ISSN: 1979-2328

Analysis of the Effect of Congestion Control DCCP CCID 2 on TCP SACK

Analisis Pengaruh *Congestion Control* DCCP CCID 2 Pada TCP SACK

Bambang Sugiantoro [1], Izza Afkarina [2]

[1,2] Magister Informatika, UIN Sunan Kalijaga Yogyakarta, Indonesia

[1] bambang.sugiantoro@uin-suka.ac.id, [2*] izzaafkarinnaa@gmail.com
**: Penulis korenspondensi* (*corresponding author*)

Keywords: Congestion Control, DCCP CCID 2, NS-2, TCP SACK

Abstract

Purpose: *Observing the effect of DCCP CCID 2 congestion control on TCP traffic using the NS-2 simulator.*

Design/methodology/approach: *Using TCP-SACK with performance testing parameters observed are congestion window, throughput, and packet drop.*

Findings/result: *DCCP CCID 2 is more able to maximize bandwidth than TCP SACK because TCP SACK has a retransmit phase where if a packet is lost, the lost packet will be sent back which can cause delays. The buffer size also does not affect the throughput value of TCP SACK because TCP SACK and DCCP CCID 2 have the same congestion control mechanism and the congestion control algorithm CCID 2 is more advanced.*

Originality/value/state of the art: *This study observes the effect of DCCP CCID 2 congestion control on TCP traffic, where the TCP SACK variation used in this study has never been used before.*

Kata kunci: Congestion Control, DCCP CCID 2, NS-2, TCP SACK

Abstrak

Tujuan: Mengamati pengaruh *congestion control* DCCP CCID 2 terhadap trafik TCP menggunakan simulator NS-2.
Perancangan/metode/pendekatan: Variasi TCP yang digunakan adalah TCP-SACK dengan parameter pengujan kinerja yang diamati yaitu *congestion window*, rata-rata *throughput*, dan *packet drop.*
Hasil: DCCP CCID 2 lebih bisa memaksimalkan *Bandwidth* dibandingkan TCP SACK karena TCP SACK memiliki fase

ISSN: 1979-2328

retransmit dimana jika ada paket hilang maka paket yang hilang tersebut akan dikirim kembali yang dapat menyebabkan *delay*. Ukuran *buffer* juga tidak mempengaruhi nilai *throughput* TCP SACK karena TCP SACK dan DCCP CCID 2 memiliki mekanisme *congestion control* yang sama dan algoritma *congestion control* CCID 2 lebih *advance*.

Keaslian/ *state of the art*: Penelitian ini mengamati pengaruh *congestion control* DCCP CCID 2 terhadap trafik TCP, dimana variasi TCP SACK yang digunakan dalam penelitian ini belum pernah digunakan sebelumnya.

1. Pendahuluan

Jaringan komputer merupakan sekumpulan komputer yang saling terhubung [1]. Terdapat dua protokol yang berada pada *transport layer* dalam jaringan komputer untuk komunikasi data, yaitu *Transmission Control Protocol* (TCP) dan *User Datagram Protocol* (UDP) [2].

TCP merupakan protokol *connection-oriented* (TCP membuat koneksi terlebih dahulu ke komputer penerima sebelum melakukan pengiriman data). Kelebihan TCP diantaranya *reliability* (TCP memiliki beberapa mekanisme dalam pengiriman data, antara lain *checksum, duplicate data detection, retransmission, sequencing,* dan *timers*) dan *byte-stream* (TCP mengirim data dalam urutan-urutan *byte*) [2]. Namun TCP memiliki kekurangan yaitu ketika jaringan padat atau antrian paket berlebihan akan menyebabkan *packet loss*. Saat *packet loss* terjadi, TCP pengirim akan berhenti mengirim data hingga paket yang hilang berhasil di transmisikan ulang dan *throughput* menjadi tidak optimal [3]. *Throughput* yang rendah yang menyebabkan *traffic* mengalami *congestion* [3]. *Congestion* merupakan kemacetan dalam proses pengiriman data [4].

UDP merupakan protokol yang melakukan komunikasi secara sederhana. Karakteristik UDP berupa [2]: *Connectionless* artinya data dikirim tanpa proses negosiasi antar komputer yang bertukar informasi. *Unreliable* yang berarti data dikirim dalam bentuk datagram tanpa nomor urut. Jika selama transmisi ada pesan-pesan yang hilang, maka protokol aplikasi yang letaknya di atas UDP harus memulihkan pesan tersebut [5]. UDP merupakan protokol yang ditujukan untuk kecepatan pengiriman data. Akibat dari kecepatan pengiriman data yang tidak dapat dikendalikan, protokol UDP akan menggunakan seluruh *bandwidth* yang ada di dalam jaringan [6].

Pada aplikasi media *streaming* dan *game* berbasis web, internet umumnya menggunakan TCP atau UDP untuk mengirim dan menerima data. TCP tidak cocok pada kondisi ini karena TCP menjamin pengiriman dan keandalan data. Artinya ketika suatu unit data hilang, hal itu berpengaruh pada pemrosesan data baru dan yang sudah ada dalam antrian yang menyebabkan *delay* [7]. Sedangkan UDP dapat mengatasi *delay*, tetapi protokol UDP tidak menerapkan *congestion control* yang menjaga kelancaran transmisi jika terjadi kemacetan [5].

