

PENINGKATAN PERFORMA PROTOKOL SCTP DENGAN MEKANISME MULTI-STREAMING DAN PERBANDINGANNYA DENGAN PROTOKOL DCCP

I Putu Eka Y. Budiana ¹, Supeno Djanali ²

^{1,2}Jurusan Teknik Informatika, Fakultas Teknologi Informasi,
Institut Teknologi Sepuluh Nopember
Email: ptekabudiana@gmail.com

Abstrak. *Teknologi streaming menjadi semakin penting dengan perkembangan internet sekarang ini karena sebagian besar pengguna tidak memiliki akses cukup cepat untuk men-download file multimedia berukuran besar dengan cepat. Protokol TCP yang berada dilapisan transport komunikasi jaringan, memiliki kelemahan untuk aplikasi-aplikasi seperti komunikasi yang bersifat real-time. karena delay-nya yang tinggi. Sedangkan protokol UDP, kurang adanya mekanisme kontrol kongesti sehingga injeksi data ke jaringan yang terlalu tinggi dapat mempercepat peningkatan kongesti jaringan. Untuk meningkatkan kualitas layanan pada streaming video MPEG-4, beberapa protokol lapisan transport telah dikembangkan oleh Internet Engineering Task Force (IETF). Dua protokol yang banyak diteliti dan dimanfaatkan adalah SCTP dan DCCP. Dalam penelitian ini keunggulan fitur multistreaming dan partial reliable dari protokol SCTP dikembangkan untuk mengirim streaming video secara rate adaptive dalam framework Evalvid-RA. Hasil modifikasi yang dinamakan E-PR-SCTP, kemudian dievaluasi performanya dengan protokol DCCP yang dilakukan pada kondisi jaringan yang sama. Dari hasil uji coba terlihat ada peningkatan performa dan kualitas distribusi video dari pengembangan protokol SCTP, dibandingkan dengan protokol DCCP maupun SCTP standar.*

Kata Kunci: *PR-SCTP, kontrol kongesti, multi-stream, Evalvid-RA.*

Streaming adalah teknik untuk mentransfer data sehingga dapat diproses sebagai aliran yang tetap dan berkesinambungan [1]. Teknologi streaming menjadi semakin penting dengan pertumbuhan internet karena sebagian besar pengguna tidak memiliki akses cukup cepat untuk men-download file multimedia berukuran besar dengan cepat. Dengan streaming, browser klien dapat mulai menampilkan data sebelum seluruh file telah terkirim.

Kebanyakan protokol transport yang digunakan untuk aplikasi internet seperti; *World Wide Web* (WWW), email, dan transfer file adalah *Transmission Control Protocol* (TCP) [2]. Protokol ini cocok digunakan pada aplikasi-aplikasi yang membutuhkan kehandalan (*reliable*) dan pengiriman *byte-byte stream* yang sesuai urutan. Akan tetapi TCP tidak cocok untuk aplikasi-aplikasi seperti komunikasi yang *real-time* karena transmisi ulang (*retransmission*) yang ketat menambah ekstra waktu *delay-nya*. Oleh karena itu *user*

datagram protocol (UDP) yang menyediakan pembentukan jaringan yang sangat simpel, umumnya digunakan sebagai protokol transport untuk aplikasi *streaming* multimedia. Namun, kinerja UDP tidak memuaskan karena beberapa keterbatasan seperti kurangnya mekanisme kontrol kongesti (banyaknya penimbunan data sehingga menimbulkan kemacetan aliran pada jaringan) [3].

Streaming Transmission Control Protocol (SCTP), diusulkan oleh *Internet Engineering Task Force* (IETF) sebagai alternatif lain protokol transport sebelumnya dengan beberapa fitur unggulan yang dimilikinya [4]. SCTP menyediakan layanan transmisi *reliable* dan mekanisme *flow control* yang serupa dengan TCP. Disamping itu, SCTP dapat mendukung transmisi *unreliable* seperti yang digunakan pada mekanisme transmisi UDP. Perbedaan utama antara SCTP dengan protokol transport sekarang ini (TCP dan UDP) adalah memungkinkan setiap asosiasi untuk memiliki dukungan layanan *multi-stream* dan *multi-*

homming. Dukungan terhadap *multistreaming* digunakan untuk mengatasi *Head of Line (HoL)* yang menjadi masalah pada TCP dengan *stream* tunggalnya. Sedangkan *multi-homming* memungkinkan untuk menggunakan lebih dari satu jalur (*path*) dengan alamat IP yang berbeda pada saat pembentukan koneksi yang dapat disamakan pada TCP [5]. Dengan demikian, SCTP bisa menyediakan berbagai aplikasi yang lebih luas, terutama untuk internet multimedia, salah satu yang dibahas adalah streaming video.

Teknologi kompresi untuk aplikasi yang luas diterapkan pada standar kompresi streaming video adalah MPEG-4. MPEG-4 menggunakan tiga jenis *frame* dalam skema pengkodean, yaitu; *intra-frame (I-frame)*, *forward predicted frames (P-frame)*, dan *bi-directional predicted frames (B-frames)* [6]. Dengan adanya perluasan protokol SCTP yaitu *Partial Reliability SCTP (PR-SCTP)*, memungkinkan protokol ini untuk mengatur tingkat reliabilitas transmisi pesan-pesan dari data *frame* stream video MPEG-4. Penelitian yang dilakukan [7] menunjukkan keuntungan performa PR-SCTP bila mengatur prioritas transmisi pada frame-frame yang paling penting. Dalam MPEG-4 misalnya, *I-frame* membawa informasi yang mandiri dan memiliki dampak yang besar jika tidak disampaikan, sementara *P-frame* atau *B-frame* dapat di-*drop* dengan dampak yang terbatas. PR-SCTP digunakan dalam konteks ini untuk memprioritaskan pesan tertentu selama terjadi kongesti jaringan. Hasil eksperimen yang dihasilkan dalam penelitian [8] dan [10] menunjukkan bahwa kemampuan PR-SCTP sangat baik bila dibandingkan pada protokol transport dengan layanan *reliable* yang tinggi seperti TCP maupun SCTP standar yang masih memanfaatkan prioritas pada setiap pesan.

