

# ANALISIS KINERJA TCP BIC UNTUK PENCEGAHAN KONGESTI PADA JARINGAN LTE DENGAN MENGGUNAKAN *NETWORK SIMULATOR 2.33*

M. Fajri Fitrianto <sup>\*)</sup>, Sukiswo, and Imam Santoso

Jurusan Teknik Elektro, Universitas Diponegoro Semarang  
Jl. Prof. Sudharto, SH, kampus UNDIP Tembalang , Semarang 50275, Indonesia

<sup>\*)</sup> Email: fazz\_antho@yahoo.com

## Abstrak

LTE merupakan teknologi telekomunikasi yang dikembangkan oleh 3GPP dari GSM. LTE berbasis flat-IP, yaitu semua informasi dilewatkan melalui teknologi IP dan berkecepatan akses yang sangat tinggi. Perkembangan ini diikuti peningkatan trafik yang menyebabkan penurunan QoS sehingga perlu dilakukan pengontrolan menggunakan BIC pada TCP dimana BIC meningkatkan window dengan algoritma additive increase dan binary search sesuai dengan ketersediaan bandwidth serta akan menurunkan window saat kongesti menjadi kali ukuran window sebelum terjadi kongesti. Berdasarkan hal tersebut maka pada penelitian ini dilakukan simulasi jaringan LTE dengan agen pembawa TCP dengan algoritma BIC dengan menggunakan NS 2.33. Kemudian dilakukan analisis throughput dan fairness dengan jumlah user 5 dan dalam satu jaringan LTE yang sama dengan variasi (Beta) yang digunakan adalah 0.6 hingga 0.9, serta analisa terhadap delay end to end dengan 6 buah user yang aktif menggunakan variasi topologi 1 eNodeB, 2 eNodeB dan 2 gateway. Dari hasil simulasi didapatkan throughput rata-rata BIC 0.9 memiliki throughput paling tinggi namun memiliki fairness yang rendah, sedangkan BIC 0.6 memiliki throughput paling rendah tetapi memiliki indeks fairness paling tinggi. Pada analisis delay end to end didapatkan delay pada topologi 1 eNodeB 4.8 ms, topologi 2 eNodeB 8.9 ms dan topologi 2 gateway 11.8 ms.

*Kata kunci : LTE, BIC, kontrol kongesti, throughput, fairness, delay end to end.*

## Abstract

LTE is telecommunication technology which was developed by 3GPP based on GSM. LTE is All-IP based technology and has very high access rate, which lead the traffic rising and QoS degradation. It's needed to control QoS using BIC algorithm on TCP. BIC will increase the window using additive increase and binary search algorithm according to the bandwidth availability, BIC will decrease the window size when congested to times the last window size before congestion. In this research LTE simulated by NS 2.33 using TCP BIC as transport agent. Throughput, fairness and end to end delay are being analyzed. To analyze throughput and fairness 5 active user in one network system with decrease factor ( ) 0.6 to 0.9. To analyze end to end delay, 3 topology (1 eNodeB, 2 eNodeB and 2 gateway). As the result, throughput BIC 0.9 has the highest throughput than the other BIC variations, but has the lowest fairness index. In the other hand, BIC 0.6 has the lowest throughput, but has the highest fairness index. In delay end to end analyze, resulting in 1 eNodeB topology is 4.8 ms, in 2 eNodeB topology is 8.9 ms and in 2 gateway topology is 11.8 ms.

*Keywords: LTE, BIC, Congestion Control, throughput, fairness, delay end to end*

## 1. Pendahuluan

*Long Term Evolution (LTE)*<sup>[1]</sup> merupakan suatu evolusi teknologi dalam dunia jaringan *mobile* yang menawarkan berbagai keuntungan, terutama dalam hal kecepatan akses. LTE memberikan tingkat kapasitas *downlink* minimal 100 Mbps dengan menggunakan OFDMA (*Orthogonal Frequency Division Multiple Access*) dan *uplink* paling sedikit 50 Mbps dengan menggunakan SC-FDMA (*Single Carrier Frequency Division Multiple Access*) dan *round trip* kurang dari 10 ms. Seiring

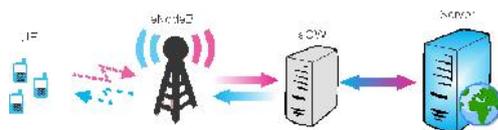
berkembangnya teknologi, jumlah trafik juga ikut meningkat. Peningkatan trafik tersebut diakibatkan oleh bertambahnya pengguna dan bertambahnya kebutuhan akses data. Hal tersebut akan menimbulkan kongesti pada jaringan, yang membuat jaringan tidak sanggup lagi melewatkan trafik. Oleh karena itu diperlukan suatu teknik dengan algoritma tertentu untuk menangani kongesti. Pada *Layer 4 OSI (Open System Interconnection)*, yaitu TCP (*Transport Control Protocol*) menerapkan algoritma tertentu untuk menangani kongesti diantaranya *Tahoe* ataupun *Reno*. Algoritma tersebut kini

