

Metode Penentuan Ukuran Congestion Window Menggunakan Parameter Decrease dan Increase

Ranny

Universitas Indonesia, Depok, Indonesia

Diterima 30 November 2012

Disetujui 15 Desember 2012

Abstract—Permasalahan pada sebuah jaringan terus berkembang seiring dengan peningkatan jumlah jaringan yang terbentuk. Peningkatan jumlah jaringan terjadi karena peningkatan jumlah dan jenis dari pengguna jaringan. Hal ini menyebabkan sistem yang ada sekarang harus terus disesuaikan dengan kebutuhan pengguna jaringan. Salah satu yang terus digunakan adalah algoritma TCP untuk mengatur pengiriman data, paket *drop* dan *congestion window*. TCP yang biasa digunakan masih dianggap kurang efektif dalam melayani pengiriman data terutama jika terjadi banyak paket *drop*. Jika banyak terjadi paket *drop* otomatis waktu yang dibutuhkan sebuah paket untuk sampai pada tujuan akan membutuhkan waktu yang lebih lama. Untuk itu dilakukan beberapa percobaan untuk mencari solusi yang bertujuan mengurangi *drop*. Salah satunya adalah dengan memodifikasi algoritma TCP. Solusi yang dijabarkan adalah dengan menambah parameter *decrease* dan *increase* yang mampu mengatur *congestion window* sedemikian sehingga mampu mengurangi *drop*.

Index Terms—*Congestion window*, paket *drop*, parameter *increase* dan *decrease*, TCP.

I. PENDAHULUAN

Pada jaringan internet yang ada sekarang, menuntut mampu menangani dan melayani berbagai kebutuhan pengguna internet. Kebutuhan dan kecepatan sebuah jaringan internet tidak hanya dilihat dari seberapa cepat sebuah jaringan mampu membuka sebuah halaman jaringan, tapi juga dilihat seberapa cepat pengguna mampu menerima dan mengirim sebuah file dalam ukuran besar. Perkembangan jaringan internet tidak sebanding dengan perkembangan kebutuhan pengguna ini. Sehingga terus dilakukn berbagai penelitian dan pengembangan pada jaringan internet agar kebutuhan pengguna dapat terpenuhi. Pengembangan jaringan internet yang dilakukan diharapkan tidak banyak merubah bentuk jaringan secara fisik namun dengan mengembangkan berbagai algoritma yang mampu menangani kebutuhan pengguna internet.

Saat ini bentuk jaringan yang ada banyak

menggunakan sistem *slow start*. Dimana sistem ini menentukan ukuran *window* (*window size*) secara perlahan untuk menghindari *congestion*. Namun pada beberapa kasus, sistem ini tidak berjalan dengan optimal. Salah satunya adalah menangani pengiriman sebuah file dalam paket yang berukuran besar pada bandwidth yang mempunyai delay yang besar. Dengan kondisi delay yang besar, maka sebuah pengiriman paket yang besar akan menyebabkan peningkatan *window size* menjadi lambat.

II. PERMASALAHAN

Pada umumnya untuk menentukan *congestion window size* tidak bergantung dari seberapa besar paket yang dikirimkan. Metode yang umumnya digunakan adalah dengan menentukan *congestion window size* dengan metode *slow start*. Pada metode ini, pada setiap awal pengiriman data yang dibagi menjadi beberapa paket, *window size* yang digunakan adalah maksimum, tapi pada saat pengiriman paket kedua, *window size* dibagi menjadi setengah ukuran maksimalnya. Hal ini dilakukan untuk mencegah terjadinya antrian yang menyebabkan *delay*. Namun, hal ini justru membuat pengiriman data secara keseluruhan menjadi lambat, karena banyak terjadi *drop* pada setiap pengiriman data. Selain *drop* juga terjadi penurunan *window size* yang disesuaikan dengan ukuran data. Semakin besar data yang dikirimkan akan semakin banyak terjadi *drop* karean *window size* yang mengalami penurunan ukuran *window*. Setelah terjadi penurunan ukuran *window* ini, otomatis sistem akan kembali berusaha untuk meningkatkan ukuran *window* sesuai dengan ukuran data dan bandwidth. Sehingga proses ini akan memerlukan waktu yang lebih panjang untuk mengirimkan sebuah data dalam satu round trip. Dalam satu round trip pengiriman data akan memakan waktu untuk *congestion window*, juga untuk melakukan *drop*. Waktu yang dibutuhkan untuk sekali round trip akan bergantung pada ukuran data dan *congestion window*.

