

SISTEM KENDALI KONGESTI DI INTERNET

Pranoto Hidaya Rusmin, Carmadi Machbub, Agung Harsoyo, dan Hendrawan

Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung, Bandung 40132, Indonesia

E-mail: pranoto; carmadi; aharsoyo@lisk.ee.itb.ac.id

Abstrak

Kongesti di internet terjadi karena jumlah pengiriman data melebihi kapasitas *router* yang ada. Namun demikian, kenyataannya tidak setiap kongesti yang terjadi disebabkan karena hal tersebut. Sebagai contoh ketika kecepatan data dari satu pengirim jauh lebih tinggi dari pengirim yang lain karena mekanisme umpan balik yang tidak tepat, maka pengirim dengan kecepatan data jauh lebih rendah dapat dikatakan mengalami kongesti. Saat ini kongesti diatasi dengan melibatkan 2 mekanisme pengendalian, yaitu *flow/congestion control* di sumber pengirim data dan *Active Queue Management (AQM)* di *router*. AQM bertugas memberikan umpan balik kepada sumber pengirim sebagai indikasi dari tingkat kongesti di *router*. Sumber pengirim akan mengirimkan data ke jaringan sesuai dengan umpan balik tersebut. Mekanisme ini dirasakan masih belum cukup untuk menuntaskan persoalan kongesti di internet. Untuk itu, dalam makalah ini akan dipaparkan sebab-sebab terjadinya kongesti, kelemahan dan teknik kendali kongesti, yang telah dikembangkan sampai saat ini. Untuk memperoleh gambaran respon sistem, digunakan Matlab untuk simulasi sistem.

Abstract

Internet Congestion Control System. Internet congestion occurs when resource demands exceeds the network capacity. But, it is not the only reason. Congestion can happen on some users because some others user has higher sending rate. Then some users with lower sending rate will experience congestion. This partial congestion is caused by inexactly feedback. At this moment congestion are solved by the involvement of two controlling mechanisms. These mechanisms are flow/congestion control in the TCP source and Active Queue Management (AQM) in the router. AQM will provide feedback to the source a kind of indication for the occurrence of the congestion in the router, whereas the source will adapt the sending rate appropriate with the feedback. These mechanisms are not enough to solve internet congestion problem completely. Therefore, this paper will explain internet congestion causes, weakness, and congestion control technique that researchers have been developed. To describe congestion system mechanisms and responses, the system will be simulated by Matlab.

Keywords: congestion control, flow control, router, AQM

1. Pendahuluan

Dalam kamus Merriam-Webster, kongesti didefinisikan sebagai terjadinya konsentrasi sesuatu di area yang sempit. Sedangkan, di dalam kamus MSN Encarta kongesti dalam komputasi didefinisikan sebagai situasi di mana jumlah informasi yang ditransfer lebih besar dari kemampuan jalur komunikasi. Satu lagi dalam referensi [1] Michael Welzl memberikan definisi kongesti sebagai berikut: Suatu jaringan dikatakan kongesti dari perspektif pengguna jika kualitas layanan dirasakan oleh pengguna menurun karena kenaikan beban jaringan.

Dari ketiga definisi di atas, dapat diperhatikan bahwa kongesti melibatkan beberapa hal berikut:

- (1) Kemampuan jalur komunikasi data dalam mentransfer data, terbatas.
- (2) Beban jaringan, berupa jumlah data yang perlu ditransfer, dapat bervariasi tergantung jumlah pengguna dan kecepatan pengiriman data.

Dalam jaringan *best effort* seperti internet, variasi beban akan mengakibatkan variasi kualitas yang dirasakan oleh pengguna. Saat beban naik, dengan kemampuan jalur komunikasi data yang terbatas dan tetap, karena jalur ini harus dibagi secara *fair* kepada seluruh pengguna, maka wajar kalau kualitas layanan yang dirasakan pengguna akan turun. Namun demikian, turunnya kualitas layanan ini belum dapat langsung dikatakan sebagai kondisi kongesti. Sebagai contoh, misal terdapat jalur komunikasi data dengan

kemampuan 100 Kbps. Saat hanya ada satu pengguna, 100 Kbps digunakan sendiri. Tetapi, ketika ada satu lagi pengguna, karena kapasitas dibagi dua, setiap pengguna akan memperoleh 50 Kbps. Pengguna pertama akan mengalami penurunan kualitas layanan, namun tidak dikatakan mengalami kongesti.