telah dikembangkan menjadi lebih dinamis yaitu algoritma kontrol kongesti penambahan biner atau yang lebih dikenal dengan BIC (*Binary Increase Congestion Control*) yang merupakan pengembangan dari algoritma *Reno* yang akan mengurangi ataupun mencegah terjadinya kongesti pada jaringan yaitu dengan mengatur ukuran window dengan algoritma yang khas yang dimiliki oleh BIC. Dengan demikian kinerja dari BIC pada jaringan LTE perlu dianalisis dengan menggunakan perangkat lunak *Network Simulator (NS) 2.33*.

Dalam penelitian sebelumnya, telah dibahas mengenai algoritma kongesti pada TCP seperti *Reno*, *New Reno*, *Vegas*, *Westwood* dan *Winsock* pada jaringan 4G<sup>[4]</sup>, namun tidak dibahas mengenai BIC. Dalam penelitian lain, membahas mengenai BIC pada jaringan *Long Fat Network*<sup>[5]</sup>, namun tidak dibahas bagaimana kinerjanya dalam 4G. Dengan demikian, maka diperlukan analisa tentang TCP pada jaringan LTE dengan kontrol kongesti penambahan biner (*Binary Increase Congestion Control*) menggunakan *Network Simulator 2.33*.

Arsitektur jaringan LTE<sup>[2]</sup> dirancang untuk tujuan mendukung trafik *packet switching* dengan mobilitas tinggi, *quality of service (QoS)*, dan *latency* yang kecil. *Packet switching* ini memperbolehkan semua layanan termasuk layanan *voice* menggunakan koneksi paket. Oleh karena itu pada arsitektur jaringan LTE dirancang sesederhana mungkin, yaitu hanya terdiri dari dua *node* yaitu *eNodeB* dan *mobility management entity/gateway (MME/GW)*. Hal ini sangat berbeda dengan arsitektur teknologi GSM dan UMTS yang memiliki struktur lebih kompleks dengan adanya *radio network controller (RNC)*. Beberapa keuntungan yang dapat diperoleh dengan hanya menggunakan *single node* pada jaringan akses adalah pengurangan *latency* dan distribusi trafik RNC untuk beberapa *eNodeB*. Pengeliminasian RNC pada jaringan akses memungkinkan karena LTE tidak mendukung *soft handover*.

*Radio Interface*<sup>[11]</sup> utama pada jaringan LTE terdiri dari *User Equipment, Enhanced Node B, Gateway, dan Server. Evolution of Packet Core Network (EPC)* akan mendukung layanan *bandwidth* yang tinggi dan *message rate* yang baik.



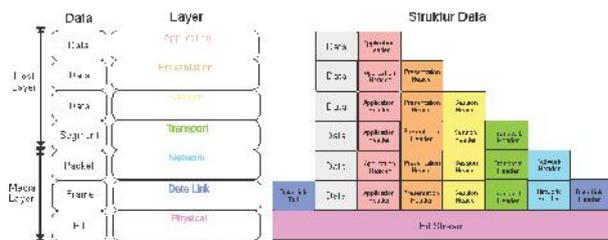
Gambar 1 Interface sederhana Jaringan LTE

Pada LTE, teknik akses<sup>[3]</sup> yang digunakan pada transmisi dalam arah *downlink* dan *uplink* berbeda. Arah *downlink* adalah arah komunikasi dari *eNodeB* ke UE, sementara arah *uplink* adalah arah dari UE menuju *eNodeB* seperti yang ditunjukkan pada Gambar 2.4. Pada arah *downlink*

teknik akses yang digunakan adalah *orthogonal frequency division modulation access (OFDMA)* dan pada arah *uplink* teknik akses yang digunakan adalah *single carrier frequency division multiple access (SC-FDMA)*. OFDMA adalah variasi dari *orthogonal frequency division modulation (OFDM)*.

FDD merupakan teknik *duplex* yang menggunakan dua frekuensi yang berbeda untuk melakukan komunikasi dalam dua arah. Dengan menggunakan FDD dimungkinkan untuk mengirim dan menerima sinyal secara simultan dengan frekuensi yang berbeda-beda. Dengan teknik ini dibutuhkan *guard frequency* untuk memisahkan frekuensi pengiriman dan penerimaan secara simultan, serta dibutuhkan proses *filtering* frekuensi yang harus akurat. Sedangkan TDD menggunakan frekuensi tunggal dan frekuensi tersebut digunakan oleh semua kanal untuk melakukan pengiriman dan penerimaan *message*.

Model OSI memiliki 7 lapis layer seperti pada gambar berikut:



Gambar 2 Media Layer dan Host Layer OSI Model

Dari gambar di atas, data asal akan mendapatkan penambahan *header* pada tiap-tiap layer hingga sampai pada *physical layer* data akan dikirimkan dalam bentuk *bit stream*.

