

SIMULASI PENGUKURAN *INTRAFAIRNESS* DAN *INTERFAIRNESS* PROTOKOL-PROTOKOL *STREAM CONTROL TRANSMISSION PROTOCOL (SCTP)* DAN *TRANSMISSION CONTROL PROTOCOL (TCP)* PADA JARINGAN *UNICAST*

Bagio Budiardjo dan Michael Thiotrisno

Departemen Teknik Elektro, Fakultas Teknik, Universitas Indonesia, Depok 16424, Indonesia
Center for Information and Communication Engineering Research (CICER),
Fakultas Teknik, Universitas Indonesia, Depok 16424, Indonesia

E-mail: bbudi@ee.ui.ac.id dan thiotrisno@yahoo.com

Abstrak

Makalah ini menguraikan analisis kinerja jaringan *unicast* pada saat protokol-protokol *SCTP (Stream Control Transmission Protocol)* dan *TCP (Transmission Control Protocol)* berada bersama dalam satu jaringan. Kedua protokol tersebut di ujicoba dengan aplikasi *File Transfer Protocol (FTP)* dan aplikasi *traffic Constant Bit Rate (CBR)*. Dari simulasi dapat disimpulkan bahwa *SCTP* mampu mencapai *throughput* yang lebih tinggi dibandingkan *TCP* baik pada aplikasi *FTP* maupun pada aplikasi *traffic CBR* dengan tingkat *fairness*, baik *intrafairness* maupun *interfairness*; yang cukup tinggi. Peningkatan waktu simulasi menghasilkan nilai kinerja jaringan yang meningkat dan relatif stabil. Peningkatan jumlah sesi pada kapasitas *bandwidth* kecil menghasilkan peningkatan total *throughput* dan utilisasi jalur. Hasil simulasi menyatakan bahwa penerapan *SCTP* dan *TCP* dalam satu jaringan *unicast* umumnya dapat dilakukan tanpa menurunkan kinerja jaringan secara keseluruhan dan masih menghasilkan indeks *fairness* yang tinggi, diatas 0.85 untuk berbagai variasi *traffic*.

Abstract

This paper elaborates the performance analysis of a unicast network when two protocols, *SCTP (Stream Control Transmission Protocol)* and *TCP (Transmission Control Protocol)* coexist in the network. These two protocols are tested using *File Transfer Protocol (FTP)* and *Constant Bit Rate (CBR)* based traffic. From the simulations, it could be concluded that *SCTP* could achieve better throughput compared to *TCP* in both *FTP* and *CBR* based traffic, with sufficiently high fairness index of both *intrafairness* and *interfairness*. The increase of simulation time yields to the increase and relatively stable network performance. The increase of the number of sessions of small bandwidth applications resulted in the increase of troughput and link utilization. The simulations show that implementation of *SCTP* and *TCP* in a unicast network generally could be done without having to degrade the network performance, with relatively high fairness index, over 0.85 for various traffic variations.

Keywords: SCTP, Interfairness, Intrafairness, TCP, Unicast

1. Pendahuluan

Upaya untuk menyempurnakan Internet dengan teknologi protokol baru yang lebih mendukung penghantaran paket dengan *Quality of Service (QoS)*, seperti misalnya *Integrated Services Architecture (ISA)* [1] atau *Differentiated Services Architecture (DSA)* [2]; sulit untuk dilakukan karena keduanya memerlukan perubahan arsitektur jaringan secara mendasar. Salah satu solusi untuk mendukung kebutuhan kualitas

layanan adalah dengan menciptakan protokol baru, yang dapat diterapkan pada Internet tanpa harus mengubah arsitektur jaringan ataupun teknologi protokolnya secara mendasar. *TCP*, yang merupakan protocol transport yang banyak digunakan saat ini; memiliki beberapa kelemahan yang mendasar, antara lain: *packet latency* yang cenderung panjang, *head of line blocking (HOL)*, *denial of service* yang disebabkan oleh mekanisme *three-way-handshake* dan pengurutan informasi yang ketat Sedangkan *UDP (User Datagram*

Protocol), merupakan protokol yang *connectionless* yang tidak memiliki mekanisme *congestion control* dan memperbaiki kesalahan. Untuk memperbaiki kelemahan itu, *Internet Engineering Task Force (IETF)* mengembangkan protokol baru sebagai protokol alternatif, yaitu *SCTP* [3]. *SCTP* dirancang sebagai protokol yang handal, *connection-oriented* seperti *TCP* akan tetapi memiliki kemampuan-kemampuan: *fast recovery*, *fast retransmit*, *Selective Acknowledgement (SACK)*, *multihoming* dan *multistreaming*. *SCTP* diharapkan lebih bertoleransi terhadap kebutuhan *QoS* dibandingkan protokol-protokol sebelumnya. *SCTP* juga dapat mendukung modus transmisi *wireless* [4].

