

Perbaikan Tcp Westwood +

Sukiswo

Abstract: TCP Westwood+ is evolution of TCP Westood. TCP Westwood+ representing one of TCP proposed for the modification of existing TCP now. Specially TCP Westwood+, TCP Westwood+ sender keeps an end to end bandwidth estimation to adjust the smart transmission rate after detection a packet loss. In this way , TCP Westwood+ can improve; repair the standard TCP (Newreno, SACK) in a lossy environment. This paper comment the mechanism congestion base control TCP Westwood+, testing performance of TCP Westwood+, enhance TCP Westwood+ by modification increase phase from additive mechanism to multiplicative mechanism, and compare with TCP Westwood+ standard.

Keywords: TCP Westwood+, end to end bandwidth estimation, estimation filter, additive mechanism, multiplicative mechanism

Algoritma kontrol kongesti TCP untuk pertama kali diperkenalkan oleh Van Jacobson [6] yang kemudian dikenal dengan TCP Tahoe. TCP ini berdasarkan mekanisme *sliding window* yang menerapkan algoritma AIMD (*Additive Increase Multiplicative Decrease*) untuk pembaharuan laju transmisi dari sumber daya jaringan yang tersedia. Penerima menyediakan umpan balik ACK, dan dipakai oleh pengirim untuk mendeteksi *segments losses* dan sebagai konsekwensinya memperbaharui laju transmisinya. Pada awalnya TCP dirancang untuk laju jaringan kawat (*wired*) dengan laju kesalahan rendah, dengan asumsi bahwa semua packet losses disebabkan kongesti jaringan. Algoritma AIMD memastikan stabilitas jaringan tetapi tidak menjamin pembagian sumber daya jaringan adil.

Algoritma TCP Westwood+ berdasarkan pada estimasi *bandwidth end-to-end* sepanjang koneksi jalur TCP [2], [7] . Estimasi diperoleh dengan memfilter kembalinya aliran paket ACK dan digunakan secara adaptif untuk menetapkan *congestion window*. Ketika 3 duplikasi ACK diterima *congestion window* (cwnd) dan *slow start threshold* (ssthresh) ditetapkan sama dengan estimasi *bandwidth* (BWE) dikalikan dengan *round trip time* minimum hasil pengukuran (RTTmin); ketika terjadi *timeout* ssthresh ditetapkan sama seperti sebelumnya dan cwnd ditetapkan sama dengan 1.

Paper ini mengulas modifikasi TCP Westwood+ fasa naik dari mekanisme aditif ke mekanisme multiplikatif, dan membandingkan kinerja (*throughput, delay, fairness*) dengan TCP Westwood+ melalui 3 skenario yaitu konfigurasi *client server*, konfigurasi *multihop* dan konfigurasi *bottleneck*.

STRATEGI EVOLUSI CONGESTION WINDOW

Koneksi TCP diaktifkan oleh *Congestion Window* (cwnd) W yang sepadan dengan satu segmen. W meningkat satu untuk tiap-tiap menerima ACK yang bukan duplikasi sampai ssthresh dicapai. Ide dasar *slow start* adalah memperoleh kenaikan window secara cepat karena bersifat eksponensial dari waktu ke hingga kapasitas pipa transmisi dicapai.

Algoritma kontrol kongesti untuk pertama kali diperkenalkan oleh Van Jacobson yang merumuskan mekanisme algoritma AIMD ketika penerima mendapati ACK atau segmen hilang sebagai berikut:

$$\text{ACK} : W \leftarrow W + \frac{1}{W} \quad (1)$$

$$\text{LOSS} : W \leftarrow \frac{W}{2} \quad (2)$$

dengan W adalah ukuran cwnd

Mekanisme AIMD menetapkan cwnd linier pada fasa naik dari waktu ke waktu. Pendeteksian *losses* melalui duplikasi ACK menyebabkan pengirim membelah cwnd menjadi setengahnya.

Sukiswo,(sukiswo@elektro.ft.undip.ac.id) adalah dosen di Jurusan Teknik Elektro Fakultas Teknik Universitas Diponegoro (Undip) Semarang Jl. Prof. Sudharto, S.H. Tembalang Semarang 50275

Fasa naik aditif dipandang konservatif dan fase turun multiplikatif dipandang agresif menyebabkan keterlambatan pada jaringan kecepatan tinggi. Mekanisme AIMD dianggap tidak efisien untuk jaringan dengan BDP besar. Dengan kata lain, sejumlah besar waktu diperlukan untuk "memugar kembali" cwnd setelah pendeteksian paket hilang.