Oleh karena itu, *Internet Engineering Task Force* (IETF) memperkenalkan protokol baru yang menerapkan mekanisme *congestion control* yang *advance*, yaitu *Datagram Congestion Control*

ISSN: 1979-2328

Protocol (DCCP). Protokol ini sangat cocok untuk *real-time application* [7]. Protokol DCCP memiliki algoritma *congestion control* yang disebut CCID (*Congestion Control* ID) agar lebih TCP-*friendly* karena TCP masih mendominasi internet saat ini [8].

Berdasarkan uraian diatas, penulis tertarik melakukan penelitian yang berfokus mengamati pengaruh *congestion control* DCCP CCID 2 terhadap trafik TCP menggunakan simulator NS2. Penelitian ini merujuk penelitian sebelumnya tentang analisis pengaruh *congestion control* DCCP CCID 2 pada TCP New-Reno dan TCP Vegas [9] tetapi menggunakan variasi TCP lain yaitu TCP-SACK dengan parameter pengujian kinerja jaringan yang sama. Parameter unjuk kerja yang diamati yaitu *congestion window*, rata-rata *throughput*, dan *packet drop.*

Latar belakang menggunakan variasi TCP-SACK dalam penelitian ini adalah penelitian sebelumnya [10] yang membahas variasi beberapa macam TCP seperti: TCP-Reno, TCP-Tahoe, TCP-Vegas dan TCP-SACK mendapatkan hasil penelitian bahwa perbandingan beberapa variasi TCP tersebut dalam kondisi jaringan seperti yang sudah disimulasikan tidak mempunyai perbedaan hasil *throughput* yang signifikan, tetapi varian TCP-SACK dan TCP-Vegas memiliki *average throughput* yang relatif besar dan TCP-Reno berada pada *average throughput* paling kecil sedangkan TCP-Tahoe berada diantara keduanya.

TCP *Selective Acknowledgments* (SACK) juga merupakan TCP yang mengatasi kekurangan dari TCP Reno dan New-Reno [11]. TCP SACK memecahkan masalah yang dihadapi oleh TCP Reno dan New-Reno seperti mendeteksi beberapa kehilangan paket di *window* yang sama. TCP SACK juga memiliki fitur *slow start*, *fast retransmit*, dan *rough grained timeout* dari varian TCP lama. Fitur utama TCP SACK adalah, ia mengakui paket secara selektif daripada kumulatif [12].

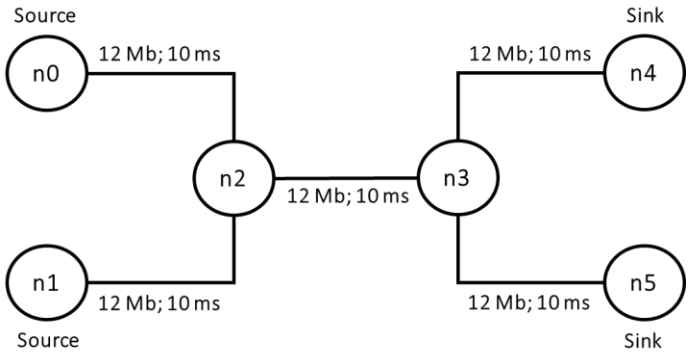
2. Metode Penelitian

Langkah-langkah yang dilakukan dalam penelitian ini dalam membangun simulasi agar dapat berjalan dan mendapatkan hasil sesuai dengan yang diharapkan adalah sebagai berikut. Tahap pertama adalah menentukan topologi yang digunakan. Hal yang menjadi pertimbangan dalam menentukan topologi adalah model jaringan *wired* atau *wireless*, jumlah *node*, posisi *node* sumber dan *node* tujuan [5]. Tahap kedua adalah menentukan parameter simulasi. Parameter simulasi bersifat konstan dan digunakan sesuai skenario pengujian yang ditentukan. Tahap ketiga adalah menentukan skenario pengujian. Skenario pengujian yang digunakan menggunakan tipe antrian *Drop Tail*. Tahap berikutnya adalah menentukan parameter kinerja. Parameter yang digunakan dalam penelitian ini adalah *throughput*, *congestion window* dan *packet drop*. Tahap selanjutnya yaitu, tahap *coding* dan menjalankan simulasi. Simulasi dibangun berdasarkan topologi, parameter, dan skenario pengujian yang telah ditentukan sebelumnya. Tahap akhir yaitu menganalisa hasil simulasi dan mendapatkan kesimpulan berdasarkan simulasi yang telah dijalankan.

2.1. Topologi Simulasi

Topologi simulasi yang digunakan dalam simulasi ini adalah topologi *dumbbell* dengan 2 *node* pengirim 2 *node router* dan 2 *node* penerima. Secara umum model topologi ini digunakan untuk menganalisis efek jalur *bottleneck* yang dilalui oleh banyak *node* pengirim. Desain topologi yang digunakan sesuai **Gambar 1** berikut.