Pada simulasi jaringan NS, digunakan jenis TCP

untuk pengaturan permasalahan ini. Pada NS, TCP telah diberikan nilai *default* dan algoritma tertentu yang dinilai cukup untuk mengatur *congestion window* secara umum. Pada paper acuan dilakukan modifikasi dari pengaturan TCP ini dengan melihat hubungan antara ukuran data dengan ukuran *window* yang mempengaruhi *congestion window*. Selain dilihat berdasarkan ukuran data, pada paper ini juga akan dilihat hubungan dari kecepatan pengaturan TCP ini dengan AQM (*Active Queue Management*).

III. INCREASE AND DECREASE CONGESTION WINDOW METHOD

Penggunaan parameter *increase* dan *decrease* ini terletak pada TCP. Pada algoritma TCP ditambahkan kedua parameter ini untuk menentukan besarnya *congestion window*. Pendekatan yang dilakukan untuk pemecahan masalah ini dengan memodifikasi algoritma serta rumus yang biasa digunakan pada TCP. Pada umumnya untuk menentukan besarnya *congestion window*, TCP tidak bergantung pada besarnya data yang dikirimkan. TCP yang biasa digunakan mengakibatkan banyak terjadi *drop* dan pemulihan ukuran *congestion window* menjadi lambat akibat *drop* yang terlalu banyak. Tujuan dari modifikasi ini adalah agar tidak terjadi *double congestion window*, yang mengakibatkan data berjalan dengan lambat. Dengan melihat peningkatan nilai dari paket data yang dikirimkan dan penurunan dari jumlah *drop* akibat dari modifikasi, akan membuktikan keefesienan dari hasil modifikasi. Berikut adalah persamaan untuk *increase* ($a(w)$) dan *decrease* ($b(w)$), pada saat dua titik (P, W) dan ($P1, W1$). Dimana $W1$ paket adalah nilai dari *congestion window*.

$$a(w) = \frac{w^2 * 2.0 * b(w) * p(w)}{2.0 - b(w)} \quad (1)$$

Dimana p adalah paket *drop* pada $W1$. Berikut persamaan untuk mendapatkan $b(w)$.

$$b(w) = (B - 0.5) \frac{\log(w) - \log(W)}{\log(W1) - \log(W)} \quad (2)$$

Dengan mengubah nilai default *increase* dan *decrease* pada TCP biasa dengan $a(w)$ dan $b(W)$ di atas, maka diharapkan akan mampu mempercepat proses pengiriman data. Hal ini dikarenakan nilai *increase* dan *decrease* yang digunakan untuk menentukan *congestion window* disesuaikan dengan paket *drop* dan data yang dikirimkan. Sehingga ukuran *congestion window* akan lebih fleksibel dan mengurangi paket *drop* karena ukuran *congestion window* berubah-ubah sesuai ukuran datanya.

Dengan adanya modifikasi di atas yang

mengakibatkan *congestion window* dapat mempercepat pemulihan paket *drop*, maka diharapkan sebuah round trip time untuk sebuah paket data pun akan menurun. Artinya waktu yang dibutuhkan sebuah data untuk sampai ke tujuan akan lebih cepat dengan pengurangan paket yang *drop*.

IV. EKSPERIMEN DAN ANALISA

Eksperimen dibagi menjadi dua yaitu dengan membandingkan nilai *congestion window* dengan nilai *decrease rate*, eksperimen kedua dengan membandingkan nilai *congestion window* dengan nilai *increase rate*. Dengan membandingkan nilai *congestion window* dengan kedua nilai parameter, maka akan terlihat pengaruh antara penambahan kedua parameter pada TCP dengan nilai *congestion window*.

Hasil eksperimen pertama (parameter *decrease* dengan *congestion window*) ditunjukkan pada Gambar 1. Pada Gambar 1 tersebut menunjukkan bahwa dengan peningkatan *congestion window*, didapatkan nilai parameter *decrease*. Terlihat bahwa nilai parameter *decrease* terus menurun. Nilai-nilai dari parameter *decrease* ini yang akan membentuk fungsi *decrease*. Sedangkan hasil eksperimen kedua (parameter *increase* dengan *congestion window*) ditunjukkan pada Gambar 2. Melalui Gambar 2 ini akan didapat nilai-nilai parameter *increase*. Nilai-nilai parameter *increase* ini yang akan digunakan pada pembentukan fungsi *increase*. Kedua hasil fungsi ini akan digunakan pada algoritma TCP yang biasa sehingga akan menghasilkan algoritma TCP yang telah dipercepat hasilnya. Untuk membandingkan antara algoritma TCP biasa dengan algoritma TCP yang telah dimodifikasi dilakukan eksperimen. Hasil eksperimen dapat dilihat pada Gambar 3. Terlihat pada Gambar 3, garis merah menunjukkan TCP yang telah dimodifikasi memiliki kemampuan pengiriman lebih banyak dibanding dari TCP yang tidak dimodifikasi. Semua eksperimen yang dilakukan menggunakan source code dari <http://acs.lbl.gov/~evandro/hstcp/simul/simul.html>:

