Metode Pengaturan *Throughput* untuk TCP Westwood+ pada Saluran *Bottleneck*

Hilal Hudan Nuha*, Fazmah Arif Y.**

*hilalnuha@gmail.com, ** fay@ittelkom.ac.id

Pasca Sarjana Teknik Informatika IT Telkom

Jln. Telekomunikasi no 1. Dayeuhkolot. Bandung

Abstrak

TCP mempunyai banyak jenis pengembangan. Untuk jaringan dengan delay tinggi seperti seluler dan satelit seperti HSDPA maka TCP Westwood sering dipakai untuk protokol transport. Varian dari Westwood sendiri yaitu Westwood+ mempunyai kelebihan karena bersifat adaptif dalam menghadapi jaringan dengan loss tinggi. Pada linux kernel 2.6 TCP westwood sudah tersedia dan tinggal diaktivasi.

TCP westwood+ sendiri mempunyai beberapa parameter diantaranya *increment factor* yang akan dipakai pada saat fase congestion avoidance pada algoritma Additive Increase. Secara default *increment factor* bernilai 1 dan bisa diubah. Pada makalah ini diujicobakan mengubah nilai *increment factor* tadi dan melihat pengaruhnya pada *throughput*. Dari hasil simulasi pada Network Simulator 2 bisa disimpulkan bahwa jika kita mengubah nilai *increment factor* tersebut pada jaringan *bottleneck* dengan nilai sembarang x^2 dengan x^2 diantara 0 sampai 1 maka pada *throughput* yang dihasilkan ternyata relatif sebanding dengan nilai x. pada kondisi *bottleneck* ternyata pengaturan besar *throughput* pada TCP Westwood+ bisa dilakukan dengan mengganti nilai *increment factor* dengan nilai tertentu sedangkan pada kondisi yang longgar atau traffik rendah maka perilaku *throughput* akan sama.

I. Pendahuluan

Internet sendiri didominasi penggunaan protokol HTTP pada layer aplikasi dan Transmission Control Protocol(TCP) adalah protokol utama yang dipakai pada internet pada layer transport. TCP sendiri mempunyai banyak jenis pengembangan. Untuk jaringan dengan delay tinggi seperti seluler dan satelit seperti HSDPA maka TCP Westwood sering dipakai untuk protokol transport. Varian dari Westwood sendiri yaitu Westwood+ mempunyai kelebihan karena bersifat adaptif dalam menghadapi jaringan dengan loss tinggi. Pada linux kernel 2.6 TCP westwood sudah tersedia dan tinggal diaktivasi. Sehingga TCP Westwood+ layak untuk dianalisis lebih lanjut karena pasar dan pemakaiannya yang bersifat luas.

Tujuan dari penelitian ini adalah mencari cara pengaturan besaran *throughput* dengan mengubah parameter yang ada pada TCP. Pada makalah ini akan dilakukan percobaan mengenai pengaruh *increment factor* pada *throughput* TCP westwood+. Hal ini sangat bermanfaat untuk provider untuk mengendalikan *throughput* pada subscriber pada kondisi sibuk dan untuk mengklasifikan QoS berdasarkan paket konsumsi yang dipakai.

II. Dasar Teori

2.1. Transmission Control Protocol(TCP) dan Congestion Control

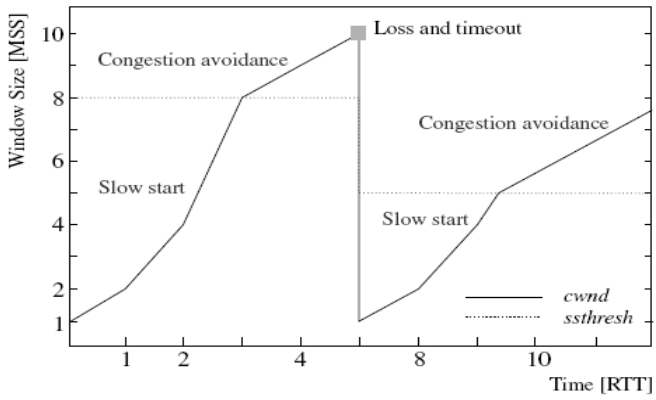
TCP dirancang untuk protocol host to host yang handal dalam jaringan komunikasi komputer dan interkoneksi dari jaringan yang ada. TCP mengintrepetasikan sebuah packet loss sebagai tanda bahwa terjadi kongesti(antrian membludak) dalam jaringan dan menanggapinya dengan mengurangi ukuran window. Ukuran window atau dinotasikan dengan cwnd adalah jumlah paket yang dikirim dalam satu kali pengiriman. Selain itu untuk menandai bahwa paket tersebut sampai maka terdapat acknowledgement sebagai penanda bahwa paket yang dikirim sudah sampai.