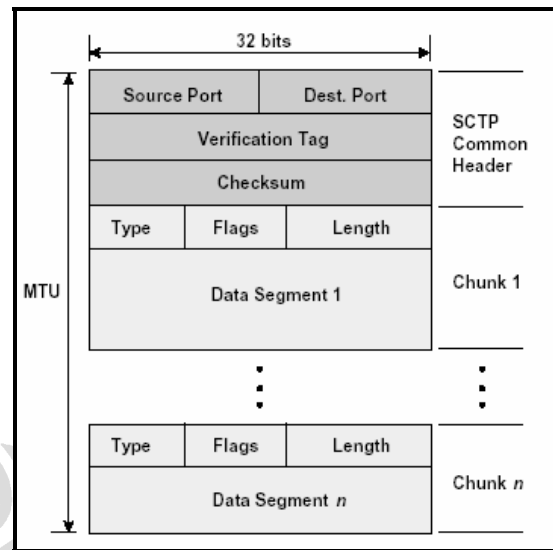
Tulisan ini menguraikan analisis kinerja jaringan *unicast*, pada saat protokol-protokol *TCP* dan *SCTP* berada bersama dalam satu jaringan. Analisis kinerja dilakukan dengan mengukur *intrafairness* antar sesama *traffic SCTP* dan *interfairness* antara *traffic SCTP* dan *traffic TCP* pada jaringan *unicast*.

Interfairness menyatakan tingkat keadilan pembagian *bandwidth* antara sesi-sesi *traffic* yang terdiri dari dua atau lebih protokol yang berbeda. Sedangkan *intrafairness* menyatakan tingkat keadilan pembagian *bandwidth* antara sesi-sesi yang terdiri dari protokol yang sama. Hal-hal lain yang dianalisis meliputi kinerja jaringan seperti *prosentase kenaikan throughput*, indeks *fairness*, dan total *throughput SCTP* dan *TCP*, dengan aplikasi *FTP* dan aplikasi berbasis *traffic CBR*. Dalam simulasi diujicobakan berbagai besaran yang meliputi: waktu simulasi, *delay* propagasi, *bandwidth* jalur *bottleneck*, dan jumlah sesi yang berpartisipasi. Analisis ini diharapkan dapat menjadi salah satu acuan dalam menerapkan protokol *SCTP* sebagai salah satu alternatif protokol Internet di masa mendatang.

Organisasi dari tulisan ini disusun sebagai berikut. Bagian 2 membahas konsep dasar protokol *SCTP*, yang meliputi format paket, mekanisme asosiasi, fitur-fitur *SCTP*, dan perbandingan dengan *TCP* dan *UDP* secara makro. Bagian 3 membahas perancangan, tahapan, topologi, dan parameter simulasi. Bagian 4 berisi hasil simulasi dan analisa, sedangkan bagian 5 memaparkan kesimpulan dari makalah ini.

2. Metode

SCTP yang dirancang oleh *IETF* diformulasikan dalam dokumen *Draft Proposed Standard SCTP* berupa *Request for Comments (RFC) 2960* pada Oktober 2000 [3]. *SCTP* dirancang sebagai protokol lapisan *transport* yang berbasis pada aplikasi *message-oriented*. *SCTP* menyediakan *acknowledgement*, *error-free non-duplicated transfer message* karena adanya *checksum*, *sequence number*, dan mekanisme *SACK*.

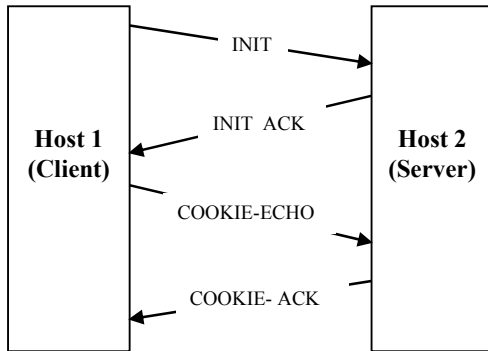


Gambar 1. Format Paket SCTP [5]

