JISKA (Jurnal Informatika Sunan Kalijaga)
Vol. 7, No. 2, MEI, 2022, Pp. 75 – 82
ISSN: 2527 – 5836 (print) | 2528 – 0074 (online)

Analisis Unjuk Kerja TCP Sack Menggunakan Antrian *Random Early Detection*

Rinny Asasunnaja [(1)*], Bambang Sugiantoro [(2)]
Magister Informatika, Fakultas Sains dan Teknologi, UIN Sunan Kalijaga, Yogyakarta
e-mail : rinnyasasun14@gmail.com, bambang.sugiantoro@uin-suka.ac.id.
* Penulis korespondensi.
Artikel ini diajukan 15 Juli 2021, direvisi 20 September 2021, diterima 28 September 2021, dan dipublikasikan 25 Mei 2022.

Abstract

As time goes by, the current technology is also developing very rapidly, the rapid development of technology, especially on the internet network is also developing so that it is expected to continue to provide the best quality to meet the growing and fast information needs. An internet network is highly demanded of its best quality, one of which is in the spotlight to be developed, namely TCP Transmission Control Protocol. The efficiency and smoothness of the data transmission process is the most important thing in a communication network. TCP is one aspect that supports today's fast internet, with TCP we can exchange information accurately, quickly, and well. In this study, researchers are interested in analyzing the performance of TCP Sack using the Random Early Detection queue. By doing this research is expected to know the performance of TCP Sack by looking at the parameter values of packet delivery ratio, throughput, end to end delay, and packet drop. Based on the results of the analysis, it can be concluded that end-to-end delay parameter testing on the TCP Sack still has poor performance, while for testing the packet delivery ratio, throughput, and packet drop TCP Sack already has good performance.

Keywords: TCP, TCP Sack, Random Early Detection, Packet Delivery Ratio, Throughput, Packet Drop

Abstrak

Seiring berjalannya waktu maka teknologi saat ini juga berkembang dengan sangat pesat, perkembangan teknologi yang pesat khususnya pada jaringan internet juga ikut berkembang sehingga diharapkan dapat terus memberikan kualitas terbaik untuk memenuhi kebutuhan informasi yang semakin besar dan juga cepat. Suat u jaringan internet sangat di tuntut kualitas terbaiknya, salah satu yang menjadi sorotan untuk dikembangkan yaitu TCP (*Transmission Control Protocol*). Efisiensi dan kelancaran proses pengiriman data adalah hal terpenting dalam jaringan komunikasi. TCP salah satu aspek yang mendukung adanya internet cepat saat ini, dengan adanya TCP maka kita dapat bertukar informasi secara tepat, cepat dan baik. Dalam penelitian ini peneliti tertarik untuk menganalisis kinerja TCP Sack menggunakan antrian *Random Early Detection*. Dengan melakukan penelitian ini diharapkan agar dapat mengetahui kinerja dari TCP Sack dengan melihat nilai dari parameter *packet delivery ratio, throughput*, *end to end delay* dan *packet drop*. Berdasarkan hasil analisis maka dapat disimpulkan untuk pengujian parameter *end to end delay* pada TCP Sack masih mempunyai kinerja yang kurang baik, sedangkan untuk pengujian terhadap parameter *packet delivery ratio, throughput* dan *packet drop* TCP Sack telah memiliki kinerja yang baik.

Kata Kunci: TCP, TCP Sack, *Random Early Detection*, *Packet Delivery Ratio*, *Throughput*, *Packet Drop*

