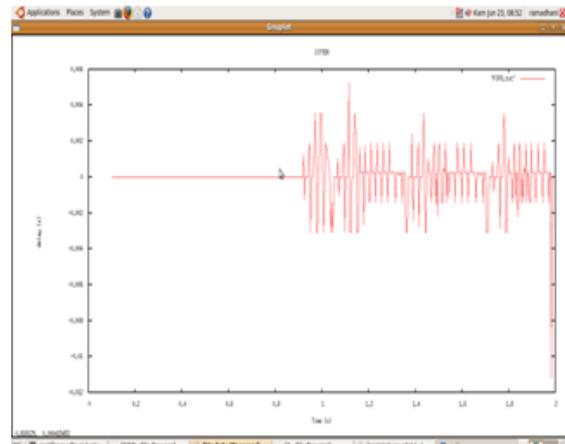
Xgraph pada kedua antrean, baru mengalami *fluktuasi* berarti pada *agent* TCP, pada gambar terlihat pada garis berwarna biru. Hal ini dikarenakan aplikasi yang digunakan *agent* TCP menggunakan aplikasi FTP yaitu menghasilkan data yang akan dikirim oleh TCP, nilai paket yang dihasilkan tidak dalam nilai bit konstan.

Pada gambar xgraph kedua antrean FIFO dan RED, terlihat antrean RED lebih mampu mengontrol trafik yang sedang tinggi, di bandingkan antrean FIFO. Hal ini dikarenakan

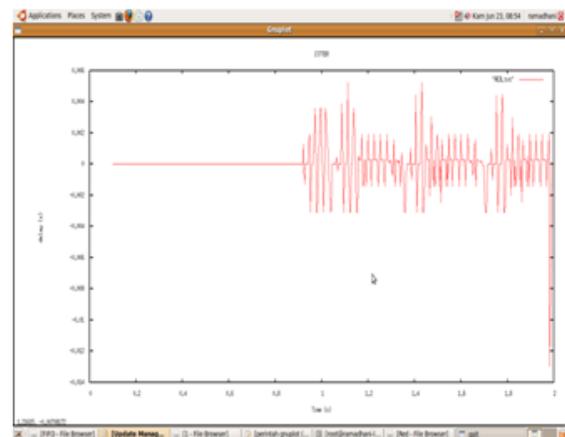
antrean ini mampu mendeteksi, sebelum terjadinya congestion.

5. Analisa Berdasarkan Jitter

Pada bagian ini akan dianalisa berdasarkan *jitter* pada kedua antrean FIFO (*First In First Out*) dan RED (*Random Early Detection*) seperti yang terlihat pada gambar dibawah ini.



Gambar 17 Jitter
pada antrean FIFO



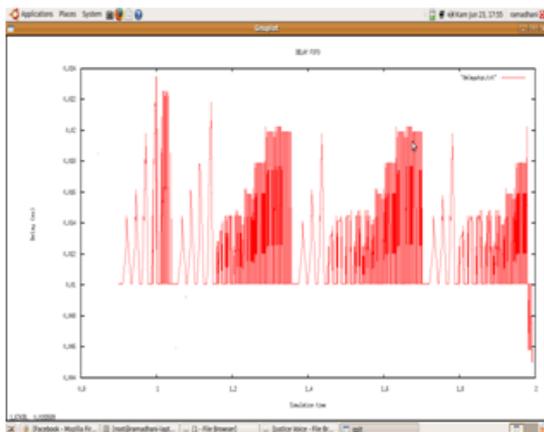
Gambar 18 Jitter
pada antrean RED

Berdasarkan gambar 17 dan gambar 18 Tampak pada kedua antrean FIFO dan RED, baru mengalami *Jitter* pada waktu 0.9 detik. Hal ini dikarenakan pada waktu 0.9 detik semua node sumber telah mengirimkan paket data. Hal inilah yang menyebabkan terjadinya variasi *delay* (*jitter*).

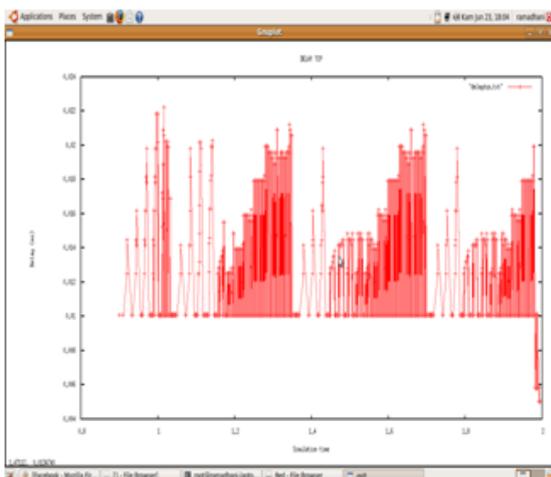
Jitter pada antrean RED lebih banyak meningkat hingga mencapai nilai 0.005ms dibandingkan antrean FIFO. Hal ini disebabkan antrean *RED*, berkerja berdasarkan mekanisme panjang antrean. Semakin panjang antrean semakin banyak mengalami variasi *delay*, yang akan mempengaruhi tinggi dan rendahnya *jitter*.

6. Analisa Berdasarkan Delay

Pada bagian ini akan dianalisa *Delay* dari masing-masing antrean. *Delay* adalah tenggang waktu yang dibutuhkan mulai mengirim data sampai dengan data diterima dari masing-masing *node*. Hal ini terlihat pada gambar :



Gambar 19
Delay antrean FIFO



Gambar 20 *Delay* antrean RED

Pada Gambar 19 dan gambar 20, terlihat grafik *delay* pada kedua antrean mulai meningkat pada waktu 0.1 detik. Hal ini disebabkan, waktu tersebut. Salah satu dari ketiga *node* sumber yaitu n0, mulai mengirimkan paket data.

Gambar *delay* pada kedua antrean terlihat mulai terjadi kerapatan pada waktu 0.9 detik. Hal ini dikarenakan, pada waktu tersebut terjadi kepadatan antrean yang membutuhkan lebih banyak *delay* dalam proses pengiriman paket data.

Pada gambar *delay* kedua antrean FIFO dan RED. Antrean FIFO terlihat lebih rapat dibandingkan *delay*, pada antrean RED. Hal ini disebabkan antrean RED mampu menghindari *congestion* yang menyebabkan, antrean ini mampu menghindari kemacetan pada saat trafik sedang tinggi. Sedangkan *delay* pada antrean FIFO lebih rapat dikarenakan, sistem antrean ini tidak memiliki pengaturan terhadap trafik. Sehingga paket berukuran kecil harus berkompetisi dengan paket data yang berukuran besar, *delay (congestion)* pada paket akan mempengaruhi paket lain. [3]

KESIMPULAN

Dapat disimpulkan hasil simulasi diketahui antrean RED terlihat lebih mampu menangani antrean ketika sedang padat. Sedangkan antrean FIFO lebih unggul ketika menangani antrean berasal dari agent UDP ketika antrean tidak terlalu padat.

DAFTAR PUSTAKA

- [1] Andi Bayu Wirawan & Eka Indarto.2004. *Mudah Membangun Simulasi dengan Network Simulator*, ANDI, Yogyakarta,
- [2] Karina Auliasari, Dewi Awirat.2006. *“Mengukur Kualitas Layanan Jaringan Komputer”*, Fakultas Teknik UGM, Yogyakarta.
- [3] Ramadhani Ulan sari. 2011. *Analisis Perbandingan Antrean menggunakan ns2.2002*