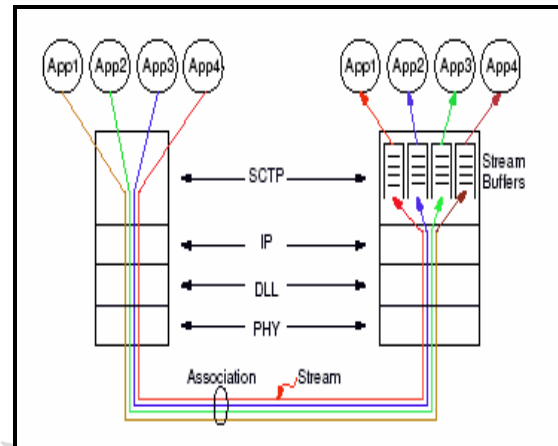
Setiap paket *SCTP* terdiri dari *SCTP common header* dan beberapa blok *chunk*. Ukuran paket *SCTP* adalah sebesar 1 *MTU* (*Maximum Transmission Unit*) sepanjang 1500 bytes (Gambar 1) [5]. *Common header* mendukung proses validasi dan koneksi pada *SCTP* sedangkan *chunk* merupakan unit informasi pada paket *SCTP* yang dapat berupa *chunk* data atau *chunk* kontrol. *Common header SCTP* berukuran 12 bytes, yang terdiri dari: alamat *port* sumber dan *port* tujuan masing-masing sebesar 16 bit, *tag* verifikasi (32 bit) dan *checksum* (32 bit).

Alamat *port* sumber (*source*) menyatakan identifikasikan asal paket, sedangkan alamat *port* tujuan adalah alamat penerima/tujuan paket *SCTP*. Setiap *endpoint* menggunakan *tag* verifikasi untuk mengidentifikasi asosiasi dimana penerima melakukan validasi terhadap pengirim paket *SCTP*. Istilah asosiasi menggambarkan koneksi antara dua *endpoints*. *Checksum* digunakan untuk menjaga *data integrity* dan menghindari duplikasi. Setiap *chunk SCTP* berukuran 8 bytes yang terdiri dari *Field Type*, *Flags*, *Length* dan *User Data*. *Field Type* digunakan untuk membedakan antara tipe *chunk* data atau *chunk* kontrol. Terdapat 13 macam *chunk* yang digunakan sebagai standar *chunk SCTP* [3]. *Field Flag* digunakan untuk menentukan bit mana yang akan digunakan dalam koneksi. *Field Length* digunakan untuk menentukan panjang *chunk*. *Field data segment* mengandung *payload* data setiap *chunk*.

Sebagai ganti dari *three-way-handshake* pada koneksi *TCP*, *SCTP* menggunakan *four-way-handshake* untuk menginisialisasi suatu asosiasi (Gambar 2). Sisi pasif



Gambar 2. Four-Way-Handshake Sctp



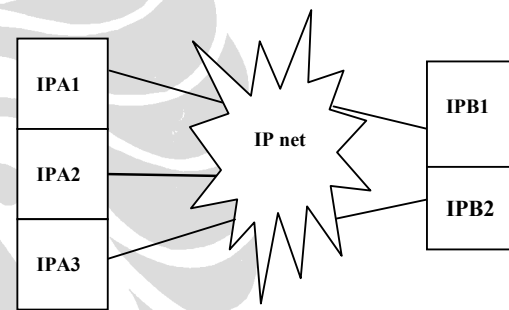
Gambar 3. Asosiasi Sctp Multiple Stream [7]

dari asosiasi (biasanya server) tidak akan mengalokasikan sumber daya sampai pesan ketiga tiba dan divalidasi.

Keempat pesan yang dipertukarkan merupakan *chunk-chunk* yang terdapat pada paket Sctp, yaitu *INIT chunk*, *INIT-ACK chunk*, *COOKIE-ECHO chunk* dan *COOKIE-ACK chunk*. Mekanisme *four-way-handshake* ini disebut mekanisme *COOKIE*, yang dapat mencegah adanya *denial of service* seperti *SYN attack* [6].

Pada awalnya, server berada dalam keadaan *CLOSED* dan menerima permintaan asosiasi melalui sebuah *INIT chunk* yang dikirim oleh client. Server yang menerima *INIT chunk* dan membalas dengan *INIT-ACK chunk* yang berisi *COOKIE*, yang menyatakan server siap untuk melakukan asosiasi. Client yang menerima *INIT-ACK chunk* lalu mengirim *COOKIE-ECHO chunk* yang berisi *COOKIE* dari *INIT ACK* yang diterima sebelumnya. Setelah menerima kiriman client, server memastikan apakah *COOKIE* tersebut tepat. Data yang ada dalam *COOKIE* digunakan untuk mengisialisasi asosiasi Sctp. Server kemudian mengirim *COOKIE-ACK chunk* ke client dan kondisi server menjadi *ESTABLISHED*. Client yang menerima *COOKIE-ACK chunk* dari server kemudian berada pula dalam kondisi *ESTABLISHED* dan siap untuk melakukan asosiasi dengan server.