Scalable TCP mencoba untuk memperbaiki kekurangan mekanisme AIMD dalam jaringan dengan BDP besar dengan menerapkan MIMD (*Multiplicative Increase Multiplicative Decrease*) seperti tergambar dalam persamaan berikut :

$$\text{ACK} : W \leftarrow W + \alpha \quad (3)$$

$$\text{LOSS} : W \leftarrow \beta \cdot W \quad (4)$$

Kecepatan kenaikan *window* ditentukan oleh parameter α sedangkan oleh parameter β bertanggungjawab untuk pengurangan *window* secara dinamis. Nilai α dan β disarankan oleh pengusul Scalable TCP masing-masing 0.01 dan 0.875 sebagai nilai *default* untuk kalibrasi.

TCP Westwood+ menggunakan peningkatan *window* aditif linier sesuai dengan TCP Standard. Jika paket hilang dideteksi, maka nilai cwnd W diperbarui sebagai berikut:

$$\square \text{CK} : W \leftarrow W + \frac{1}{W} \quad (5)$$

$$\text{LOSS} : W \leftarrow \max\left(2, \frac{BWE \cdot RTT_{\min}}{\text{seg_size}}\right) \quad (6)$$

dimana BWE merupakan estimasi *bandwidth end-to-end* yang tersedia, RTT_{\min} merupakan waktu pulang pergi minimum yang diukur selama koneksi dan seg_size ukuran segmen TCP, dalam bit

Ada usulan lain yang mengkombinasikan mekanisme peningkatan multiplikatif dan mekanisme pengurangan adaptif pada TCP Westwood+ sehingga disebut MIAD (*Multiplicative Increase Adaptive Decrease*).

Perbaikan yang dapat dilakukan adalah dengan merancang filter estimasi *bandwidth* yang menangani algoritma MIAD dan melakukan perbaikan pada fasa naik misalnya dengan menerapkan fungsi logaritmik seperti pada BIC TCP, dimana cwnd W akan bernilai seperti tergambar dalam persamaan berikut :

$$\square \text{CK} : W \leftarrow \max\left(2, \frac{BWE \cdot RTT_{\min}}{\text{seg_size}}\right) \quad (7)$$

$$\text{LOSS} : W \leftarrow W + \max\left(\frac{W_{\max} - W}{\alpha \cdot W}, \frac{1}{W}\right) \quad (8)$$

Ukuran *window* maksimum W_{\max} didefinisikan sebagai nilai congestion *window* terakhir pada saat kejadian kehilangan paket dideteksi. Pada dasarnya hal ini menggambarkan estimasi kasar dari keseluruhan ukuran pipa (kanal). Persamaan (8) menseleksi nilai W dengan memilih point antara W sekarang dan W_{\max} yang diatur oleh parameter α . Parameter α mengendalikan tingkat keagresifan algoritma naik secara dinamis: dinaikkan dengan faktor dua dalam kasus paket *drop* dideteksi sebelum W mencapai nilai W_{\max} , sementara akan dikurangi dengan faktor yang sama untuk setiap peningkatan tanpa *error* dalam interval yang sama. Batas terendah W (lowerbound) adalah 2 (*initial value*) dengan menganggap bahwa nilai dibawah 2 menyebabkan kenaikan yang terlalu agresif.

Menurut persamaan (8) pertumbuhan nilai congestion *window* agresif ketika W bernilai rendah, sementara menjadi lamban ketika mendekati W_{\max} . Bagaimanapun, untuk memastikan desain fungsi yang agresif sebagai standard mekanisme TCP, kenaikan *window* minimum untuk satu per *window* :

$$W \leftarrow W + \max\left(\frac{W_{\max} - W}{\alpha \cdot W}, \frac{1}{W}\right)$$

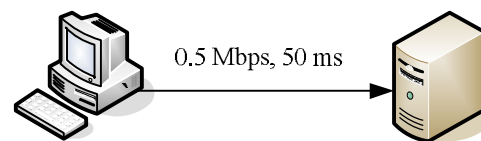
Fungsi yang dipakai pada paper ini sama dengan yang dipakai pada BIC TCP [3] dan TCP Logwestwood+ [1] tetapi dengan modifikasi pada nilai α . Pada simulasi ini digunakan nilai $\alpha = 0.01$

EVALUASI PERFORMANSI

Kinerja TCP Westwood+ yang diperbaiki (*enhancements*) dianalisa dengan simulator jaringan ns-2 versi 2.29.3. Modul TCP Westwood+ diperoleh dari [12], dan fungsi logaritmik fasa naik diperoleh dari [3] dan [1] yang dimodifikasi.