ISSN: 1979-2328



Gambar 1. Desain Topologi Simulasi *Dumbbell*

Topologi ini menggunakan 6 *node* yang disimbolkan dengan n0, n1, n2, n3, n4 dan n5. *Node* n0 dan n1 bertindak sebagai *node* pengirim, *node* n2 dan n3 bertindak sebagai *node router*, *node* n4 dan n5 bertindak sebagai *node* penerima. *Node* n0 dipasangkan dengan *node* n4 dan *node* n1 dipasangkan dengan *node* n5.

2.2. Parameter Simulasi

Parameter simulasi yang digunakan pada penelitian ini sesuai dengan skenario pengujian ditunjukkan pada **Tabel 1.**

Tabel 1. Parameter Simulasi Yang Digunakan Sesuai Skenario Pengujian

Parameter Simulasi	**Nilai**
Link Source n0, n1 – Router n2	Bandwidth: 12 Mbps Delay: 10 ms
Link Router n2 – Link Router n3	Bandwidth: 12 Mbps Delay: 10 ms
Link Router n3 – Sink n4, n5	Bandwidth: 12 Mbps Delay: 10 ms
Protocol Transport	TCP SACK, DCCP CCID 2
Model Antrian	Drop Tail
Ukuran Buffer pada Router	5, 10, 15
Trafik	TCP --- FTP Size: 1000 Byte DCCP --- CBR Size: 1000 Byte Rate 13 Mbps

Aplikasi yang digunakan TCP adalah FTP yang mewakili aplikasi yang berbasis nrt-VBR (*non real-time Variable Bit Rate*) yang bersifat *bursty* dan tidak sensitif terhadap *delay*. Sedangkan DCCP menggunakan trafik CBR (*Constant Bit Rate*) yang mewakili trafik *real-time* dengan *bit-rate* yang tetap.

Penentuan *buffer* menggunakan perhitungan *Bandwidth Delay Product* pada *link router* n2 dan *link router* n3 dengan perhitungan sebagai berikut:

Bandwidth Delay Product:
12 Mb

10 ms
12 Mb x 0,01 s = 0,12 Mbps

Buffer:
1500 *Byte* = 0,012 Mb
0,012 Mb x 5 = 0,06 Mb (<*Bandwidth Delay Product*)
0,012 Mb x 10 = 0,06 Mb (<*Bandwidth Delay Product*)
0,012 Mb x 15 = 0,06 Mb (<*Bandwidth Delay Product*)

2.3. Skenario Pengujian

Skenario pengujian TCP SACK vs DCCP CCID 2 digunakan untuk membandingkan kinerja dari TCP SACK ketika menggunakan *buffer* (5, 10, 15) dengan jenis antrian *Drop Tail* ketika dihadapkan dengan DCCP CCID 2. Sebuah trafik FTP yang menggunakan TCP SACK akan dijalankan dari *node* n0 sebagai pengirim (TCP *Source*) menuju *node* n4 sebagai penerima (TCP *Sink*). Sebuah trafik CBR yang menggunakan protokol CCID 2 dijalankan dari *node* n1 sebagai pengirim (DCCP *Source*) menuju *node* n5 sebagai penerima (DCCP *Sink*). Kedua trafik dari n0 dan n1 dialirkan menuju *node* n2 kemudian diteruskan ke *node* n3, selanjutnya trafik dari *node* n3 akan diteruskan ke *node* n4 dan n5. Pada *node* n2 dan n3 menggunakan antrian *Drop Tail* dengan *buffer* 5, 10, 15.

2.4. Parameter Kinerja Jaringan

Parameter kinerja jaringan yang digunakan pada penelitian ini adalah sebagai berikut.

2.4.1. Throughput

Throughput merupakan jumlah data per satuan waktu yang dikirim di dalam sebuah jaringan, dari suatu titik jaringan ke titik jaringan yang lain. Pengukuran *throughput* bertujuan untuk mengetahui kehandalan jaringan dalam meneruskan paket yang datang hingga sampai di tujuan [13]. Rumus menghitung *throughput* adalah sebagai berikut [14].

$$Throughput = \frac{ukuran\ data\ diterima}{waktu\ pengiriman\ data} \quad (1)$$

2.4.2. *Congestion Window* (CWND)

Congestion window merupakan satuan dari jumlah paket yang dapat dikirim oleh TCP. Nilai *congestion window* akan di set menjadi 1 setiap memulai pengiriman data dalam suatu koneksi, sehingga *sender* hanya bisa mengirimkan 1 *segment* dalam sekali kirim [15].

2.4.3. *Packet Drop*

Packet drop merupakan paket yang dibuang karena gagal ditransmisikan untuk mencapai destinasi disebabkan *buffer overflow* (*congestion*) dan *bit error* (pada jaringan *wireless*). Semakin banyak paket yang dibuang menunjukkan keadaan jaringan yang memiliki masalah [14].