1. PENDAHULUAN

Seiring berjalannya waktu maka teknologi saat ini juga berkembang dengan sangat pesat, perkembangan teknologi yang pesat khususnya pada jaringan internet juga ikut berkembang sehingga diharapkan dapat terus memberikan kualitas terbaik untuk memenuhi kebutuhan informasi yang semakin besar dan juga cepat. Suatu jaringan internet sangat dituntut kualitas terbaiknya, salah satu yang menjadi sorotan untuk dikembangkan yaitu TCP (*Transmission Control Protocol*) (Cesarius Agni Christian Kurniawan, 2017). Efisiensi dan kelancaran dalam

proses pengiriman data adalah hal terpenting dalam jaringan komunikasi. TCP adalah salah satu aspek yang mendukung adanya internet cepat saat ini, untuk mencapai tujuan internet cepat tersebut maka dibutuhkan kinerja dari protokol TCP. TCP mempunyai peran penting dalam jaringan komunikasi, dengan adanya TCP maka kita dapat bertukar informasi secara tepat, cepat dan baik (Antonius Cahyo Gumilang, 2017).

Transmission Control Protocol (TCP) merupakan suatu protokol yang berada pada *layer* ke-4 *Open System Interconnection* (OSI) dan TCP adalah salah satu protokol utama dalam protokol di internet yang menyediakan pengiriman paket yang dapat diandalkan, sehingga mampu membuat aliran data yang diterima oleh TCP *receiver* tidak rusak (M. Ryandy Ghonim Asgar, 2018). Saat ini pengiriman data melalui jaringan kabel masih mengalami adanya kemacetan sehingga data yang sudah dikirim perlu dikirim ulang. TCP adalah salah satu protocol transport yang sering digunakan dan TCP yang akan bertanggung jawab atas paket-paket yang dikirim. Jika ada paket yang hilang atau di drop maka TCP lah yang akan bertanggung jawab untuk mengirimi ulang paket-paket yang hilang tersebut. Pada protocol transport TCP juga akan mengalami penurunan kinerja ketika banyaknya paket yang hilang (Fransiskus Pando Kristianto, 2017).

Berdasarkan penelitian sebelumnya yang dilakukan oleh Guntur Wahyu Pamungkas, Widhi Yahya dan Heru Nurwasinto yang berjudul "*Analisis Perbandingan Kinerja TCP Vegas dan TCP New Reno Menggunakan Antrian Random Early Detection dan Droptail*", peneliti membandingkan kinerja dari TCP Vegas dan TCP New Reno dengan menggunakan antrian *Random Early Detection* dan menggunakan antrian *Droptail*. Hasil yang didapatkan dari penelitian tersebut adalah, dengan menggunakan antrian *Random Early Detection* maka hasilnya menunjukkan kinerja TCP Vegas lebih baik dari TCP NewReno terutama pada *packet delivery ratio, delay* dan juga pada *packet drop* (Pamungkas et al., 2018).

TCP bergantung pada batas waktu untuk melakukan pemulihan awalnya, TCP sendiri sudah melalui banyak revisi yaitu TCP Tahoe, TCP Reno, TCP New Reno dan TCP Sack. Semua varian TCP tersebut mecoba agar lebih cepat dalam melakukan pemulihan data terhadap *packet loss.* Diantara semua varian TCP tersebut TCP Sack diterima sebagai skema yang paling stabil, adil dan efesien (Kothari & Dasgupta, 2006). Oleh karena latar belakang permasalahan tersebut maka peneliti tertarik untuk meneliti kinerja TCP Sack menggunakan antrian *Random Early Detection*. Peneliti akan melakukan simulasi untuk mengetahui kinerja dari TCP Sack menggunakan Network Simulator 2.35 (NS-2). Pengujian dalam penelitian ini akan dilakukan sebanyak 4 kali dengan skema penambahan nilai *min tresh*. Dengan melakukan penelitian ini diharapkan dapat mengetahui kinerja dari TCP Sack dengan melihat nilai dari parameter *packet delivery ratio, throughput*, *end to end delay* dan *packet drop*.