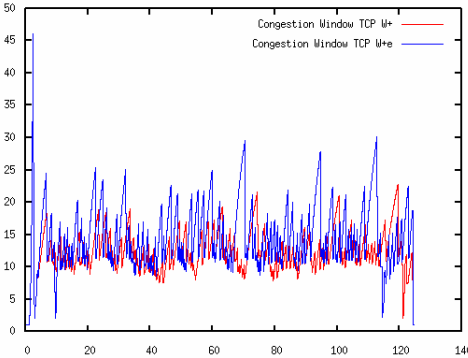
Skenario 1

Skenario simulasi 1 ditunjukkan pada Gambar 1. Skenario 1 merupakan konfigurasi *client server* dengan 2 node, yang dihubungkan 1 link dengan bit rates 0.5 Mbps dan propagation *delay* 50 ms.

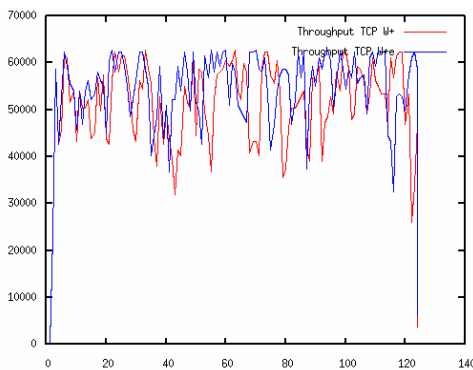


Gambar 1 Konfigurasi Client Server

Simulasi dengan NS 2 menghasilkan grafik congestion window vs waktu simulasi seperti Gambar. 2, *throughput* vs waktu simulasi seperti Gambar. 3 dan menghasilkan Tabel 1 untuk *error rate* yang divariasi, plot byte diterima vs error rate seperti Gambar. 4 dan plot *delay* vs *error rate* seperti Gambar. 5. Simulasi dijalankan dengan TCP Westwood+ (TCP W+) dan TCP Westwood+ enhancement (TCP W+e), dan hasilnya dibandingkan.



Gambar 2 Perbandingan Congestion Window TCP Westwood+ dan TCP Westwood+e



Gambar 3 Perbandingan Throughput TCP Westwood+ dan TCP Westwood+e

Table 1 Perbandingan Kinerja TCP Westwood+ dan TCP Westwood+e

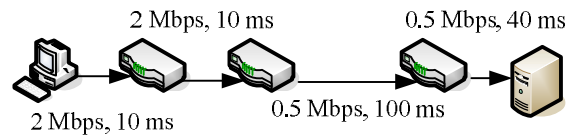
Error Rate	TCP Westwood +		TCP Westwood +e	
	Received Bytes	Average Delay end to end (sec)	Byte diterima	Average Delay end to end (sec)
0.01	6385904	0.059472	6795568	0.059472
0.02	4030336	0.059472	4973392	0.059472
0.03	2770560	0.059472	4012576	0.059472
0.04	2192768	0.059472	3385056	0.059472
0.05	1885520	0.059472	2919744	0.059472
0.06	1747584	0.059472	2745104	0.059472
0.07	1533872	0.059472	2225328	0.059472
0.08	1375216	0.059472	2114032	0.059472
0.09	1171568	0.059472	365264	0.059472
0.10	1080400	0.059472	1037776	0.059472

Gambar 2 menunjukkan perbandingan *congestion window* TCP W+ dan TCP W+e secara jelas pada konfigurasi *client server*. Gambar 2 juga menunjukkan ukuran *window* maksimum TCP W+e lebih besar dari TCP W+ dan berarti rata-rata ukuran *window* TCP W+e lebih baik dari TCP W pada konfigurasi *client server*.