Layer pada LTE tersusun atas 3 layer, seperti pada gambar berikut:



Gambar 3 Layer LTE

Fungsi masing-masing layer pada layer LTE<sup>[15]</sup> seperti yang terlihat pada Gambar 3 adalah:

1) *Layer 1 (PHY)*

Physical *layer* melakukan deteksi kesalahan pada saluran transportasi dan mengindikasikan ke *layer* di atasnya, serta menginformasikan pula modulasi dan demodulasi, sinkronisasi waktu dan frekuensi, pengolahan radio frekuensi dan MIMO *processing* jika digunakan.

2) *Layer 2 (MAC dan RLC)*

MAC akan melakukan koreksi kesalahan melalui HARQ (*Hybrid Automatic Repeat Request*) dan penjadwalan *user*. RLC akan mengoreksi kesalahan melalui ARQ (*Automatic Repeat Request*), Retransmisi, deteksi dan pemulihan kesalahan protokol serta mengirim PDU (*Packet Data Unit*) ke *layer* atasnya.

3) *Layer 3 (RRC)*

RRC memiliki fungsi utama melakukan pembentukan, pemeliharaan dan pelepasan koneksi antara UE dan E-UTRAN dalam eNodeB, serta melakukan manajemen QoS dan mobilitas.

*Transmission Control Protocol (TCP)* adalah suatu protokol yang berada di lapisan *transport* yang berorientasi sambungan (*connection oriented*) dan dapat diandalkan (*reliable*). TCP dispesifikasikan dalam RFC 793, karakteristik TCP antara lain:

1) *Connection oriented*

Sebelum *message* dapat ditransmisikan antara dua *host*, dua proses yang berjalan pada lapisan aplikasi harus melakukan negosiasi untuk membuat sesi koneksi terlebih dahulu. Koneksi TCP ditutup dengan menggunakan proses terminasi koneksi TCP (*TCP connection termination*).

2) *Full Duplex*

Untuk setiap *host* TCP, koneksi yang terjadi antara dua *host* terdiri atas dua buah jalur, yakni jalur keluar dan jalur masuk, maka *message* pun dapat secara simultan diterima dan dikirim. *Header TCP* berisi nomor urut (*TCP sequence number*) dari *message* yang ditransmisikan dan sebuah *acknowledgment* dari *message* yang masuk.

3) *Reliable*

*Message* yang dikirimkan ke sebuah koneksi TCP akan diurutkan dengan sebuah nomor urut paket dan akan mengharapkan paket *positive acknowledgment* dari penerima. Jika tidak ada *acknowledgment* dari penerima, maka segmen TCP (*protocol message unit* dalam protokol TCP) akan ditransmisikan ulang. Pada pihak penerima, segmen-segmen duplikat akan diabaikan dan segmen-segmen yang *message* yang tidak sesuai dengan urutannya akan diletakkan di belakang untuk mengurutkan segmen-segmen TCP. Untuk menjamin integritas setiap segmen TCP, TCP mengimplementasikan penghitungan *TCP Checksum*.

4) *Byte Stream*

TCP melihat *message* yang dikirimkan dan diterima melalui dua jalur masuk dan jalur keluar sebagai

sebuah *byte stream* yang berdekatan. Nomor urut TCP dan nomor *acknowledgment* dalam setiap *header TCP* didefinisikan juga dalam bentuk *byte*. Meski demikian, TCP tidak mengetahui batasan pesan-pesan di dalam *byte stream* TCP tersebut.

5) *Flow Control*

Untuk mencegah *message* terlalu banyak dikirimkan pada satu waktu, yang akhirnya membuat "macet" jaringan *internetwork IP*, TCP mengimplementasikan layanan *flow control* yang dimiliki oleh pihak pengirim yang secara terus menerus memantau dan membatasi jumlah *message* yang dikirimkan pada satu waktu. Untuk mencegah pihak penerima untuk memperoleh *message* yang tidak dapat disangganya (*buffer*), TCP juga mengimplementasikan *flow control* dalam pihak penerima, yang mengindikasikan jumlah *buffer* yang masih tersedia dalam pihak penerima.

6) *One to one send*

hal ini karena memang TCP harus membuat sebuah sirkuit logis antara dua buah protokol lapisan aplikasi agar saling dapat berkomunikasi. TCP tidak menyediakan layanan pengiriman *message* secara *one-to-many*. Sehingga pengiriman dilakukan sesuai permintaan dari masing-masing *user* yang meminta layanan TCP.