Apabila diasumsikan beban cenderung bertambah dan mekanisme kendali kongesti mampu membagi kapasitas secara *fair*, maka kapasitas yang diperoleh masing-masing pengguna cenderung mengecil. Sampai pada batas tertentu, kapasitas yang diperoleh pengguna akan dirasakan tidak layak lagi. Sebagai contoh, tranfer file dirasakan layak oleh pengguna di atas 5 Kbps. Kurang dari 5 Kbps pengguna akan merasakan waktu tunda yang sudah tidak dapat diterima lagi. Oleh karena itu, kongesti dapat didefinisikan lebih tepat sebagai berikut: Suatu jaringan dikatakan kongesti dari perspektif pengguna jika karena kenaikan beban jaringan, kualitas layanan yang dirasakan oleh pengguna kurang dari batas kualitas minimum yang masih dapat diterima oleh pengguna.

Saat alokasi pada tiap pengguna telah mencapai batas minimum, namun beban tetap bertambah, alokasi akan semakin kecil. Kalau hal ini tetap dibiarkan, alokasi akan mencapai nilai cukup kecil sedemikian sehingga dirasakan pengguna tidak dapat melakukan komunikasi data. Kondisi inilah yang harus dihindari, selain menjamin pembagian kapasitas secara *fair*. Saat indeks bernilai 1, alokasi kapasitas kepada seluruh pengguna dapat dikatakan *fair*.

Beberapa hal berikut merupakan kesalahan pemahaman mengenai penyebab dan solusi kongesti di internet [2].

1. Kongesti disebabkan jumlah penyimpan data yang sedikit. Masalah kongesti dapat diselesaikan ketika harga memori cukup murah, memungkinkan penggunaan memori dalam jumlah besar. Penggunaan memori yang besar hanya bermanfaat mengatasi kongesti dalam waktu singkat dan menimbulkan waktu tunda yang lebih besar. Antrian panjang dan waktu tunda yang besar sangat dihindari oleh banyak aplikasi.
2. Kongesti disebabkan oleh kapasitas *router* yang rendah. Masalah ini akan dapat diselesaikan ketika terdapat *router* dengan kapasitas tinggi. Padahal, kapasitas *router* yang tinggi dapat mengakibatkan kongesti semakin parah di *router/switch*.
3. Kongesti disebabkan oleh kecepatan prosesor yang rendah. Ketika kecepatan prosesor dapat ditingkatkan, kongesti akan dapat diatasi. Namun demikian, ketika beberapa *router* dengan kecepatan prosesor lebih tinggi mentransmisikan data ke satu tujuan, akan mengakibatkan tujuan kelebihan beban.

Ketiga dugaan penyebab dan solusi kongesti di atas merupakan solusi statis. Padahal kongesti merupakan

persoalan dinamik. Oleh karena itu, solusi statis tidak akan dapat menyelesaikan persoalan kongesti ini.

2. Metode Penelitian

Formulasi Persoalan. Dalam referensi [3], disebutkan bahwa persoalan kongesti merupakan persoalan optimasi. Dapat diperhatikan dalam persamaan (1) – (3) berikut ini.

$$\max_{\{x_r\} \in S} \sum_r U_r(x_r) \quad (1)$$

$$\sum_{r:l \in r} x_r \leq c_l, l \in L \quad (2)$$

$$x_r \geq 0, r \in S \quad (3)$$

dengan L adalah himpunan semua *Router* dan S adalah himpunan semua pengirim. Misalkan masing-masing pengguna r memiliki fungsi kegunaan $U_r(x_r)$ ketika kecepatan pengiriman data sebesar x_r dialokasikan kepadanya. Persamaan (1) menyatakan bahwa agregat dari kegunaan semua pengguna harus dimaksimalkan. Syaratnya, agregat dari semua kecepatan pengiriman data pengguna kurang dari sama dengan kapasitas *router* dan kecepatan pengiriman data pengguna lebih dari sama dengan nol, dinyatakan dalam persamaan (2) dan persamaan (3).

Dalam kenyataannya, kongesti terjadi bukan di *router*. Akan tetapi, berada di *router/switch* yang bertugas mengirimkan data melalui *router* lain agar sampai ke tujuan. Agregat kecepatan pengiriman data yang masuk ke *router* tidak boleh melebihi kecepatan layanan *router*. Kalau melebihi, antrian di *router* akan berkembang, sampai penuh, mengakibatkan paket data terbuang. Oleh karena itu, persamaan (2) diubah menjadi persamaan (4) berikut ini.

$$\sum_{r:l \in r} x_r \leq Q_l, l \in R_t \quad (4)$$

dengan Q_l merupakan kecepatan layanan di *router* ke- l . R_t merupakan himpunan semua *router*.

