

Studi Network Congestion Dengan TCP Tahoe

Misbahul Fajri¹

¹Fakultas Ilmu Komputer, Universitas Mercu Buana
Jl. Raya Meruya Selatan, Kembangan, Jakarta Barat
¹fajri@mercubuana.ac.id

Abstract

The internet and intranets are increasingly dense in traffic due to the development of computer services and their users to date. In 1986, the first time there was congestion on the internet network, namely with a drastic decrease in traffic and a collapse, a few years later the theory of Congestion Avoidance and Control was proposed by Jacobson. Congestion is the sending of data packets on a link that exceeds its bandwidth capacity, this will have a negative impact on network performance, namely; many lost packets, decreased link utility (low throughput), high queue waiting times, and even deadlocks (congestion collapse), these were studied in this study because they were very challenging to analyze and develop. TCP Congestion Control is a TCP protocol that is applied to end devices, namely on the server and client, slow-start protocol and congestion avoidance and Fast Retransmit is the beginning of the previous research, bottleneck network topology where the mechanism is implemented at the Transport Layer with FTP service traffic. In this research, further study and analysis of the TCP Tahoe algorithm is carried out, which implements the four standard scenarios according to the TCP mechanism. This research simulation uses the OPNET simulator with dumbbell network and discarding packets.

Keyword: TCP Congestion Control, Network Traffic Management, Tahoe

Abstrak

Intenet maupun intranet semakin padat trafiknya dikarenakan perkembangan layanan komputer dan penggunaannya sampai saat ini. Pada tahun 1986 terjadi pertama kali kemacetan pada jaringan internet, yaitu dengan trafik menurun secara drastis dan terjadi kolaps, bebrapa tahun kemudian diusulah teorinya yaitu Congestion Avoidance and Control oleh Jacobson. Kemacetan adalah pengiriman paket data pada suatu link yang melebihi kapasitas bandwidthnya, ini akan berdampak negatif pada kinerja jaringan, yaitu; banyak paket yang hilang, menurunnya utilitas link (low throughput), tingginya waktu tunggu antrian, dan bahkan terjadi deadlock (congestion collapse), hal tersebut yang dipelajari pada penelitian ini karena sangat menantang untuk dianalisa dan dikembangkan. TCP Congestion Control adalah protokol TCP yang diterapkan pada end device yaitu di server dan client, protokol slow-start dan congestion avoidance serta Fast Retransmit merupakan awal dalam penelitian sebelumnya, topologi jaringan bottleneck yang dimana dimplementasikan mekanisme tersebut pada Transport Layer dengan trafik layanan FTP. Pada penelitian ini dilakukan studi lanjut dan analisa TCP Tahoe algortima, dimana mengimplementasikan keempat skenario standar sesuai mekanisme TCP. Simulasi penelitian ini menggunakan simulator OPNET dengan jaringan dumbbell dan paket discarding.

Keyword: TCP Congestion Control, Network Traffic Management, Tahoe

I. Pendahuluan

Dengan berkembangnya teknologi informasi dan layanannya, membuat penggunaan akses semakin tinggi pada jaringan computer yaitu internet, sehingga pengiriman trafik padat pada lalu-lintas data komunikasi di jaringan *local* maupun internet. Pada tahun 1986 terjadi pertama kali kemacetan jaringan internet, yang mengakibatkan lumpuh total (*congestion collapse*) karena penurunan drastis kecepatan pengiriman paket data. Hal tersebut karena belum adanya pengaturan untuk mengatasi keadaan trafik yang sangat tinggi, teori *Congestion Avoidance and Control* [1] yang dibuat oleh Jacobson pada tahun 1988 untuk mengatasi kemacetan

tersebut. Kemacetan adalah pengiriman paket data pada suatu link yang melebihi kapasitas *bandwidth*nya, ini akan berdampak negatif pada kinerja jaringan, yaitu; banyak paket yang hilang, menurunnya utilitas link (*low throughput*), tingginya waktu tunggu antrian, dan bahkan terjadi *deadlock (congestion collapse)*. Congestion sangat penting untuk diperbaiki yang apabila tidak dapat diatasi akan menghambat perkembangan informasi, oleh karena itu perkembangan penelitian di bidang ini berkembang sampai saat ini, munculnya metode-metode baru dari yang simpel hingga kompleks dan juga untuk berbagai media komunikasinya, tetapi metode-metode tersebut mempunyai keunggulan dan kekurangannya, dikarenakan mempunyai karakteristik yang berbeda pula dalam penerapannya, hal ini sangat menarik dan menantang untuk mempelajari bidang ini yang merupakan pembahasan dalam penelitian ini. Ada dua mekanisme *Congestion Control*, yaitu *TCP Congestion Control* dan *Active Queue Management (AQM)* [2], yang implementasinya berbeda. AQM diimplementasikan pada intermediate network yang berfungsi untuk penanganan *Buffer overflow*, dimana penyebab padatnya trafik pada sebuah node dan menjadi masalah dalam pengiriman paket data yang mengakibatkan memburuknya kinerja jaringan [3] yaitu diantaranya pada *buffer overflow* pada router. Empat mekanisme yang dimiliki oleh *TCP congestion control* adalah; *slow start*, *congestion avoidance*, *fast retransmit*, dan *fast recovery*. Keempat mekanisme tersebut diimplementasikan menjadi *TCP Tahoe*, mekanisme ini sangat ampuh untuk mengatasi kepadatan jaringan dibandingkan dengan *TCP control* standar sebelumnya.