TCP BIC<sup>[5]</sup> merupakan varian TCP yang meningkatkan *congestion window*-nya sebanyak 1 setiap 1xRTT (*Round-trip Time*) dan akan berkurang setengah pada *loss event*. BIC biasanya digunakan dalam *Long Distance Network*. BIC memiliki fitur<sup>[5]</sup> yang khas yaitu *Scalability*, *RTT Fairness*, *TCP Friendliness* dan *Convergence*. *Scalability* dari BIC dapat menskalakan *bandwidth* yang ada sesuai dengan kebutuhan dan kondisi jaringan. Hal ini salah satu keuntungan dari penerapan kontrol terhadap kongesti, sehingga *bandwidth* akan termanfaatkan dengan baik. Untuk *congestion window* yang besar, BIC memiliki RTT yang proporsional. Sehingga *fairness index* dapat dinyatakan dengan persamaan berikut:

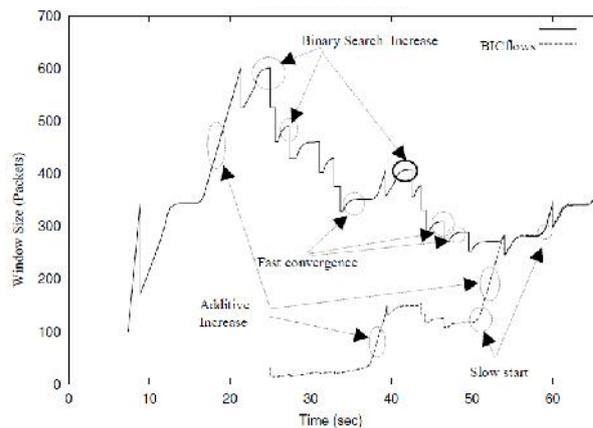
$$\text{Fairness Index} = \frac{w_2/RTT_2}{w_1/RTT_1} = \left( \frac{RTT_1}{RTT_2} \right)^{\frac{1}{1-d}}$$

Dengan  $W_1$  adalah *window* pengguna pertama,  $W_2$  adalah *window* pengguna kedua sedangkan  $RTT_1$  adalah RTT rata-rata dari pengguna pertama,  $RTT_2$  adalah RTT rata-rata pengguna kedua dan  $d$  adalah konstanta *protocol congestion avoidance* yang nilainya antara 0,5 hingga 1. Besarnya nilai  $d$  untuk AIMD<sup>[5]</sup> adalah 0,5 HSTCP<sup>[5]</sup> adalah 0,82 dan STCP<sup>[5]</sup> adalah 1.

Dalam BIC, *friendliness* selalu dijadikan acuan dalam pengontrolan terhadap kongesti, yaitu keseimbangan penanganan antar *user*. BIC mempunyai *bandwidth fairness* yang lebih baik untuk berbagai skala waktu dan memiliki *convergence* yang cepat ke ukuran *window* yang lebih baik.

Selain karakteristik tersebut, BIC memiliki algoritma yang khas dalam penanganan kongesti, yaitu dengan menggunakan *Binary Search Increase*, dimana titik awal untuk melakukan pencarian biner ini dimulai dengan menentukan *window* minimum saat ini (*current minimum window size*  $W_{min}$ ) dan ukuran *window* maksimum (*maximum window size*  $W_{max}$ ). Umumnya,  $W_{max}$  merupakan ukuran *window* sesaat sebelum pemulihan cepat (*fast recovery*) terakhir atau dengan kata lain saat terjadinya paket hilang yang terakhir.  $W_{min}$  merupakan ukuran *window* setelah terjadinya *fast recovery*. Algoritma tersebut akan terus berulang menghitung titik tengah antara  $W_{max}$  dan  $W_{min}$ , titik tersebut akan dianggap sebagai  $W_{max}$  baru apabila saat itu paket hilang, dan akan dianggap sebagai  $W_{min}$  baru jika tidak demikian. Proses tersebut akan berlanjut hingga perbedaan antara  $W_{max}$  dan  $W_{min}$  berada dalam batas (*threshold*) yang ditentukan, yang biasa disebut sebagai penambahan minimum (*minimum increment*  $S_{min}$ ). Saat jarak ke titik tengah dari titik minimum sekarang terlalu tinggi, ukuran *window* akan bertambah secara langsung menjadi seukuran dengan titik tengah tersebut, hal ini mungkin akan memberikan beban lebih terhadap jaringan. Untuk menghindari hal tersebut, BIC menetapkan ambang (*threshold*) yang dinamakan dengan nilai penambahan maksimum (*maximum increment*,  $S_{max}$ ). jika perbedaan antara titik minimum dan titik tengah lebih dari *threshold*, *window* akan bertambah sebesar  $S_{max}$  sampai jarak semakin kecil dan kurang dari  $S_{max}$ . Saat ukuran *window* sekarang bertambah melebihi ukuran *window* maksimum saat ini ( $W_{max}$ ), algoritma pencarian biner akan mencari nilai maksimum *window* yang baru dengan menjalankan strategi *slow start*<sup>[4]</sup>, sehingga saat nilai *window* sekarang lebih tinggi dari  $W_{max}$ , tapi lebih kecil dari  $W_{max}+S_{max}$ , congestion *window* akan bertambah untuk setiap RTT sebagai berikut  $W_{max}+S_{min}$ ,  $W_{max}+2S_{min}$ ,  $W_{max}+4S_{min}$ .....,  $W_{max}+S_{max}$ . *Convergence*<sup>[4]</sup> BIC sangat cepat dalam memperbaiki ukuran *window*. Dengan menggunakan  $\log(d)-\log(S_{min})$  untuk memperoleh *window* maksimum setelah terjadi pengurangan *window* sebanyak  $d$ .

