

Analisis Pengaruh Ukuran Window Pada Pengendali Kemacetan di SCTP Menggunakan Fitur Multihoming

Agus Halid*¹, Reza Pulungan²

¹Program Studi S2/S3 Ilmu Komputer, FMIPA UGM, Yogyakarta

²Jurusan Ilmu Komputer dan Elektronika, FMIPA UGM, Yogyakarta

e-mail: *agushalid@kharisma.ac.id, pulungan@ugm.ac.id

Abstrak

Stream Control Transmission Protocol (SCTP) merupakan protokol yang mirip dengan Transmission Control Protocol (TCP) dan User Datagram Protocol (UDP). SCTP merupakan protokol yang bersifat reliable dan connectionless. Protokol ini memiliki kemampuan multistreaming dan multihoming dalam melakukan transmisi data.

Penelitian ini merupakan pemodelan terhadap SCTP menggunakan simulator OPNET yang dapat menjadi akselerasi bagi peneliti dalam bidang jaringan. SCTP pada simulator dibangun dengan melakukan modifikasi terhadap TCP. Pemodelan dimulai dengan membangun skenario jaringan dan menentukan bandwidth pada jalur yang akan dilewati oleh paket data.

Modifikasi ukuran window dalam penelitian ini menggunakan nilai 1 MMS, 2 MMS hingga 10 MMS pada pengendali kemacetan. Tujuannya adalah untuk melihat pengaruh modifikasi ukuran window terhadap nilai packet loss, delay dan throughput. Hasil pengukuran memperlihatkan bahwa nilai throughput tertinggi terdapat pada Skenario Kedua sebagaimana diperlihatkan pada Tabel 6.4 dengan nilai throughput sebesar 433.566,0244 bit/s. Penggunaan ukuran window dalam pengendali kemacetan dimaksudkan untuk menghindari banjir data pada sisi endpoint yang dapat menyebabkan packet loss.

Kata kunci—Pengendali kemacetan, throughput, delay, packet loss, ukuran window, multihoming, SCTP

Abstract

Stream Control Transmission Protocol (SCTP) is a protocol that is similar to the Transmission Control Protocol (TCP) and User Datagram Protocol (UDP). SCTP is a protocol that is both reliable and connectionless. This protocol has the ability multistreaming and multihoming in the transmit data.

This research is the modeling of the SCTP using OPNET simulator that can be accelerated for researchers in the field of networking. SCTP on the simulator was built to perform modifications to TCP. Modeling starts with building a network scenarios and determine the bandwidth on the path that will be passed by data packets.

Modification of window size in this research using 1 MMS, 2 MMS up to 10 MMS on congestion control. The aim is to disclose the effect of modification of the window size to the value packet loss, delay and throughput. The measurement results show that the throughput rate is highest in the Second Scenario as shown in Table 6.4 with throughput value of 433.566,0244 bits/s. Using window size in congestion control is intended to prevent a flood of data on the endpoint that can lead to packet loss.

Keywords—Congestion control, throughput, delay, packet loss, window size, multihoming, SCTP

1. PENDAHULUAN

Jaringan dapat diartikan sebagai suatu sistem yang didesain untuk dapat berbagi sumber daya (*resource sharing*) seperti *printer*, *processor* ataupun komunikasi data (data, informasi, suara, video). Sebagaimana dipaparkan dalam [1], penambahan jumlah layanan dan aplikasi yang meluas, telah ditawarkan kepada pengguna dimanapun berada melalui akses jaringan *wireless*. Penambahan layanan dan aplikasi ini, tidak terlepas dari kebutuhan *bandwidth* dan jaminan bahwa paket yang dikirim, sampai di tujuan dalam kondisi baik. Semakin besar *bandwidth* maka jumlah potongan paket yang dapat ditransmisikan juga akan semakin besar. Transmisi paket yang tidak terkontrol akan menyebabkan terjadinya penumpukan paket, dan pada saat tertentu akan dimusnahkan [2]. Pemusnahan paket disebut juga dengan *packet loss* atau *packet dropped*, sehingga untuk mengurangi kejadian tersebut, perlu dilakukan pengaturan terhadap jumlah paket yang akan ditransmisikan. Pengaturan transmisi paket, dilakukan dalam pengendali kemacetan dengan menentukan ukuran *window*. Transmisi paket yang baru, akan dilakukan setelah pengirim memperoleh *acknowledge* dari penerima, jika tidak maka akan dilakukan transmisi ulang. Transmisi ulang terjadi karena paket yang dikirim dianggap tidak sampai pada tujuan. Paket dianggap sampai ditujuan bilamana pengirim menerima *acknowledge* dari penerima.

Penyebab lain kemacetan dalam sistem jaringan dapat dipengaruhi oleh protokol (*protocol*) yang digunakan. Protokol yang sering digunakan dalam jaringan yaitu Transmission Control Protocol (TCP) dan User Datagram Protocol (UDP). Fungsi utama dari protokol TCP adalah untuk menjaga *reliable* data, selain itu juga menjadi pengendali kemacetan dalam sistem jaringan, sehingga kemacetan yang parah dapat dihindari [3]. Protokol UDP merupakan protokol yang bersifat *connectionless*, di mana pengiriman data tidak memerlukan persetujuan (*handshaking*) sebelum dua proses saling berkomunikasi. Protokol UDP tidak menjamin suatu paket pernah dikirimkan, begitu juga dengan paket yang dibuang (*packet loss*) pada sisi penerima [4].

Penelitian [5] memaparkan bahwa protokol TCP merupakan protokol yang memiliki pengendali kemacetan, namun protokol UDP tidak [4], sehingga protokol ini tidak *user-friendly*. Penggunaan protokol UDP biasanya ditemukan pada aplikasi yang bersifat *streaming*. Penelitian sebelumnya, [6] juga memaparkan bahwa protokol TCP dengan menggunakan *multiple TCP connection* akan membutuhkan *resource* dalam membuka dan menutup koneksinya. Berbeda dengan SCTP yang merupakan protokol dengan kemampuan untuk melakukan transmisi data secara *streaming* dan memiliki pengendali kemacetan [5].