2. METODE PENELITIAN

2.1 TCP (*Transmission Control Protocol*)

TCP merupakan salah satu protokol yang terletak pada *layer transport* dalam OSI (*Organization for Standardization*) *layer*. OSI adalah sebuah model arsitektural jaringan yang dikembangkan oleh badan *International Organization for Standardization* di Eropa pada tahun 1977, model ini juga disebut dengan "model tujuh lapis OSI" (OSI *seven layer*), Model OSI dibuat untuk mengatasi berbagai kendala *internetworking* akibat perbedaan arsitektur dan protokol jaringan (Baiq Alung Septiya Nurmala, 2020). TCP (*Transmission Contr0x0ol Protocol*) adalah standar komunikasi data yang digunakan oleh komunitas internet dalam proses pertukaran data dari satu komputer ke komputer lainnya dalam jaringan internet. Contoh sederhana dalam kehidupan, kita membuat sebuah dokumen Word pada aplikasi ini tidak ada hubungan dengan OSI akan tetapi bila program tersebut dikaitkan dengan jaringan misalnya dokumen Word akan dikirimkan ke *email* maka OSI model akan berpengaruh disini.

TCP adalah sebuah standar jaringan terbuka yang bersifat independen terhadap mekanisme transport jaringan fisik yang digunakan, sehingga TCP dapat digunakan di mana saja. Skema

pengalamatan yang digunakan protokol ini sangat sederhana yang sering disebut sebagai alamat IP (IP *address*) yang akan mengizinkan hingga beberapa ratus juta komputer untuk dapat saling berhubungan satu sama lainnya di internet (Syaifuddin et al., 2016).

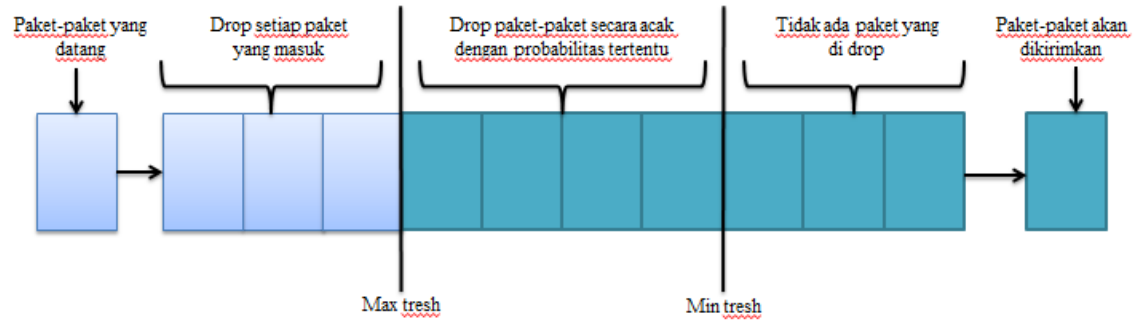
TCP adalah protokol yang terletak pada lapisan ke 4 dari OSI *layer* yang akan bertanggung jawab menyedikan layanan komunikasi *end-to-end* atau *host-to-host* antar lapisan *application*. TCP pada umumnya digunakan ketika protokol lapisan *application* membutuhkan layanan transfer yang bersifat handal. Contoh dari protokol yang menggunakan TCP adalah *HTTP* dan *FTP*. *Header* TCP berisi urut (TCP *sequence number*) dari data yang ditransmisikan dan sebuah *acknowledgment* dari data yang masuk. Data yang dikirimkan ke sebuah protokol TCP akan diurutkan menggunakan nomor urut paket dan akan mengharapkan paket *positive acknowledgment* dari penerima. Jika tidak ada paket *acknowledgment* dari penerima maka segmen TCP akan ditransmisikan ulang (Marlina Nathalia, 2016).

2.2 TCP Sack

Di dalam TCP terdapat salah satu pengirim yang bernama SACK (*Selective Acknowledgment*), jadi SACK adalah strategi yang mengoreksi dalam menghadapi kehilangan beberapa segmen. Dengan menggunakan *Selective Acknowledgment* (SACK) maka penerima data dapat menginformasikan pengirim tentang semua segmen yang telah berhasil dikirim sehingga pengirim hanya perlu mengirimi ulang segmen-segmen yang hilang saja (Aloysius Gilang Pradipta, 2015). TCP SACK dipilih untuk implementasi karena memungkinkan kita untuk menerima *selective acknowledgment* seperti contohnya yaitu ketika satu paket yang dikirim hilang maka server akan mengetahui semua paket yang berhasil sampai ke tujuan sehingga ia akan memberitahukan ke pengirim bahwa ada paket yang hilang (Minakhmetov et al., 2018).