Dari Tabel 1 diperoleh rata-rata kenaikan *byte* diterima TCP W+e terhadap TCP W+ sebesar 26,7 %. Sedangkan *delay* rata-rata baik TCP W+ maupun TCP W+e tidak berubah (tetap) untuk *error rate* yang bervariasi. Jadi untuk skenario 1 TCP W+e secara signifikan menambah jumlah *byte* yang diterima tanpa menambah *delay*.

Skenario 2

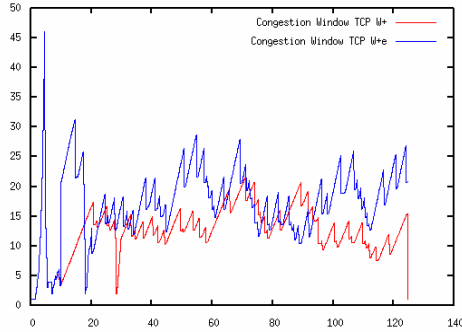
Simulasi skenario 2 ditunjukkan pada Gambar. 6 dan Gambar 11. Gambar.6 menggunakan variasi *bandwidth* dan *delay* propagasi sementara Gambar 11. menggunakan *bandwidth* seragam (sama). Gambar 6 merupakan konfigurasi *multihop* yang terdiri dari 5 node, yang dihubungkan 4 link dengan *bit rates* 2 Mbps, *delay* propagasi 10 ms; 0.5 Mbps , *delay* propagasi 100 ms, dan 0.5 Mbps, *delay* propagasi 40 ms



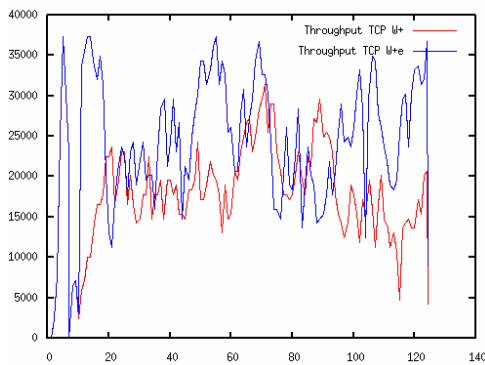
Gambar 4 Konfigurasi Multihop : Variasi Bandwidth

Simulasi dengan NS 2 menghasilkan grafik congestion window vs waktu simulasi seperti Gambar. 7, *throughput* vs waktu simulasi seperti Gambar. 8 dan menghasilkan Tabel 2 untuk *error rate* yang divariasi, plot byte diterima vs error rate seperti Gambar. 9 dan plot *delay* vs *error rate* seperti Gambar. 10. Simulasi dijalankan dengan TCP Westwood+ (TCP W+) dan TCP Westwood+ enhancement (TCP W+e), dan hasilnya dibandingkan.

Gambar 7 menunjukkan perbandingan *congestion window* TCP W+ dan TCP W+e secara jelas pada konfigurasi multihop dengan variasi *bandwidth*. Gambar 7 juga menunjukkan ukuran *window* maksimum TCP W+e lebih besar dari TCP W+ dan berarti rata-rata ukuran *window* TCP W+e lebih baik dari TCP W pada konfigurasi *multihop* dengan variasi *bandwidth*.



Gambar 5 Perbandingan Congestion Window TCP Westwood+ dan TCP Westwood+e

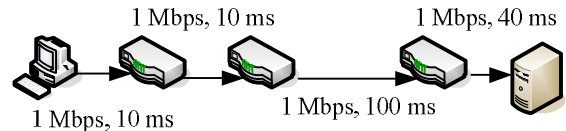


Gambar 6 Perbandingan Throughput TCP Westwood+ dan TCP Westwood+e

Dari Tabel 2 diperoleh rata-rata kenaikan *byte* diterima TCP W+e terhadap TCP W+ sebesar 8,7 %. Sedangkan *delay* rata-rata baik TCP W+ maupun TCP W+e berubah-ubah untuk *error rate* yang bervariasi. Pada konfigurasi multihop dengan variasi *bandwidth* TCP W+e menaikkan jumlah *byte* dengan penambahan *delay* 0.023611detik atau sebesar 12 %. *Delay* rata-rata TCP W+ sebesar 0.198191 detik dan TCP W+e sebesar 0.221803 detik.