Van Jacobson memperkenalkan *Congestion Control* sebagai solusi atas terjadinya kongesti dalam jaringan. Setidaknya ada dua bagian dalam algoritma kontrol kongesti yang pertama adalah fase *Slowstart Algorithm* dimana pertama kali koneksi dibuka maka nilai cwnd atau W berlaku persamaan berikut:

W <- W +1 per ACK

Atau W <- **2**W per RTT

Sehingga ukuran cwnd meningkat secara cepat dan tumbuh secara eksponensial. Hal ini dimaksudkan agar pengiriman data segera dimulai dengan ukuran sebesar-besarnya sampai batas tertentu. Batas inilah yang disebut sebagai *ssthresh* atau *slowstart threshold*. Ketika cwnd mencapai nilai lebih dari *ssthresh* maka TCP memasuki fase *Congestion Avoidance* yang mempunyai persamaan berbeda tergantu dari variansi TCP yang dipakai.



Gambar 2.1 Ilustrasi Ukuran W pada TCP reno

Gambar 2.1 menggambarakan ukuran W pada fase slowstart dan pada congestion avoidance. Pada TCP secara umu, algoritma congestion avoidance adalah sebagai berikut

- ACK: W← W + 1/W →(additive increase)
- LOSS: W← W/2 →(multiplicative decrease)

Sehingga a sering disebut *increment factor*. Algoritma congestion avoidance TCP reno termasuk pada Additive Increase Multiplicative Decrease(AIMD), yang mempunyai pola:

- ACK: W← W + a/W
- LOSS: W← b*W

Nilai a dan b bervariasi misalkan pada TCP Reno nilai **a**=1, **b**= ½ . Misalkan pada kondisi tersebut kita simbolkan *throughput* dengan nilai Th. Pada penelitian sebelumnya jika kita mensubtitusi a dengan x^2=**(** ½$)^2$=¼ ,maka *throughput* akan berubah menjadi ½ Th atau setengah dari *throughput* sebelumnya. Hal ini berarti jika kita mengganti nilai **a** dengan x^2, dimana x^2berada pada 0 sampai 1 maka *throughput* akan berubah nilainya dengan x kali Th (Honda, 2008).

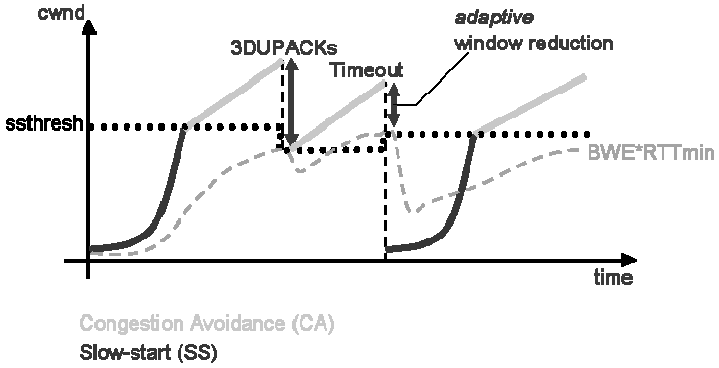
2.2 TCP Westwood+

TCP westwood+ merupakan pengembangan dari TCP westwood dan mempunyai kelebihan dalam hal mengatasi jaringan dengan loss tinggi dan delay yang panjang seperti satelit dan seluler. Dalam linux kernel 2.4, TCP westwood+ sudah terpasang dan bisa dipakai dengan konfigurasi tertentu.