3. Hasil dan Pembahasan

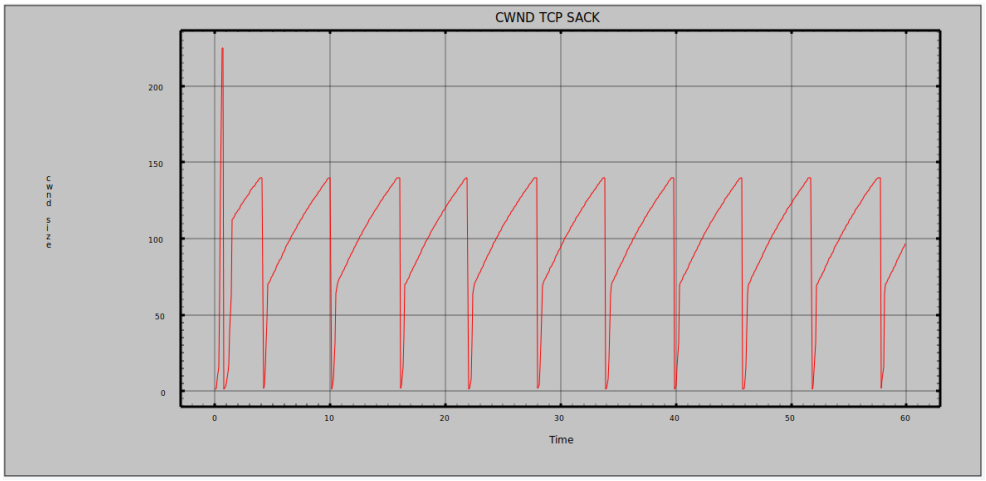
Simulator yang digunakan pada simulasi dalam penelitian ini menggunakan Network Simulator NS 2. Modul protokol TCP, maupun DCCP sudah tersedia di dalam NS 2. Setelah simulasi dijalankan didapatkan *output* berupa file *trace* dengan ekstensi ".tr", file *network animator* dengan ekstensi ".nam" dan file *xgraph* dengan ekstensi ".xg". Penelitian ini hanya

ISSN: 1979-2328

menggunakan awk dan Microsoft Excel untuk mengolah nilai CWND, rata-rata *throughput* dan *packet drop* hasil simulasi semua skenario yang telah dibuat sebelumnya.

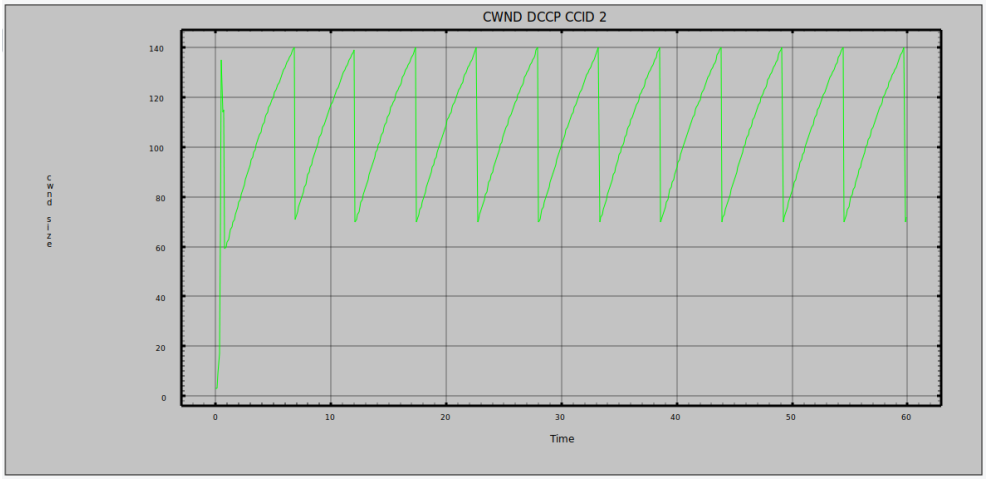
3.1. Pengambilan Data

Pengambilan data dilakukan sesuai dengan skenario simulasi yang telah dirancang sebelumnya seperti pada **Gambar 2** dan **Gambar 3** berikut.



Gambar 2 Grafik CWND TCP SACK dalam 1 kali Pengujian per Skenario

Gambar diatas merupakan grafik dari CWND TCP SACK dalam satu skenario pengujian yang berjalan sendiri sebelum diganggu dengan trafik DCCP. Pada skenario ini terlihat TCP SACK berjalan dan memaksimalkan penggunaan *resource* yang ada.