Table 2 Perbandingan Kinerja TCP Westwood+ dan TCP Westwood+e

Error Rate	TCP Westwood +		TCP Westwood +e	
	Received Bytes	Average Delay end to end (sec)	Received Bytes	Average Delay end to end (sec)
0.01	2266768	0.207104	3044656	0.217724
0.02	1364560	0.200373	1685424	0.206095
0.03	1042512	0.199536	1329632	0.202672
0.04	849520	0.198524	1101120	0.205923
0.05	699152	0.197954	671328	0.211771
0.06	613312	0.197253	935952	0.209192
0.07	509712	0.196139	383024	0.224022
0.08	503792	0.196197	420912	0.234894
0.09	435120	0.195794	100048	0.245640
0.10	410848	0.195934	568320	0.235044



Gambar 7 Konfigurasi Multihop bandwidth Seragam

Konfigurasi pada Gambar 11. menggunakan *bandwidth* seragam (sama). Gambar 11 merupakan konfigurasi multihop yang terdiri dari 5 node, yang dihubungkan dengan 4 link dengan bit rates 1 Mbps, *delay* propagasi 10 ms; 1 Mbps, *delay* propagasi 100 ms, dan 1 Mbps, *delay* propagasi 40 ms dan *bandwidth* divariasi mulai dari 1 Mbps sampai 10 Mbps dengan *error rate* konstan 0.01.

Simulasi dengan NS 2 menghasilkan Table 3 ketika *bandwidth* divariasi, plot *byte* diterima vs *error rate* seperti Gambar. 12 dan plot *delay* vs *error rate* seperti Gambar. 13. Simulasi dijalankan dengan TCP Westwood+ (TCP W+) dan TCP Westwood+ enhancement (TCP W+e), dan hasilnya dibandingkan.

Tabel 3 menunjukkan perbandingan kinerja TCP W+ dan TCP W+e secara jelas pada konfigurasi multihop dengan variasi *bandwidth*. Tabel 3 juga menunjukkan ukuran window maximum TCP W+e lebih besar dari TCP W+ dan berarti ukuran rata-rata window TCP W+e lebih baik dari TCP W pada konfigurasi multihop dengan variasi *bandwidth*.

Table 3 Perbandingan Kinerja TCP Westwood+ dan TCP Westwood+e

Bdan width (Mbps)	TCP Westwood +		TCP Westwood +e	
	Received Bytes	Average Delay end to end (sec)	Received Bytes	Average Delay end to end (sec)
1	3295704	0.178941	3765752	0.178941
2	3454952	0.169470	4068264	0.169470
3	3252488	0.166314	4057016	0.166314
4	3269656	0.164735	4049320	0.164735
5	3363784	0.163788	4142856	0.163788
6	3406408	0.163157	3945720	0.163157
7	3536648	0.162706	4086024	0.162706
8	3453768	0.162368	4147000	0.162368
9	3441928	0.162105	4240536	0.162105
10	3404040	0.161894	4257112	0.161894

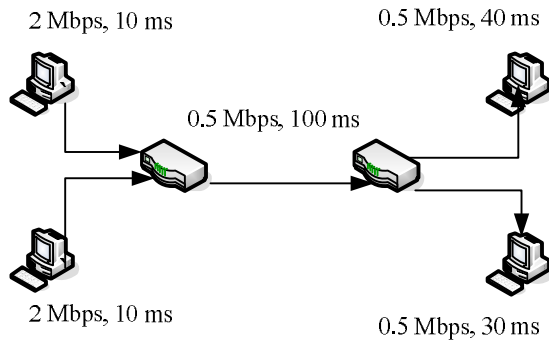
Dari Tabel 3 diperoleh rata-rata kenaikan *byte* diterima TCP W+e terhadap TCP W+ sebesar 20,35 %. Sedangkan *delay* rata-rata baik TCP W+ maupun TCP W+e tidak berubah untuk *bandwidth* yang sama. Pada konfigurasi multihop dengan variasi *bandwidth* TCP W+e menaikkan jumlah *byte* tanpa penambahan *delay*.

Gambar. 12 menunjukkan jumlah byte diterima TCP W+e selalu lebih besar dari TCP W+, hal ini berarti bahwa byte diterima TCP W+e pada jaringan multihop tanpa *bottleneck* selalu lebih baik dari TCP W+.