Informasi atau data dapat dikirim melalui beberapa *stream (multiple stream)* dalam satu asosiasi. Data dari beberapa *upper layer application* dapat di-multiplex dalam satu *channel* yang disebut asosiasi (Gambar 3) [7]. Paket yang hilang pada sebuah *stream* tidak mengganggu pengiriman paket *stream* yang lain. Paket yang mengikuti paket yang hilang pada *stream* tersebut akan ditampung di *buffer stream* penerima, sampai paket yang hilang ditransmisi ulang dan tiba di sisi penerima. Kemudian, antrian paket ini dikirim ke *upper layer application*. Dengan demikian, Sctp mendukung



Gambar 4. Analogi Multihoming Sctp [5]

pengiriman paket yang tidak teratur dalam beberapa *stream*. Kedua fitur Sctp ini menghilangkan masalah *Head of Line (HOL) Blocking* dan *delay* transmisi yang sering timbul pada TCP.

Setiap *endpoint Sctp* dapat memiliki lebih dari satu *network interface* (pada tingkatan *Internet Protocol*). Pada saat asosiasi dilakukan, maka *network interface* dari tiap *host* yang terhubung akan membentuk jalur-jalur yang dapat dilalui paket Sctp (Gambar 4) [5]. Ada 6 kemungkinan jalur : IPA1-IPB1, IPA1-IPB2, IPA2-IPB1, IPA2-IPB2, IPA3-IPB3. Manfaat *multihoming* adalah untuk penyeleksian dan pengawasan jalur transportasi data, dan untuk menghindari *redundancy* data.

Fitur *multihoming Sctp* tidak berarti Sctp mendukung pembagian beban (*load sharing*). Paket Sctp tetap dirutekan ke tujuannya melalui jalur primer. Jalur-jalur lainnya berfungsi sebagai cadangan, yang akan dimanfaatkan bila jalur lain mengalami kegagalan.

Tabel 1. Perbandingan SCTP, TCP dan UDP [9]

Feature	SCTP	TCP	UDP
State required at endpoints	√	√	
Reliable Data Transfer	√	√	
Multi-homed Host Support	√		
Multi-Stream Support	√		
Security against SYN attacks	√		
Unordered Data Delivery	√		√
Failure detect. (Heartbeat)	√		
Message bundling	√		
Congest. ctrl. & avoidance	√	√	
Message boundary conserve.	√		√
Supports Unlimited numbers of SACK	√		

Pengecekan terhadap aktif atau tidaknya jalur pengiriman paket saat terjadi kegagalan, pada SCTP dilakukan dengan mekanisme HEARTBEAT.

Pada TCP, apabila paket dengan urutan tertentu tidak sampai di tujuan maka tidak akan ada ACK. Pengirim akan mendeteksi terjadinya *time-out*. Pada SCTP, fungsi paket ACK ini digantikan oleh SACK (*Selective Acknowledgement*). SACK merupakan salah satu dari *control chunk SCTP*. SACK tidak hanya berfungsi sebagai *acknowledgement*, tetapi juga berfungsi pada mekanisme *fast retransmit* dan *fast recovery*. SACK mampu melaporkan kepada pengirim jumlah paket yang hilang melalui *GapAck Block*, yang dapat mempercepat deteksi kehilangan paket untuk transmisi ulang, meningkatkan *throughput*, utilisasi *bandwidth*. Jumlah SACK pada SCTP dapat mencapai 19380 blok [4], sedangkan pada TCP hanya 3-4 blok.

Algoritma *congestion control SCTP* berbasis pada RFC 2581 yang merupakan *congestion control TCP* [3]. SCTP menggunakan mekanisme *end-to-end window based flow control* dan *congestion control* yang mirip dengan *congestion control TCP* [8]. Ini memastikan adanya *fairness* antara kedua protokol jika berada pada jaringan yang sama [7].

Pada SCTP juga terdapat dua fase yaitu *slow start* dan *congestion avoidance*, yang diterapkan pada *endpoint* pengirim untuk mengontrol jumlah data yang dapat ditransmisikan. Seperti pada TCP, pada *endpoint SCTP* juga digunakan tiga parameter pengontrol kecepatan transmisi [3] : *Receiver advertised window* (*rwnd*, ukuran *buffer*, dalam *bytes*), *congestion control window* (*cwnd*, dalam *bytes*), jumlah paket maksimum pengirim dan *slow start threshold* (*ssthresh*). Variabel *partial_byte_ack* (*pba*); digunakan selama fase *congestion avoidance*, memfasilitasi perubahan *cwnd*.

Congestion window pada SCTP meningkat berdasarkan jumlah bytes yang diterima (*acknowledged*), bukan berdasarkan kepada jumlah *acknowledgement* yang diterima seperti pada TCP [8]. Fase *fast recovery* dapat dicapai dengan penggunaan fasilitas SACK [4].