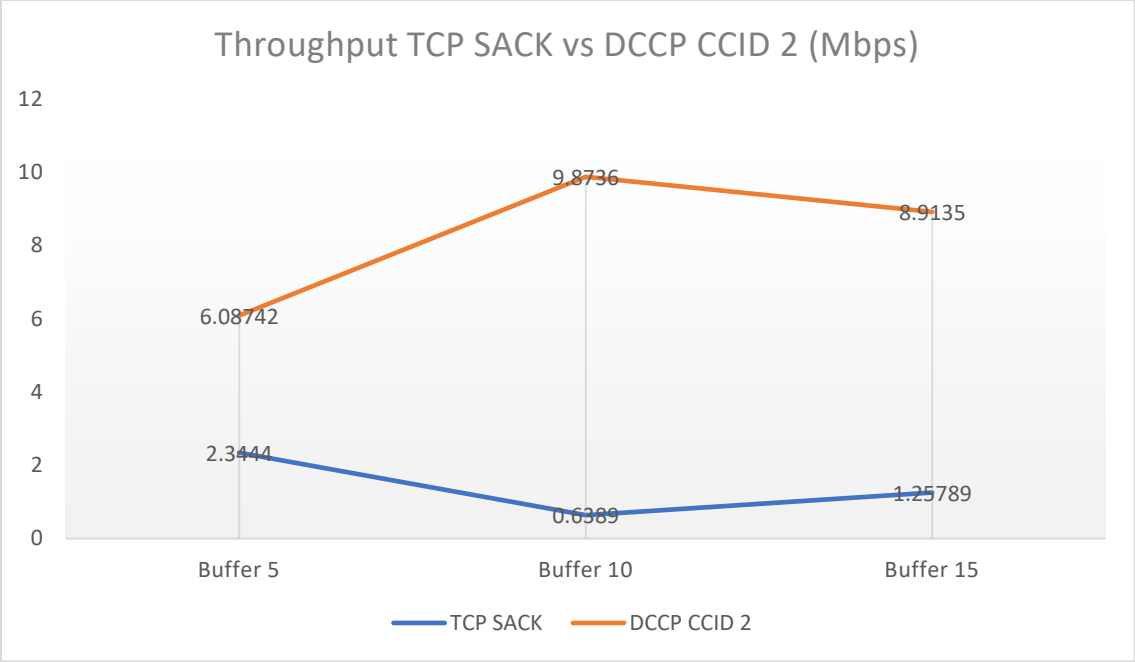
Gambar 3 Grafik CWND DCCP CCID 2 dalam 1 kali Pengujian per Skenario

Gambar diatas merupakan grafik dari CWND DCCP CCID 2 dalam satu skenario pengujian yang berjalan tanpa adanya gangguan. Pada skenario ini terlihat DCCP CCID 2 berjalan dan memaksimalkan penggunaan *resource* yang ada.

ISSN: 1979-2328

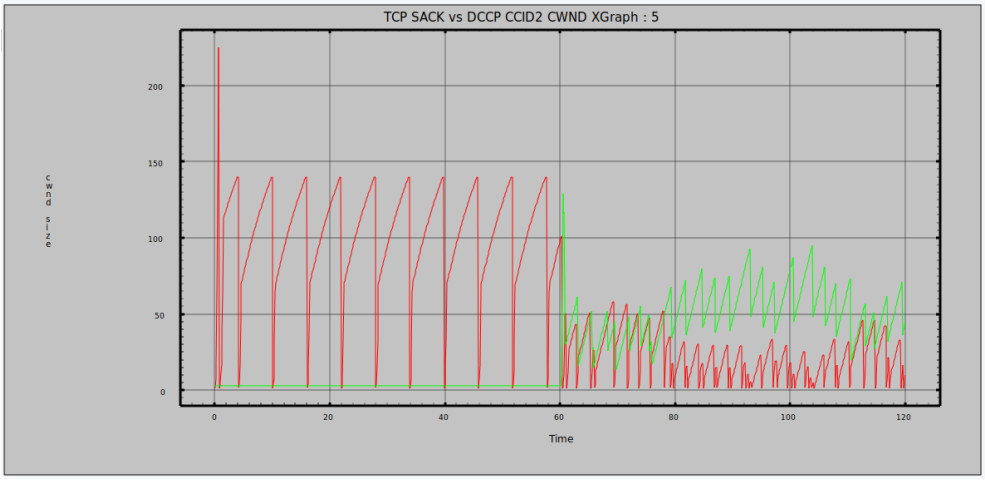
3.2. Analisis Data Hasil Simulasi

Simulasi pada penelitian ini menggunakan model antrian *Drop Tail*. Pada antrian *Drop Tail*, ketika antrian penuh, paket yang baru datang akan dibuang sampai antrian memiliki ruang untuk menampung paket yang akan datang.



Gambar 6 Grafik *Throughput* pada Skenario TCP SACK vs DCCP CCID 2 *Buffer* 5,10,15