Seiring makin tingginya pemanfaatan multimedia di internet, perkembangan lapisan transport semakin cepat untuk memenuhi kebutuhan akan kualitas layanan yang lebih baik. *Datagram Congestion Control Protocol (DCCP)*, merupakan protokol transport baru yang dikembangkan dengan alasan sama seperti SCTP [11]. Ide utama dari DCCP adalah tetap menggunakan mekanisme aliran paket-paket yang tidak *reliable* seperti UDP (tidak adanya retransmisi dalam kasus paket drop). Menggunakan SCTP standar memiliki karakteristik yang sama dengan TCP

yaitu transmisi yang *reability* dan *in-order*. Sedangkan DCCP sudah peduli terhadap *congestion control* dan *flow control* yang tidak terdapat pada UDP. Pada penelitian ini memanfaatkan *Congestion Control Identifier 3 (CCID 3)* yang berbasis *TCP Friendly Rate Control (TFRC)*. TFRC menggunakan mekanisme kontrol kongesti *rate-controlled* dan *equation-based* dalam rangka untuk mendapatkan rate yang halus, yang layak untuk pemanfaatan *bandwidth* yang berimbang saat berkompetisi dengan aliran TCP lain.

Meneliti performa dari suatu parameter-parameter penting pada *streaming* multimedia (baik pada lapisan transport maupun aplikasi), dengan melakukan uji coba pada jaringan multimedia yang sebenarnya (*testbed*) memberikan hasil yang lebih realistis daripada simulasi. Akan tetapi membutuhkan biaya dan upaya yang sangat besar untuk membangun dan mengkonfigurasi perangkat keras maupun topologi jaringan. Sehingga simulasi jaringan seperti *network simulator* sangat menarik untuk dijadikan alternatif karena biaya, skalabilitas, fleksibilitas dan pengembangan yang cepat. Salah satu *network simulator* yang luas digunakan untuk penelitian jaringan komputer selama ini adalah Network Simulator versi 2 (NS2). Keuntungan menggunakan NS2 ini karena berbasis *open source*, sehingga mudah dikembangkan. Untuk mengembangkan *streaming video* pada NS2 timbul masalah bagaimana cara untuk menambahkan pembangkit trafik video dari sumber ke simulator ini.

Menurut [12], untuk mengevaluasi transmisi video, ada tiga cara yang berbebeda; *real bit stream*, *video traffic model*, dan *traffic trace*. Metode *real bit stream* menggunakan output yang aktual dari encoding video yang dipergunakan untuk evaluasi transmisi video. Real Tracer salah satu tool untuk mengukur performa real video. Satu keuntungan dari metode ini bahwa kualitas suatu video untuk dapat dievaluasi secara visual. Metode *video traffic model* dengan cara menangkap bit-bit stream real video secara matematis. Metode ini biasanya dibangun berdasarkan properti statistika dari sebuah set contoh-contoh *video-trace* pada *traffic real-video*. *Transform Expand Sample (TES)* adalah sebuah contoh dari jenis metodologi ini untuk *generate* data yang mendekati cocok dengan observasi-observasi

mana saja yg diberikan pada sebuah seri waktu. Metode ini memiliki kekurangan dalam memvisualisasikan video hasil uji transmisi. Metode *traffic trace* adalah sebuah abstraksi dari arus real video. Biasanya memberikan nomor frame, tipe frame (I, B, P) dan ukuran frame dalam sebuah file teks untuk menjelaskan karakteristik *traffic real video*. Keuntungan menggunakan *traffic trace* adalah aman dari hak cipta karena tidak memuat informasi video yang aktual. Salah satu contoh metode ini adalah *frame work EVALVID* [13], yang kemudian dikembangkan menjadi EVALVID-RA[14] untuk mendukung distribusi video secara rate adaptive.

Penelitian[15] membandingkan efektivitas transmisi UDP, DCCP, dan SCTP pada IEEE 802.11 WLAN. Dari hasil yang didapat menunjukkan SCTP kurang optimal dibandingkan protokol DCCP karena mekanisme SCTP standar yang masih menggunakan stream tunggal layaknya TCP dengan reliable yang ketat. Dengan mengadaptasi *framework EVALVID-RA* untuk menunjang *streaming* video MPEG-4 pada protokol SCTP di *Network Simulator versi 2* (NS2), protokol SCTP yang dikembangkan pada penelitian ini yang diberi nama E-PR-SCTP. Adapun tujuan pada penelitian ini adalah;

1. Menerapkan mekanisme *multi-streaming* pada modul SCTP standar dengan distribusi paket secara *partial reliable* untuk transmisi *streaming* video MPEG-4 pada melalui NS2.
2. Meningkatkan kehandalan layanan protokol SCTP untuk transmisi video MPEG-4 akibat gangguan di jalur lintasan pengiriman dengan mekanisme prioritas paket.

Penelitian Terkait

Studi perbandingan mengenai performa PR-SCTP dengan protokol-protokol TCP dan UDP sudah dilakukan oleh [8]. Uji coba dilakukan pada jaringan bergerak dengan dilalui lalu lintas video MPEG-4. Fokus pada penelitian ini adalah performa protokol transport atas *delay*-nya selama *handoff*. Ketika transmisi PR-SCTP diatur menjadi *unreliable*, tingkat performanya mirip pada UDP. Sedangkan dengan mengatur prioritas *reliabel* pada *frame*

tertentu, tingkat *delay* PR-SCTP lebih rendah dibandingkan dengan TCP.