Sebagaimana dipaparkan dalam [7,8,9] bahwa pembagian beban pada SCTP menggunakan lebih dari satu jalur sumber dan tujuan yang *multihomed* melalui *Current Multipath Transfer (CMT)* bertujuan untuk meningkatkan *throughput* pada aplikasi jaringan. CMT merupakan pengiriman data baru secara simultan dari sumber ke tujuan melalui dua jalur atau lebih. Penelitian [9] melakukan modifikasi pengendali kemacetan dengan membagi beban menggunakan algoritma baru yang disebut dengan *Split Fast Retransmit*. Menurut [8] *sharing buffer* dapat menurunkan *throughput* dengan menggunakan CMT, sehingga *receive buffer (rbuf)* akan melakukan pemblokiran bilamana sebuah aliran didistribusikan ke sejumlah jalur *end-to-end* oleh *application layer*, di sisi lain *application layer* penerima, memiliki ruang *buffer* yang terbatas. Terjadinya *packet loss* pada SCTP yang menyebabkan penurunan kinerja di jaringan *wireless heterogen*, bukan hanya disebabkan oleh *Bit Error Rate (BER)* yang tinggi ataupun kemacetan, sehingga dibutuhkan mekanisme untuk membedakan berbagai jenis *packet loss* [1].

Masalah lain yang dapat terjadi pada SCTP *multihoming* diantaranya adalah waktu perpindahan (transisi) antar jalur. Penelitian [10] memaparkan bahwa perpindahan jalur secara tradisional didasarkan pada jumlah pengiriman ulang *timeout* yang berurutan. Dasar ini dianggap tidak selalu tepat dalam SCTP *multihoming*. Aspek kunci optimasi dalam penelitian ini adalah waktu perpindahan antar jalur yang tersedia. Berbeda dengan penelitian yang dilakukan pada jalur yang *heterogen*, sebagaimana diungkapkan [11,12] bahwa dibutuhkan mekanisme yang efisien untuk meningkatkan konektivitas jaringan dan mempertahankannya.

Modifikasi ukuran *window* diawal secara signifikan mengurangi lonjakan *traffic* video [13]. Meskipun tidak semua hasil yang diperoleh menunjukkan pengukuran yang sama, dalam artian modifikasi ukuran *window* tidak menyelesaikan sama sekali masalah lonjakan *traffic* video. Ukuran *window* yang digunakan dalam penelitian ini meliputi 3*MMS (default), 5*MMS, 7*MMS, 10*MMS, 15*MMS, dan 20*MMS.

Penggunaan pengendali kemacetan pada SCTP menurut [14], didasarkan pada ukuran *Maximum Transmission Unit* (MTU) dari *ethernet* yang digunakan. Nilai MTU yang ada saat ini adalah rata-rata 1.500 byte. Potongan data yang diijinkan tidak melebihi nilai MTU dari *ethernet* yang digunakan, termasuk IP header dan SCTP header. Besarnya potongan data yang kemudian dikenal dengan istilah *Maksimum Segmen Size* (MMS) pada penelitian sebelumnya adalah sebesar 1.460 byte. Sedangkan ukuran *window* diawal yang digunakan sebesar 4.380 byte. Penelitian ini akan melakukan analisa terhadap pengaruh ukuran *window* pada pengendali kemacetan di protokol SCTP, dimana ukuran *window* yang digunakan adalah 1*MMS, 2*MMS, hingga 10*MMS.

2. METODE PENELITIAN

Metodologi yang diterapkan dalam penelitian ini adalah :

- Mengumpulkan referensi
 - Referensi tentang SCTP dikumpulkan dari tahun 2012 hingga tahun 2014, baik dari jurnal, internet maupun buku; dan
 - Memilah referensi yang ada hubungannya dengan SCTP dan kemacetan dalam jaringan.
- Membangun skenario SCTP
 - Menganalisa kebutuhan node yang digunakan untuk membangun SCTP pada *simulator* OPNET;
 - Merancang arsitektur jaringan berdasarkan kebutuhan rancangan topologi. Tahap ini menempatkan *node* yang digunakan untuk merancang SCTP pada OPNET menggunakan Connection Node (CN) sebagai *server*, Mobile Node (MN) sebagai *endpoint*, Home Agen (HA) sebagai *router* yang mengatur informasi lokasi terakhir dari MN, Foreign Agent (FA) merupakan *router* yang dikunjungi oleh MN dan menyediakan layanan *routing* kepada MN saat registrasi, dan Access Point (AP) yang menghubungkan MN dengan jaringan;
 - Menentukan besar *bandwith* yang digunakan. Tahap ini merupakan penentuan besar bandwidth pada skenario yang dibuat untuk melihat nilai *packet loss*, *delay* dan *throughput*; dan
 - Menentukan parameter yang akan diukur. Parameter yang dimaksudkan dalam penelitian ini sebagaimana terdapat dalam *simulator* OPNET.
- Melakukan pengukuran skenario
 - Pengukuran dengan modifikasi ukuran *window* menggunakan *bandwidth* sebesar 1 Mbps; dan
 - Pengukuran dengan modifikasi ukuran *window* menggunakan *bandwidth* sebesar 11 Mbps.

3. HASIL DAN PEMBAHASAN

Simulasi dilakukan dengan menggunakan *simulator* OPNET. Pengujian simulasi ini bertujuan untuk mengetahui pengaruh ukuran *window* pada pengendali kemacetan dengan melihat perbedaan *packet loss* (*packet dropped*), *delay* dan *throughput* pada SCTP, yang menggunakan fitur *multihoming*.

3.1 Pengujian Skenario

Ukuran *window* yang digunakan sebagai nilai *default* adalah 1.460 byte. Nilai 1.460 byte menjadi nilai ukuran *window (default)* yang akan dibandingkan dengan nilai $2 \times$ Maksimum Segmen Size (MSS) atau sebesar 2.920 byte. Nilai ukuran *window* di awal adalah 1.460 byte, sedangkan nilai ukuran *window* tertinggi yang digunakan adalah 14.600 byte.