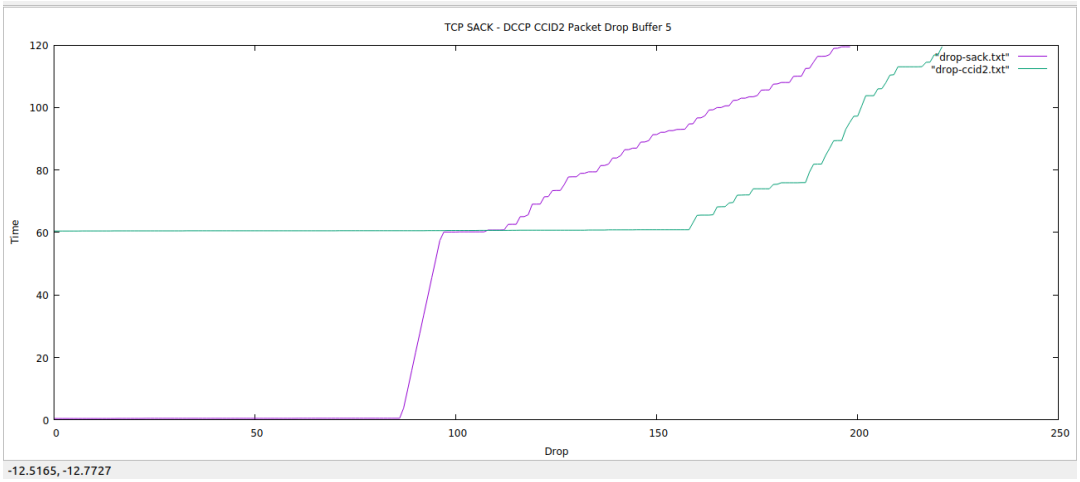
Grafik *throughput* diatas menunjukkan bahwa DCCP CCID 2 lebih unggul daripada TCP SACK karena TCP dan DCCP memiliki *congestion control* sehingga ketika *buffer* antrian penuh dan terjadi paket drop, keduanya mampu mendeteksi adanya *congestion* dan menurunkan cwnd. Akan tetapi, *Recovery* congestion DCCP lebih cepat dibandingkan TCP sehingga DCCP mengirim lebih banyak paket dari pada TCP. DCCP CCID 2 lebih memakan *bandwidth* yang ada menjadikan nilai *throughput* TCP SACK lebih sedikit.



Gambar 7 Grafik CWND pada Skenario TCP SACK vs DCCP CCID 2 *Buffer* 5

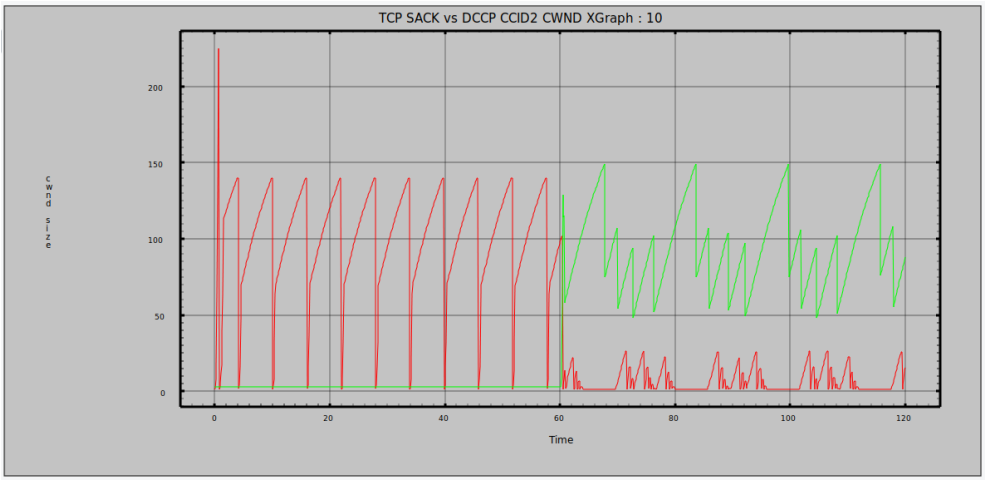
ISSN: 1979-2328

Pada skenario TCP vs DCCP diatas terlihat bahwa cwnd DCCP lebih besar di banding cwnd TCP. DCCP mengirim paket lebih banyak dari TCP sehingga *buffer* antrian lebih banyak digunakan DCCP.



Gambar 8 Grafik *Packet Drop* pada Skenario TCP SACK vs DCCP CCID 2 *Buffer* 5

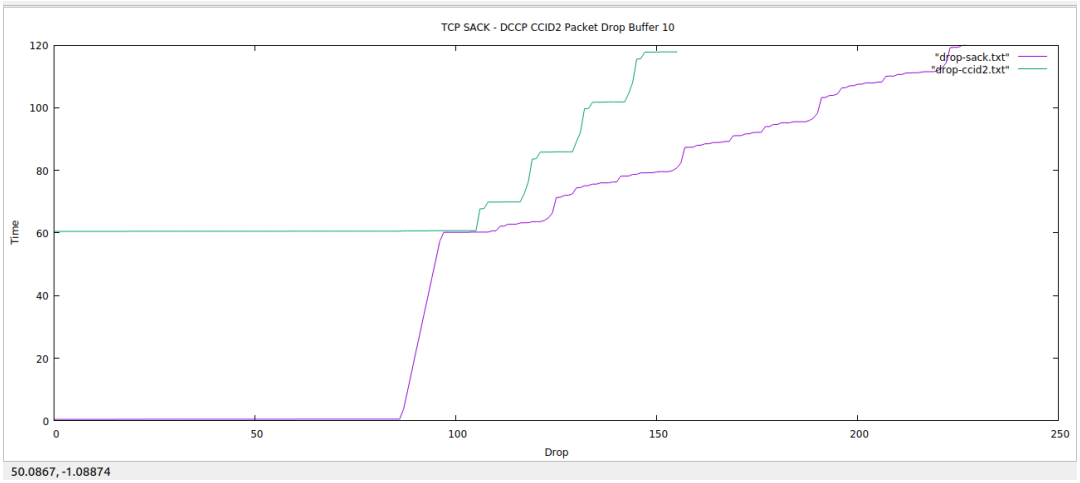
Pada *buffer* 5, TCP SACK yang awalnya berjalan sendiri mengalami *packet drop* pada detik ke 60 karena adanya trafik pengganggu yaitu DCCP CCID 2 dan langsung memenuhi *bandwidth* yang ada, sehingga TCP SACK hanya memperoleh sebagian *bandwidth*.