Namun penelitian yang dilakukan pada [15] hasil simulasi NS2 protokol SCTP pada jaringan *wireless* LAN menunjukkan, untuk lalu lintas video dengan kecepatan sampai 4 Mbps, *throughput* SCTP dan DCCP performanya sama dan tidak ditemukan adanya paket yang hilang. Namun dengan kecepatan diatas 4 Mbps, performa *throughput* SCTP menurun diantara protokol DCCP dan UDP dengan rata-rata kehilangan paket paling tinggi. Selanjutnya dengan diberikan latar belakang lalu lintas lain, performa DCCP lebih baik dari pada SCTP.

Beberapa penelitian sebelumnya, lebih cenderung memanfaatkan modul SCTP standar yang disediakan simulator jaringan NS2. Akhir-akhir ini beberapa penelitian fokus pada pemanfaatan fitur *multi-streaming*. Penelitian [9] misalnya, dengan memberikan kebijakan yang berbeda pada masing-masing *stream*-nya. Sedangkan pada [10] mengembangkan kontrol transmisi untuk mengatur layaknya *frame-frame* untuk ditransmisikan dan mengatur penempatan *frame-frame* video MPEG pada *stream* yang berbeda, sesuai dengan prioritas *frame*. Dari hasil evaluasi, performa pengembangan SCTP ini lebih baik dari pada PR-SCTP maupun SCTP.

Penelitian [7] mengembangkan *forward_tsn chunk* dari diagram modul PR-SCTP dengan mengatur persentase pesan-pesan penting. Pada penelitian ini, membandingkan PR-SCTP dengan protokol SCTP, TCP dan UDP pada lingkungan uji coba *test-bed* hanya dengan aliran tunggal. Dari analisa uji rata-rata protokol UDP tetap lebih unggul dari protokol lainnya. Namun dengan mengatur *important message* dari 20% sampai 1%, PR-SCTP dapat mendekati performa UDP dan lebih unggul performanya dengan protokol yang *fully reliable* seperti TCP dan SCTP.

METODOLOGI

Video Sumber

Dari penelitian ini mengambil empat referensi model video berdasarkan penelitian [16] dengan kompleksitas gerakan yang berbeda-beda mulai dari kompleksitas gerakan yang rendah sampai kompleksitas gerakan yang tinggi guna melihat pengaruh *congestion*

terhadap kualitas layanan lalu lintas video melalui protokol transport. Empat jenis video yang digunakan pada penelitian ini adalah **akiyo**, **coastguard**, **news**, dan **mobile**. Adapun karakteristik dari masing-masing video seperti pada tabel 1.

Tabel 1. Karakteristik video

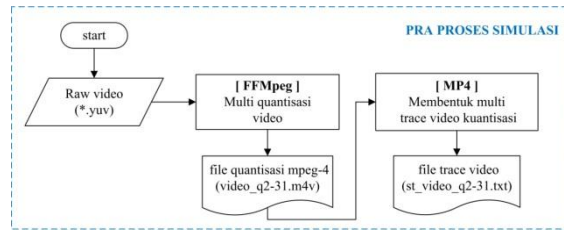
File	Akiyo	Coastguard	News	Mobile
<i>Gerakan</i>	Rendah	Tinggi	Rendah	Sedang
<i>Resolusi</i>	CIF (352x288)	CIF (352x288)	CIF (352x288)	CIF (352x288)
<i>Durasi</i>	61,2 s	61,2 s	61,2 s	61,2 s
<i>Panjang frame</i>	300	300	300	300

Sesuai dengan rekomendasi ITU [17], panjang durasi video yang akan diujikan paling sedikit 5 detik. Pada penelitian ini, semua video sumber digabung menjadi satu rangkaian dengan judul **videosumber.yuv** sehingga mendapatkan panjang durasi 244,8 detik.

Dalam transmisi video, gambar – gambar dari *raw frame* tidak dikirim dalam bentuk aslinya. Gambar-gambar dari video sumber pertama-tama di-*encode* dalam format MPEG-4 kemudian dilakukan *abstraction* untuk mendapatkan file *trace* video. Pada penelitian ini, metode yang dilakukan untuk transmisi video mengadopsi penelitian [14] dimana mendukung transmisi video secara *rate adaptive* pada jaringan. Untuk mendukung transmisi video secara *rate adaptive*, video sumber dibentuk kedalam 30 macam *encode* MPEG4 dengan tingkat skala kuantisasi yang berbeda-beda mulai dari 2 sampai 31. Skala kuantisasi 2 menghasilkan kualitas video dan *bit rate* yang paling tinggi, sedangkan kuantisasi 30 menghasilkan kualitas video dan *bit rate* yang paling rendah. Masing-masing *traffic trace* memberi informasi dari ketiga tipe *frame* yang dibentuk dari encode MPEG4 yang kemudian dibagi menjadi *Group Of Picture* (GOP). Adapun parameter pengaturan *encoder* video adalah sebagai berikut;

Tabel 2. Parameter video codec

Parameter	Setting
Quantisasi	2 - 31
B-frame	2
VOP	IBBPBBPBBPBB
GOP	12
FPS	30
Codec	MPEG-4



Gambar 1. Alur pra proses untuk membentuk kuantisasi video MPEG-4 dan *trace* video