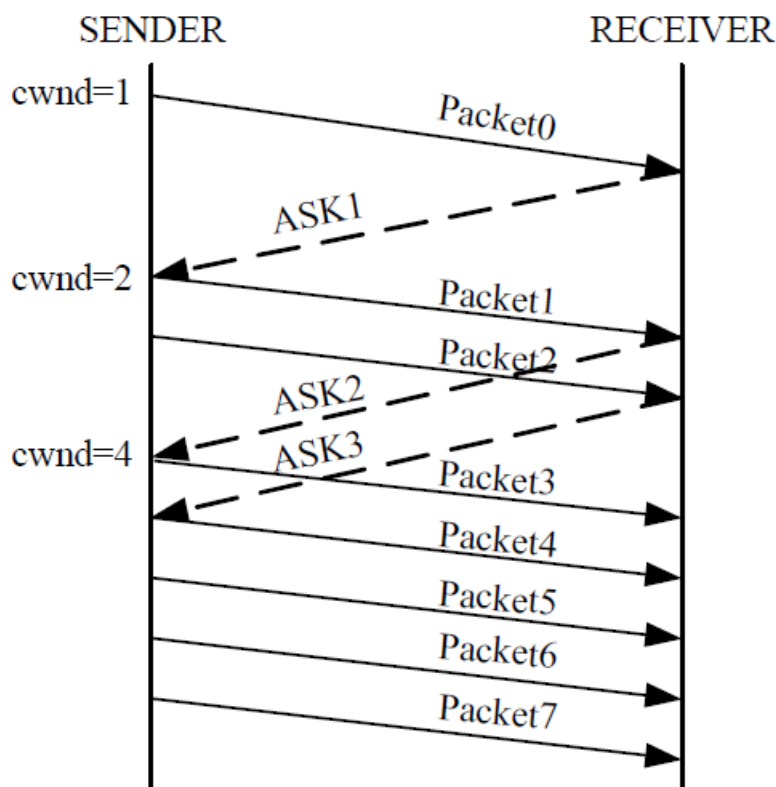
Pada penelitian ini dilakukan analisa kinerja protokol *TCP Congestion Control* dengan menggunakan metode *TCP Tahoe*, dan disimulasikan untuk uji pada beberapa skenario drop paket yang dilalui layanan FTP pada topologi *bottleneck network*, keluaran simulasi menghasilkan karakteristik kinerja *TCP Control* yang akan dianalisa. Penggunaan simulator jaringan OPNET, metode *TCP Control* diimplementasikan pada topologi jaringan *bottleneck*, yang kemudian dibandingkan mekanisme standar *TCP Control* yaitu *slow start* dan *congestion avoidance*. Trafik FTP digunakan dalam test skenario pengiriman paket data. Pada Simulasi ini akan didapat karakteristik penggunaan *buffer* dalam penanganan antrian, jumlah paket yang dibuang, *congestion window size*, dan waktu tunda.

II. Metodologi Penelitian

Algoritma *TCP Control – Slow Start*

Untuk memecahkan masalah kemacetan pada jaringan, algoritma *slow start* disertakan. Dasar pendekatannya adalah dengan ide *congestion window (cwnd)* [4]. Ketika koneksi baru dibuat, *cwnd* di inisial satu paket. Setiap waktu sebuah paket dengan nomor berurut n datang pada penerima, penerima mengkonfirmasi paket n dengan pengiriman *acknowledgment (ACK)* paket, yang mengandung informasi nomor paket berikutnya, dimana sedang ditunggu yang dikirim balik ke pengirim. TCP menggunakan kedatangan ACK sebagai trigger paket baru yang ditransmisikan, sebagai contoh, setiap waktu sebuah ACK diterima, *congestion window* dinaikan dengan satu paket. Pengirim menstop kenaikan *window size* ketika waktu mencapai batas kapasitas jaringan [5]. Batas didefinisikan sebagai minimum *window* yang pengirim dapat transmisikan dan *window* yang penerima dan terima.

Pada Gambar 1. ditunjukkan bahwa TCP memulai koneksi antara pengirim dan penerima *window size* di set satu paket dan pengirim menunggu ACKnya. Pada step berikutnya, setelah penerimaan ACK, *congestion window* di set dari satu ke dua, dan dua paket dapat dikirim. Pada saat setiap dua paket diterima dengan baik, *congestion window* dinaikan menjadi empat. Ini dapat diestimasi sebagai suatu pertumbuhan eksponensial.



Gambar 1. Kenaikan Congestion Window [3]