2.3 Antrian *Random Early Detection*

Antrian adalah salah satu fungsi dari *router* yang menyimpan paket-paket sebelum di transmisikan. Dalam istilah lain antrian adalah sederetan paket data yang masuk kedalam ruang *buffer* untuk menunggu giliran diproses dan kemudian paket data tersebut akan di transmisikan kembali (Moch Ibnu Rian Febriansah, 2020). Manajemen antrian yang digunakan dalam penelitian ini adalah *Random Early Detection* (RED). Metode RED digunakan untuk mencegah terjadinya *congestion,* maka ia akan mengendalikan ukuran rata-rata dari antrian yang dibuatnya. RED dapat mendeteksi lebih awal kondisi jaringan yang akan mengalami *congestion* (Halim Agung, 2017). Manajemen antrian RED akan menetapkan nilai *max tresh* dan nilai *min tresh*. Jika rata-rata pergerakannya dibawah nilai *max tresh* maka segmen tersebut akan langsung di *drop*, jika rata-rata pergerakannya dibawah nilai *min tresh* maka segmen tersebut akan dilayani dan jika nilainya diantara *max tresh* dan *min tresh* maka mekanisme *drop* dapat menggunakan nilai probabilitas untuk menentukan apakah data tersebut akan di *drop* atau tidak (Jefry Septriajaya, 2019). Ilustrasi untuk antrian *Random Early Detection* ditunjukkan pada Gambar 1.



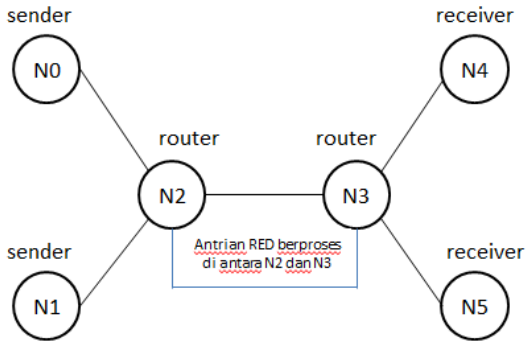
Gambar 1 Gambaran pada Antrian RED

Floyed dan Jacobson mengusulkan antrian RED untuk mengelola terjadinya kemacetan (*congestion*) pada tahap awal. Oleh karena itu *drop packet* secara acak dalam algoritma RED

bisa menjadi solusi yang baik untuk menghindari terjadinya *burst traffic* dan sinkronisasi global (Mahmoud Baklizi, 2020).

2.4 Topologi Simulasi

Dalam penelitian ini digunakan topologi simulasi pengujian *dumbbell*, dengan 2 *node* sebagai pengirim, 2 *node* sebagai *router* dan 2 *node* sebagai penerima. Berikut gambar untuk topologi simulasi dalam penelitian ini.



Gambar 2 Topologi *Dumbbell*

Pada topologi ini menggunakan 6 *node*, penjelasannya adalah sebagai berikut:

- N0 dan N1 berfungsi sebagai *node* pengirim (*sender*).
- N2 dan N3 berfungsi sebagai *router*
- N4 dan N5 berfungsi sebagai *node* penerima (*receiver*)
- Paket dikirimkan dari N0 menuju N4 melalui N2 dan N3 (N0- N2- N3- N4)
- Paket dikirimkan dari N1 menuju N5 melalui N2 dan N3 (N1- N2- N3- N5)

Topologi *dumbbell* umumnya digunakan untuk mempelajari efek jalur *bottleneck* yang dilewati oleh banyak *node* pengirim, efek *bottleneck* terjadi karena adanya perbedaan antara data yang masuk terlampau besar daripada kapasitas yang dapat ditampung pada *node* (Arie Dharma Putra, 2019).