3.1.1 Skenario Pertama

Skenario Pertama menggunakan tiga buah jalur. Ketiga jalur pada Skenario Pertama menggunakan kabel (link model) dengan nilai *bandwidth* masing-masing percobaan 100 Mbps yang dihubungkan dari AP hingga ke HA. Berbeda dengan *bandwidth* pada AP yang menuju ke MN, nilai *bandwidth*nya diset dengan 11 Mbps. Secara lengkap hasil pengukuran dengan menggunakan Skenario Pertama, baik dengan menggunakan *bandwidth* 1 Mbps maupun 11 Mbps sebagaimana diperlihatkan pada Tabel 1 dan Tabel 2 dengan setiap percobaan dilakukan selama 60 menit atau 3.600 detik menggunakan ukuran *window* yang dimulai dari 1.460 byte hingga 14.600 byte.

Tabel 1 diperoleh dari hasil pengujian Skenario Pertama dengan memberikan beban *bandwidth* pada sisi MN sebesar 1 Mbps untuk setiap WLAN yang digunakan, dalam penelitian digunakan tiga buah WLAN yang terhubung dengan tiga buah AP. Berdasarkan Tabel 1 diperlihatkan bahwa nilai *delay* semakin meningkat seiring dengan peningkatan nilai ukuran *window*. Jumlah rata-rata *packet loss* sebagaimana diperlihatkan Tabel 1 sebesar 398,065778 bit/s, dan paling kecil *packet loss* dengan ukuran *window* sebesar 7.300 byte. Nilai *throughput* pada bagian awal dan akhir mengalami peningkatan, hal ini relevan dengan pemaparan [13], yang dalam penelitiannya melakukan modifikasi ukuran *window* untuk mengurangi lonjakan trafik di awal.

Tabel 1 Hasil Pengukuran Skenario Pertama (*Bandwidth* 1 Mbps)

Ukuran <i>Window</i>	Hasil Pengukuran		
	<i>Packet Loss</i> (bit/s)	<i>Delay</i> (s)	<i>Throughput</i> (bit/s)
1.460	404,60444	0,01622	18.607,12000
2.920	409,64444	0,01584	17.636,98667
4.380	408,88889	0,01625	17.454,39111
5.840	405,77778	0,01633	16.010,02667
7.300	356,28444	0,02099	18.325,68000
8.760	413,19111	0,02068	16.956,92444
10.220	392,84444	0,02163	17.659,00444
11.680	399,47556	0,02434	16.490,48000
13.140	400,78222	0,02512	17.679,48444
14.600	389,16444	0,02874	19.535,68889

Hasil pengukuran sebagaimana diperlihatkan Tabel 2 diperoleh dengan menaikkan *bandwidth* MN sebesar 11 Mbps untuk setiap WLAN yang digunakan. Nilai *packet loss* semakin menurun seiring dengan kenaikan nilai ukuran *window*. Nilai *bandwidth* pada sisi *server* sangat berpengaruh pada pengiriman data menuju ke *client*, sehingga menyebabkan antrian paket pada sisi *client*, dimana pada kondisi tertentu paket tersebut akan dimusnahkan (*packet loss*), selain itu dapat disebabkan karena diskoneksi saat perpindahan kanal (*handoff*) yang berada diluar jangkauan [1]. Nilai *delay* mengalami kenaikan seiring dengan tingginya nilai ukuran *window*. Peningkatan nilai *delay* pada Skenario Pertama disebabkan karena jarak, antrian, jumlah *node* yang dilibatkan maupun transmisi ulang [15], yang pada penelitian ini peningkatan nilai *delay* disebabkan oleh jarak, antrian dan terjadinya transmisi ulang yang dapat diamati berdasarkan besarnya nilai *packet loss* pada pengukuran.

Tabel 2 Hasil Pengukuran Skenario Pertama (*Bandwidth* 11 Mbps)

Ukuran <i>Window</i>	Hasil Pengukuran		
	<i>Packet Loss</i> (bit/s)	<i>Delay</i> (s)	<i>Throughput</i> (bit/s)
1.460	412,80889	0,00943	18.610,06667
2.920	417,36889	0,00926	18.611,09333
4.380	404,24889	0,00918	15.306,34667
5.840	386,51556	0,00978	16.217,04000
7.300	374,22222	0,01298	19.759,47556
8.760	412,36444	0,01040	15.288,46222
10.220	393,96444	0,01308	17.847,20889
11.680	397,83111	0,01436	18.111,41333
13.140	393,09333	0,01457	17.414,56000
14.600	390,23111	0,01466	16.488,72000

Berdasarkan pada Skenario Pertama (Tabel 1 dan Tabel 2), nilai *packet loss* terkecil terjadi pada nilai ukuran *window* 7.300 byte dengan nilai *packet loss* masing-masing 356,28444 bit/s dan 374,22222 bit/s. Nilai *delay* terkecil pada Tabel 1 terdapat pada nilai ukuran *window* 2.920 byte yaitu sebesar 0,015840 s (15,84 ms), sedangkan pada Tabel 2 terdapat pada nilai ukuran *window* 4.380 byte yaitu sebesar 0,009184 s (9,184 ms). Nilai *throughput* tertinggi pada Tabel 1 dengan ukuran *window* 14.600 byte yaitu sebesar 19.535,68889 bit/s, sedangkan pada Tabel 2 dengan ukuran *window* 7.300 byte yaitu sebesar 19.759,47556 bit/s.