Secara grafis, keunikan algoritma yang digunakan dalam penentuan ukuran *window* TCP BIC terlihat seperti gambar berikut:



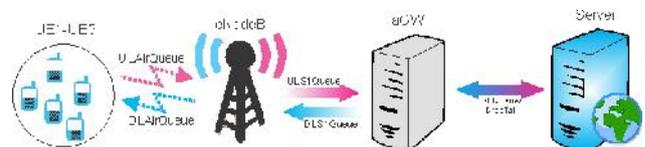
Gambar 4 Karakteristik Window BIC

## 2. Metode

Pada tugas akhir ini akan membuat simulasi jaringan LTE yang akan diimplementasikan di perangkat lunak *Network Simulator 2.33*. Seluruh simulasi yang dibuat digunakan untuk menganalisis tiga pembahasan utama yaitu *throughput*, *fairness*, dan juga *delay end-to-end*. *Node-node* akan difungsikan sebagai UE (*user*), eNodeB, gateway dan *server*.

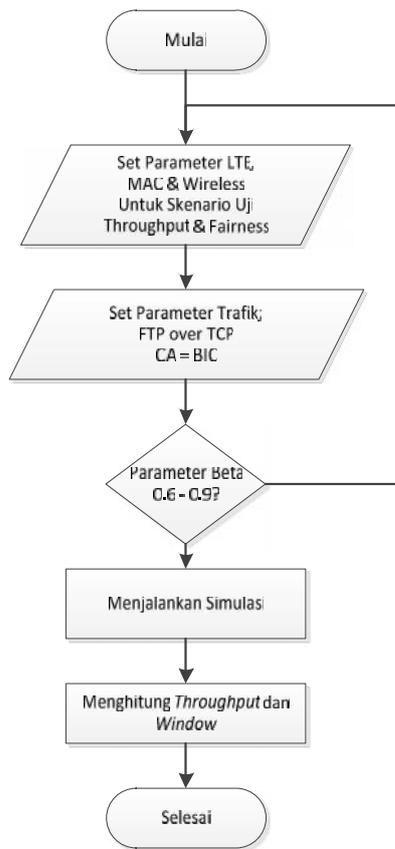
### Skenario Pengujian *Throughput* dan *fairness*

Pada skenario ini topologi yang digunakan tersusun atas 5 UE, 1 eNodeB, 1 aGW dan 1 *Server*.



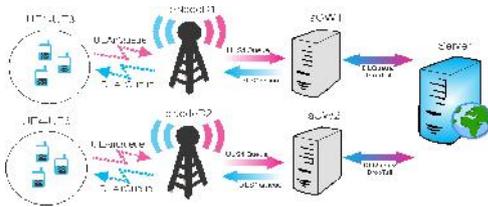
Gambar 8 Simulasi pengujian *throughput* dan *fairness*

Adapun diagram alir untuk pengujian *throughput* dan *fairness* yang dilakukan untuk melakukan pengujian menggunakan *Network Simulator 2.33* adalah sebagai berikut:

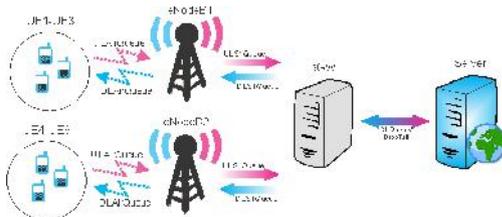


Gambar 5 Diagram alir pengujian *throughput* dan *fairness* Skenario Pengujian *Delay end to end*

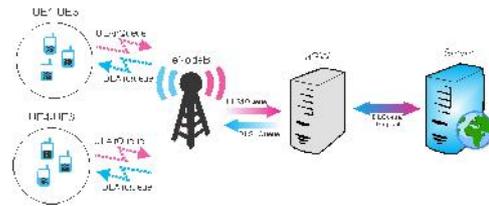
Pada skenario ini digunakan tiga variasi topologi yang berbeda yaitu topologi dengan 1 eNodeB, 2 eNodeB dan 2 gateway seperti pada gambar 6, gambar 7 dan gambar 8. Masing-masing topologi tersebut susun atas 6 buah UE dimana 3 UE mengirimkan data ke 3 UE lainnya.