Tabel 1 Simulasi Antrian RED

Parameter Simulasi	Nilai
Model Antrian	*Random Early Detection*
Link node $n0 - n2$	*Bandwidth*: 7 mB, *Delay*: 12 ms
Link node $n1 - n2$	*Bandwidth*: 7 mB, *Delay*: 12 ms
Link node $n2 - n3$	*Bandwidth*: 5 mB, *Delay*: 7 ms
Link node $n3 - n4$	*Bandwidth*: 7 mB, *Delay*: 12 ms
Link node $n3 - n5$	*Bandwidth*: 7 mB, *Delay*: 12 ms
Nilai *min tresh*	6, 10, 14, 18
Nilai *max tresh*	60
Ukuran *Buffer*	60
Waktu Simulasi	200

Parameter pada penelitian ini digunakan sebagai nilai yang berfungsi untuk melakukan proses komputasi saat simulasi berjalan. Besar data yang digunakan untuk file transfer protocol (FTP) dimulai dengan *start* perhitungan 0,1 dan perhitungan selesai pada 200 dalam satuan kB. Dalam penelitian ini peneliti melakukan pengujian sebanyak empat kali dengan skema penambahan nilai *min thresh* yaitu 6 paket, 10 paket, 14 paket dan 18 paket. Format dari data yang diuji berupa angka dan untuk nilai *min tresh* yang digunakan bebas akan tetapi tetap menggunakan kelipatan dari nilai yang sama, pada penelitian ini peneliti menggunakan kelipatan 4 untuk nilai *min tresh* yang diberikan.

Untuk nilai *max tresh* yang telah ditetapkan dalam penelitian ini adalah 60. Peneliti menggunakan manajemen antrian RED oleh karena itu perlu ditetapkannya nilai *max tresh* agar mengetahui apakah data akan diproses atau tidak karena nilai *max tresh* akan berpengaruh terhadap *output* yang dihasilkan. Jika data kurang dari *min tresh* maka data akan diproses, jika data diantara *min tresh* dan *max tresh* maka data akan ditandai dan di *drop* secara acak dan jika data melebihi nilai *max tresh* maka data akan langsung di *drop*.

2.5 Parameter Simulasi

Pada penelitian ini telah ditentukan parameter-parameter jaringannya. Parameter unjuk kerja TCP Sack menggunkan antrian *Random Early Detection* yang digunakan dalam penelitian adalah sebagai berikut.

3. HASIL DAN PEMBAHASAN

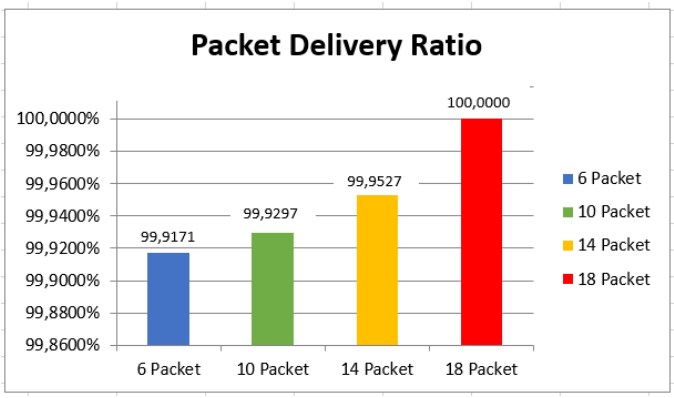
Dalam penelitian ini peneliti melakukan pengujian terhadap kinerja dari TCP Sack dengan menggunakan antrian *Random Early Detection*. Peneliti telah melakukan pengujian sebanyak empat kali dengan skema penambahan nilai *min thresh* sehingga peneliti mendapatkan data yang akan dianalisis. Pengujian terhadap parameter yang dilakukan bertujuan untuk mengetahui kinerja dari TCP Sack, parameter kinerja yang digunakan dalam penelitian ini adalah *packet delivery ratio*, *throughput, end to end delay* dan *packet drop*. Berikut analisis hasil yang telah dilakukan terhadap kinerja TCP Sack.