TCP westwood+ mempunyai Slowstart Algorithm yang sama dengan TCP Reno yaitu :

W ← W +1 per ACK atau W ← 2W per RTT

Algoritma Slowstart membuat W meningkat secara cepat dan eksponensial sampai W menyentuh nilai ssthresh. Pada kondisi tersebut TCP masuk pada algoritma Congestion avoidance. Gambar 2.2 menggambarkan tingkah laku TCP westwood+ pada jaringan.



Gambar 2.2 Ilustrasi Ukuran W pada TCP Reno

Berbeda dengan TCP Reno, TCP westwood+ mempunyai algoritma Congestion Avoidance yang termasuk kedalam Addittive Increase Adaptive Decrease(AIADD) karena mempunyai tingkah laku yang berbeda dengan Reno pada kondisi loss yang mana pada westwood+ bersifat adaptif.

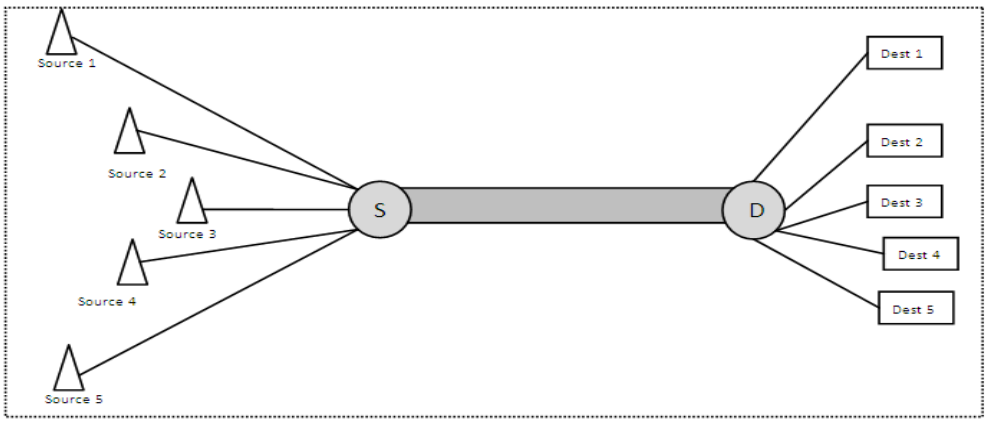
- ACK: W← W + a/W, →(additive increase, **a**=1)
- LOSS: W← max(2, BWE . RTTmin / Seg_Size), →(adaptive decrease)
    - BWE : available bandwidth end-to-end
    - RTTmin : minimum Round Trip Time measured during connection and
    - Seg_size : TCP segment (bit)

Pada penelitian sebelumya(Honda, 2008), telah dilakukan modifikasi TCP Reno sehingga *throughput* bisa diskalakan atau dikendalikan dengan mengubah *increment factor*. Pada makalah ini akan dilakukan hal yang sama pada TCP westwood+ dimana akan diubah algoritma Additive Increase dengan mengganti nilai a dengan x^2. Sehingga algoritma akan berubah menjadi berikut.

- ACK: W← $(W + Dn^2/W)$
- LOSS: W← (max(2, BWE . RTTmin / Seg_Size))
  Diharapkan dengan modifikasi di atas dapat juga mengubah nilai thoughput pada jaringan.

III. Rancangan Percobaan

Pada percobaan kali ini akan diujicobakan algoritma di atas pada *Network Simulator* 2. Akan dilakukan 4 kondisi *bottleneck* yang berbeda untuk melihat perilaku TCP westwood+ dengan modifikasi *increment factor*. Gambar 3.1 akan menggambarkan topologi jaringan pada percobaan.