Gambar 6 Topologi 2 gateway

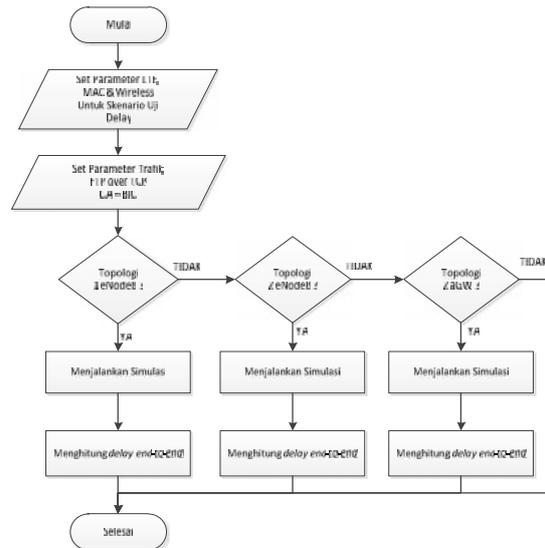


Gambar 7 Topologi dua eNodeB



Gambar 8 Topologi satu eNodeB

Adapun diagram alir skenario ini sebagai berikut:

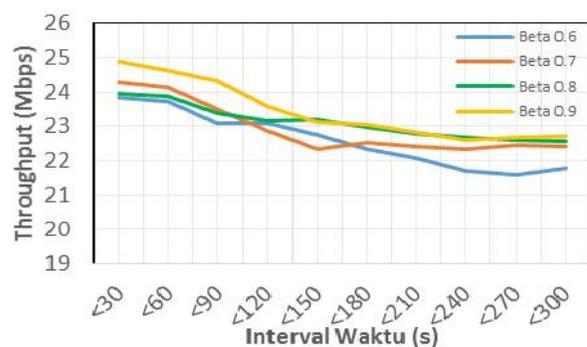


Gambar 9 Diagram alir pengujian *delay end to end*

### 3. Hasil dan Analisis

#### Throughput Sistem

Setelah dilakukan simulasi *throughput* sistem dihitung secara statistik dari hasil *trace file* tiap variasi BIC.

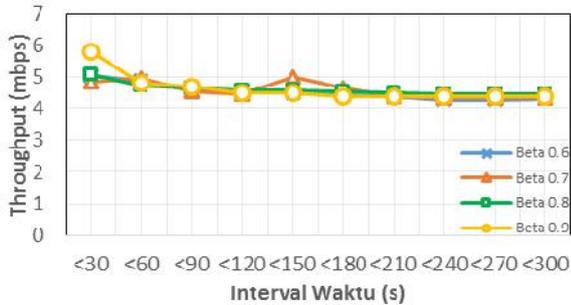


Gambar 10 Grafik *Throughput* Jaringan Tiap Variasi BIC

Dari gambar diatas terlihat bahwa BIC dengan *Beta* 0.9 memiliki *throughput* paling tinggi dibandingkan dengan varian BIC lainnya.

**Throughput User**

Setelah dilakukan simulasi *throughput user* dihitung secara statistik dari hasil *trace file* tiap variasi BIC.



**Gambar 11** Grafik *Throughput* Rata-rata Tiap User

Dari gambar 11, terlihat bahwa pada BIC dengan *Beta* 0.9 memiliki *throughput* terbaik. Secara keseluruhan *throughput* tiap variasi BIC setelah system stabil berada pada 4 Mbps hingga 5 Mbps.

**Fairness Sistem**

*Fairness index* dari sistem dapat diketahui dengan menggunakan *MaxMin Fairness* seperti berikut

$$\text{MinMax Fairness} = \frac{\text{Throughput}_{\min}}{\text{Throughput}_{\max}}$$

Dari data hasil pengujian *throughput* sistem diperoleh:

**Tabel 3** *Throughput* rata-rata sistem dengan varian

**Fairness Window**

Dengan membandingkan *window* hasil simulasi diperoleh:

**Tabel 5** Indeks *Window Fairness* Tiap Variasi BIC

Interval (s)	Window index BIC				Rata-rata	Deviasi (%)			
	eta 0.6	eta 0.7	eta 0.8	eta 0.9		eta 0.6	eta 0.7	eta 0.8	eta 0.9
0-30	1.10	1.06	1.04	1.05	1.06	4%	1%	2%	1%
31-60	1.07	1.10	1.03	1.05	1.06	1%	4%	3%	1%
61-90	0.97	1.11	0.96	1.03	1.02	4%	9%	6%	1%
91-120	1.03	0.97	1.02	0.96	0.99	3%	2%	2%	3%
121-150	1.04	0.94	1.02	0.98	0.99	5%	5%	2%	2%
151-180	0.90	1.00	0.98	1.02	0.98	7%	3%	0%	4%
181-210	0.97	0.92	1.01	0.97	0.97	0%	5%	4%	0%
211-240	0.90	0.98	0.97	0.97	0.95	5%	2%	1%	2%
241-270	0.96	0.97	1.00	0.97	0.98	2%	0%	2%	0%
271-300	1.06	0.94	0.97	1.01	1.00	6%	5%	3%	1%
Deviasi Rata-rata						4%	4%	3%	2%