Tabel 3. Encoding rate video MPEG-4

Q skala	Encoding rate (kb/s)	Q skala	Encoding rate (kb/s)
2	19343,5	17	1667,9
3	12603,4	18	1557,5
4	9484,4	19	1450,4
5	7299,5	20	1358,8
6	5976,5	21	1277,8
7	4965,5	22	1206,6
8	4286,1	23	1138,2
9	3685,5	24	1077,5
10	3254,3	25	1019,9
11	2880,3	26	980,0
12	2596,2	27	928,6
13	2348,6	28	892,5
14	2142,7	29	853,9
15	1958,9	30	822,3
16	1817,0	31	787,2

Gambar 1. adalah alur dalam membentuk kuantisasi video MPEG-4 dan file-file *trace* video MPEG-4 yang terkuantisasi. File *trace* video MPEG-4 terdiri dari informasi karakteristik video, dimana terdapat informasi mengenai nomor *frame*, tipe *frame*, kapasitas *frame*, fragmentasi-fragmentasi dalam segmen, dan jadwal waktu tiap *frame* video. File-file *trace* video MPEG-4 digunakan sebagai pembangkit lalu lintas video dari server ke penerima pada saat simulasi jaringan di NS2, berkat adanya informasi penjadwalan waktu dan kapasitas paket frame yang harus terkirim.

Berdasarkan informasi pada Tabel 3., besarnya *encoding rate* dari hasil proses pada masing - masing skala kuantisasi berbeda-beda. Semakin besar skala kuantisasi, semakin kecil rate dari paket video yang dikirimkan.

Metode Transmisi Video

Pada saat simulasi di NS2, batasan waktu diberikan pada masing-masing *frame* video yang akan didistribusikan agar dapat diukur

masa *frame* sampai ke tujuan. Dengan menggunakan metode [10] dalam menentukan batasan waktu, diperoleh marginal waktu yang dimiliki masing-masing *frame* dalam menentukan *stream* yang akan dilalui. Aplikasi akan menetapkan batasan waktu yang berbeda-beda sesuai dengan data *frame*-nya. Untuk menetapkan batasan waktu, diperlukan waktu maksimum yang diprediksi untuk setiap data dalam setiap pengirimannya.

Berikut adalah parameter-parameter yang terlibat dalam menentukan batasan waktu setiap data *frame* video yang akan ditransmisikan;

- Waktu *Delay* (T_{DL}) merupakan masa dimana saat aplikasi video mulai mengirim pesan data *frame* kemudian sampai ke penerima untuk diputar ulang (*play back*).
- Waktu pembentukan data (T_{CD}) merupakan waktu yang dibentuk saat data dibuat dan diterima oleh protokol.
- Waktu *playback* (T_{PB}) merupakan nilai waktu yang dimiliki data yang didapat dari penjumlahan waktu tercipta dengan waktu *delay*. Apabila data tiba pada saat melebihi masa *playback time*, maka data tidak dapat dimainkan sehingga data dibuang. Adapun persamaan waktu *playback* adalah;

$$T_{PB} = T_{CD} + T_{DL} \quad (1)$$

- Waktu *marginal playback* (T_{MG}) adalah waktu yang tersisa yang dimiliki data saat masih dalam proses pengiriman menuju aplikasi penerimaan. Waktu *marginal playback* didapat dari selisih dari estimasi waktu *playback* (T_{PB}) data dengan waktu yang sedang berjalan (T). Besarnya waktu *marginal playback* ini sebagai parameter *sub-layer* untuk menyeleksi data apakah data akan dikirim atau tidak.

$$T_{MG} = T_{PB} - T \quad (2)$$

SCTP tidak akan mentransmisi ulang bila masa *lifetime* data sudah kadaluarsa. SCTP akan mengirim pesan-pesan secara *multiple stream* sesuai dengan alokasi yang sudah ditentukan sebelumnya dengan berdasarkan tipe pesan dan panjang waktu marginalnya (T_{MG}). Klasifikasi panjang waktu marginal dikelompokkan dalam tiga kasus sesuai dengan tipe pesannya, yaitu :

$$\begin{aligned} T_{MG} &\geq Th_1 \\ T_{MG} &\geq Th_2, \\ T_{MG} &\geq Th_3, \text{ dimana: } Th_1 > Th_2 > Th_3 \end{aligned}$$

Dengan menggunakan metode (Kim, K.H, dkk., 2010), nilai-nilai parameter Th_1 , Th_2 dan

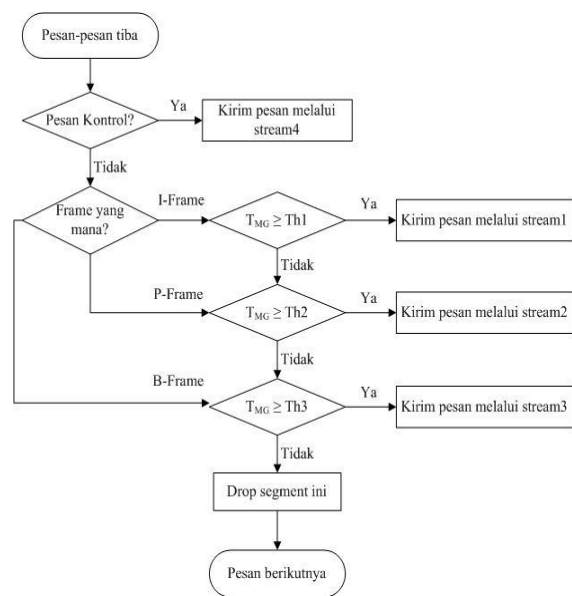
Th_3 diatur dengan menggunakan nilai dari *Retransmission Timeout* (RTO) yang dibentuk oleh PR-SCTP secara terus-menerus. Sehingga batas ambang yang dibijaksanai dari kondisi jaringan adalah:

$$Th_1 = RTO * 4 + \alpha_1, \text{ untuk stream I} \quad (3)$$

$$Th_2 = RTO * 2 + \alpha_2, \text{ untuk stream P} \quad (4)$$

$$Th_3 = RTO * 1 + \alpha_3, \text{ untuk stream B} \quad (5)$$

Th_1 yang diatur menjadi lebih dari empat kali RTO supaya dapat memungkinkan transmisi dilakukan empat kali. Demikian juga nilai-nilai Th_2 dan Th_3 diatur yang lebih dari dua kali lipat RTO dan satu kali lipat RTO. Dimana dalam metode ini, nilai α diseti 1/8