3. 1.2 Skenario Kedua

Skenario Kedua juga menggunakan tiga buah jalur, namun pada skenario ini, *router* dihilangkan, sehingga AP (FA) langsung dihubungkan dengan HA. Ketiga jalur pada Skenario Kedua menggunakan kabel (*link model*) dengan nilai *bandwidth* masing-masing percobaan 100 Mbps yang dihubungkan dari AP hingga ke HA. Berbeda dengan *bandwidth* pada AP(FA) yang menuju ke MN, nilai *bandwidth*nya diset dengan 11 Mbps. Secara lengkap hasil pengukuran dengan menggunakan Skenario Kedua, baik dengan menggunakan *bandwidth* 1 Mbps maupun 11 Mbps, sebagaimana diperlihatkan pada Tabel 3 dan Tabel 4 dengan setiap percobaan dilakukan selama 60 menit atau 3.600 detik menggunakan ukuran *window* yang dimulai dari 1.460 byte hingga 14.600 byte.

Hasil pengukuran sebagaimana diperlihatkan pada Tabel 3 dengan beban *bandwidth* pada MN sebesar 1 Mbps untuk setiap WLAN yang digunakan. Berdasarkan pada Tabel 3, nilai *packet loss* terendah terlihat pada ukuran *window* 14.600 byte sebesar 250,89778 bit/s. Peningkatan ukuran *window* dalam pengendali kemacetan mengakibatkan *packet loss* pada *router* yang memiliki ukuran *buffer* kecil, atau *router* mengalami kemacetan [16], namun kenaikan ukuran *window* hanya terjadi sekali (di awal koneksi), sisanya diatur dalam mekanisme pengendali kemacetan.

Tabel 3 Hasil Pengukuran Skenario Kedua (*Bandwidth* 1 Mbps)

Ukuran <i>Window</i>	Hasil Pengukuran		
	<i>Packet Loss</i> (bit/s)	<i>Delay</i> (s)	<i>Throughput</i> (bit/s)
1.460	267,78667	0,01216	421.251,07111
2.920	262,50667	0,01182	376.873,15333
4.380	257,68000	0,01425	374.633,40444
5.840	259,09333	0,01532	363.822,80667
7.300	264,13333	0,01556	367.648,32667
8.760	270,14222	0,01837	390.362,00000
10.220	270,39111	0,01956	384.458,92222
11.680	262,60444	0,02061	365.538,41556
13.140	276,81778	0,02321	381.120,39778
14.600	250,89778	0,02615	419.595,21778

Nilai *delay* sebagaimana diperlihatkan pada Tabel 3 semakin meningkat seiring dengan peningkatan nilai ukuran *window*. Peningkatan *delay* terjadi karena protokol melakukan transmisi ulang akibat *packet loss* pada jaringan [15]. Berdasarkan pada hasil pengukuran (Tabel 3), nilai *throughput* mengalami peningkatan, selain karena *bandwidth* pada jaringan yang lebar (100 Mbps), juga karena jumlah *node* yang dilibatkan. Meskipun nilai *packet loss* dan nilai *delay* cukup tinggi, namun menurut pengamatan yang dilakukan, tingginya *packet loss* disebabkan jumlah data yang ditransmisikan cukup besar sehingga menyebabkan antrian yang penuh. Paket yang tidak dapat masuk kedalam antrian akan dibuang (*dropped*) atau dengan kata lain paket hilang (*packet loss*) [4].

Tabel 4 Hasil Pengukuran Skenario Kedua (*Bandwidth* 11 Mbps)

Ukuran <i>Window</i>	Hasil Pengukuran		
	<i>Packet Loss</i> (bit/s)	<i>Delay</i> (s)	<i>Throughput</i> (bit/s)
1.460	277,58222	0,00337	349.960,82667
2.920	262,00889	0,00387	382.106,94000
4.380	248,20444	0,00387	423.802,93111
5.840	260,40889	0,00447	391.418,23778
7.300	243,68000	0,00490	433.566,02444
8.760	251,47556	0,00490	404.192,96667
10.220	268,35556	0,00562	384.822,80667
11.680	243,44000	0,00592	410.417,51778
13.140	268,38222	0,00631	411.182,21778
14.600	278,04444	0,00678	344.030,23111

Hasil pengukuran sebagaimana diperlihatkan pada Tabel 4 diperoleh dengan beban *bandwidth* pada MN sebesar 11 Mbps untuk setiap WLAN yang digunakan. Berdasarkan pada Tabel 4, nilai *packet loss* terendah terlihat pada ukuran *window* 11.680 byte sebesar 243,44000 bit/s. Peningkatan ukuran *window* dalam pengendali kemacetan mengakibatkan *packet loss* pada *router* yang memiliki ukuran *buffer* kecil, atau *router* mengalami kemacetan [16], namun kenaikan ukuran *window* hanya terjadi sekali (di awal koneksi), sisanya diatur dalam mekanisme pengendali kemacetan.

Nilai *delay* sebagaimana diperlihatkan pada Tabel 4 semakin meningkat seiring dengan peningkatan nilai ukuran *window*. Peningkatan *delay* terjadi karena protokol melakukan transmisi ulang akibat *packet loss* pada jaringan [15].

Berdasarkan pada hasil pengukuran (Tabel 4), nilai *throughput* mengalami peningkatan, selain karena *bandwidth* pada jaringan yang lebar (100 Mbps), juga karena jumlah *node* yang dilibatkan. Meskipun nilai *packet loss* dan nilai *delay* cukup tinggi, namun menurut pengamatan yang dilakukan, tingginya *packet loss* disebabkan jumlah data yang ditransmisikan cukup besar sehingga menyebabkan antrian yang penuh. Paket yang tidak dapat masuk kedalam antrian akan dibuang (*dropped*) atau dengan kata lain paket hilang (*packet loss*) [4].

3. 1.3 Skenario Ketiga

Skenario Ketiga menggunakan tiga buah jalur, namun pada skenario ini, tidak lagi menggunakan *link model*, sehingga AP (FA) langsung dihubungkan dengan HA dengan menggunakan *wireless*. Ketiga jalur pada Skenario Ketiga menggunakan *wireless* dengan nilai *bandwidth* masing-masing percobaan 11 Mbps yang dihubungkan dari AP hingga ke HA. Berbeda dengan *bandwidth* pada AP (FA) yang menuju ke MN, nilai *bandwidth*nya diset dengan 11 Mbps. Secara lengkap hasil pengukuran dengan menggunakan Skenario Ketiga, baik dengan menggunakan *bandwidth* 1 Mbps maupun 11 Mbps, sebagaimana diperlihatkan pada Tabel 5 dan Tabel 6 dengan setiap percobaan dilakukan selama 60 menit atau 3.600 detik menggunakan ukuran *window* yang dimulai dari 1.460 byte hingga 14.600 byte.