3.1 Packet Delivery Ratio

Menganalisis *packet delivery ratio* bertujuan untuk mengetahui seberapa besar jumlah paket yang berhasil diterima dari sumber pengirim paket. Perhitungan *packet delivery ratio* ditunjukkan pada Pers. (1).

$$Paket\ delivery\ ratio = \frac{paket\ diterima}{paket\ dikirim} \times 100\% \qquad (1)$$

Hasil pengujian terhadap *packet delivery ratio* dari kinerja TCP Sack menggunakan antrian RED ditunjukkan pada Gambar 3.



Gambar 3 *Packet Delivery Ratio* Menggunakan Antrian RED

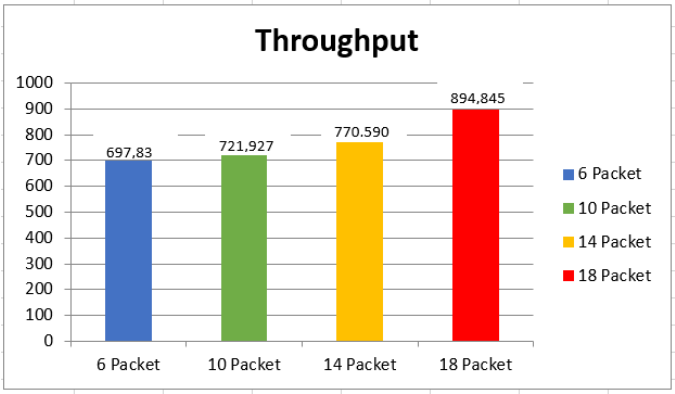
Dari grafik pada Gambar 3 dapat disimpulkan bahwa semakin besar nilai *min thresh* yang diberikan maka *packet delivery ratio* yang dihasilkan juga semakin besar. Pada pengujian ini nilai rata-rata yang dihasilkan untuk *packet delivery ratio* yaitu 99.95%.

3.2 Throughput

Menganalisis parameter *throughput* pada penelitian ini bertujuan untuk mengetahui seberapa banyak paket yang dapat diterima dalam kurun waktu yang telah ditentukan. Perhitungan *throughput* ditunjukkan pada Pers. (2).

$$Throughput = \frac{paket\ diterima}{waktu\ pengiriman} \times ukuran\ paket \qquad (2)$$

Hasil pengujian terhadap *thoroughput* dari kinerja TCP Sack menggunakan antrian RED Perhitungan *throughput* ditunjukkan pada Gambar 4.



Gambar 4 *Throughput* pada Antrian RED

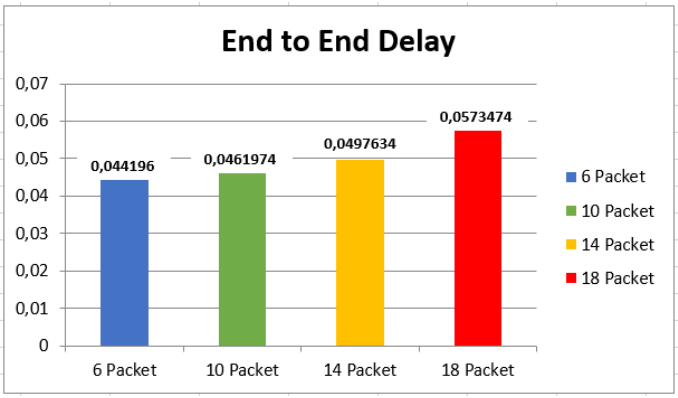
Dari grafik pada Gambar 4 dapat disimpulkan bahwa semakin besar nilai *min thresh* yang diberikan maka *throughput* yang dihasilkan juga semakin meningkat. Pada pengujian ini nilai rata-rata yang dihasilkan untuk *throughput* yaitu 771,298 Mbit.