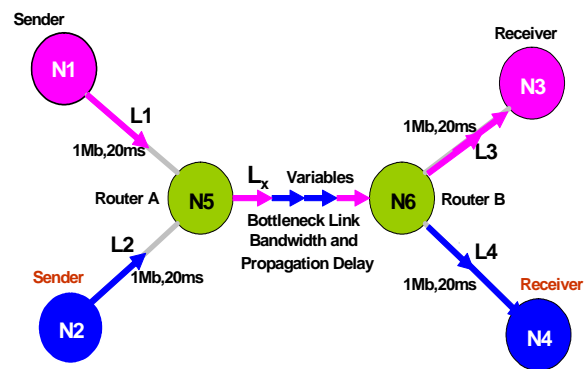
Secara umum, perbandingan antara SCTP, TCP dan UDP sebagai protokol lapisan *transport* dalam hal koneksi, layanan, dan transmisi data pada jaringan komputer, tertera di Tabel 1 [9].

Perangkat lunak simulasi adalah *Network Simulator* (NS-2) versi 2.26 [10] dengan *patch module SCTP* dari University of Delaware [11], yang dijalankan pada platform *Linux RedHat 7.2*. Tujuan utama simulasi ini adalah untuk menganalisa pengaruh penerapan protokol SCTP dan TCP terhadap kinerja jaringan *unicast* yang mengalami kelangkaan *bandwidth*.

Topologi jaringan yang digunakan dalam simulasi umumnya sama yaitu jaringan *unicast* akan tetapi jumlah sesi atau source bervariasi yaitu 2 sesi, 4 sesi dan 8 sesi. Begitu pula dengan jalur *bottleneck Lx* yang bervariasi berdasarkan kapasitas *bandwidth* dan *delay* propagasi. Topologi simulasi tertera pada Gambar 5. dibawah ini.

Sesi-sesi saling berbagi *bandwidth* jalur *bottleneck* yang dibentuk antara *router A* dan *router B*. Peningkatan jumlah sesi dimaksudkan untuk mengamati peningkatan partisipasi *endpoints* pada jaringan *unicast* dengan jalur yang langka *bandwidth* (*bottleneck*).

Variabel dalam simulasi ini, antara lain: ukuran *bandwidth* jalur *bottleneck*, jumlah sesi yang mengalir pada jalur *bottleneck* dan waktu simulasi. Waktu pengamatan umumnya 200-1000 detik.



Gambar 5. Topologi Simulasi Jaringan Unicast. $N_i = \text{node}_i$.

Traffic generator yang digunakan pada simulasi adalah CBR (Constant Bit Rate) yang mewakili real-time traffic dengan bit-rate yang tetap dan aplikasi FTP, yang mewakili aplikasi yang berbasis nrt-VBR (non-real time Variable Bit Rate) yang bersifat bursty dan tidak sensitif terhadap delay [12].

Adapun beberapa besaran kinerja jaringan yang diamati adalah: total throughput, indeks fairness, dan peningkatan persentase throughput SCTP terhadap TCP. Karena terbatasnya ruang dalam makalah ini, besaran kinerja lainnya seperti packet loss rate dan utilisasi jalur bottleneck tak dapat dipresentasikan.

Total throughput didefinisikan sebagai total paket (dalam bits) yang tiba di seluruh penerima per satuan waktu. Total throughput dinyatakan dalam Mbps (Mega bits per second), yang mengindikasikan pemanfaatan bandwidth jaringan per detik. Besaran ini digunakan untuk mengukur presentase peningkatan throughput dari traffic kedua jenis protokol.

Secara kuantitatif, indeks fairness berada diantara 0 dan 1. Nilai 1 menyatakan pembagian bandwidth yang adil bagi setiap sesi. Bila ada n sesi yang mengalir melalui sebuah jalur bottleneck dengan kapasitas bandwidth B, maka setiap sesi memperoleh bandwidth sebesar B/n. Persamaan (1) dan (2) di bawah ini digunakan untuk menghitung indeks fairness :

Bila R_i^{τ} menyatakan laju transmisi paket dari sesi i, menggunakan paket sepanjang b byte selama waktu τ maka [13] :

$$R_i^{\tau} = \frac{b \langle \text{packets sent in the interval } (t, t + \tau) \rangle}{\tau} \quad (1)$$

dan index fairness F_{τ} dapat dihitung dengan persamaan [13] :

$$F_{\tau}(t) = \frac{\left(\sum_{i=0}^n X_i^{\tau}(t) \right)^2}{n \sum_{i=0}^n \left(X_i^{\tau}(t) \right)^2} \quad (2)$$

dimana $X = R/B$ adalah rasio antara laju transmisi dengan bandwidth yang ada sedangkan n adalah banyaknya sesi yang memperebutkan bandwidth yang tersedia. Semakin tinggi indeks fairness, semakin adil pembagian bandwidth diantara sesi-sesi yang berkompetisi.