Tabel 5 Hasil Pengukuran Skenario Ketiga (*Bandwidth* 1 Mbps)

Ukuran <i>Window</i>	Hasil Pengukuran		
	<i>Packet Loss</i> (bit/s)	<i>Delay</i> (s)	<i>Throughput</i> (bit/s)
1.460	251,73778	0,00862	21.989,00889
2.920	261,79556	0,00959	23.884,68444
4.380	251,74222	0,00784	16.770,82667
5.840	228,03111	0,01052	22.386,21778
7.300	251,52889	0,01050	21.460,09778
8.760	237,98222	0,01077	19.068,71111
10.220	261,35556	0,01029	17.035,91556
11.680	247,03556	0,01311	22.439,59556
13.140	239,03556	0,01345	20.560,27111
14.600	248,18222	0,01386	20.071,16889

Berdasarkan hasil pengukuran yang dilakukan sebagaimana diperlihatkan pada Tabel 5 dengan menggunakan Skenario Ketiga. Pengujian yang diperlihatkan pada Tabel 5 menggunakan 1 Mbps pada masing-masing WLAN MN, sedangkan pada HA dan AP (FA) menggunakan *bandwidth* sebesar 11 Mbps. Tabel 5 memperlihatkan bahwa *packet loss* tertinggi terjadi pada ukuran *window* 2.920 byte sebesar 261,79556 bit/s yang berbanding lurus dengan nilai *throughput* sebesar 23.884,68444 bit/s, dimana pada jaringan terjadi antrian yang penuh sehingga paket yang baru tiba, tidak memiliki tempat sehingga dibuang dengan kata lain *packet loss* [4]. Faktor yang menyebabkan perbedaan nilai *packet loss* diantaranya *bit error* pada jaringan *wireless*, diskoneksi pada saat terjadinya perpindahan kanal, dan kemacetan karena antrian yang berlebihan dalam jaringan [1]. Berbeda dengan nilai *delay*, semakin tinggi nilai ukuran *window*, nilai *delay* juga semakin tinggi sebagaimana diperlihatkan Tabel 5. Semakin besar nilai ukuran *window* maka semakin besar pula paket yang dapat ditransmisikan, dimana hal ini akan berpengaruh pada antrian yang terdapat dalam jaringan, sehingga berpengaruh pada transmisi ulang. *Delay* juga dapat disebabkan oleh *node* yang digunakan, dalam hal ini digunakan *node* yang dihubungkan dengan *wireless* [15]. Tingginya nilai *bandwidth* dalam suatu jaringan akan sulit dipertahankan apabila terdapat nilai *delay*, dimana paket yang ditransmisikan memiliki waktu yang telah berlalu sementara paket tersebut masih dalam perjalanan dari sumber ke tujuan [17].

Tabel 6 Hasil Pengukuran Skenario Ketiga (*Bandwidth* 11 Mbps)

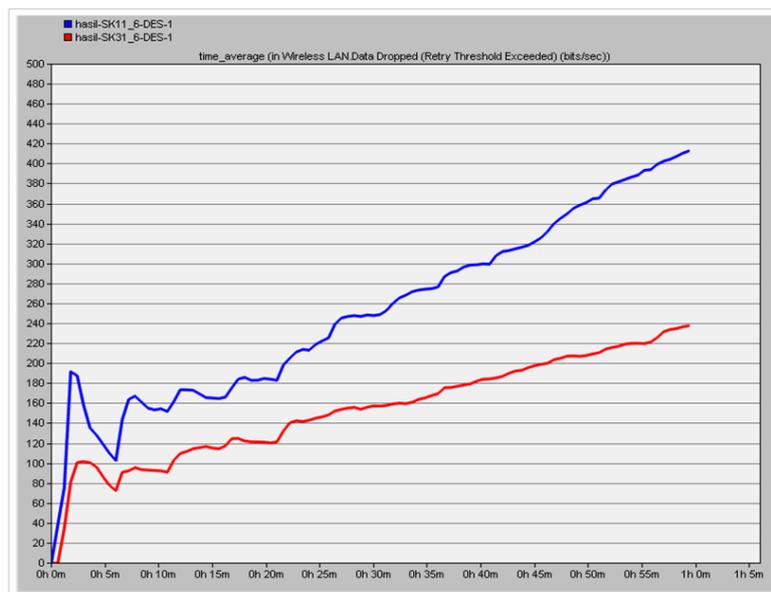
Ukuran <i>Window</i>	Hasil Pengukuran		
	<i>Packet Loss</i> (bit/s)	<i>Delay</i> (s)	<i>Throughput</i> (bit/s)
1.460	294,90222	0,00279	19.199,86667
2.920	298,49333	0,00299	21.816,40889
4.380	303,18667	0,00311	19.458,47111
5.840	279,16000	0,00317	18.942,39111
7.300	291,58667	0,00334	19.415,44000
8.760	277,19556	0,00360	19.665,91111
10.220	287,89333	0,00350	17.510,98222
11.680	302,83556	0,00373	18.058,67111
13.140	285,16000	0,00406	19.009,36889
14.600	315,84000	0,00412	18.539,50667