Persamaan (3) menyatakan kecepatan pengiriman data di tiap pengirim dapat sama atau lebih dari nol. Untuk menghindari kongesti karena kecepatan pengiriman data terlalu kecil, kecepatan pengiriman data perlu dibuat sama atau lebih dari batas minimal kualitas layanan, x_{\min} . Untuk itu, persamaan (3) diubah menjadi persamaan (5) berikut ini.

$$x_r \geq x_{\min}, r \in S \quad (5)$$

Selain persoalan optimasi alokasi kapasitas, terdapat persoalan *fairness* dalam pengalokasiannya. Ketika sejumlah pengirim berebut kapasitas di satu *router*, dengan semua pengirim memiliki hak yang sama, tiap pengirim akan menerima alokasi yang sama. Indeks Jain dalam persamaan (6) berikut ini merupakan ukuran dari *fairness* alokasi kapasitas.

$$f(x) = \frac{(\sum_{i=1}^n x_i)^2}{n \sum_{i=1}^n x_i^2} \quad (6)$$

Indek akan bernilai 1, kalau alokasi kepada semua pengirim persis sama.

Di internet, tiap pengirim berebut kapasitas dari beberapa *router* dalam jalur komunikasinya. Alokasi yang diterima oleh satu pengirim dengan pengirim lain belum tentu sama. Hal ini karena tiap pengirim akan memiliki satu alokasi kapasitas minimal dari satu di antara beberapa *router* dalam jalur komunikasi. Inilah yang disebut sebagai *max-min fairness*.

Taksonomi Kendali Kongesti. Pertumbuhan jaringan yang semakin besar dan aplikasi yang digunakan semakin berkembang menjadikan kendali kongesti bagian penting dalam menjaga komunikasi data di internet dari terjadinya kongesti. Untuk itu telah dilakukan berbagai penelitian untuk memperoleh hasil terbaik. Dari referensi [4], berikut ini merupakan taksonomi kendali kongesti.

Router-Centric versus Host-Centric

- Pada *router-centric* tiap *router* memutuskan paket yang dibuang dan paket yang diteruskan. Selain itu, tiap *router* memberikan informasi kepada pengirim jumlah paket yang boleh dikirimkan.
- Pada *host-centric* pengirim selalu mengamati kondisi jaringan dan mengubah jumlah paket yang dikirimkan sesuai kondisi jaringan.

Reservation-Based versus Feedback-Based

- Pada *reservation-based* pengirim meminta kepada jaringan agar dialokasikan sejumlah kapasitas untuk pengiriman data.
- Pada *feedback-based* pengirim akan memulai mengirim data dan mengubah kecepatan pengiriman data sesuai dengan umpan balik yang diterima dari jaringan.

Implicit feedback versus Explicit feedback

Pada umpan balik implisit, perubahan jumlah data yang dikirimkan atau kecepatan pengiriman data ditentukan oleh keberadaan *acknowledgment* atau *time out*. Berbeda dengan umpan balik eksplisit, tiap *router* akan memberikan umpan balik kepada pengirim.

Window-Based versus Rate-Based

Pada *Window-based* digunakan *window advertisement* dalam jaringan untuk membatasi dan mengalokasikan jumlah *buffer*. Sedangkan, pada *rate-based* pengendalian pengiriman data dilakukan dengan mengendalikan kecepatan pengiriman data.

Yang menjadi sumber data adalah pengirim. Agar jumlah data dalam jaringan dapat berubah, pengirim harus dapat mengubah jumlah data yang dikirimkan.

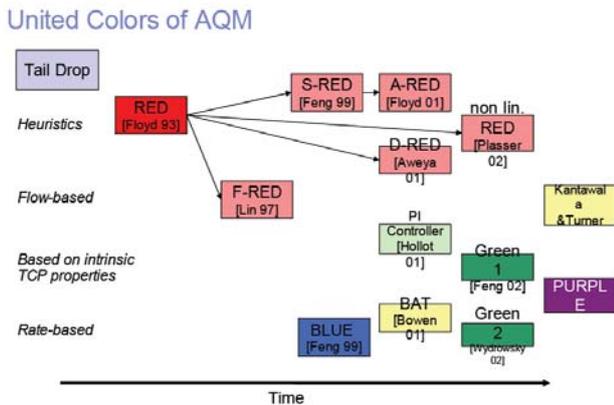
Untuk melakukan ini, pengirim membutuhkan informasi kondisi jaringan. *Router-centric* dan *host-centric* keduanya diperlukan. Dalam hal ini, *router-centric* bukan menentukan paket yang dibuang atau yang diteruskan. Tetapi, memberikan informasi kondisi jaringan dalam bentuk umpan balik. Dibandingkan dengan cara reservasi, umpan balik jauh lebih sederhana dalam implementasi dan protokol. Pada awalnya, internet dengan protokol TCP/IP menggunakan umpan balik implisit. Saat ini, mulai dikembangkan umpan balik eksplisit. Hal ini karena umpan balik eksplisit dapat memberitahukan pengirim agar mengurangi jumlah data yang dikirimkan, sebelum *buffer* penuh.