1. behavior.tcl
2. utils.tcl
3. web.tcl

V. SIMPULAN

Berdasarkan hasil percobaan dapat disimpulkan dengan memodifikasi algoritma TCP dengan menambahkan fungsi parameter *increase* dan *decrease* mampu mempercepat proses pengiriman. Hal ini dikarenakan kedua parameter mampu mengurangi

paket *drop* yang terjadi pada sebuah pengiriman. Paket *drop* dapat dikurangi karena *congestion window* yang terbentuk dapat diatur sesuai dengan nilai parameter *increase* dan *decrease* yang digunakan. Untuk itu dapat disimpulkan dengan mengubah nilai parameter *increase* dan *decrease* yang tetap menjadi sebuah fungsi akan mampu mempercepat proses pemulihan *congestion window* dan mengurangi paket *drop*.

UCAPAN TERIMA KASIH

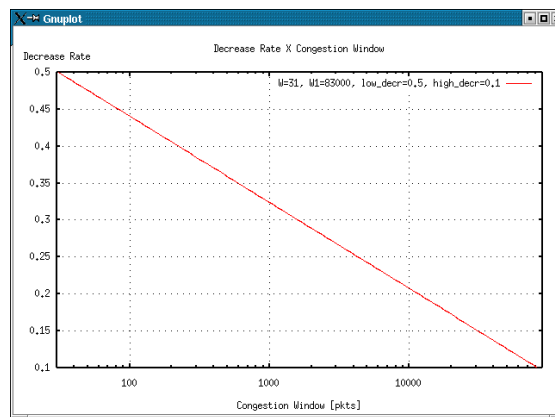
Penulis mengucapkan terima kasih kepada Bapak Yazid, PhD yang telah banyak memberikan arahan

dalam penyusunan makalah ini. Juga kepada seluruh anggota kelas Topik Dalam Jaringan Komputer Magister Ilmu Komputer Ganjil 2011/2012.

DAFTAR PUSTAKA

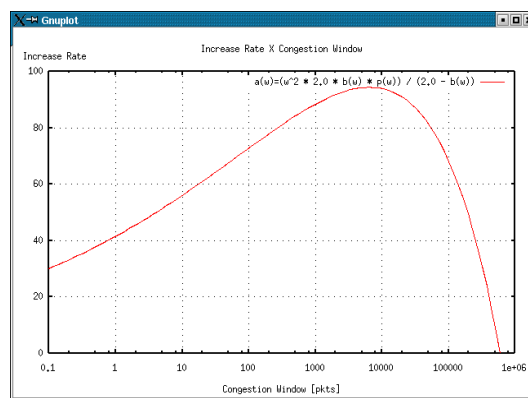
1. D. Nandita, "Rate Control Protocol (Rcp): Congestion Control To Make Flows Complete Quickly," The Department Of Electrical Engineering and The Committee On Graduate Studies, Stanford University (October 2007).
2. E. deSouza, <http://acs.lbl.gov/~evandro/hstcp/simul/simul.html> (2011).

LAMPIRAN



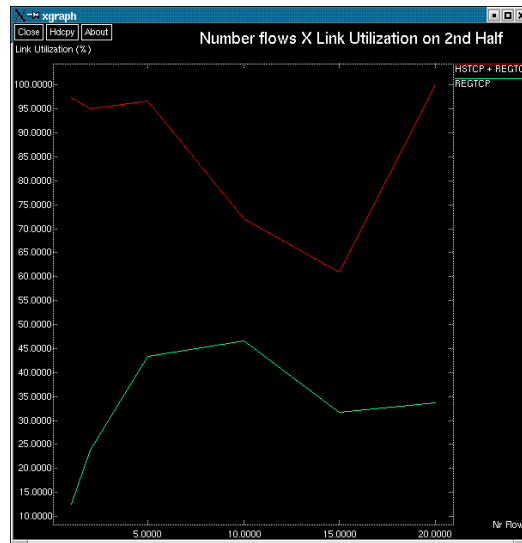
Gambar 1 Grafik perbandingan nilai *Congestion window* dengan *Decrease rate*

Sumber: <http://acs.lbl.gov/~evandro/hstcp/graph/decr-eq.png>



Gambar 2 Grafik perbandingan nilai *Congestion window* dengan *Increase Rate*

Sumber: <http://acs.lbl.gov/~evandro/hstcp/graph/incr-eq.png>



Gambar 3 Grafik perbandingan antara HSTCP dengan REGTCP pada saat perubahan nilai *increase*

Sumber: <http://acs.lbl.gov/~evandro/hstcp/simul/42/graphics.html>