Dari tabel 5, jika *window index* lebih dari 1, maka user mendapatkan kanal lebih besar dari yang lainnya. Dari tabel tersebut terlihat bahwa deviasi rata-rata dari BIC *Beta* 0.9 lebih kecil dibandingkan yang lainnya, hal ini

No	Variasi BIC	MinMax Fairness
1	Beta 0.6	0.999996
2	Beta 0.7	0.999999
3	Beta 0.8	0.999999
4	Beta 0.9	0.999998

Dari table di atas, terlihat bahwa *fairness index* BIC dengan *Beta* 0.7 dan 0.8 memiliki indeks paling tinggi dibandingkan dengan variasi BIC lainnya.

**Fairness User**

*Fairness index* tiap user dapat diketahui dengan menggunakan *Jain's Fairness Index* ataupun dengan membandingkan *window* tiap koneksi. Seperti ditunjukkan oleh persamaan berikut ini.

$$\text{Jain's Fairness Index} = \frac{(\sum_{i=1}^n x_i)^2}{n \cdot (\sum_{i=1}^n x_i^2)}$$

$$\text{Fairness Index} = \frac{W_i}{W}$$

Dari hasil pengujian *throughput user* diperoleh:

**Tabel 4** *Throughput* rata-rata tiap user dengan varian

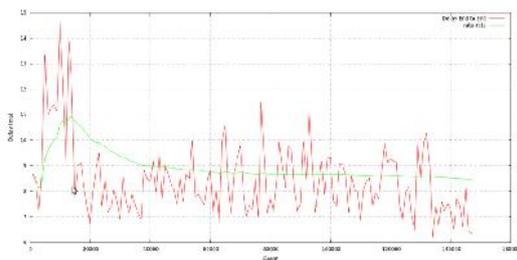
Beta	Throughput (Mbps)					Jain Fairness
	UE1	UE2	UE3	UE4	UE5	
0.6	4.8	4.6	3.5	4.9	4.8	0.986976
0.7	4.5	5.9	2.9	4.3	5.4	0.952124
0.8	3.6	6.2	3.4	4.6	5.3	0.951092
0.9	3.9	6.2	3.7	4.3	5	0.962934
Rata-Rata						0.963281

Dari tabel di atas, terlihat bahwa *fairness index* BIC dengan *Beta* 0.6 terbaik dibandingkan dengan variasi BIC lainnya.

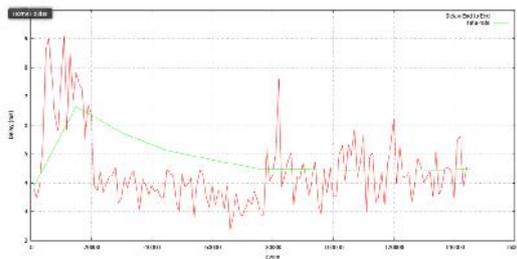
menunjukkan bahwa berdasarkan ukuran *window*, BIC *Beta* 0.9 mampu menangani trafik user dengan baik, hal ini dikarenakan faktor penurunan *window* yang dilakukan lebih sedikit.

**Delay end to end**

Dari simulasi untuk pengujian *delay end to end* dari tiap topologi yang dijadikan variasi dalam simulasi yang dilakukan dengan menggunakan *Network Simulator 2.33* diperoleh hasil sebagai berikut:



**Gambar 17** Grafik hasil simulasi *Delay end-to-end* 2 eNodeB



**Gambar 18** Grafik hasil simulasi *Delay end-to-end* 1 eNB

Dari grafik di atas, jika dihitung rata-rata *delay* maka diperoleh hasil sebagai berikut:

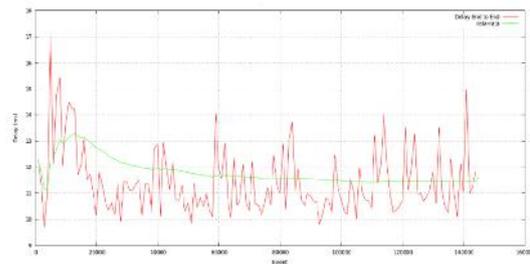
**Tabel 5** Perbandingan *delay* tiap topologi

Skenario Topologi	Delay End to end rata-rata (ms)
2 gateway (aGW)	11.8363089
2 eNodeB (eNB)	8.92106315
1 eNodeB (eNB)	4.81487492

Dari gambar dan tabel di atas, terlihat bahwa pada skenario 1 *eNodeB* memiliki *delay end to end* rata-rata paling kecil, sehingga dengan kata lain pada skenario ini kerja dari TCP BIC dalam pengontrolan kongesti paling baik dibandingkan dengan skenario lainnya.