Perubahan pengiriman jumlah data dari pengirim lebih sederhana berbasis kecepatan pengiriman data, dibandingkan dengan berbasis window. Untuk aplikasi audio video *streaming* basis kecepatan pengiriman data lebih tepat dilihat dari karakteristik aliran data yang terus menerus. Saat ini internet menggunakan basis window untuk perubahan jumlah pengiriman data. Hal ini dilakukan karena berpegang pada prinsip konservasi, yaitu pengirim dapat mengirimkan data ketika sejumlah data telah berhasil ditransmisikan jaringan sampai ke tujuan (keluar dari jaringan) [1]. Sehingga, seolah-olah jumlah data dalam jaringan tetap dalam jumlah setimbangnya. Kelemahan utama basis window ini menimbulkan ledakan trafik (*burst traffic*). Dengan adanya variasi waktu tunda pada tiap jalur komunikasi data, basis window ini pada kondisi tertentu dapat mengakibatkan paket yang dibuang cukup besar.

Teknik Kendali Kongesti saat ini. Sampai saat ini teknik yang diimplementasikan dalam pengendalian kongesti masih berdasarkan *Additive Increase Multiplicative Decrease* (AIMD), yang melibatkan umpan balik implisit atau eksplisit [4]. Saat tidak ada kongesti jumlah data yang dikirimkan meningkat. Sebaliknya, ketika terdapat kongesti jumlah data yang dikirimkan dengan cepat diturunkan. RFC 2581 mendefinisikan 4 algoritma terpisah untuk kendali aliran data dan kendali kongesti pada TCP. Ditambah RFC3168 tentang ECN. Kelima algoritma tersebut adalah

1. *slow start*;
2. *congestion avoidance*;
3. *fast retransmit*;
4. *fast recovery*;
5. *explicit congestion notification (ECN)*

ECN dihasilkan dari teknik AQM yang diimplementasikan di *router*. Saat ini paling banyak diimplementasikan adalah *Random Early Detection* (RED). Teknik AQM yang telah dikembangkan sampai saat ini dapat diperhatikan dalam Gambar 1 dan dapat dibaca lebih rinci dalam referensi [1,3].



Gambar 1. Teknik AQM

Sistem kendali kongesti yang diimplementasikan saat ini, dapat diklasifikasikan dalam kendali ON-OFF. Sistem kendali seperti ini memiliki overshoot dan respon yang beresilasi. Kedua respon ini memiliki dampak buruk bagi komunikasi, yaitu buffer overflow dan kehilangan data. Selain itu, karena tidak terdapat batasan alokasi, memungkinkan alokasi kapasitas sekecil mungkin, sampai dirasakan tidak nyaman oleh pengguna.

3. Hasil dan Pembahasan

Teknik Kendali Kongesti Yang Dikembangkan. Dari kelemahan teknik kendali kongesti saat ini, telah dikembangkan teknik baru, yang dinamakan White, yang dapat diperhatikan dalam Gambar 2 [5].

Dengan fungsi alih sebagai berikut.

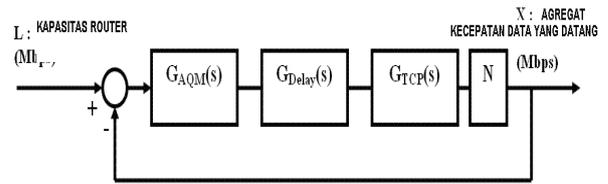
$$G_{AQM}(s) = \frac{K_2}{s} \tag{7}$$

$$G_{Delay}(s) = \frac{(1 - \frac{sR}{2})}{(1 + \frac{sR}{2})} \tag{8}$$

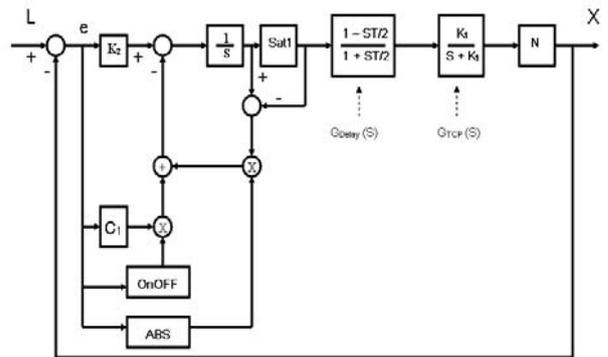
$$G_{TCP}(s) = \frac{K_1}{s + K_1} \tag{9}$$

Teknik di atas dapat membagi kapasitas router dengan alokasi yang fair. Namun demikian, terdapat overshoot yang harus segera diatasi. Untuk itu, ditambahkan teknik sliding dan anti-windup untuk mengatasi overshoot dan saturasi di pengendali [6]. Teknik ini dinamakan NewWhite. Sistem ini dapat diperhatikan dalam Gambar 3.