Gambar 3.1 Desain Percobaan

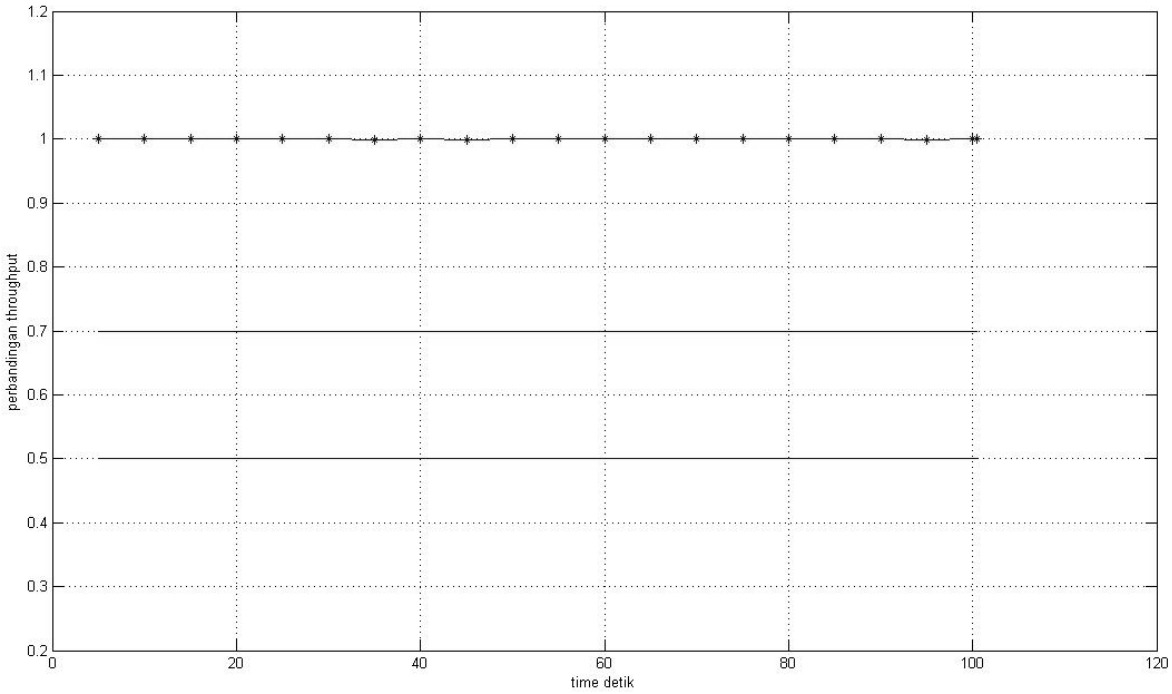
Tujuan dari percobaan ini adalah mencari hubungan antara pengaruh nilai *increment factor*(a) pada *throughput* jaringan pada berbagai ukuran *bottleneck*. Terdapat 5 buah koneksi yang menghubungkan antara source dengan destination. Source 1 dan 2 akan mempunyai x=0,2; 0.5; dan 0,7 dan a=x^2.sedangkan koneksi yang lainnya yaitu 3,4,dan 5 akan mempunyai nilai normal a=1. Sambungan antara source dengan node S bernilai 3Mbps begitu pula untuk Node D dengan destination sedangkan node S dengan D mempunyai *bottleneck* 15, 5, 1, dan 0.5 Mbps.

Akan dibandingkan *throughput* untuk koneksi dengan a kurang dari 1 dan a bernilai normal atau bernilai 1 pada masing-masing kondisi *bottleneck* yang akan disample *throughput*nya setiap 5 detik selama 100 detik.