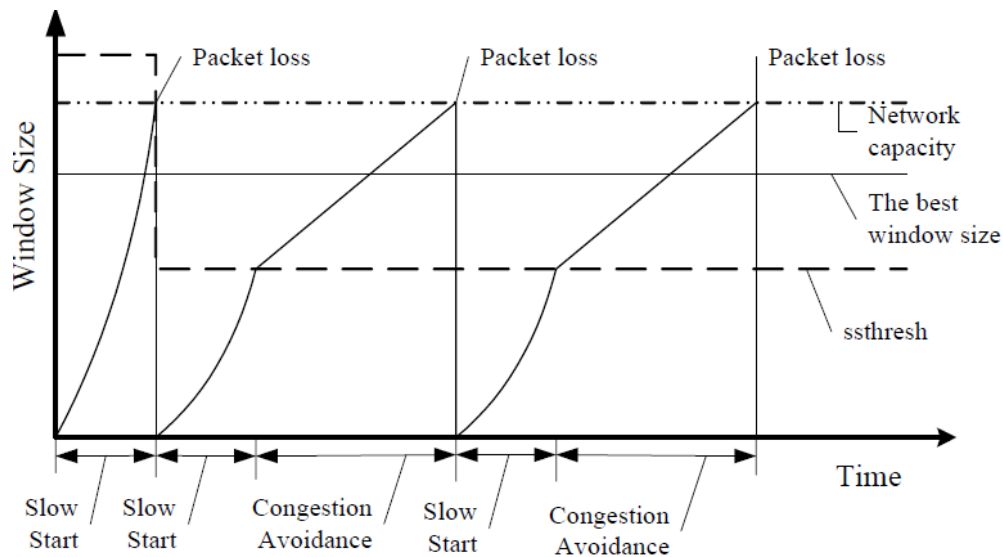
Congestion Avoidance

Congestion Avoidance adalah algoritma yang mencoba memecahkan masalah dengan membuang paket. Kemacetan timbul dikarenakan paket yang dikirim ke router membanjiri daya tampung buffer pada router. Indikasi yang umum adanya paket yang hilang adalah *timeout* dan menerima duplikat ACK. *Congestion avoidance* dan *slow start* merupakan mekanisme control yang berbeda yang bekerja sama. Mekanisme kombinasi control didahului dua parameter disesuaikan jumlah data yang diterapkan pada jaringan: *congestion window (cwnd)* dan *slow start threshold size (ssthresh)*. *Window size* didefinisikan oleh formula sebagai berikut [3].

$$\text{Window size} = \min(\text{advertized_window}, \text{cwnd})$$

Dimana *cwnd* adalah *window* yang pengirim dapat transmisikan, *advertized_window* adalah *flow control window*, dimana dikirim dari bagian penerima.

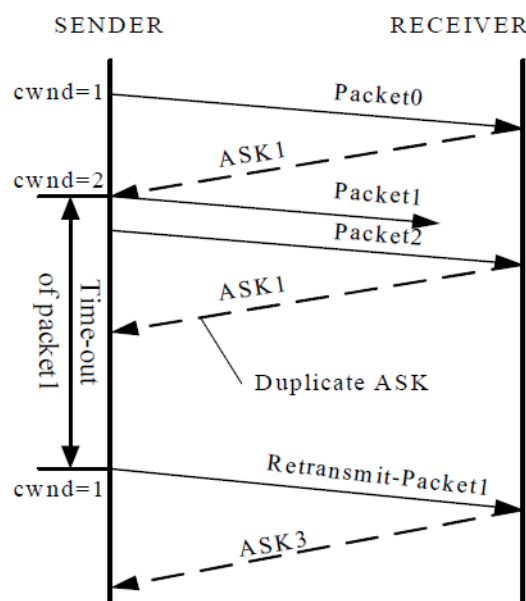
Ketika koneksi baru mulai, TCP set *cwnd* satu paket, *ssthresh* nilai tinggi (65535 bytes), dan memulai mode *slow start*. Pada saat macet (diindikasikan oleh *timeout* atau penerimaan duplikat ACK), setengah nilai *window size* di set pada *ssthresh*, dan *cwnd* di set satu paket. TCP mentrigger *Slow Start* pada permulaan pengiriman dan kenaikan eksponensial *window*: kirim satu *segment*, kemudian dua, kemudian empat, dan seterusnya setiap ACK diterima. TCP bekerja pada mode *slow start* sampai *window size* mencapai *ssthresh*. Setelah TCP menentukan *congestion avoidance*, maka nilai *cwnd* akan ditambah menjadi $\text{segsize} * \text{segsize} / \text{cwnd}$ pada setiap ACK diterima, dimana *segsize* adalah *segment size* dan *cwnd* yang diatur dalam *byte*. Ini merupakan kenaikan yang linier *cwnd*, dibandingkan *slow start* yang merupakan kenaikan eksponensial.



Gambar 2. Grafik Slow Start dan Congestion Avoidance

Fast Retransmit

TCP versi lama mendeteksi kepadatan jaringan dan paket yang hilang dengan mekanisme *timeout*. Ketika sebuah paket dikirim, TCP menset timernya dengan *retransmission timeout period* (RTO) [3]. Jika penerima menerima paket dengan baik, TCP segera membuat *acknowledgment* (ACK) sebagai *corresponding* sebelum melewati *timer*. TCP berasumsi bahwa *network* baik. Setelah TCP secara otomatis menginformasikan timer paket ACK diterima dan menunggu paket ACK lainnya. Pada kasus, bahwa TCP tidak menerima paket ACK dalam periode RTO, pengirim akan mentransmisikan ulang paket yang *expired*. Selanjutnya, *TCP slow-start* dan set *cwnd* diset ke 1 dan *ssthresh* di set ke $(old\ cwnd/2)$ dapat dilihat pada gambar 3. dibawah ini.



Gambar 3. Mekanisme paket *lost* dengan RTO

Fast Recovery

Fast recovery berlaku setelah adanya *congestion* (kemacetan). Beberapa tahun yang lalu beberapa modifikasi algoritma TCP *fast recovery* telah diperkenalkan seperti Tahoe, Reno dan Vegas, yang berbeda pada dasar mekanisme *congestion control*.