Sampai sejauh ini, alokasi kapasitas secara fair dan overshoot dapat diatasi. Namun demikian, alokasi kapasitas masih memungkinkan kurang dari batas



Gambar 2. Diagram Blok Sistem Kendali Kongesti



Gambar 3. Teknik Kendali Sliding dan Anti-Windup

minimal layanan, yang dirasakan cukup layak bagi pengguna. Oleh karena itu, perlu dikembangkan lebih lanjut teknik blocking, untuk membatasi pengguna pada batas alokasi minimal.

Dalam referensi [7] telah dikembangkan pula pengendali kongesti dengan pengendali proporsional-integral dan anti-windup. Dalam makalah tersebut, pengendali proporsional-integral digunakan untuk menghitung probabilitas kongesti. Hal ini berbeda dengan pengendali integral dalam NewWhite, yang digunakan untuk menghitung alokasi kapasitas. Sedangkan, anti-windup dalam referensi [7] digunakan untuk mencegah nilai probabilitas melebihi nilai 1. Hampir serupa, dalam NewWhite anti-windup berfungsi membatasi alokasi kapasitas sampai batas kapasitas router.

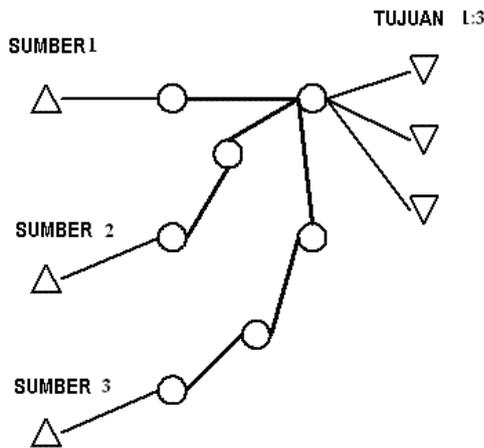
Skenario Simulasi. Dipilih AQM GREEN pada router dan AIMD pada pengguna dari teknik yang telah ada untuk teknik yang dikembangkan. Untuk simulasi ini, keluaran algoritma GREEN berupa probabilitas diberi batasan [0 1]. Sedangkan, keluaran integrator algoritma NewWhite diberi batasan [0 100]. Simulasi dilakukan menggunakan SIMULINK, MATLAB.

Skenario 1

Jalur komunikasi data dari pengirim ke penerima dapat melewati satu atau lebih router. Hal ini akan membuat pengiriman data memiliki Round Trip Time (RTT) yang berbeda-beda. Skenario 1 ini dibuat dengan tujuan mengetahui kinerja sistem terhadap variasi RTT. Dalam skenario 1 digunakan topologi dalam Gambar 4 dan tahapan simulasi pada Tabel 1 dan Tabel 2.

Skenario 2

Untuk skenario 2 menggunakan topologi Gambar 5 dan tahapan simulasi dalam Tabel 3.



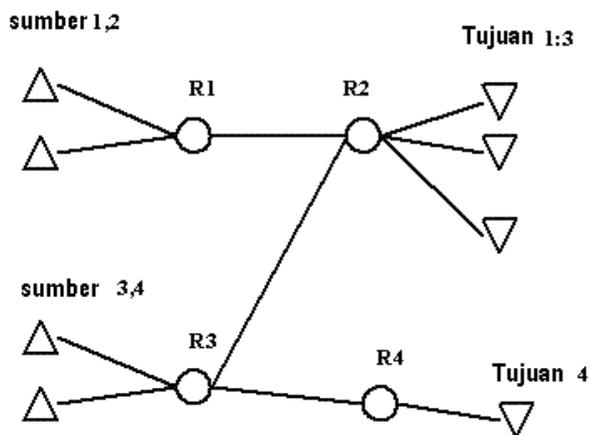
Gambar 4. Topologi 1

Asumsi:

- Kapasitas Router 100 Kbps
- RTT 1 (Sumber 1 ke Tujuan 1): 50 mS
- RTT 2 (Sumber 2 ke Tujuan 2): 250 mS
- RTT 3 (Sumber 3 ke Tujuan 3): 500 mS

Tabel 1. Tahapan simulasi Skenario 1a

Kelompok	Jumlah Pengirim	Permulaan (detik ke)	RTT (m detik)
S1	1	0	50
S2	1	250	250
S3	1	500	1000



Gambar 5. Topologi 2

Tabel 2. Tahapan simulasi Skenario 1b

Kelompok	Jumlah Pengirim	Permulaan (detik ke)	RTT (m detik)
S1	200	0	50
S2	300	250	250
S3	500	500	1000