**4. Kesimpulan**

Hasil dari penelitian ini adalah TCP BIC dengan *Beta* 0.9 memiliki rata-rata *throughput* jaringan paling tinggi dibandingkan dengan BIC *Beta* 0.8, BIC *Beta* 0.7 dan BIC *Beta* 0.6. TCP BIC dengan *Beta* 0.9 memiliki rata-rata *throughput* tiap *user* paling tinggi jika dibandingkan dengan variasi BIC lainnya yang digunakan. Semakin besar parameter *Beta* yang digunakan, semakin tinggi *throughput* yang akan



**Gambar 16** Grafik hasil simulasi *Delay end-to-end* 2 aGW

didapatkan, namun semakin kecil *fairness user* yang dapat dicapai. *Fairness* jaringan yang dimiliki oleh BIC *Beta* 0.7 dan BIC *Beta* 0.8 lebih baik dibandingkan BIC *Beta* 0.6 dan BIC *Beta* 0.9 dengan selisih *indeks fairness* 0.000001 hingga 0.000003. Berdasarkan *throughput*, *fairness* tiap *user* yang dimiliki oleh BIC *Beta* 0.6 lebih baik dibandingkan dengan varian BIC lainnya karena pembagian kanal yang dilakukan antar *user* lebih seimbang. Berdasarkan ukuran *window*, *fairness* dari BIC *Beta* 0.9 memiliki deviasi rata-rata paling kecil, karena lebar *window* yang berbanding lurus dengan tingginya *throughput*. BIC *Beta* 0.9 memiliki lebar *window* yang lebih dibandingkan dengan varian BIC lainnya, namun memiliki jumlah data hilang lebih tinggi karena *event* kongesti yang dialami lebih sering dibandingkan lainnya. Dari sisi *delay end to end*, topologi dengan 1 *eNodeB* memiliki *delay end to end* terkecil, hal ini menjelaskan bahwa kontrol kongesti TCP BIC yang digunakan paling baik pada skenario 1 *eNodeB*. Untuk itu pada penelitian selanjutnya dapat dilakukan analisa terhadap parameter *QoS* lainnya seperti *Jitter*, *Goodput*, *Delay* antrian dan lainnya, selain itu juga dengan menambah jumlah *user*, topologi yang digunakan, ataupun jenis aplikasi trafik yang digunakan seperti VoIP, HTTP, atau trafik Non-GBR lainnya.

**Referensi**

- [1] Playtoni, Kh. *Design and Development of Handoff Management System in LTE Networks using Predictive Modelling*. 2009.
- [2] Elmannai, Wafa. *TCP-UB: A New Congestion Aware Transmission Control Protocol Variant*. *International Journal of Computer Networks & Communications*. USA. 2012
- [3] Khlaif K Alenazi, Saud. *Analysis and Taxonomy of Network Quality of Service (QoS) Concepts in the Long Term Evolution/System Architecture Evolution (LTE/SAE) System*. University of Southern Queensland. 2010
- [4] Xu, Lisong. Harfoush, Khaled. Rhee, Injong. *Binary Increase Congestion Control (BIC) for Fast Long Distance Network*. North Carolina State University. 2004
- [5] Abed, Ghassan. *A Realistic Model and Simulation Parameters of LTE-Advanced Network*. Malaysia. 2012
- [6] Elmannai, Wafa. *A High Performance and Efficient TCP Variant*. University of Massachusetts. 2012

- [7] Singh, Amandeep. *Overview of the Evolved Packet Core Network. Thesis Masters of Engineering.* University of Alberta. 2009.
- [8] Sanchez, Juan Jesus. *Analysis of SC-FDMA and OFDMA Performance over Fading Channels.* Universidad de Malaga. 2011
- [9] Hua, Wu. *Analysis of TCP BIC Congestion Control Implementation.* Southeast University Nanjing, China.
- [10] Arianfar, Somaya. *TCP's Congestion Control Implementation in Linux Kernel.* Aalto University.
- [11] Simoneau, Paul. *The OSI Model: Understanding the Seven Layers of Computer Networks.* 2006
- [12] Jie, Li. *Performance Evaluation of Different TCP Congestion Control Schemes in 4G System,* Vaasa University of Applied Science. 2013
- [13] McKinney, Gordon. *TCP/IP State Transition Diagram (RFC793).* Publishing Company. 2002.
- [14] Rachmat, M. *Teknologi Jaringan Akses LTE (Long Term Evolution). Makalah Teknik Telekomunikasi Jurusan Elektro Universitas Hasanuddin.* 2013.
- [15] Cahyo, Nur. *Evaluasi Kinerja Penjadwalan Weighted Fair Queueing (WFQ) Dengan Adaptive Modulation And Coding (AMC) dalam Jaringan Mobile WIMAX.* 2013
- [16] Kusuma, Uke Kurniawan dan Tim. *Fundamental Teknologi Seluler LTE.* Rekayasa Sains Bandung. 2012