Berdasarkan hasil pengukuran yang dilakukan sebagaimana diperlihatkan pada Tabel 6 dengan menggunakan Skenario Ketiga. Pengujian yang diperlihatkan pada Tabel 6 menggunakan 11 Mbps pada masing-masing WLAN MN, sedangkan pada HA dan AP (FA) menggunakan *bandwidth* sebesar 11 Mbps. Tabel 6 memperlihatkan bahwa *packet loss* tertinggi terjadi pada ukuran *window* 14.600 byte sebesar 315,84000 bit/s. Faktor yang menyebabkan perbedaan nilai *packet loss* diantaranya *bit error* pada jaringan *wireless*, diskoneksi pada saat

terjadinya perpindahan kanal, dan kemacetan karena antrian yang berlebihan dalam jaringan [1]. Berbeda dengan nilai *delay*, semakin tinggi nilai ukuran *window*, nilai *delay* juga semakin tinggi sebagaimana diperlihatkan Tabel 6. Semakin besar nilai ukuran *window* maka semakin besar pula paket yang dapat ditransmisikan, dimana hal ini akan berpengaruh pada antrian yang terdapat dalam jaringan, sehingga berpengaruh pada transmisi ulang. Penyebab lain terjadinya *delay* adalah penggunaan *node*, dalam hal ini digunakan *node* yang dihubungkan dengan *wireless* [15] Tingginya nilai *bandwidth* dalam suatu jaringan akan sulit dipertahankan apabila terdapat nilai *delay*, dimana paket yang ditransmisikan memiliki waktu yang telah berlalu sementara paket tersebut masih dalam perjalanan dari sumber ke tujuan [17]. Nilai *throughput* tertinggi sebagaimana diperlihatkan Tabel 6 terdapat pada ukuran *window* 2.920 byte sebesar 21.816,40889 bit/s.

3.2 Analisis Perbandingan Hasil

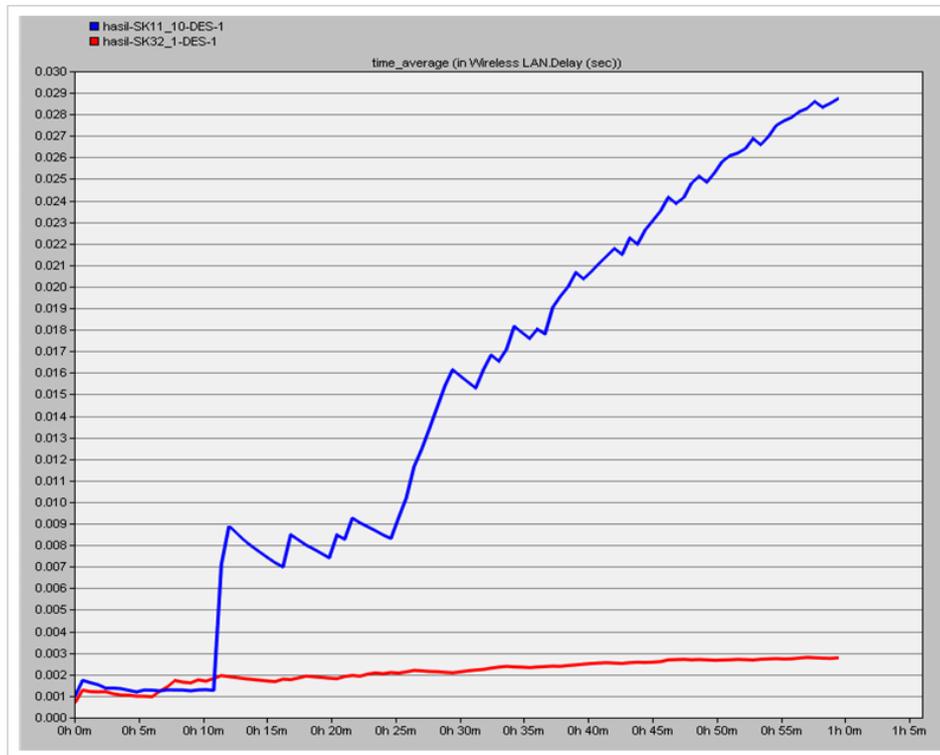
Berdasarkan pengukuran yang dilakukan pada *simulator*, maka tingkat *packet loss* terendah berada pada Skenario Ketiga dengan *bandwidth* WLAN MN sebesar 1 Mbps. Nilai tersebut diperoleh dari pengukuran sebagaimana diperlihatkan Tabel 5 dengan ukuran *window* sebesar 8.760 byte dengan nilai *packet loss* sebesar 237,9822222 bit/s. Secara rata-rata, nilai *packet loss* terendah terdapat pada Tabel 5. Sebaliknya, nilai *packet loss* tertinggi terlihat pada Skenario Pertama dengan *bandwidth* WLAN MN sebesar 1 Mbps (Tabel 1). Nilai tersebut diperoleh dari pengukuran dengan ukuran *window* 8.760 byte sebesar 413,1911111 bit/s. Perbandingan *packet loss* tertinggi dan terendah sebagaimana diperlihatkan Gambar 1.



Gambar 1 Perbandingan *Packet Loss*

Berdasarkan pada Gambar 1 terlihat bahwa Skenario Pertama dengan *bandwidth* sebesar 1 Mbps percobaan 6 (SK11-6) menunjukkan nilai *packet loss* yang tertinggi, sebaliknya Skenario Ketiga dengan *bandwidth* sebesar 1 Mbps percobaan 6 (SK31-6) yang memperlihatkan nilai *packet loss* terendah. Tingginya *packet loss* pada Skenario Pertama didasarkan pada jumlah antrian yang besar pada jaringan menggunakan *bandwidth* 100 Mbps atau kabel sebagai *link model*, sehingga pada saat transmisi melalui *wireless* dengan *bandwidth* yang telah ditentukan (1 Mbps) menuju ke MN menyebabkan antrian. Perbedaan *bandwidth* pada kedua sisi dalam suatu jaringan akan menyebabkan antrian paket, dimana pada saat tertentu paket yang ada dalam antrian akan dibuang dengan kata lain paket hilang (*packet loss*) [4]. Paket hilang terjadi ketika satu atau lebih paket data transmisi ke dalam jaringan dan mengalami kegagalan mencapai tujuannya [1].