Table 3. Tahapan Simulasi Skenario 2

Waktu (detik)	Tahap
0	S1 ke D1
100	S4 ke D4
400	S2 ke D2
2000	S3 ke D3
3500	S3 ke D3 berakhir
4250	S2 ke D2 berakhir

Keterangan: Si: Sumber (pengirim) ke i,
Di: Tujuan ke i, i=1..4

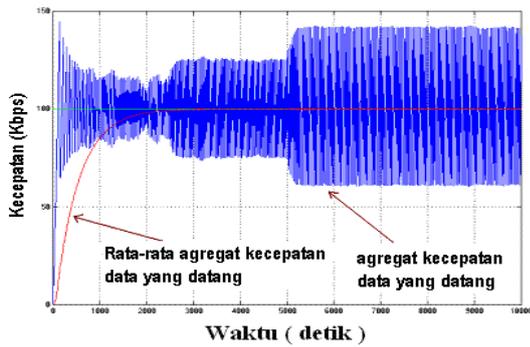
Hasil Simulasi

Skenario 1a : GREEN. Dapat diperhatikan dalam Gambar 6 bahwa teknik GREEN sensitif terhadap perbedaan waktu tunda. Semakin besar waktu tunda akan mengakibatkan osilasi dengan amplituda semakin besar. Untuk pembagian alokasi teknik GREEN berhasil membagi secara *fair* kepada ketiga pengirim, ditunjukkan dalam Gambar 7.

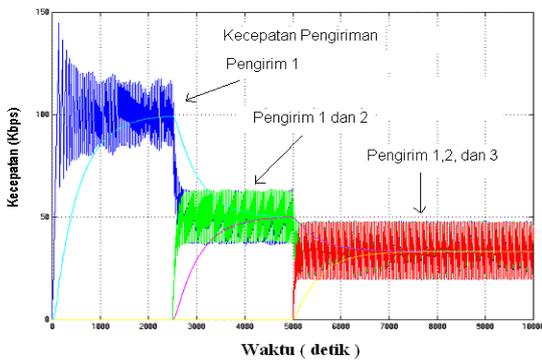
Skenario 1a : NewWhite. Dalam Gambar 8 dapat diperhatikan bahwa teknik NewWhite berhasil mengatasi *overshoot* dengan segera menurunkan alokasi kapasitas kepada setiap pengirim. Dihasilkan alokasi kapasitas yang *fair*, ditunjukkan dalam Gambar 9.

Skenario 1b : GREEN. Pada skenario ini, pada akhirnya total pengguna yang mengirmkan data adalah 1000. Dapat diperhatikan dalam Gambar 10 bahwa teknik GREEN berhasil membagi secara *fair* kepada semua pengirim, yaitu 100 bps, dengan rata-rata nilai ditunjukkan dalam Gambar 11.

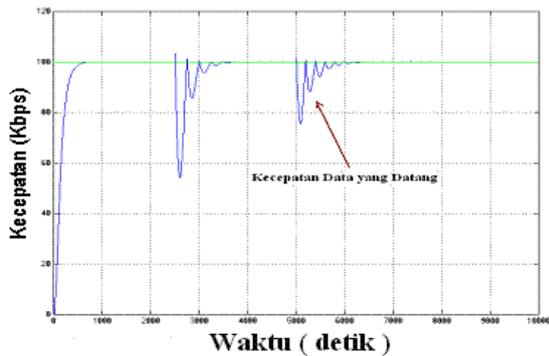
Skenario 1b : NewWhite. Dari Gambar 12 diperlihatkan bahwa untuk jumlah pengguna yang besar, teknik NewWhite menunjukkan perilaku osilasi, dengan rata-rata di bawah 100 bps, yang ditunjukkan Gambar 13. Hal ini terjadi karena saat berusaha mengatasi *overshoot*, sinyal kendali yang diberikan berupa alokasi



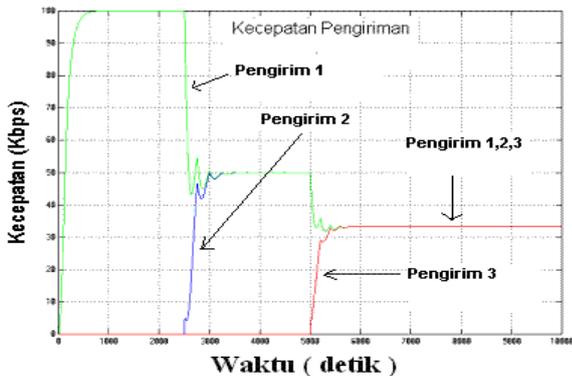
Gambar 6. Kapasitas Router dan Agregat Kecepatan Data Yang Datang