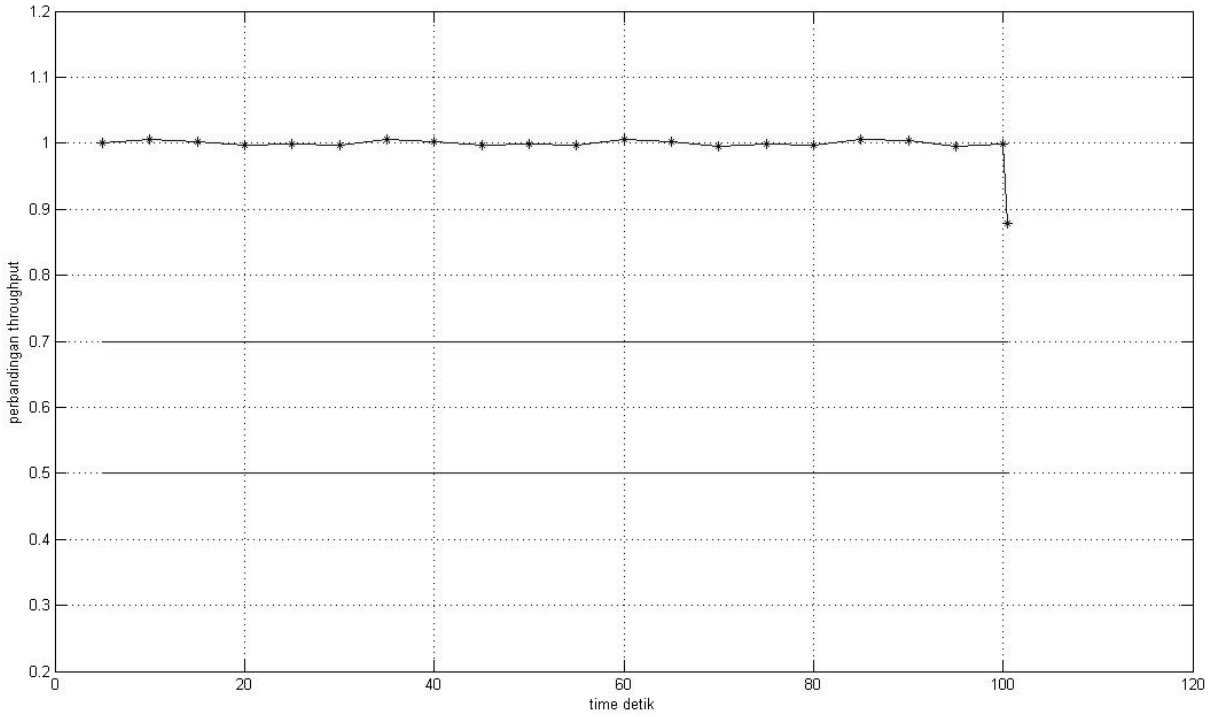
IV. Analisis Hasil

Pada sekenario pertama akan terdapat 5 buah *throughput* untuk masing masing koneksi. Agar *throughput* tersebut bisa dibandingkan maka *throughput* koneksi yang diatur (*source* 1 dan 2) akan diambil rata-ratanya pada untuk setiap detik dan dibandingkan dengan rata-rata througput koneksi yang normal(source 3,4,dan 5).

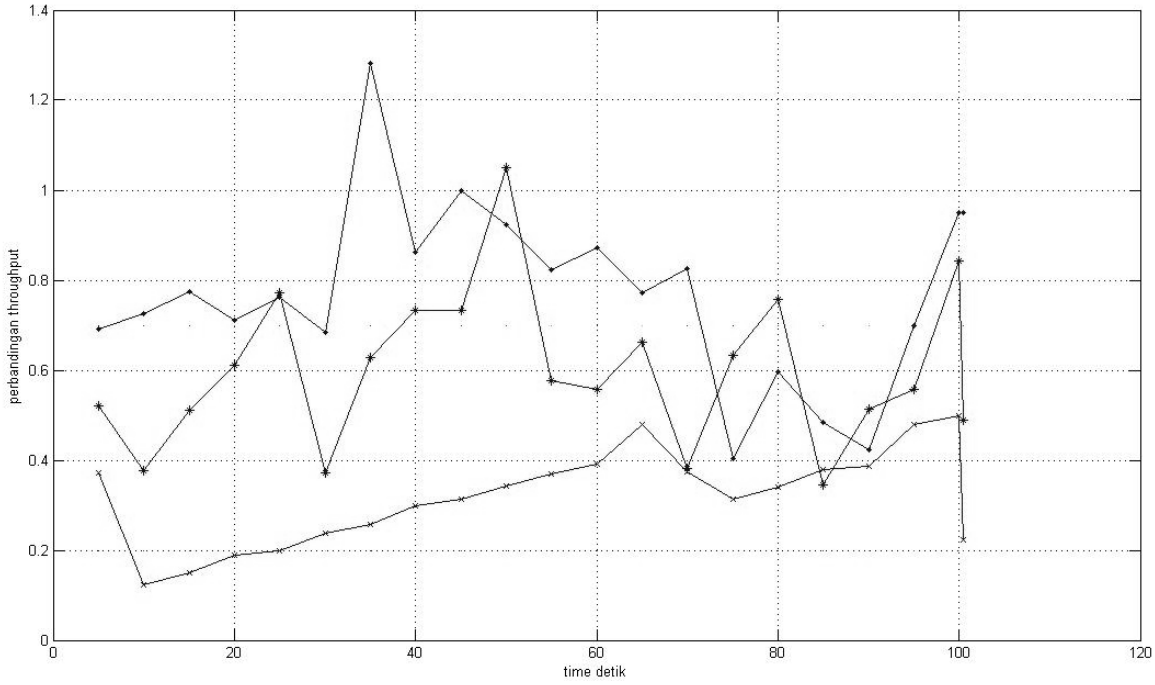
Gambar 4.1 dan 4.2 adalah grafik perbandingan rata-rata *throughput* dengan nilai x=0.2, 0.5, dan 0.7 dan x=1 pada berbagai kondisi *bottleneck*. Untuk kurva x=0.2 disimbolkan dengan tanda silang (**x**), kurva x=0.5 disimbolkan dengan tanda bintang (*****), dan kurva x=0.7 disimbolkan dengan tanda titik(**.**).