Gambar 2. Metode Transmisi Data [10]

Adapun algoritma transmisi paket-paket video frame dapat dilihat pada Gambar 2. file-file *st_video_q2-31.txt* yang diperoleh pada tahap pra proses, akan dipanggil untuk mendapatkan informasi kapasitas rate tiap frame dan penjadwalan waktu untuk transmisi. Marginal waktu tiap frame yang diperoleh, sesuai dengan keadaan jaringan. Frame-frame video diatur prioritas pengiriman, dimana frame I diupayakan harus sampai di tujuan, kemudian diikuti dengan frame P dan B.

HASIL DAN PEMBAHASAN

Skenario Uji Coba

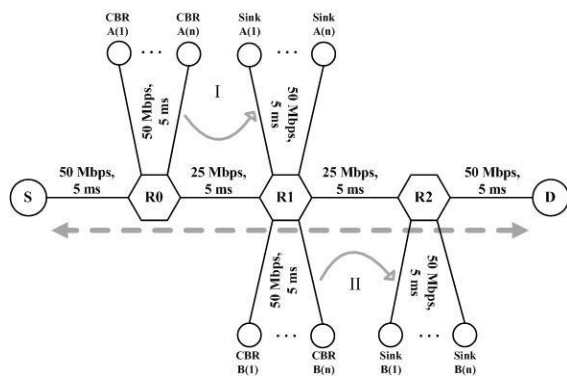
Skenario ujicoba dilakukan dengan dua model uji untuk melihat kehandalan layanan

distribusi protokol transport dalam mengangkut paket-paket video. Semua protokol masing-masing dilakukan pengujian pada kondisi jaringan yang sama agar mendapatkan penilaian perbandingan yang adil. Adapun skenario kondisi uji coba yang dibentuk :

1. Dengan cara meningkatkan beban pada *background traffic*.
2. *Error model* dengan peningkatan *packet loss ratio* pada router.

Skenario I

Tahap pertama ini, menguji protokol pada jaringan dengan kondisi peningkatan kepadatan lalu lintas jaringan pada *background*-nya dengan cara menambah jumlah node. Pada Gambar 3., skenario uji I digunakan untuk menciptakan kondisi jaringan agar tampak nyata, dimana terdapat dua bagian kelompok *background traffic* CBR yang dapat diatur jumlah node-node pada kedua sisi-nya. Oleh karena itu, pada skenario peningkatan beban *background traffic* dilakukan dengan cara menambah jumlah node yang sama pada masing-masing kelompok sehingga dengan cara ini dapat digunakan sebagai parameter pengujian dari segi jumlah node keseluruhan pada bagian *background*-nya. Masing-masing node didesain dengan memiliki ukuran paket sebesar 1500 byte. Sehingga dengan penambahan jumlah node, ada peningkatan beban paket-paket yang diinjeksikan pada bagian *background* dengan rate yang konstan. Pada penelitian thesis ini, maksimum jumlah *node background* yang digunakan sebagai parameter analisa sampai dengan 60 node, karena berdasarkan hasil pengujian lebih dari atau sama dengan 60 node, jumlah paket *frame* video yang diperoleh terlalu sedikit dari jumlah *frame* video awal. Sehingga parameter jumlah *node background* yang digunakan adalah 10, 20, 30, 40, 50, dan 60. Dari uji coba pertama ini



Gambar 3. Topologi skenario uji I

akan didapat informasi pengaruh *background traffic* terhadap kehandalan protokol dari segi analisa *delay*, *jitter*, *throughput*, dan *frame drop*.

Tiap-tiap protokol transport yang dipakai sebagai pengujian dan analisa, disimulasikan secara bergantian agar mendapatkan keadaan lingkungan jaringan yang adil.

Delay

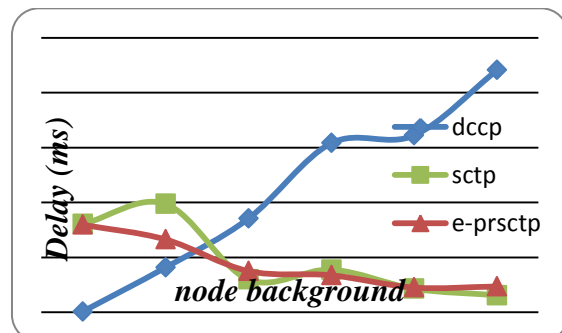
Hasil analisa yang ditunjukkan pada Tabel 4. memperlihatkan rata-rata *delay* pada tiap-tiap protokol akibat pengaruh penambahan jumlah node yang terhubung pada jaringan.

Tabel 4. Hasil perbandingan rata-rata *Delay*

Node <i>Background</i>	DCCP (ms)	SCTP (ms)	E-PRSCTP (ms)
10	0,06	8,03	7,97
20	4,08	6,87	6,63
30	8,53	2,98	3,75
40	15,40	3,88	3,37
50	16,11	2,14	2,25
60	22,05	1,55	2,35

Dari grafik diatas terlihat bahwa, *delay* antara protokol SCTP dan E-PRSCTP pada saat beban *background* jaringan rendah, tampak lebih tinggi dari protokol DCCP. Semakin banyak node yang dilibatkan, kedua protokol SCTP dan E-PRSCTP memiliki *delay* makin kecil, sedangkan pada protokol DCCP *delay*-nya meningkat.