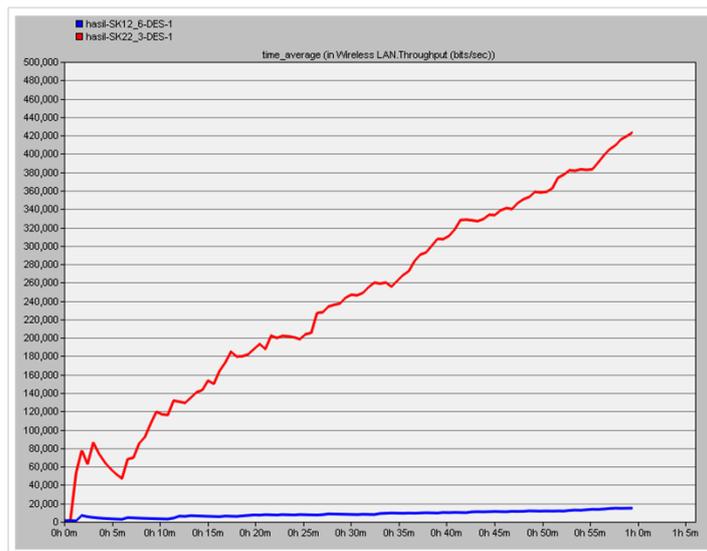
Pengukuran dengan menggunakan *simulator* juga memperlihatkan nilai *delay* tertinggi dari semua percobaan sebagaimana diperlihatkan pada Skenario Pertama (Tabel 1) dengan ukuran *window* 14.600 byte sebesar 0,028744 s (28,744 ms). Nilai terendah terlihat pada Skenario Ketiga (Tabel 6) dengan ukuran *window* 1.460 byte sebesar 0,002795 s (2,795 ms). Perbandingan *delay* tertinggi dan terendah sebagaimana diperlihatkan Gambar 2.



Gambar 2 Perbandingan Delay

Berdasarkan pada Gambar 2 terlihat bahwa Skenario Pertama dengan *bandwidth* WLAN MN sebesar 1 Mbps percobaan 10 (SK11-10) menunjukkan nilai *delay* yang tertinggi, sebaliknya Skenario Ketiga dengan *bandwidth* WLAN MN sebesar 11 Mbps percobaan 1 (SK32-1) yang memperlihatkan nilai *delay* terendah. Tingginya *delay* pada Skenario Pertama didasarkan pada jumlah antrian ataupun *node* yang dilibatkan. Antrian yang panjang akan menyebabkan paket hilang dan pengendali kemacetan akan melakukan transmisi ulang. Waktu yang dibutuhkan untuk melakukan transmisi ulang merupakan salah satu penyebab *delay* pada jaringan komputer [15]. Begitu pula paket yang gagal mencapai tujuannya pada waktu tertentu akan ditransmisi ulang oleh pengendali kemacetan.

Hasil pengukuran yang dilakukan juga memperlihatkan bahwa nilai *throughput* paling tinggi terdapat pada Skenario Kedua (Tabel 4) dengan ukuran *window* 4.380 byte sebesar 423.802,9311 bit/s. Nilai *throughput* terendah pada Skenario Pertama (Tabel 2) dengan ukuran *window* 8.760 byte sebesar 15.288,4622 bit/s. Perbandingan *throughput* tertinggi dan terendah sebagaimana diperlihatkan Gambar 3.



Gambar 3 Perbandingan Throughput

Berdasarkan pada Gambar 3 terlihat bahwa Skenario Kedua dengan *bandwidth* sebesar 11 Mbps percobaan 3 (SK22-3) menunjukkan nilai *throughput* yang tertinggi, sebaliknya Skenario Satu dengan *bandwidth* sebesar 11 Mbps percobaan 6 (SK12-6) yang memperlihatkan nilai *throughput* terendah. Selain karena pengaruh *link model* dengan *bandwidth* 100 Mbps yang digunakan, tingginya *throughput* pada Skenario Kedua juga dipengaruhi oleh *bandwidth* yang ada pada WLAN MN yaitu 11 Mbps.

Secara keseluruhan, hasil pengukuran terbaik terlihat pada Skenario Kedua dengan melihat *throughput* dari Tabel 3 dan Tabel 4. Adapun nilai *packet loss* yang paling tinggi dapat disebabkan oleh berbagai faktor seperti antrian yang berlebihan karena kemacetan [1], [4], ataupun karena transmisi yang telah dikirim namun tidak pernah sampai pada jaringan yang dituju [4]. Begitu pula dengan *delay* yang lebih tinggi dari percobaan lain, dapat dipengaruhi oleh jarak, antrian, ataupun penggunaan *node* serta terjadinya transmisi ulang [15].

4. KESIMPULAN

Berdasarkan uraian yang telah dijelaskan pada bab sebelumnya, maka dapat ditarik kesimpulan sebagai berikut :

- SCTP fitur *multihoming* dapat dimodelkan pada OPNET dengan modifikasi TCP menggunakan model *Mobile Node (MN)*, *Home Agent (HA)*, *Foreign Agent (FA)* dan *Correspondent Node (CN)*, sedangkan untuk berkomunikasi dengan *Access Point (AP)* yang *overlap* dengan AP asalnya, MN membutuhkan *Agent Solicitation* sebagaimana telah digunakan dalam [18,19,20];
- Modifikasi ukuran *window* dilakukan pada pengendali kemacetan yang ada pada OPNET berdasarkan [21,16], dimana modifikasi ukuran *window* yang digunakan dalam penelitian ini adalah 1.460 byte (1 segmen), 2.920 byte (2 segmen) hingga 14.600 byte (10 segmen) dengan nilai MSS sebesar 1.460 byte [14]. Modifikasi ukuran *window* pada pengendali kemacetan berpengaruh terhadap *packet loss*, *delay* dan *throughput* sebagaimana diperlihatkan Tabel 1, Tabel 2, Tabel 3, Tabel 4, Tabel 5 dan Tabel 6;
- Skenario Kedua merupakan skenario yang memiliki nilai *throughput* yang terbaik dengan *bandwidth* 11 Mbps pada WLAN MN sebagaimana diperlihatkan Tabel 4, walaupun Tabel 5 dan Tabel 6 memperlihatkan nilai *packet loss* dan *delay* yang lebih rendah. Peningkatan nilai *packet loss* pada Skenario Kedua disebabkan karena jumlah data yang dikirimkan ke tujuan menemukan antrian yang penuh tanpa ada tempat untuk menyimpan paket yang baru,

dalam waktu tertentu akan dibuang (*dropped*) atau disebut juga dengan *packet loss* [4]. Berbeda dengan nilai *delay*, selain disebabkan oleh jumlah antrian yang terdapat pada jaringan, juga karena terjadinya transmisi ulang pada *link layer* [15]; dan

- Berdasarkan pengukuran yang telah dilakukan, nilai *packet loss*, *delay* dan *throughput*, selain dipengaruhi oleh media transmisi yang digunakan juga dipengaruhi oleh jumlah *node* [11,12,15].