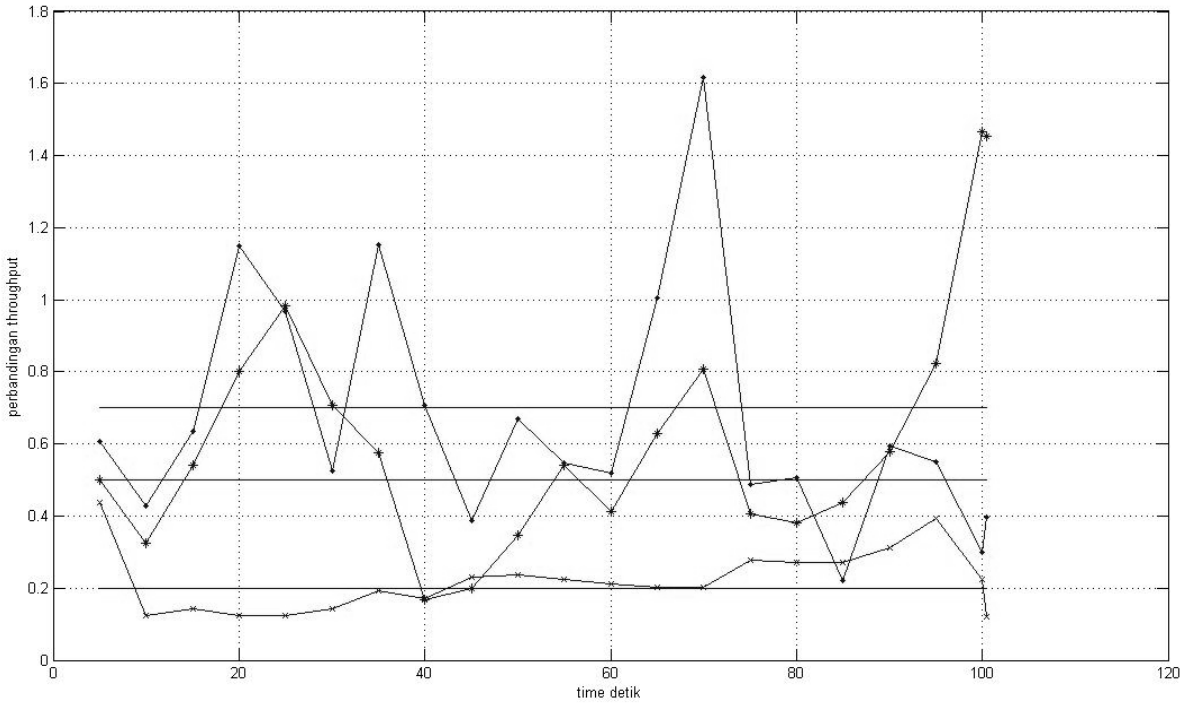
Gambar 4.1 Grafik Perbandingan Rata-Rata Throughput dengan Skala dan Normal(bottleneck=15Mbps).