Gambar 7. Kecepatan Pengiriman Data



Gambar 8. Kapasitas Router dan Agregat Kecepatan Data Yang Datang



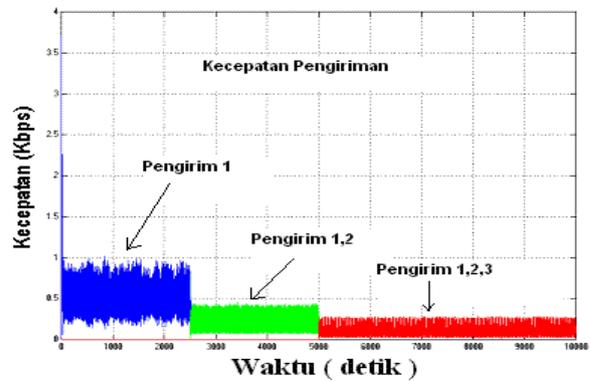
Gambar 9 Kecepatan Pengiriman Data

kapasitas tidak tepat. Teknik *sliding* dalam *NewWhite* ini perlu disempurnakan untuk mengatasi persoalan ini.

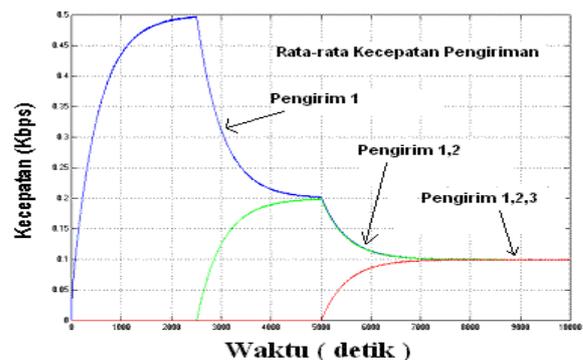
Dari simulasi skenario 1 ini dapat diketahui satu sebab kongesti yang harus diatasi adalah jumlah pengguna yang banyak, yang mengakibatkan alokasi kapasitas terlalu kecil. Jika kondisi ini tidak diantisipasi, maka memungkinkan tiap pengguna akan memperoleh alokasi mendekati nol.

Skenario 2 : AIMD-GREEN. Gambar 14 di bawah ini merupakan hasil simulasi *AIMD-GREEN* untuk skenario 2. Perlu diketahui bahwa respon yang ditunjukkan merupakan rata-rata dari respon sebenarnya, yang berhasil. Dalam skenario 2 ini, mulai detik ke 2000 seharusnya pengirim 1 dan pengirim 2 menurunkan kecepatan, agar pengirim 3 dapat menaikkan kecepatan sampai ketiganya mencapai kecepatan yang sama yaitu 1/3 kapasitas total. Namun hal ini tidak terjadi, karena ketidaktepatan umpan balik yang diperoleh pengguna. Persoalan ini akan dikaji lebih rinci dalam pembahasan mengenai umpan balik dalam sistem kendali kongesti.

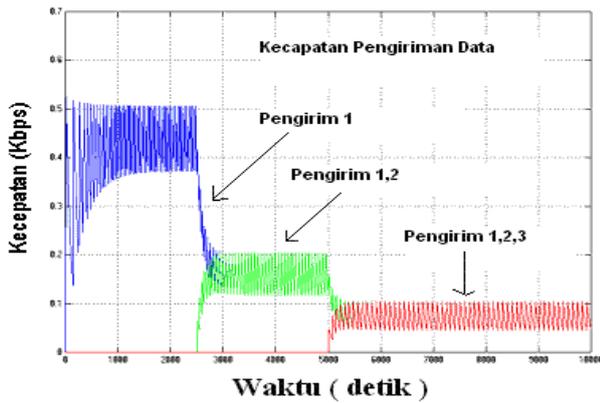
Skenario 2 : NewWhite. Ditunjukkan dalam Gambar 15, *NewWhite* berhasil mengalokasikan kapasitas secara fair tanpa terjadi overshoot yang berlebihan, kurang dari 5% dari kapasitas maksimum.