Pada simulasi interfairness yang diamati adalah parameter persentase peningkatan throughput SCTP terhadap TCP, seperti yang dilakukan oleh

Atiquzzaman dan Ivanic [7]. Adapun persamaannya adalah [7] :

$$\delta = \frac{\lambda_1 - \lambda_2}{\lambda_2} \times 100 \quad (3)$$

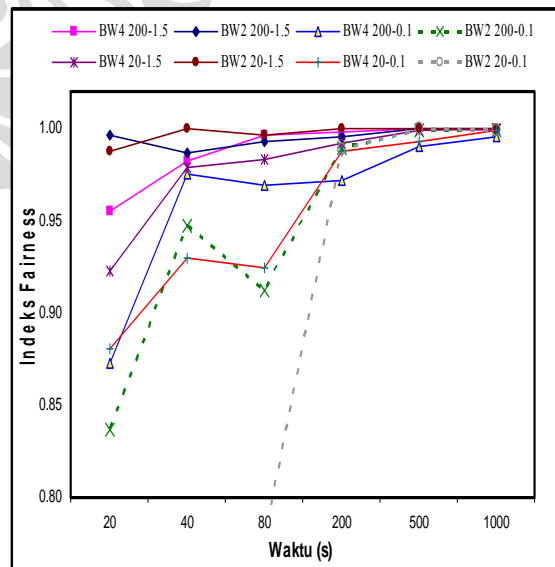
dimana δ adalah persentase peningkatan throughput, λ_1 adalah total throughput sesi SCTP dan λ_2 adalah total throughput sesi TCP.

Parameter yang ditetapkan dalam simulasi sesuai dengan parameter NS-2, antara lain :

- a. Traffic CBR: sending rate 0.75Mbps-1.0Mbps; set random variable, ukuran paket CBR: 1000 bytes
- b. Aplikasi FTP: ukuran paket FTP, default NS-2.26; no random variable
- c. duplex link: link antar router (jalur bottleneck), bandwidth 0.1Mbps-1.5Mbps, delay propagasi 20ms-200ms, loss prob. 0.1%
- d. Router: ukuran buffer adalah 20 paket; queuemangement adalah drop tail
- e. Agent TCP: ukuran paket 1000 bytes, window 20 paket, dan parameter lain sesuai default NS-2.26.
- f. Agent SCTP: Paket 1 MTU = 1500 bytes; Parameter lain disesuaikan dengan default NS-2.26.

4. Hasil dan Pembahasan

Pada gambar diperlihatkan bahwa peningkatan indeks fairness antar protokol SCTP, sejalan dengan lamanya waktu simulasi (1000 second/detik). BW4 200-1.5 berarti ada 4 sesi yang aktif, dengan b/w bottleneck 1.5 Mbit dan delay jaringan 200 ms.



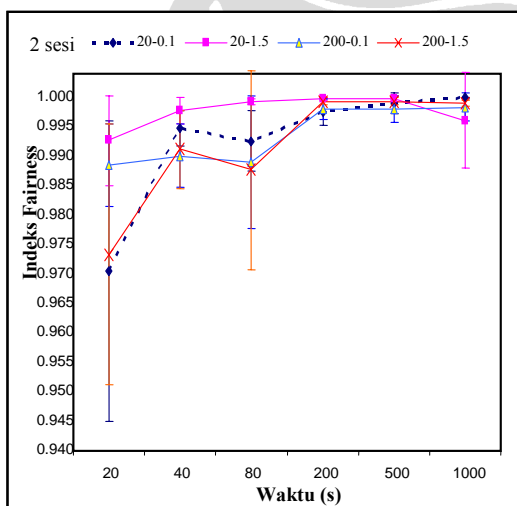
Gambar 6. Indeks Fairness FTP terhadap waktu untuk dua jenis bandwidth dan dua jenis sesi

Delay propagasi yang relatif panjang (dari 20 ms ke 200 ms), mampu mengangkat indeks *fairness* terutama pada awal simulasi walaupun tidak signifikan. Hal ini diakibatkan oleh panjangnya *delay* sehingga pengiriman paket dari setiap sesi tidak maksimum (masalah *timeout* dan jarak). Karenanya, kompetisi antar sesi tidak tinggi. Sesi mampu berbagi *bandwidth* secara lebih adil.

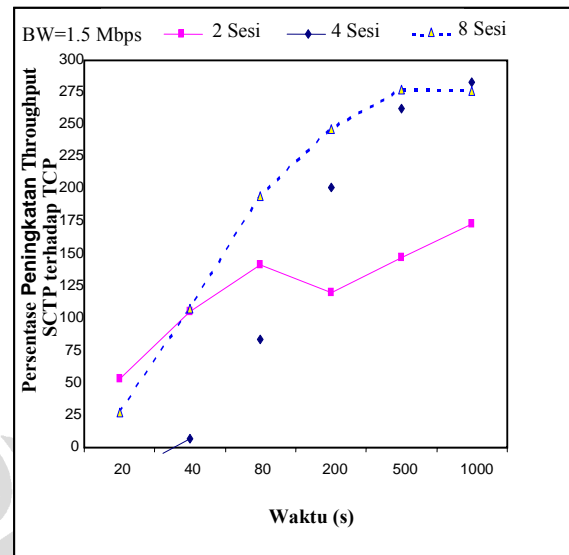
Peningkatan jumlah sesi pada ukuran *bandwidth* yang kecil menyebabkan indeks *fairness* yang menurun, namun tidak signifikan. *Intrafairness* pada simulasi ini umumnya cukup tinggi karena seluruh sesi menerapkan algoritma *congestion control* yang sama.