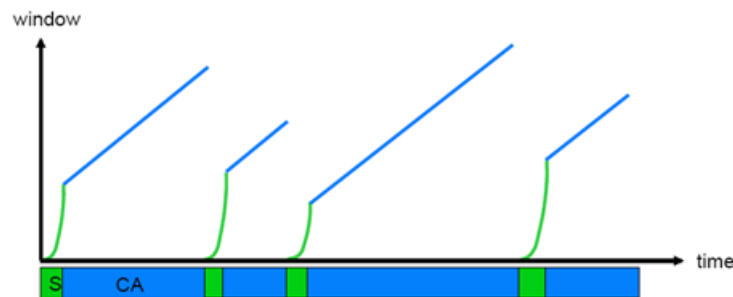
Algoritma Tahoe beroperasi sebagai berikut [3]:

- Setelah fast retransmit TCP menset *window size* ke 0 dan *ssthresh* menjadi *window size/2*.
- TCP memulai *slow start*.
- Ketika *window size* mencapai *ssthresh*, TCP mentrigger *congestion avoidance*.

TCP Tahoe

Algoritma Tahoe adalah suatu protokol TCP yang telah banyak digunakan dalam pengaturan *windows size* untuk menjamin komunikasi *end-to-end* [6]. karakteristiknya TCP Tahoe ditunjukkan pada Gambar 4, yang beroperasi sebagai berikut:

- Setelah *fast retransmit* TCP menset *window size* 1 dan *ssthresh* menjadi *window size/2*.
- TCP memulai *slow start*.
- Ketika *window size* mencapai *ssthresh*, TCP mentrigger *congestion avoidance*.



Gambar 4. Congestion Window TCP Tahoe

Simulator Jaringan dan Konfigurasi

Analisa kinerja jaringan dapat dilakukan dengan simulasi model. *Analytical modeling* dan *computer simulation* adalah dua bentuk umum untuk simulasi jaringan. Tools untuk melakukan simulasi topologi jaringan yang kompleks dapat dilakukan dengan software simulasi. Pada penelitian ini digunakan software simulasi OPNET [7], merupakan perangkat lunak yang berdaya-guna pada penelitian industri maupun akademik, yang pada penelitian ini menggunakan software ini. OPNET (*Optimized Network Engineering Tool*) menawarkan kondisi penelitian/pengembangan yang menyeluruh untuk berbagai spesifikasi untuk simulasi dan analisa kinerja jaringan komunikasi, serta mendukung berbagai jangkauan sistem komunikasi dari single LAN sampai jaringan satelit.

Konfigurasi jaringan penelitian dan model simulasi OPNET ditunjukkan pada Gambar 5. Pada Diagram tersebut topologi jaringan yang diimplementasikan pada OPNET merupakan konfigurasi jaringan *bottleneck* yang simpel, yaitu mempunyai dua buah router dan terkoneksi dengan node subnet [8]. Node subnet memiliki sebuah router dan host yang mempunyai layanan TCP. Konfigurasi koneksi antara LAN ke WAN dapat merupakan akses dial-up atau line jaringan ISP.

Tahapan Penelitian

1. Studi literatur.
Pada fase awal penelitian, melakukan *literature review* berdasarkan konsep penelitian terkait yang didapat dari artikel jurnal dan seminar, buku, dan internet. Disamping itu mempersiapkan kebutuhan dalam penelitian.
2. Identifikasi Masalah dan Definisi Skenario

Implementasi mekanisme TCP Tahoe untuk mengatasi masalah dengan mempelajari karakteristik kerjanya, serta simulasi membandingkan dengan mekanisme standar TCP Congestion Control, dalam penelitian ini akan dilakukan pada skenario Layanan FTP dan Paket Drop. Karakteristik kinerja yang dianalisa adalah; *Buffer Usage, Traffic Dropped, Congestion Window, Delay, Utilization*. Skenario-skenario yang telah ditentukan diharapkan dapat kinerja dan karakteristik TCP Tahoe pada penelitian ini.

3. Perancangan dan Implementasi

Pada tahap ini perancangan topologi jaringan bottleneck (dummbell) diimplementasi modelnya pada simulator jaringan OPNET dengan menggunakan metode *Modeling and Simulation cycle* [9]. Konfigurasi jaringannya sebagai berikut:

- Topoogi jaringan bottleneck ditunjukkan pada Gambar 5., yaitu dengan adanya cloud jaringan.
- Menggunakan 2 buah network yang terhubung pada cloud jaringan dengan Link PPP DS3, masing network mempunyai *router device*.
- Satu client dengan layanan FTP clien, link 100MB terhubung ke *router east*
- Satu Server dengan FTP *Application Server*, link 100MB terhubung ke *router west*.