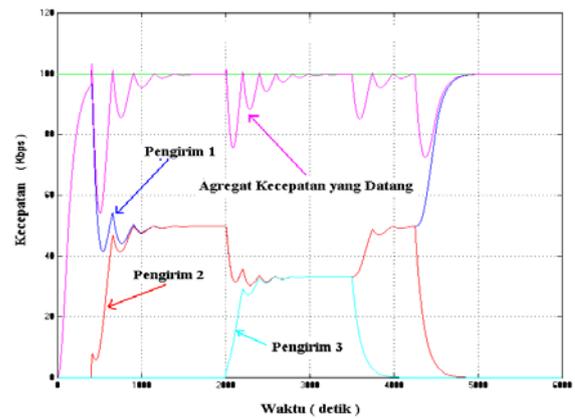
Gambar 10. Kapasitas Router dan Agregat Kecepatan Data Yang Datang



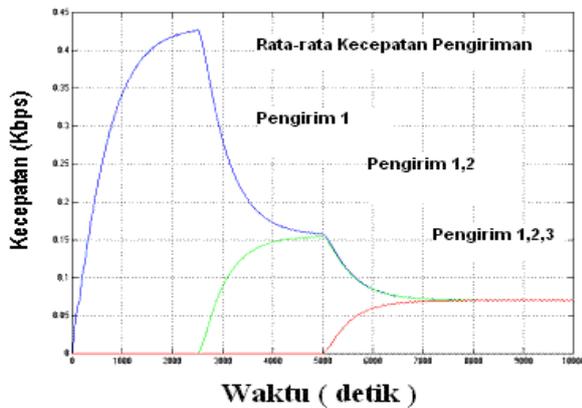
Gambar 11. Kecepatan Pengiriman Data



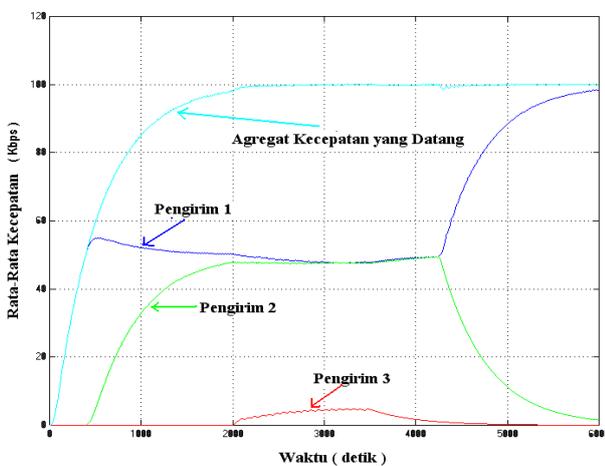
Gambar 12. Kapasitas Router dan Agregat Kecepatan Data Yang Datang



Gambar 15. Alokasi Kapasitas di R2 (NewWhite)



Gambar 13. Kapasitas Router dan Agregat Kecepatan Data Yang Datang



Gambar 14. Alokasi Kapasitas di R2 (AIMD-GREEN)

4. Kesimpulan

Teknik *AIMD-GREEN* memiliki kelemahan berupa respon yang berosilasi, yang semakin tinggi sebanding dengan besar RTT dan jumlah pengguna. Namun demikian, diperhatikan dari rata-rata alokasi, berhasil membagi alokasi secara *fair*. Teknik *NewWhite* berhasil mengalokasikan kapasitas secara *fair* tanpa osilasi dan *overshoot* dapat diatasi. Tetapi, untuk jumlah pengguna yang banyak, teknik *sliding* yang digunakan belum sempurna. Hal ini menimbulkan osilasi di bawah nilai set point. Selain itu, kedua teknik masih membuka peluang terjadinya kongesti karena alokasi kapasitas yang terlalu kecil.

Teknik *sliding* yang digunakan dapat menjamin agregat kecepatan data yang datang, tidak akan melebihi kapasitas *router*. Dengan kata lain, untuk alokasi kapasitas pengguna yang terbatas, akan dihasilkan agregat yang terbatas pula. Hal ini akan menjamin kestabilan sistem. Selain itu, perlu dikembangkan pula teknik untuk membatasi jumlah pengguna. Dengan teknik ini dapat diperoleh jaminan bahwa kongesti dapat dihindari.

Daftar Acuan

- [1] M. Welzl, Network Congestion Control, managing internet traffic, John Wiley & Sons, 2005.
- [2] J. Kalunga, What is Congestion Control in Broadband Networks? [Http://Cnx.org](http://Cnx.org), 2006.
- [3] R. Srikant, The Mathematics of Internet Congestion Control (Systems and Control: Foundations and Applications), Springer Verlag, 2004.
- [4] Cui-Qing, V.S.R. Alapati, A Taxonomy for Congestion Control Algorithms in Packet Switching Networks, IEEE Network, 1995.

- [5] P.H. Rusmin, C. Machbub, A. Harsoyo, 9th Int. Conf. Control, Automation, Robotics and Vision Singapore, 2006.
- [6] P.H. Rusmin, C. Machbub, A. Harsoyo, Proceedings EECCIS, Malang, Indonesia, 2006.
- [7] H. Lim, K.J. Park, E.C. Park, C.H. Choi, In Conference on Information Sciences and Systems, Princeton University (2002) 20-22.