3.3 End to End Delay

Menganalisis parameter *end to end delay* bertujuan untuk mengetahui seberapa banyak waktu yang dibutuhkan oleh sumber pengirim paket ke penerima paket. Perhitungan *end to end delay* ditunjukkan pada Pers. (3).

$$Delay = \frac{waktu\ diterima - waktu\ dikirim}{jumlah\ paket\ dikirim} \qquad (3)$$

Hasil pengujian terhadap *end to end delay* dari kinerja TCP Sack menggunakan antrian RED ditunjukkan pada Gambar 5.



Gambar 5 *End to End Delay* pada Antrian RED

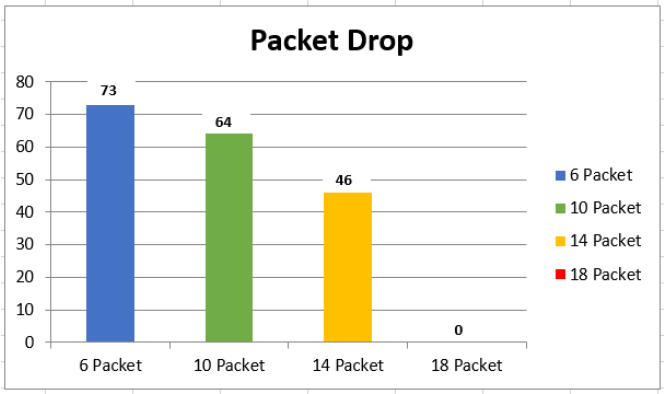
Dari grafik pada Gambar 5 dapat disimpulkan bahwa semakin besar nilai *min thresh* yang diberikan maka *end to end delay* yang dihasilkan juga semakin meningkat. Pada pengujian ini nilai rata-rata yang dihasilkan untuk *end to end delay* yaitu 0,0493761.

3.4 Packet Drop

Menganalisis parameter *packet drop* bertujuan untuk mengetahui seberapa banyak paket yang hilang saat dikirimkan dari sumber pengirim paket ke penerima paket. Perhitungan *packet drop* ditunjukkan pada Pers. (4).

$$Paket\ drop = \frac{paket\ dikirim - paket\ diterima}{paket\ dikirim} \times 100\% \qquad (4)$$

Hasil pengujian terhadap *packet drop* dari kinerja TCP Sack menggunakan antrian RED ditunjukkan pada Gambar 6.



Gambar 6 *Packet Drop* pada Antrian RED

Dari grafik pada Gambar 6 dapat disimpulkan bahwa semakin besar nilai *min thresh* yang diberikan maka *packet drop* yang dihasilkan juga semakin rendah. Pada pengujian ini nilai rata-rata yang dihasilkan untuk *packet drop* yaitu 45,75.

4. KESIMPULAN

Berdasarkan analisis yang dilakukan terhadap kinerja dari TCP Sack menggunakan antrian *Random Early Detection* dengan parameter pengujian yang digunakan yaitu *packet delivery ratio, throughput, end to end delay,* dan *packet drop* didapatkan hasil dari pengujian bahwa semakin besar nilai *min thresh* yang diberikan maka hasil untuk *packet delivery ratio, throughput,* dan *end to end delay* yang dihasilkan juga semakin meningkat. Pada parameter pengujian *packet delivery ratio* dihasilkan nilai rata-rata yaitu 99,95%, pada parameter pengujian *throughput* dihasilkan nilai rata-rata 771,298 Mbit, sedangkan pada parameter pengujian *end to end delay* nilai rata-rata yang dihasilkan yaitu 0,0493761. Sedangkan untuk parameter kinerja dari *packet drop,* semakin besar nilai *min thresh* yang diberikan maka *packet drop* yang dihasilkan semakin rendah dengan nilai rata-rata yang didapatkan sebesar 45,75. Jadi dapat disimpulkan untuk pengujian parameter *end to end delay* maka TCP Sack masih mempunyai kinerja yang kurang baik. Sedangkan untuk pengujian terhadap parameter *packet delivery ratio, throughput,* dan *packet drop* TCP Sack telah memiliki kinerja yang baik.