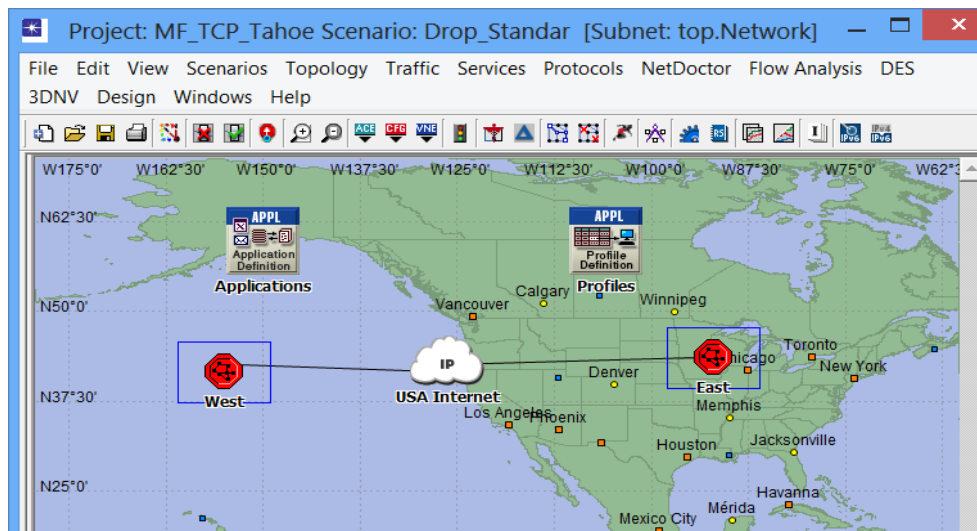
4. Uji Coba Simulasi dan Analisa

Pada tahap akhir ini dijalankan skenario simulasi yang sudah dirancang, yaitu tiga buah skenario uji coba terhadap mekanisme TCP Tahoe, dan satu skenario uji coba yang membandingkan dengan standar TCP. Tahapan selanjutnya menganalisa kinerja dan karakteristik skenario-skenario yang telah didefinisikan pada tahap sebelumnya. Skenario dan mekanisme *TCP Control* yang disimulasikan dengan perbandingan kinerja sebagai berikut :

- Trafik data layanan FTP dan karakteristiknya
- Kondisi saat Paket Drop dan RTO.

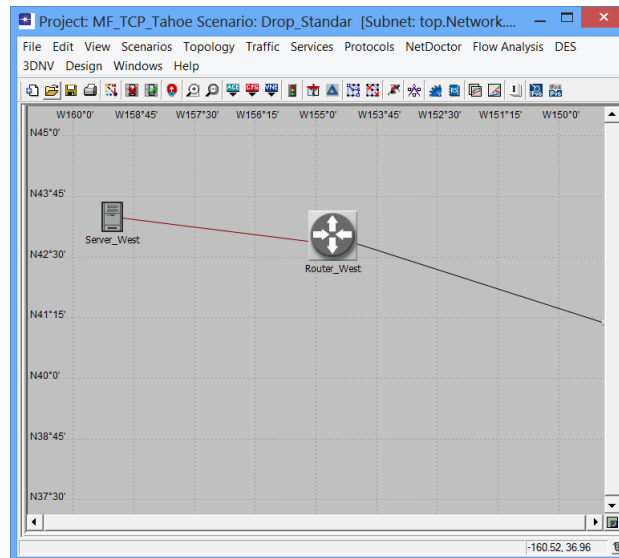
III. Hasil dan Pembahasan

OPNET Modeler [10] adalah simulator yang digunakan dalam penelitian ini. Rancangan pada simulator dengan mengimplemetasikan topologi jaringan dengan skenario dapat dilihat pada gambar 5. dibawah ini:

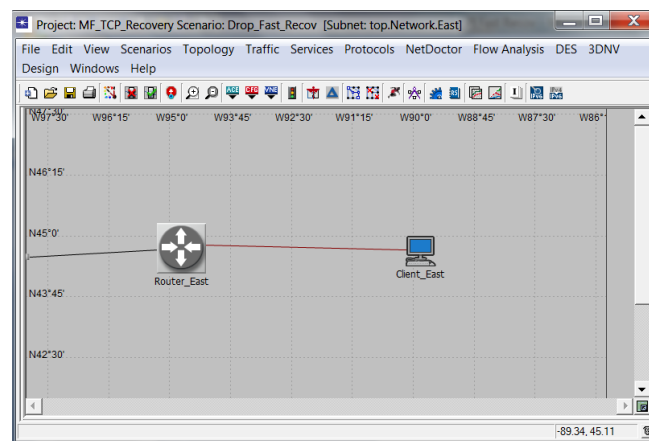


Gambar 5. Rancangan simulasi jaringan dengan OPNET Modeler.

Dari Gambar 5. Topologi jaringan yang dibentuk dengan cloud jaringan IP yang terhubung dengan *West Network* yang terdiri dari Router dan server FTP yang dapat dilihat pada Gambar 6, dan terhubung juga dengan *East Network* yang terdiri dari router dan client FTP yang dapat dilihat detailnya pada Gambar 7. Uji coba dilakukan dengan menjalankan layanan FTP pada setiap skenario yaitu mekanisme yaitu Drop paket 0,5% pada Metode Standar dan Tahoe. Layanan trafik FTP dengan 10 MB dijalan pada setiap uji coba.