Gambar 9 Grafik CWND pada Skenario TCP SACK vs DCCP CCID 2 *Buffer* 10

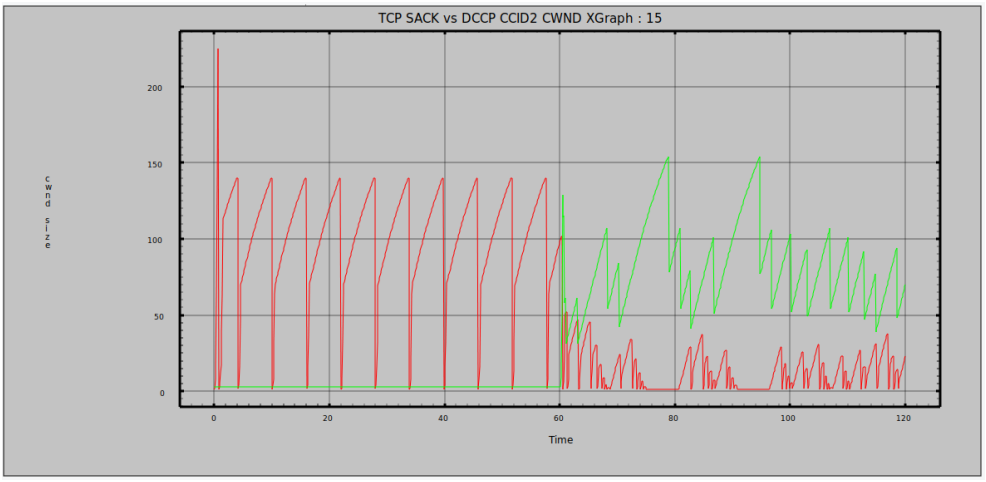
Pada skenario TCP vs DCCP di *buffer* 10 diatas terlihat bahwa cwnd DCCP tetap lebih besar di banding cwnd TCP. DCCP mengirim paket lebih banyak dari TCP sehingga *buffer* antrian lebih banyak digunakan DCCP.

ISSN: 1979-2328



Gambar 10 Grafik *Packet Drop* pada Skenario TCP SACK vs DCCP CCID 2 *Buffer* 10

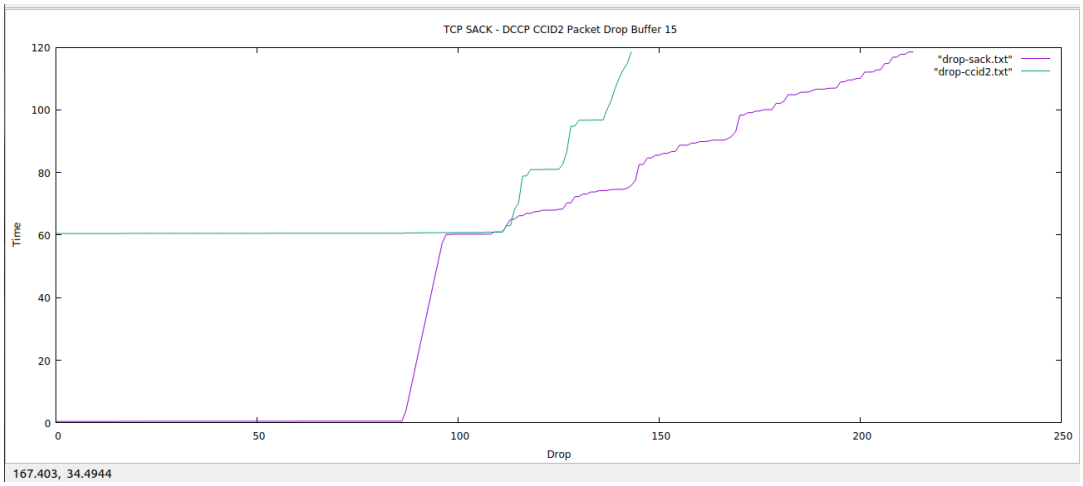
Pada *buffer* 10, DCCP CCID 2 tetap mendominasi *bandwidth* yang ada. Sehingga pada detik ke 60 TCP SACK langsung mengalami *drop* dan hanya mendapat sebagian *bandwidth*.



Gambar 11 Grafik CWND pada Skenario TCP SACK vs DCCP CCID 2 *Buffer* 15

Pada skenario TCP vs DCCP di *buffer* 15 diatas terlihat bahwa cwnd DCCP juga lebih besar di banding cwnd TCP. DCCP mengirim paket lebih banyak dari TCP sehingga *buffer* antrian lebih banyak digunakan DCCP.

ISSN: 1979-2328



Gambar 12 Grafik *Packet Drop* pada Skenario TCP SACK vs DCCP CCID 2 Buffer 15

Pada *buffer* 15 TCP SACK tetap tidak bisa bersaing dengan DCCP CCID 2 dikarenakan TCP SACK dan DCCP CCID 2 memiliki mekanisme *congestion control* yang sama sedangkan algoritma *congestion control* DCCP CCID 2 lebih *advance*.