Kesimpulan sementara dari hasil analisa *delay* ini, protokol E-PRSCTP dapat mereduksi tingkat *delay* dari protokol SCTP standar pada node 10 sebesar 0,06ms, pada node 20 sebesar 0,24ms, dan pada node 40 sebesar 0,51ms.



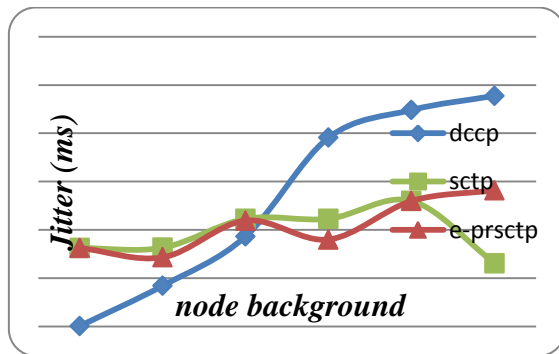
Gambar 4. Grafik rata-rata *delay* terhadap peningkatan jumlah node

Jitter

Hasil *jitter* juga sama seperti yang dilakukan pada analisa *delay*. Data yang diperoleh tiap-tiap uji pada peningkatan jumlah node, diambil rata-rata hasil *jitter*-nya.

Tabel 5. Hasil perbandingan rata-rata *Jitter*

Node Background	DCCP (ms)	SCTP (ms)	E-PRsctp (ms)
10	0,04	8,12	8,13
20	4,21	8,15	7,15
30	9,33	11,14	10,96
40	19,55	11,14	9,00
50	22,41	12,99	12,99
60	23,88	6,55	14,09



Gambar 5. Grafik rata-rata *jitter* terhadap peningkatan jumlah node

Dari data Tabel 5. diatas, tampak *jitter* yang ditimbulkan oleh E-PRsctp ada penurunan signifikan pada node 40, dibandingkan dengan SCTP, namun ada peningkatan pada node 60, sedangkan pada SCTP pada node 60 ini rata-rata *jitter*-nya menurun. Sedangkan DCCP, semakin banyak jumlah node, *jitter* yang ditimbulkan semakin meningkat. Kesimpulan sementara dari hasil *jitter* ini, protokol E-PRsctp dapat mereduksi

tingkat *jitter* dari protokol SCTP standar pada node 20 sebesar 1ms, pada node 30 sebesar 0,18ms, dan pada node 40 sebesar 2,14ms.

Frame Drop

Pada analisa ini, menghitung rata-rata *frame I*, *frame P*, dan *frame B* yang hilang akibat adanya peningkatan kepadatan jaringan yang ditimbulkan oleh *background* jaringan, saat paket-paket video yang dikirim dari sumber ke tujuan melewati jalur *bottleneck* ini. Adapun hasil dapat dilihat pada Tabel 6. dibawah ini.

Dari hasil pengamatan, semakin besar adanya beban lalu lintas data pada jaringan, protokol E-PRsctp sedikit lebih baik dalam menyelamatkan paket *frame I* yang hilang daripada protokol DCCP pada penambahan jumlah node mulai dari 40, 50, dan 60.

Throughput

Analisa ini digunakan untuk melihat seberapa besar paket-paket data video yang dapat diangkut sampai ke tujuan. Data *throughput* diperoleh dari menangkap paket-paket yang sampai pada node tujuan yang diangkut oleh protokol masing-masing yang diujikan. Dari informasi data *trace*, kemudian diambil analisa rata-rata dari keseluruhan *throughput*. Adapun hasil dari keseluruhan rata-rata *throughput* pada masing-masing protokol seperti Tabel 7. dibawah ini.

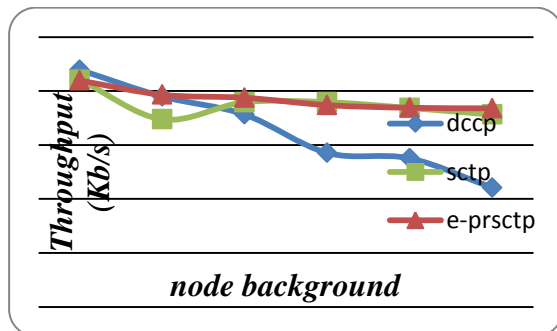
Throughput yang dihasilkan oleh E-PRsctp cenderung stabil pada peningkatan beban *background traffic* jaringan. Sedangkan pada protokol DCCP yang diberi peningkatan jumlah *background*, cenderung menurunkan nilai rata-rata *throughput*-nya. Kesimpulan sementara dari hasil ini adalah peningkatan *throughput* protokol E-PRsctp terhadap

Tabel 6. Hasil perbandingan rata-rata *frame drop*

Node Background	DCCP % loss			SCTP % loss			E-PRsctp % loss		
	I	P	B	I	P	B	I	P	B
10	0	0	0	0	0	0	0	0	0
20	1,9	1,8	2,0	16,8	13,6	16,2	9,9	7,7	9,0
30	15,7	13,4	16,1	63,5	61,5	64,1	55,8	49,4	51,1
40	61,7	55,9	58,0	63,5	61,6	64,1	60,5	59,3	60,9
50	70,5	70,6	72,4	70,6	74,6	76,2	69,1	70,5	72,4
60	69,6	69,2	71,0	77,3	80,3	81,3	69,1	70,2	72,3