Pada konfigurasi multihop : *bandwidth* seragam TCP W+e dapat menaikkan byte yang diterima tanpa menambah *delay*. *Delay* rata-rata TCP W+ sebesar 0.165548 sementara *delay* rata-rata TCP W+e juga sebesar 0.165548. Gambar 13 menunjukkan nilai *delay* TCP W+ dan TCP W+e tepat berimpit. Dengan menaikkan *bandwidth* akan menyebabkan *delay* terus berkurang hingga mendekati nilai 0.16 detik.

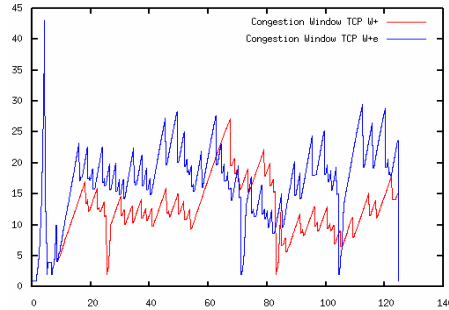
Skenario 3

Simulasi skenario 3 menggunakan konfigurasi *bottleneck* seperti ditunjukkan pada Gambar. 14 dan Gambar. 19. Gambar. 14 terdiri dari 6 node, yang dihubungkan 5 link dengan bit rates 2 Mbps, *delay* propagasi 50 ms; 0.3 Mbps, *delay* propagasi 100 ms; 0.5 Mbps, *delay* propagasi 40 ms; dan 0.5 Mbps, *delay* propagasi 30 ms.

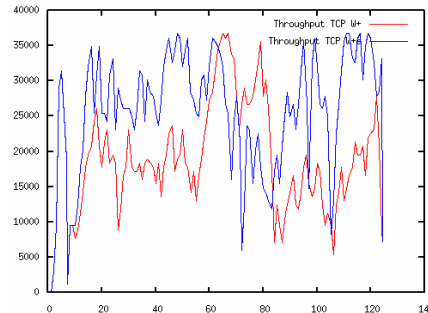


Gambar 8 Konfigurasi Bottleneck : 2 Sumber TCP

Simulasi dengan NS 2 menghasilkan grafik *congestion window* vs waktu simulasi seperti Gambar. 14, *throughput* vs waktu simulasi seperti Gambar. 15 dan menghasilkan Tabel 4 untuk *error rate* yang divariasi, plot byte diterima vs *error rate* seperti Gambar. 16 dan plot *delay* vs *error rate* seperti Gambar. 17. Simulasi dijalankan dengan TCP Westwood+ (TCP W+) dan TCP Westwood+ enhancement (TCP W+e), dan hasilnya dibandingkan.



Gambar 9 Perbandingan Congestion Window TCP Westwood+ dan TCP Westwood+e



Gambar 10 Perbandingan Throughput TCP Westwood+ & TCP Westwood+e

Table 4 Perbandingan Kinerja TCP Westwood+ dan TCP Westwood+e

Error Rate	TCP Westwood +		TCP Westwood +e	
	Received Bytes	Average Delay end to end (sec)	Received Bytes	Average Delay end to end (sec)
0.01	2385760	0.195110	32358912	0.204339
0.02	1447440	0.188281	1947720	0.191003
0.03	1202352	0.188345	1446296	0.196712
0.04	989824	0.187999	1271656	0.190468
0.05	813408	0.186812	692088	0.203736
0.06	726384	0.185935	814632	0.213995
0.07	674288	0.185433	882120	0.212861
0.08	592000	0.184867	726424	0.229451
0.09	554112	0.184340	388392	0.217766
0.10	520960	0.183899	190664	0.224451

Gambar 14 menunjukkan perbandingan *congestion window* TCP W+ dan TCP W+e secara jelas pada konfigurasi *bottleneck*. Gambar 14 juga menunjukkan ukuran *window* maksimum TCP W+e lebih besar dari TCP W+ dan berarti rata-rata ukuran *window* TCP W+e lebih baik dari TCP W+ pada konfigurasi *bottleneck*.