4. Kesimpulan dan Saran

Kesimpulan pada skenario ini, DCCP CCID 2 lebih bisa memaksimalkan *bandwidth* yang ada daripada TCP SACK. DCCP CCID 2 tidak memperdulikan *paket drop* sementara TCP SACK memiliki fase *retransmit* dimana jika ada paket hilang maka paket yang hilang tersebut akan dikirim kembali yang dapat menyebabkan *delay*. Ukuran *buffer* juga tidak mempengaruhi nilai *throughput* TCP SACK karena TCP SACK dan DCCP CCID 2 memiliki mekanisme *congestion control* yang sama dan algoritma *congestion control* CCID 2 lebih *advance*.

Saran untuk pengembangan penelitian ini dimasa yang akan datang adalah mengganti variasi TCP atau DCCP lain yang memiliki algoritma *congetion control* berbeda dengan CCID 2 sebagai trafik pengganggu atau mengganti jenis antrian yang digunakan.

1. Daftar Pustaka

[1] A. P. Sujana, "Perangkat Pendukung Forensik Lalu Lintas Jaringan," *J. Tek. Komput. Unikom – Komputika*, vol. 3, no. 1, hal. 31–37, 2014.

[2] P. V. B. Romony, L. Sitanayah, dan J. B. Sanger, "Perbandingan Quality of Service Protokol Komunikasi Data Pada Sistem Deteksi Asap Rokok Berbasis Internet of Things," *J. Ilm. Realt.*, vol. 16, no. 1, hal. 19–24, 2020, doi: 10.52159/realtech.v16i1.129.

[3] K. K. Hartono, A. G. Putrada, dan M. Abdurohman, "Simulasi Perbandingan Performa Multipath TCP dan TCP Pada Jaringan Wired," in *e-Proceeding of Engineering*, 2019, vol. 6, no. 2, hal. 9514–9527.

[4] A. Firnanda, T. Y. Arif, dan Syahrial, "Analisis TCP Cubic dan Simulasi untuk Menentukan Parameter Congestion Window dan Throughput Optimal pada Jaringan Nirkabel Ad Hoc," *J. Rekayasa Elektr.*, vol. 13, no. 2, hal. 65–69, 2017, doi:

ISSN: 1979-2328

10.17529/jre.v13i2.4874.

[5] A. D. Putra, "Analisis Pengaruh Congestion Control DCCP CCID 3 Terhadap TCP Westwood," Universitas Sanata Dharma, 2019.

[6] Y. Mardiana dan J. Sahputra, "Analisa Performansi Protokol TCP, UDP dan SCTP Pada Lalu Lintas Multimedia," *J. Media Infotama*, vol. 13, no. 2, hal. 73–84, 2017, doi: 10.37676/jmi.v13i2.455.

[7] A. Qureshi, A. Qamar, I. A. Halepoto, S. Ashfaque, dan M. Lohana, "Analysis of Congestion Control Techniques for Time-Critical Applications," *QUEST Res. J.*, vol. 17, no. 2, hal. 43–49, 2019.

[8] R. B. I. S. H. F. M. Hale, "ANALISIS PENGARUH CONGESTION CONTROL DCCP CCID2 TERHADAP TCP TAHOE," Universitas Sanata Dharma, 2016.

[9] A. Kriswandaru, "Analisis Pengaruh Congestion Control DCCP CCID 2 Pada TCP Newreno Dan TCP Vegas Di Jaringan Kabel," Sanata Dharma University, 2018.

[10] M. Taruk dan A. Ashari, "Analisis Throughput Varian TCP Pada Model Jaringan WiMAX," *IJCCS (Indonesian J. Comput. Cybern. Syst.*, vol. 10, no. 2, hal. 115–124, 2016, doi: 10.22146/ijccs.15529.

[11] V. Singla, A. Kumar, dan R. Singla, "Performance Evaluation of TCP variants over different MANET routing protocols," *J. Emerg. Technol. Innov. Res.*, vol. 6, no. 1, hal. 223–226, 2019.

[12] H. Kaur dan G. Singh, "Measuring Performance of Variants of TCP Congestion Control Protocols," *Indian J. Comput. Sci. Eng.*, vol. 8, no. 3, hal. 285–296, 2017.

[13] I. Nurhaida dan I. Ichsan, "Congestion Control Pada Jaringan Komputer Berbasis Multi Protocol Label Switching (MPLS)," *Simetris J. Tek. Mesin, Elektro dan Ilmu Komput.*, vol. 11, no. 1, hal. 77–88, 2020, doi: 10.24176/simet.v11i1.3671.

[14] K. R. A. Manibuy, "Analisis Perbandingan Untuk Kerja TCP Tahoe Dan TCP Newreno Pada Jaringan Wired Dan Wireless," Universitas Sanata Dharma, 2017.

[15] F. P. Kristianto, "Analisis Unjuk Kerja TCP Sack Pada Jaringan Wired," Universitas Sanata Dharma, 2017.