Tabel 7. Hasil perbandingan rata-rata *Throughput*

Node Background	DCCP (Kb/s)	SCTP (Kb/s)	E-PRISCTP (Kb/s)
10	878,49	841,74	838,87
20	779,80	695,05	786,07
30	712,65	758,67	775,06
40	570,56	758,67	747,74
50	550,66	737,12	737,93
60	441,82	712,53	735,94



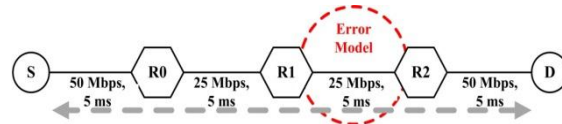
Gambar 6. Grafik rata-rata *throughput* terhadap peningkatan jumlah node

protokol SCTP standar pada node 20 sebesar 91,02 Kb/s , node 30 sebesar 16,39 Kb/s, node 50 sebesar 0,81 Kb/s, dan node 60 sebesar 23,41 Kb/s.

Skenario II

Pada uji coba simulasi ini, lalu lintas dari kedua *background* CBR tidak diaktifkan. Hanya ditambahkan loss rate pada link R1-R2.

Tujuannya adalah untuk mengetahui kehandalan masing-masing protokol dalam kualitas distribusi paket video terhadap peningkatan *loss rate* pada router.



Gambar 7. Topologi skenario model II

Pada skenario ini analisa yang dilakukan adalah PSNR dan MOS.

PSNR

Analisa ini digunakan untuk menghitung kemungkinan maksimum sinyal energi untuk energi noise. Adapun persamaan dari PSNR seperti dibawah ini,

$$PSNR(s,d) = 20\log \frac{V_{peak}}{MSE(s,d)} \text{ in dB}$$

Dimana,

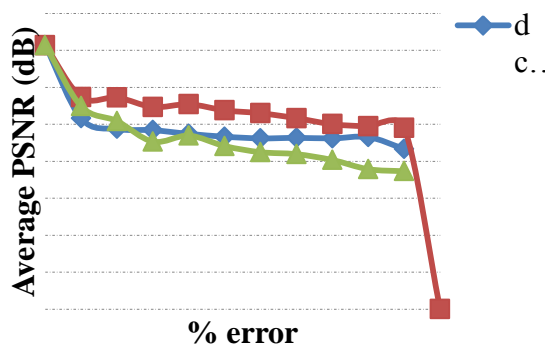
$$V_{peak} = 2^k - 1, \text{ kedalaman bit warna}$$

$$MSE(s,d) = \text{mean square error dari } s \text{ dan } d$$

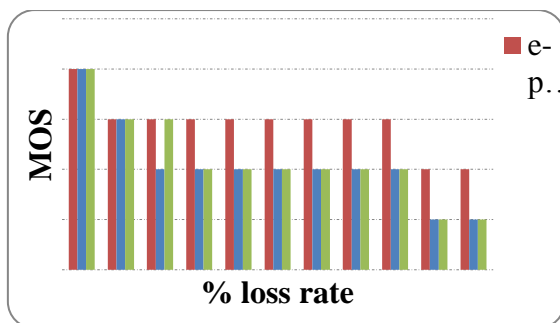
Pada Tabel 8. adalah hasil rata-rata PSNR yang didapat dari masing-masing protokol yang diujikan pada skenario model II. PSNR diperoleh dari hasil perbandingan video sumber sebelum simulasi dengan video hasil simulasi yang sudah diperbaiki (dengan *tool fixyuv_ra*). Dari Gambar 8. nilai rata-rata PSNR dari e-prisctp lebih tinggi dari protokol SCTP standar dan protokol DCCP.

Tabel 8. Pengaruh *loss rate* terhadap kualitas distribusi video

<i>error rate</i>	DCCP		E-PRISCTP		SCTP	
	PSNR(dB)	Fix Frame	PSNR(dB)	Fix Frame	PSNR(dB)	Fix Frame
0%	35,70	1200	35,68	1200	35,68	1200
1%	25,81	1198	28,69	1199	27,45	1199
2%	25,37	1197	28,64	1198	25,44	1199
3%	24,21	1197	27,35	1198	22,67	1199
4%	24,72	1195	27,72	1198	23,48	1198
5%	23,33	1195	26,90	1198	22,06	1198
6%	23,11	1193	26,51	1198	21,24	1195
7%	23,18	1195	25,84	1198	20,97	1198
8%	23,11	1059	25,02	1198	20,18	1196
9%	23,28	854	30,72	1198	18,94	1191
10%	21,68	634	24,53	1198	18,66	1196



Gambar 8. Grafik rata-rata PSNR



Gambar 9. Grafik MOS

MOS

Mean Opinion Score (MOS) dapat juga disebut sebagai pengukuran kualitas secara subyektif yang dapat diterapkan pada video (awalnya digunakan untuk kualitas audio pada telepon). Karena MOS mengasumsikan pada nomor tunggal antara 1 sampai 5, maka mudah untuk memahami baik proses maupun grafikny. Tabel 9. berikut adalah asumsi kualitas video berdasarkan hasil pengukuran PSNR.

Tabel 9. Konversi skala PSNR ke MOS

PSNR [dB]	MOS
> 37	5 (Excellent)
31 – 37	4 (Good)
25 – 31	3 (Fair)
20 – 25	2 (Poor)
< 20	1 (Bad)

Dari hasil analisa MOS, rata-rata hasil yang diperoleh E-PR SCTP dalam kualitas layanan video transport saat terdapat loss rate, mendapat kategori **sedang (fair)** dan turun menjadi **sangat kurang (poor)** pada loss rate 9% dan 10%. Sedangkan untuk DCCP, mendapat kategori sedang (*fair*) dalam kualitas layanan transport pada loss rate 1% dan 2%,

kemudian turun menjadi **sangat kurang (poor)** sampai pada loss rate 8%, kemudian menjadi **buruk (bad)** dari loss rate 9% dan 10%. Pada SCTP hampir sama seperti pada DCCP, hanya perbedaan terjadi antara loss rate 1% dan 2%.