Dari Tabel 4 diperoleh rata-rata kenaikan *byte* diterima TCP W+e terhadap TCP W+ sebesar 7,6 %. Sedangkan *delay* rata-rata baik TCP W+ maupun TCP W+e berubah-ubah untuk

error rate yang bervariasi. Pada konfigurasi bottleneck dengan variasi *error rate* TCP W+e menaikkan jumlah *byte* dengan penambahan *delay* 0.021376 detik atau sebesar 2,1 %. *Delay* rata-rata TCP W+ sebesar 0.187102 detik dan TCP W+e sebesar 0.208478 detik.

Penambahan *delay* rata-rata *end-to-end* disebabkan ukuran window TCP W+e lebih besar TCP W+ sementara dari link node 2 menuju node 3 terjadi *bottleneck* dan jumlah *buffer* hanya 20, hal ini berarti bahwa paket akan diantri di *buffer*, yang akhirnya menambah *delay*.

KESIMPULAN

Modifikasi fasa naik TCP Westwood+ dari aditif menjadi multiplikatif (TCP Westwood+e) dalam jaringan kabel (*wired*) secara umum dapat menaikkan *throughput* byte yang diterima. Pada konfigurasi *client server* menaikkan *throughput* 26,7 %, konfigurasi multihop dengan variasi *error rate* menaikkan *throughput* 8,3 %, konfigurasi *multihop* dengan variasi *bandwidth* menaikkan *throughput* 20,35 % dan konfigurasi *bottleneck* menaikkan *throughput* 7,6 %.

Perbaikan *throughput* pada konfigurasi *client server* tidak disertai dengan penambahan *delay*. Perbaikan *throughput* pada konfigurasi *multihop* dengan variasi *error rate* menambah *delay* 12 % dan perbaikan *throughput* pada konfigurasi *multihop* dengan variasi *bandwidth* tidak menambah *delay*. Perbaikan *throughput* pada konfigurasi *bottleneck* menambah *delay* 2,1 %. Penambahan *delay* disebabkan karena pembatasan *buffer* di node. Modifikasi fasa naik dari aditif menjadi multiplikatif tidak mengurangi *fairness* TCP Westwood+e.

DAFTAR RUJUKAN

- Dzmitry Kliazovich, Fabrizio Granelli, Daniele Miordani, "TCP Westwood+ Enhancement in High-Speed Long-Distance Networks" in Proc. of IEEE ICC 2006
- L. A. Grieco dan S. Mascolo, "Performance evaluation dan Perbandingan of Westwood+, New Reno, dan Vegas TCP congestion control," ACM Comp. Comm. Rev., vol. 34, no. 2, pp. 2538, Apr. 2004.
- Lisong Xu; K. Harfoush, dan Injong Rhee, "Binary increase congestion control (BIC) for fast long-distance networks", in Proc. of

- IEEE INFOCOM, vol. 4, pp. 2514 – 2524, March 2004, Hong Kong.
- T. Kelly, "Scalable TCP: improving performance in highspeed wide area networks," ACM Computer Communication Review, no. 2, April 2003.
- S. Floyd, "HighSpeed TCP for large congestion windows," RFC 3649, December 2003.
- V. Jacobson, "Congestion Avoidance dan Control", in Proc. of ACM SIGCOMM, pp. 314 – 329, August 1988, Stanford, CA.
- S. Mascolo, L. A. Grieco, R. Ferorelli, P. Camarda, G. Piscitelli, "Performance Evaluation of Westwood+ TCP Congestion Control, Extended version of the paper "Live Internet Measurements Using Westwood+ TCP Congestion Control", IEEE Globecom 2002 conference.
- Chadi Barakat, Eitan Altman, dan Walid Dabbous, "On TCP Performance in a Heterogeneous Network: A Survey, IEEE Magazine, INRIA, January 2000.
- NS-2 Simulator. <http://www.isi.edu/nsnam/ns>
- Eitan Altman, Tania Jimenez, "NS Simulator for beginners", Univ. de Los Andes, Venezuela dan ESSI, Sophia-Antipolis, France, December 2003
- S. Mascolo, C. Casetti, M. Gerla, M. Sanadidi, dan R. Wang, "TCP Westwood: End-to-end Bandwidth Estimation of Efficient Transport over Wired dan Wireless Networks", in Proc. of ACM Mobicom, July, 2001, Rome, Italy.
- TCP Westwood+ modules for ns2, <http://193.204.59.68/mascolo/tcp%20westwood/modules.htm>