5. SARAN

Pengembangan penelitian lebih lanjut, diberikan saran sebagai berikut :

- Pengukuran hanya dilakukan untuk melihat *packet loss*, *delay* dan *throughput*, sehingga untuk penelitian selanjutnya dapat menambahkan keamanan sistem jaringan sebelum melakukan pengukuran; dan
- Penelitian selanjutnya dapat melakukan pengukuran dengan membedakan penyebab terjadinya *packet loss*, misalnya karena antrian yang panjang atau karena jangkauan jaringan. Begitu pula dengan pengukuran *delay* yang dapat disebabkan oleh jarak, *node* yang dilibatkan atau transmisi ulang yang dilakukan oleh protokol.

DAFTAR PUSTAKA

- [1] Liu, T., Xu, C., Guan, J. dan Zhang, H., 2012, Loss Detection Mechanism in SCTP Heterogeneous Wireless Networks, in *Business Computing and Global Informatization (BCGIN)*, 2012 Second International Conference on, pp. 679–682.
- [2] Tomsho, G., 2011, *Guide to Networking Essentials*, 6th Edition, 6th edn, Course Technology Press, Boston, MA, United States.
- [3] Postel, J., 1981, Transmission Control Protocol, RFC 793 (Standard). Updated by RFCs 1122, 3168. URL: <http://www.ietf.org/rfc/rfc793.txt>.
- [4] Kurose, J. dan Ross, K., 2013, *Computer Networking: A Top-down Approach*, Always learning, Pearson. URL: <http://books.google.co.id/books?id=gbAhngEACAAJ>
- [5] Natarajan, P., Baker, F., Amer, P. D. dan Leighton, J. T., 2009, Sctp: What, Why, and How, *IEEE Internet Computing* 13(5), 81–85
- [6] Seo, D.-W., Kim, J.-G. dan Kim, J. G., 2008, Analysis of Dynamic Congestion Control of Sctp, in *Convergence and Hybrid Information Technology, 2008. ICHIT '08*. International Conference on, pp. 687–693.
- [7] Iyengar, J., Amer, P. dan Stewart, R., 2004, Retransmission Policies for Concurrent Multipath Transfer Using Sctp Multihoming, in *Networks, 2004. (ICON 2004)*. Proceedings. 12th IEEE International Conference on, Vol. 2, pp. 713–719 vol.2.
- [8] Liu, J., Zou, H., Dou, J. dan Gao, Y., 2008, Reducing Receive Buffer Blocking In Concurrent Multipath Transfer, in *Circuits and Systems for Communications, 2008. ICCSC 2008*. 4th IEEE International Conference on, pp. 367–371.
- [9] Joe, I. dan Yan, S., 2009, Sctp Throughput Improvement with Best Load Sharing Based on Multihoming, in *NCM*, pp. 138–142.

-
- [10] Budzisz, L., Ferrus, R., Grinnemo, K., Brunstrom, A. dan Casadevall, F., 2007, An Analytical Estimation of the Failover Time in SCTP Multihoming Scenarios, *in Wireless Communications and Networking Conference, 2007.WCNC 2007*. IEEE, pp. 3929–3934.
- [11] Siddiqui, F. dan Zeadally, S., 2006, SCTP multihoming support for handoffs across heterogeneous networks, *in Communication Networks and Services Research Conference, 2006. CNSR 2006*. Proceedings of the 4th Annual, pp. 8 pp.–250
- [12] Shieh, C.-S., Lin, I.-C. dan Lai, W. K., 2008, Improvement of SCTP Performance in Vertical Handover, *in ISDA (3)*, pp. 494–498.
- [13] Eklund, J., Grinnemo, K.-J. dan Brunstrom, A., 2012, On the Use of an Increased Initial Congestion Window to Improve mSCTP Handover Performance, *in AINA Workshops*, pp. 1101–1106
- [14] Stewart, R., 2007, RFC4960: Stream Control Transmission Protocol. Updated by RFCs 6096, 6335.
- [15] Welzl, M., 2005, Network Congestion Control: Managing Internet Traffic (Wiley Series on Communications Networking & Distributed Systems), John Wiley & Sons.
- [16] Chu, H. J., Dukkupati, N., Cheng, Y. dan Mathis, M., 2013, Increasing TCP's Initial Window. URL: <http://www.rfc-editor.org/rfc/rfc6928.txt>
- [17] Patel, S., Gupta, P. K., Garg, A., Mehrotra, P. dan Chhabra, M., 2012, Comparative Analysis of Congestion Control Algorithms Using ns-2.
- [18] Perkins, C., 2002, IP Mobility Support for IPv4. [Standards Track RFC 3344]. URL: <http://www.ietf.org/rfc/rfc3344.txt>
- [19] Perkins, C., 2010, IP Mobility Support for IPv4, Revised. [Standards Track RFC 944]. URL: <http://www.ietf.org/rfc/rfc5944.txt>
- [20] Jerjees, Z., 2010, Design of Interface Selection Protocols for Multi-homed Wireless Networks, *Master's thesis, Brunel University, United Kingdom*.
- [21] Allman, M., Floyd, S. dan Partridge, C., 2002, Increasing TCP's Initial Window, RFC 3390 (Proposed Standard). URL: <http://www.ietf.org/rfc/rfc3390.txt>