Simulasi *Intrafairness Traffic CBR* hanya dilakukan pada jaringan *unicast* dengan 2 sesi *SCTP* yang diberikan *traffic generator CBR*. Umumnya, hasil yang dicapai pada simulasi *intrafairness traffic CBR* sesuai dengan perkiraan semula. Peningkatan waktu simulasi menghasilkan nilai parameter jaringan yang cenderung tidak fluktuatif.

Total *throughput* jaringan dengan 2 sesi protokol *SCTP* mengalami peningkatan, seiring dengan peningkatan periode simulasi. Gambar 7. menunjukkan bahwa umumnya indeks *fairness* antar protokol *SCTP* dalam membawa paket *traffic CBR* cukup tinggi dan mengalami peningkatan untuk peningkatan waktu simulasi. Jika dibandingkan dengan data indeks *intrafairness FTP* maka hasil yang diperoleh pada simulasi ini menunjukkan bahwa kemampuan *SCTP* lebih baik dalam membawa paket layanan *realtime bandwidth*.



Gambar 7. Indeks *Fairness traffic CBR* terhadap Waktu untuk dua jenis bandwidth



Gambar 8. Persentase Peningkatan Total *Throughput SCTP* terhadap *TCP* dengan beberapa sesi aktif

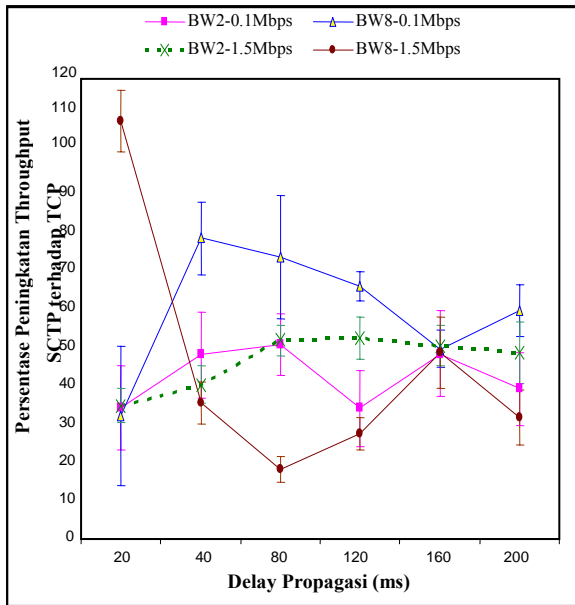
Simulasi pengukuran *interfairness* aplikasi *FTP*, diimplementasikan dengan membebani jaringan *unicast* dengan *traffic* dari protokol-protokol yang berbeda, yaitu *SCTP* dan *TCP*. Periode simulasi ini adalah 1000 detik dengan mempertimbangkan bahwa komponen jaringan telah mencapai kinerja yang relative stabil, tidak fluktuatif.

Dari hasil simulasi, dapat dianalisis lebih jauh pengaruh perubahan *delay* propagasi jalur *bottleneck* yang dilalui seluruh sesi dalam jaringan terhadap nilai-nilai parameter jaringan seperti *index fairness* dan persentase peningkatan *throughput SCTP* terhadap *TCP*.

Gambar 8 menunjukkan bahwa ketiga seri grafik *interfairness* baik 2, 4 maupun 8 sesi menunjukkan persentase peningkatan *throughput SCTP* terhadap *TCP* yang cukup besar antara 150-275% pada kondisi jaringan yang telah menyesuaikan diri terhadap perubahan beban.

Peningkatan yang dicapai oleh *SCTP* dibandingkan *TCP* pada simulasi ini lebih dikarenakan pengembangan mekanisme *congestion control SCTP*.

Spesifikasi simulasi *Interfairness Traffic CBR* mirip dengan simulasi *interfairness* aplikasi *FTP*. Perbedaannya hanya pada jenis aplikasi dan *traffic* yang ditempatkan pada *agent-agent SCTP* dan *TCP* pada *NS-2*. Periode simulasi *interfairness* ini adalah



Gambar 9. Persentase Peningkatan Total Throughput Sctp terhadap Tcp untuk berbagai sesi dan bandwidth

200 detik. Simulasi ini sepengetahuan penulis merupakan simulasi yang baru dan belum pernah dilakukan sebelumnya. Peneliti sebelumnya tidak menggunakan *traffic* CBR dalam simulasi.