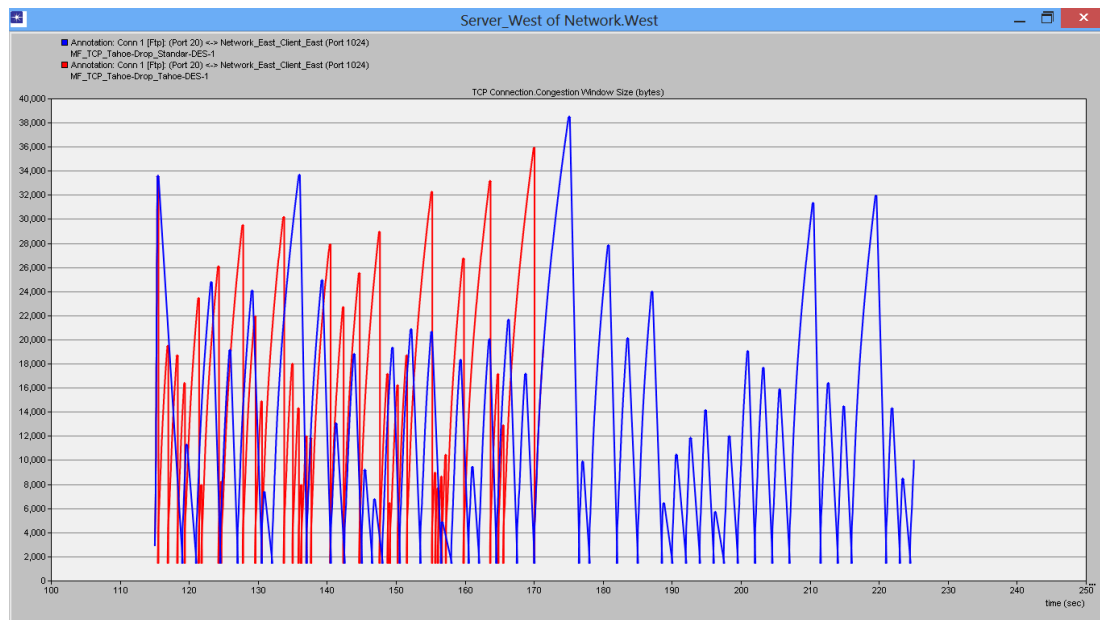
Gambar 6. Implementasi topologi jaringan *West Network*.



Gambar 7. Implementasi topologi jaringan *East Network*.

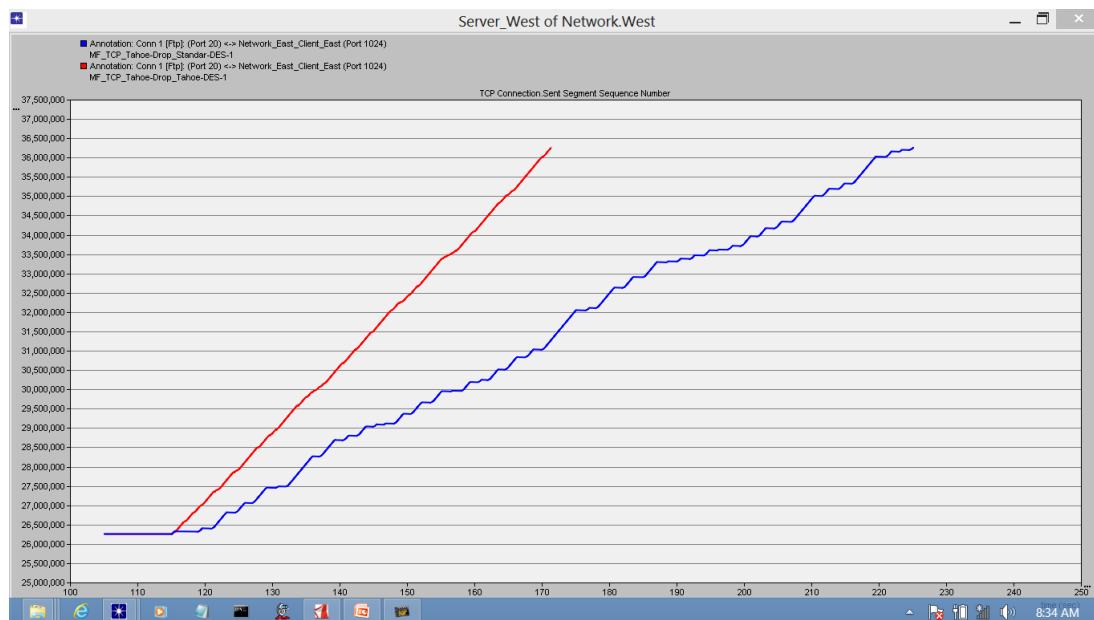
Analisa Perbandingan TCP Congestion Control Standar dengan Tahoe

Perbandingan kedua skenario untuk *Congestion Window Size* dapat dilihat pada Gambar 8. Fluktuasi terjadi pada skenario Standar dan Tahoe, untuk skenario Standar pengiriman paket data membutuhkan waktu yang sangat lama yaitu 225 detik, sedangkan Tahoe menyelesaikan pengiriman dalam waktu $\frac{1}{2}$ nya sekitar 170 detik. Pada TCP Standar Congestion setiap paket loss data akan dikirimkan kembali setelah timeout, sedangkan pada Tahoe paket loss segera dikirim tanpa menunggu *timeout* dan mereset *congestion windows* ke 1 apabila telah menerima ACK sekurangnya 3 paket terlihat ada perbaikan yang baik pada Tahoe yang menerapkan *Fast Retransmit*.