SIMPULAN

Kesimpulan yang dapat diambil berdasarkan hasil rangkaian uji coba dan analisa penelitian yang dilakukan terhadap metode yang diusulkan adalah sebagai berikut:

1. Baik SCTP maupun E-PR SCTP pada *traffic background* yang rendah menunjukkan tingginya nilai rata-rata *delay* dibandingkan dengan DCCP, sedangkan untuk *traffic background* yang tinggi nilai rata-rata *delay* menurun, sedangkan pada protokol DCCP cenderung naik.
2. Untuk analisa *jitter*, rata-rata nilai *jitter* dari E-PR SCTP lebih kecil dari pada SCTP, sedangkan untuk DCCP cenderung meningkat setiap ada peningkatan beban *background traffic*.
3. Paket-paket *frame* yang hilang pada E-PR SCTP lebih sedikit dari SCTP standar, berarti penggunaan fitur *multi-streaming* dan *partial reliable* untuk *streaming* video MPEG-4 lebih baik dari SCTP standar.
4. Terdapat peningkatan *throughput* yang diperoleh dari E-PR SCTP dibandingkan dengan SCTP.
5. Kualitas layanan pengiriman video dengan menggunakan E-PR SCTP lebih baik dari pada protokol SCTP maupun DCCP pada uji coba kedua yang memakai *error model*.
6. Secara rata-rata protokol E-PR SCTP performanya lebih baik dari SCTP standar pada uji coba dengan *background traffic* khususnya pada node 10 sampai 40. Sedangkan dari uji coba dengan adanya gangguan pada lintasan jaringan, E-PR SCTP jauh lebih baik dari SCTP standar maupun dengan DCCP.

DAFTAR PUSTAKA

[1] Mack, S. (2002), “*Streaming Media Bible*”, Hungry Minds, Inc, New York hal: 30 – 32

[2] Forouzan, B.A. (2006), “*TCP/IP Protocol Suite*”, McGraw-Hill, New York

- [3] Natarajan, P., Iyengar, J., Amer, P., and Stewart, R. (2006) "SCTP: An Innovative Transport Layer Protocol For Web", Proc. WWW 2006, hal: 615–624.
- [4] Stewart, R., et al. (2007), "Stream Control Transmission Protocol", IETF RFC 4960.
- [5] Tsunoda, H., Kato, N., Jamalipour, A., Nemoto, Y. (2007), "Performance Evaluation of SCTP with Adaptive Multistreaming over LEO Satellite Networks", *Satellite and Space Communications*, IEEE Explore, hal: 150-154
- [6] Molteni, M., Villari, M. (2002), "Using SCTP with Partial Reliability for MPEG-4 Multimedia Streaming", Proc. European BSD Conf., hal: 2-3
- [7] Rajiullah, M., Brunstrom, A. (2011), "On the Effectiveness of PR-SCTP in Networks with Competing Traffic", *Computers and Communications (ISCC)*, IEEE Symposium
- [8] Wang, H., Jin, Y., Wang, W., Ma, J., Zhang, D. (2003), "The performance comparison of PRSCTP, TCP and UDP for MPEG-4 multimedia traffic in mobile network", in *Proc. Of International Conference on Communication Technology (ICCT)*, Beijing, China, hal: 403-406
- [9] Wang, L., Kawanishi, K., and Onozato, Y. (2008), "MPEG-4 Optimal Transmission over SCTP Multi-streaming in 802.11 Wireless Access Media", *Information and Telecommunication Technologies*, APSITT. 7th Asia-Pacific Symposium
- [10] Kim, K.H., Jeong, K.M., Kang, C.H., Seok, S.J., (2009), "A transmission control SCTP for real-time multimedia streaming", *Computer Networks*, Elsevier, hal: 1418–1425
- [11] Yuan, C.L. (2008), "DCCP: Transport Protocol with Congestion Control and Unreliability", *Internet Computing*, IEEE Computer Society
- [12] Zhou, R., Kyung, S.J. (2008), "Adaptive MPEG-4 Video Streaming Over IP Networks", *International Technical Conference on Circuits/Systems, Computers and Communication (ITC-CSCC)*
- [13] Chih-Heng Ke. (2004), "How to Evaluate MPEG Video Transmission Using the NS2 Simulator"
- [14] Lie, A., Klaue J. (2007), "Evalvid-RA: Trace Driven Simulation of Rate Adaptive MPEG-4 VBR video", *ACM/Springer Multimedia Systems Journal autumn*
- [15] Nosheen, S., Malik, S.A, Zikria, Y.B., Afzal, M.K. (2007), "Performance Evaluation of DCCP and SCTP for MPEG4 Video over Wireless Networks", *Multitopic Conference*, IEEE International
- [16] Talaat, M.A., Koutb, M.A., Sorour, H.S. (2009), "PSNR Evaluation of Media Traffic over TFRC", *International Journal of Computer Networks & Communication (IJCNC)*.
- [17] ITU-R Recommendation BT.500-11, (2002), "Methodology for the subjective assessment of the quality of television pictures", *International Telecommunication Union*, Geneva, Switzerland
- [18] CIF YUV video sequences, [online], Retrieved on Oct 26st at 16:55 (2011).
- [19] <http://media.xiph.org/video/derf/>