DAFTAR PUSTAKA

Agung, H. (2017). Analisis Metode RED dan PCQ Pada Mikrotik Desa Wisata Cibuntu-Kuningan. *Jurnal Ilmiah Dasi*, *18*(2), 13–18.

Baklizi, M. (2020). Weight Queue Dynamic Active Queue Management Algorithm. *Symmetry*, *12*(12), 2077. https://doi.org/10.3390/sym12122077

Febriansah, M. ibnu R. (2020). Analisis Bottleneck dan Bufferbloat pada AQM Droptail,RED dan SFQ di Komunikasi Data TCP Newreno. *Jurnal Repositor*, *2*(9), 1213–1224. https://doi.org/10.22219/repositor.v2i9.748

Kothari, N. J., & Dasgupta, K. S. (2006). Performance enhancement of SACK TCP protocol for wireless network by delaying fast recovery. *2006 IFIP International Conference on Wireless and Optical Communications Networks*, 5 pp. – 5. https://doi.org/10.1109/WOCN.2006.1666618

Kristianto, F. P. (2017). *Analisis Unjuk Kerja TCP Sack pada Jaringan Wired*. Universitas Sanata Dharma.

Kurniawan, Y. (2017). *Analisis Perbandingan Unjuk Kerja TCP Reno pada Router Droptail dan Random Early Detection*. Universitas Sanata Dharma.

Minakhmetov, A., Ware, C., & Iannone, L. (2018). TCP Congestion Control in Datacenter Optical Packet Networks on Hybrid Switches. *Journal of Optical Communications and Networking*, *10*(7), B71. https://doi.org/10.1364/JOCN.10.000B71

Nathalia, M. (2016). *Analisis Unjuk Kerja TCP Reno Di Jaringan Single Hop Wireless Link*. Universitas Sanata Dharma.

Nirmala, B. A. S. (2020). *Analisis Perbandingan Kinerja Tcp Dan Udp Pada Jaringan Mpls Dan Non-Mpls Dengan*. Universitas Mataram.

Pamungkas, G. W., Yahya, W., & Nurwarsito, H. (2018). Analisis Perbandingan Kinerja TCP Vegas Dan TCP New Reno Menggunakan Antrian Random Early Detection Dan Droptail. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer (J-PTIIK)*, *2*(10), 3239–3248.

Pradipta, A. G. (2015). *Analisis perbandingan unjuk kerja TCP pada koneksi wired dan wireless dengan dan tanpa sack option*. Universitas Sanata Dharma.

Putra, A. D. (2019). *Analisis Pengaruh Congestion Control DCCP CCID 3 Terhadap TCP Westwood*. Universitas Sanata Dharma.

Rozak, N., Prasetyo, Y., & Mulyana, R. (2017). Perancangan Enterprise Architecture Pada Fungsi Operasional Dan Pelayanan Publik Perum Bulog Divre Jawa Barat Menggunakan Framework Togaf Adm. *EProceedings of Engineering*, *4*(3), 4542–4550.

Septriajaya, J. (2019). *Pengaruh Buffer Pada TCP Vegas Terhadap TCP Westwood di Antrian Droptail dan Random Early Detection*. Universitas Sanata Dharma.

Syaifuddin, M., Andika, B., & Ginting, R. I. (2016). Analisis Celah Keamanan Protocol TCP/IP. *Jurnal Ilmiah Sains Dan Teknologi (SAINTIKOM)*, *16*(2), 130–135.

Vinet, L., & Zhedanov, A. (2011). A 'missing' family of classical orthogonal polynomials. *Journal of Physics A: Mathematical and Theoretical*, *44*(8), 085201. https://doi.org/10.1088/1751-8113/44/8/085201