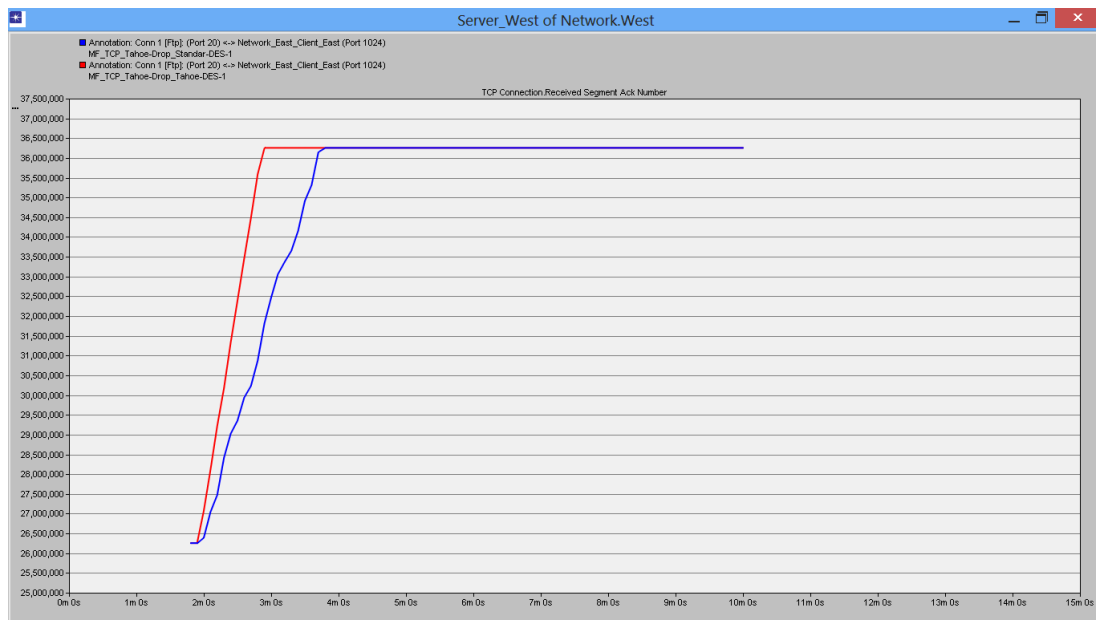
Gambar 8. Perbandingan Congestion Window.

Sent segment sequence number dapat dilihat pada Gambar 9. menunjukkan bahwa pada skenario Tahoe terjadi peningkatan pengiriman paket yang stabil dan cepat dibandingkan skenario Standar, dimana terjadinya hanya beberapa kali *timeout* saat pengiriman paket. Mekanisme Tahoe mempunyai *Sent Segment Number* diatas Mekanisme Standar, dengan demikian ia menunjukkan perbaikan yang baik dalam implementasinya.



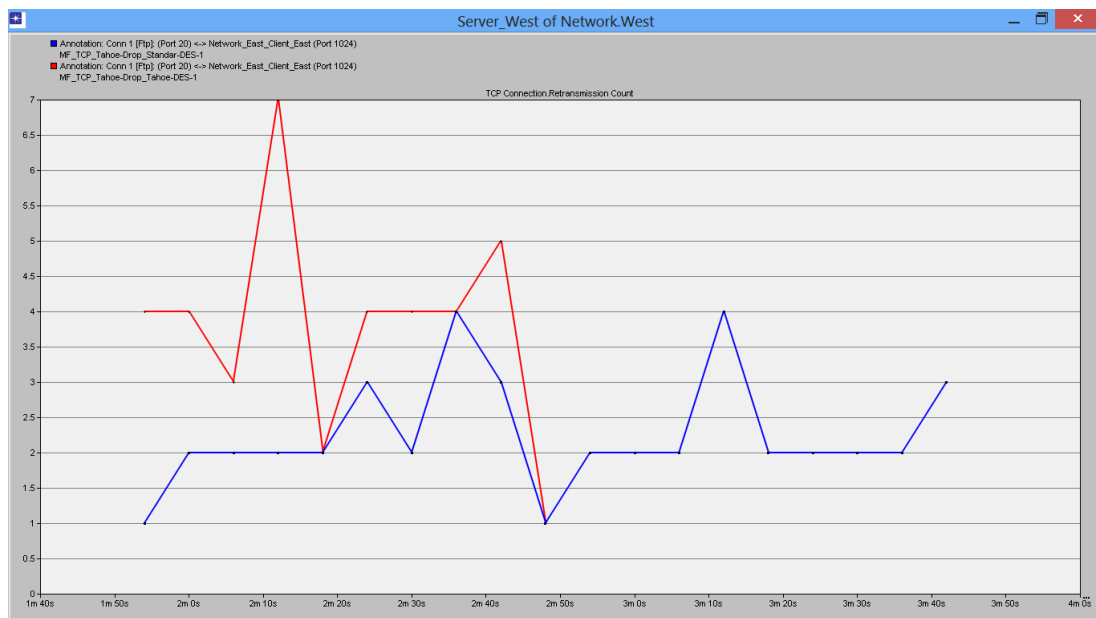
Gambar 9. Perbandingan Sent Segment

Pada kedua skenario yang diperlihatkan pada Gambar 10, merupakan grafik *Receive Acknowledgment*, dapat disimpulkan bahwa mekanisme TCP Standar mempunyai ACK lebih tinggi dengan banyaknya pekaat yang tidak terkirim ke *end host* dibandingkan dengan mekanisme TCP Tahoe.