Gambar 4.2 Grafik Perbandingan Rata-Rata Throughput dengan Skala dan Normal(bottleneck=5Mbps)



Gambar 4.3 Grafik Perbandingan Rata-Rata Throughput dengan Skala dan Normal(bottleneck=1Mbps)



Gambar 4.4 Grafik Perbandingan Rata-Rata Throughput dengan Skala dan Normal(bottleneck=0.5Mbps)

Gambar 4.1 dan 4.2 memperlihatkan kondisi *throughput* pada kondisi traffik lebih rendah dari *bottleneck*. Bisa dilihat pada semua kurva berimpit dan bernilai 1 yang berarti semua koneksi mempunyai nilai yang sama dalam hal *throughput*. Hal ini juga berarti pengubahan nilai *increment factor* tidak berpengaruh pada *throughput* jaringan pada kondisi *bottleneck* besar. Hal ini disebabkan karena kongesti atau kepadatan tidak terjadi pada *bottleneck* S ke D.

Hal tersebut berbeda dengan yang terjadi pada gambar 4.3 dan 4.4. terlihat jelas bahwa tiap koneksi dengan *increment factor* berbeda mempunyai kecenderungan *throughput* berbeda. Misalnya kurva dengan x=0.2 cenderung di bawah kurva x=0.5 dan x=0.7 dimana $a=x^2$. Secara nilai sendiri, kurva x=0.7 mendekati nilai 0.7 begitu juga untuk kurva 0.2 dan 0.5. hal ini terjadi karena kongesti atau kepadatan jaringan terjadi begitu sering sehingga setiap kali source akan melakukan peningkatan nilai cwnd kongesti terjadi dan adaptive decrease pun dipanggil sekali lagi setelah itu naik ke additive increase dengan nilai *increment factor* kecil sehingga membutuhkan waktu yang lama dalam meningkatkan jumlah paket yang dikirim yang berakibat pada *throughput* yang kecil pula. Semakin kecil *increment factor* akan semakin kecil pula perbandingan *throughput* dengan koneksi yang mempunyai *increment factor* normal.

V. Kesimpulan dan Saran

Dari beberapa percobaan di atas ada beberapa hal yang bisa menjadi penekanan dari hasil analisis. Yang pertama yaitu bahwa pada kondisi dimana trafik jaringan rendah, nilai *increment factor* tidak berpengaruh pada *throughput* sebuah node. Yang kedua, yaitu pada kondisi *bottleneck* yang sempit dimana kongesti atau kepadatan traffik mudah terjadi maka nilai *increment factor* dari suatu node dengan koneksi TCP akan mempengaruhi *throughput*. Semakin kecil nilai *increment factor*, semakin kecil pula *throughput* pada *bottleneck*. Nilai *throughput* tersebut cenderung sebanding dengan x dimana $x=a^2$ dan a adalah *increment factor*. Hal ini bisa dimanfaatkan untuk mengatur nilai *throughput* dari suatu koneksi pada *bottleneck*.

Saran untuk implementasi dan percobaan berikutnya adalah:

a. Pengaruh *Buffer* pada *Queue* pada *router* dan jumlah *user* yang mengakses *Bottleneck* terhadap metode ini.

b. Pengaruh *Packet Loss Rate* pada *wireless* terhadap *throughput* pada kondisi *increment factor* yang berubah ubah.

VI. Daftar Pustaka

Allman, M. 1999. "*TCP Congestion Control*". RFC2581.
Assaad, Mohamad. "*TCP Performance over UMTS-HSDPA Systems*".
DARPA. 1981. "*Transmission Control Protocol*".
Heckmann, Oliver. "*The Competitive Internet Service Provider*".
Honda, Michio. 2008. "*Bidimensional-Probe Multipath Congestion Control for Shared Bottleneck Fairness*". Master Thesis.
Issariyakul, Teerawat. "*Introduction to Network Simulator NS2*".
Padhye, Jitendra. "*TCP Throughput modeling*"
Sanjaya. 2007. "*Analisis TCP Westwood pada HSDPA*". Master Thesis. IT Telkom
Sukiswo. "*Perbaikan Tcp Westwood +*".
Van. Jacobson, 1988. "*Congestion avoidance and control*", *ACM SIGCOMM*, August

Biodata Pengirim:

Hilal Hudan Nuha ST.

NIM: 213090008

Mahasiswa Pasca Sarjana Teknik Informatika IT Telkom

Alamat: Jalan telekomunikasi no 1 Dayeuhkolot Kab. Bandung

No HP: 085724041947