Dengan pertimbangan bahwa untuk menjadi alternatif protokol, maka Sctp pun harus mampu mentransmisikan paket yang termasuk dalam aplikasi multimedia atau layanan CBR. Pada Gambar 9 dapat diamati bahwa dalam mentransmisikan paket dengan layanan atau *traffic* CBR maka Sctp mampu mencapai *throughput* yang lebih besar dibandingkan Tcp. Peningkatan *throughput* Sctp terhadap Tcp ini tidak dipengaruhi oleh *delay* propagasi. Pada aplikasi dengan *delay* propagasi rendah maupun tinggi, Sctp mampu mencapai *throughput* yang lebih besar dibandingkan Tcp.

Sctp lebih mampu bertoleransi terhadap peningkatan jumlah sesi dan lebih mampu membagi *bandwidth* terhadap sesi secara lebih proporsional. Hal ini secara implisit mengindikasikan bahwa Sctp lebih tanggap terhadap perubahan kondisi jaringan, dan tetap menjaga *fairness* penggunaan *bandwidth* antar sesi. Sctp lebih toleran terhadap kebutuhan QoS dibandingkan Tcp.

4. Kesimpulan

Sctp terbukti mampu mewujudkan *Intrafairness* yang tinggi (0.85-0.95) setelah jaringan mencapai titik

kestabilan, walaupun jaringan mengalami kelangkaan *bandwidth* (*bottleneck*).

Tingkat *interfairness* antara Tcp dan Sctp relatif tinggi. Hal ini diakibatkan oleh diadopsinya algoritma *congestion control* Tcp. Sctp mampu mencapai *throughput* yang lebih tinggi dibandingkan Tcp pada aplikasi *FTP* dan *traffic* CBR, disekitar 175-275 % pada berbagai variasi dari kapasitas *bandwidth*.

Peningkatan waktu simulasi umumnya menghasilkan nilai parameter kinerja jaringan yang tidak fluktuatif. Hal ini menunjukkan bahwa jaringan memerlukan waktu adaptasi terhadap algoritma *congestion control* baik Sctp maupun Tcp. Peningkatan *throughput* Sctp terhadap Tcp tidak dipengaruhi oleh *delay* transmisi paket. Pada aplikasi dengan *delay* transmisi rendah maupun tinggi Sctp mampu mencapai *throughput* yang lebih besar dari Tcp untuk mentransmisikan *traffic* CBR. Sctp secara implisit lebih toleran terhadap kebutuhan QoS dibandingkan dengan Tcp.

Daftar Acuan

1. R. Braden et al., IntServ. in the Internet Architect.: an Overview, *IETF RFC 1633*, June 1994.
2. S. Blake et al, An Architect. for DiffServ., *IETF RFC 2475*, December 1998
3. R. Stewart et. al., Stream Control Transmission Protocol, *IETF RFC 2960*, October 2000.
4. S. Fu and M. Atiquzzaman, Improving End-to-End Throughput of Mobile IP Using Sctp, *Submitted for 2003 Worksh. on High Pero. Switch. & Nets.*, <http://www.ce.ou.edu/atiq/papers> - accessed November 2003.
5. K. Zhang, Multi-homing Feature Studying of Sctp Using NS-2 Network Simulator, *CSCI 8211 Class Project*, Univ. of Minnesota, 2003.
6. I. A. Rodriguez, Sctp: The Design of a New Reliable Transp. Prot. for IP Nets., *Submitted for Master's Thesis*, Dept. of Elec. & Comm. Eng., Helsinki Univ. of Tech., Feb. 2002.
7. M. Atiquzzaman dan W. Ivancic, Eval. of Sctp Multistreaming Over Satellite Links, <http://www.cs.ou.edu/~atiq/papers-> accessed Oct. 2003.
8. R. Alamgir, M. Atiquzzaman, dan W. Ivancic, Effect of Congestion control on the Perf. of Tcp & Sctp over Satellite-Nets., <http://esto.nasa.gov/conferences/estc-2002/papers> - accessed Oct. 2003.
9. R.R. Stewart, Q. Xie, *Introduction to Sctp*, Addison Wesley, Boston, 2002, p.12.
10. The NS-2: <http://www.isi.edu/nsnam/ns>, accessed November 2003.

11. NS-2 SCTP Module: <http://pel.cis.udel.edu>, accessed November 2003.
12. Williams Stallings, *High Speed Networks & Internets: Performance & Quality of Service*, 2nd Eds., Prentice Hall Inc., 2002, p.105.
13. D. Chiu, R. Jain, Analysis of the Increase and Decrease Alg. for Congest. Avoid. in Computer Networks, *Computer Networks and ISDN Systems*, 17, (1989) 1-14.