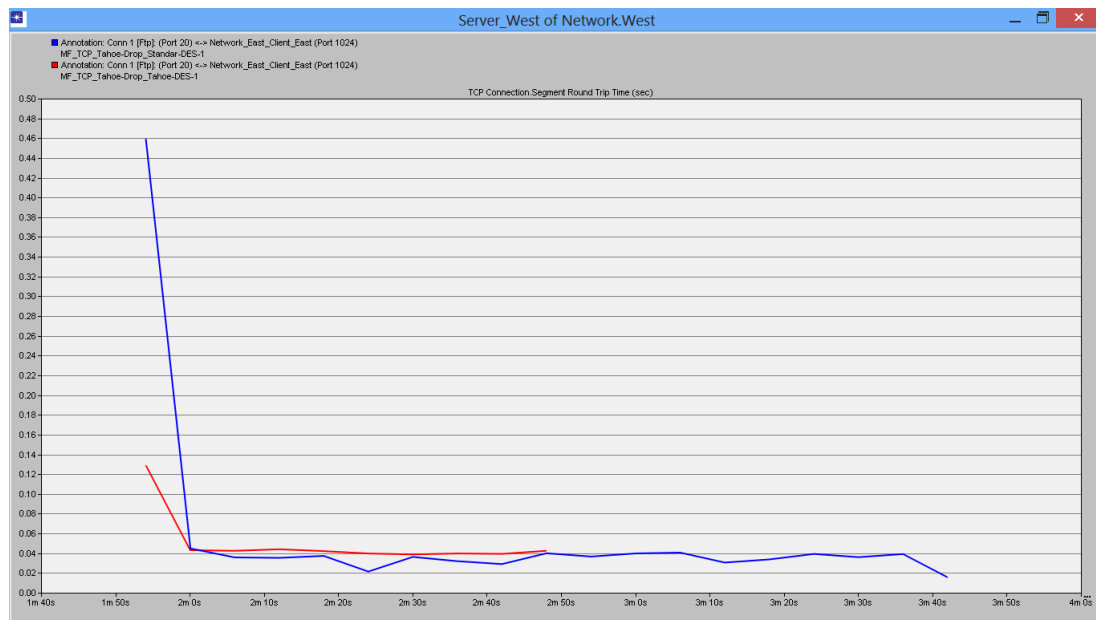
Gambar 10. Perbandingan Received Acknowledgment.

Grafik jumlah paket retransmit terjadi fluktuasi pada kedua sekario ini dapat dilihat pada Gambar 11. skenario Tahoe mempunyai jumlah *retransmit* yang besar tetapi dengan demikian proses lebih cepat dikirimm, sedangkankan Mekanisme Standar mempunyai jumlah paket *retransmit* di bawah sekenario Tahoe sepanjang waktu pengiriman paket 10 MB FTP, tetapi proses pengirimannya lama.



Gambar 11. Perbandingan Jumlah Retransmit

Terlihat RTT yang dialami Tahoe sangat rendah dibandingkan dengan *Conngestion Control* Standar dimana sangat tinggi diawal pengiriman ini ditunjukkan pada Gambar 12.



Gambar 12. Perbandingan *Round Trip Time* (RTT)

IV. Kesimpulan

Pada penelitian ini telah berhasil mensimulasikan mekanisme *TCP Congestion Control* dengan metode mekanisme TCP Tahoe pada jaringan *boottleneck* dengan paket drop 0.5 persen, dan menganalisa karakteristiknya dengan menjalankan trafik data layanan FTP 10 Mbs. Protokol TCP Tahoe tersebut dibandingkan dengan menggunakan protocol TCP Standar, kedua skenario mempunyai grafik *congestion windows size* yang berfluktuasi karena kepadatan trafik data, dimana TCP Tahoe memperbaiki *congestion window* yang lebih baik dengan kecepatan sekitar 2 kali lipat dari TCP standar. Grafik *Sequence segment number* menunjukkan adanya *timeout* penerimaan *acknowledgment number* saat terjadinya paket dibuang, TCP Tahoe mempunyai hanya beberapa kali *timeout*, sedangkan mekanisme Standar melakukan pengiriman paket loss setelah terjadi *timeout* sehingga membutuhkan waktu yang lama. Jumlah *Receive Acknowledgment* pada mekanisme Standar mempunyai grafik yang lebih banyak dibandingkan dengan protokol Tahoe. Jumlah Retransmision pada Tahoe lebih banyak diawal pengiriman dibandingkan dan mempunyai sedikit waktu pengiriman. *Round Trip Time* sangat kecil dibandingkan dengan Standar yang dialami Tahoe diawal pengiriman. Secara keseluruhan dari keempat karakteristik parameter yang diamati perbaikan dengan mekanisme Tahoe lebih baik dibandingkan *Congestion Control* Standar.

Daftar Pustaka

- [1] V. Jacobson, "Congestion avoidance and control," in *ACM SIGCOMM computer communication review*, 1988, pp. 314-329.
- [2] T. Álvarez, V. Álvarez, and L. Nicolás, "UNDERSTANDING CONGESTION CONTROL ALGORITHMS IN TCP USING OPNET," *EDULEARN10 Proceedings*, pp. 3155-3162, 2010.
- [3] Y. N. Reddy and P. Srinivas, "A Routing Delay Predication Based on Packet Loss and Explicit Delay Acknowledgement for Congestion Control in MANET," *International Journal of Communication Networks and Information Security*, vol. 10, p. 447, 2018.
- [4] Kirov, Georgi. A Simulation Analysis of the TCP Control Algoritms. International Conference on Computer Systems and Technologies – CompSysTech'2005.
- [5] Ryu, Seungwan, et all. Advance in Internet Congestion Control. IEEE Communication, Volume 5, No. 1, Third Quarter 2004.
- [6] E. Abolfazli and V. Shah-Mansouri, "Dynamic adjustment of queue levels in TCP Vegas-based networks," *Electronics Letters*, vol. 52, pp. 361-363, 2016
- [7] Opnet Technologies. OPNET Modeler documentation.

- [8] Zhu, Chengyu, et all. A Comparison of Active Queue Management Algorithms Using OPNET Modeler. School of Information Technology and Engineering University of Ottawa, Canada.
- [9] Aboelala, Emad, "Network Simulation Experiments Manual". Morgan Kaufmann Publishers. ISBN: 978-0-12-373974-2, 2008.
- [10] Chang, Xinjie. "Network Simulations With Opnet". Proceedings of the 1999 Winter Simulation Conference.