



Available online at : <http://bit.ly/InfoTekJar>

InfoTekJar : Jurnal Nasional Informatika dan Teknologi Jaringan

ISSN (Print) 2540-7597 | ISSN (Online) 2540-7600



PERFORMANCE COMPARISON TCP VARIANT PADA MODEL JARINGAN WIRED

Naufal Faiz Alfarizi, Bambang Sugiantoro

UIN Sunan Kalijaga, Jl. Laksda Adisucipto, Daerah Istimewa Yogyakarta, 55281, Indonesia

KEYWORDS

TCP, tahoe, reno, sack.

CORRESPONDENCE

Phone: +62 8994064333

E-mail: 20206052001@student.uin-suka.ac.id

A B S T R A K

Pada dasarnya, TCP (*Transmission Control Protocol*) merupakan protokol jaringan komputer yang terletak pada lapisan transport layer yang memiliki sifat *reliable* dan *end to end* dalam pengiriman paket pada jaringan. Sehingga data yang mengalir dapat dibaca oleh penerima paket sebagai TCP *receiver* tanpa duplikasi dan berurutan. TCP menyediakan layanan *flow control* yang memiliki fungsi untuk memastikan pengirim paket sebagai TCP *sender* tidak mengirimkan paket lebih cepat daripada yang dapat ditampung oleh TCP *receiver*. Varian TCP yang digunakan dalam penelitian ini merupakan TCP Tahoe, TCP Reno dan TCP sack dengan menggunakan metode antrian *Random Early Detection* dan *Droptail* pada model jaringan *wired*. Berdasarkan hasil analisa dan pengujian menggunakan parameter *packet delivery ratio*, *throughput*, *end to end delay* dan *packet drop* TCP Sack memiliki kinerja yang lebih baik dengan mengungguli pada parameter *packet delivery ratio* dan *packet drop* menggunakan antrian *Random Early Detection* dan *Droptail* sedangkan, TCP Reno ternyata juga memiliki kinerja yang lebih baik dengan mengungguli pada parameter nilai *throughput* dan *end to end delay* yang lebih rendah dari *variant* lainnya.

PENDAHULUAN

Protokol pada jaringan komputer secara umum merupakan sebuah aturan yang harus diikuti oleh pengirim dan penerima paket dalam jaringan komputer. TCP atau *Transmission Control Protocol* merupakan salah satu *protocol* pada jaringan komputer yang terletak pada lapisan transport layer dengan sifat *reliable* dan *end to end* dalam pengiriman pada jaringan. Dengan menggunakan *flow control*, *sequence number*, *acknowledgment* dan *timer* TCP memastikan bahwa data yang dikirimkan dari proses pengiriman hingga penerima tanpa duplikasi dan berurutan [1].

Pada era teknologi yang semakin berkembang, model jaringan *wired* atau jaringan komputer yang menggunakan kabel sebagai media pendistribusian informasi masih sering digunakan namun, cenderung pada dalam jangkuan *local* seperti dalam satu area gedung atau ruangan. Pada awalnya TCP dirancang untuk keperluan *control protocol* pada jaringan *wired*, TCP juga mengimplementasikan mekanisme *congestion control* dengan tujuan melakukan control pada lalu lintas data pada jaringan komputer sehingga masalah *congestion* tidak terjadi.

Congestion dapat menyebabkan beberapa masalah seperti meningkatnya *packet loss*, melambatnya transmisi dan yang paling parah dapat menyebabkan kelumpuhan jaringan [2].

Pada penelitian ini menggunakan *network simulator 2* atau biasa disebut ns2 untuk merancang sebuah topologi sedemikian yaitu *dumb-bell* yang dapat diterapkan pada varian TCP seperti TCP Tahoe, TCP Reno dan TCP Sack dengan menggunakan metode antrian *Random Early Detection* dan *Droptail*. Pengujian dilakukan dengan menggunakan skema *min thresh* dan *max thresh* pada antrian *Random Early Detection* dan skema peningkatan *buffer* pada antrian *Droptail*. Dengan tujuan melakukan *performance comparison* antara varian TCP serta menghasilkan kesimpulan terkait manajemen antrian yang terbaik dalam mengatasi *congestion* dengan menggunakan nilai parameter seperti *packet delivery ratio*, *throughput*, *end to end delay* dan *packet drop* yang dihasilkan dari masing – masing varian TCP.

TINJAUAN PUSTAKA

TCP (*Transmission Control Protocol*)

TCP (*Transmission Control Protocol*) merupakan sebuah protokol jaringan komputer yang terletak dibagian lapisan transport layer pada model OSI dengan tujuan untuk memungkinkan kumpulan komputer yang saling terhubung dapat berkomunikasi, bertukar informasi dan bertukar data dalam satu konektivitas jaringan.

Protocol TCP memiliki sifat *Connection Oriented* dan *Reliable*. *Connection Oriented* merupakan adanya antara dua *host* untuk membuka sesi komunikasi sebelum melakukan pertukaran data. Sedangkan, *Reliable* merupakan data yang dikirim melalui jaringan secara berurut dengan sebuah penomoran pada paket yang dikirimkan dan paket yang diterima dapat dijawab dengan *acknowledgment* yang juga merupakan sebuah indikasi bahwa data yang dikirim telah diterima oleh TCP *receiver*.

Jaringan Wired

Jaringan *wired* atau jaringan kabel merupakan teknologi jaringan komputer yang menggunakan kabel sebagai media transmisi untuk bertukar informasi dan data. Jaringan *wired* berlisensi *standart IEEE 802.3*. Dalam *IEEE 802.3* mayoritas merupakan teknologi jaringan *Local Area Network (LAN)*. Pada jaringan *wired*, tingkat kestabilan koneksi jaringan menjadi sebuah keunggulan tersendiri yang tidak ditemui pada jaringan lain. Hal tersebut disebabkan pada jaringan *wired* tidak adanya interferensi atau gangguan penurunan jaringan. Selain itu, pada jaringan *wired* memiliki karakteristik *ulimited bandwidth* dan *lowest error - rate* [3].

TCP Tahoe

TCP Tahoe adalah algoritma yang lebih sederhana dari varian TCP lainnya, TCP Tahoe mengacu pada algoritma *congestion control* sebagai TCP yang disarankan oleh Van Jacobson. TCP Tahoe didasarkan tiga algoritma *congestion control* yaitu, *Slow Start (SS)*, *Congestion Avoidance (CA)* dan *fast retransmit*. Tetapi, TCP Tahoe tidak menggunakan algoritma *fast recovery*. Pada fase *congestion avoidance*, TCP Tahoe memberlakukan duplikasi tiga ACK sama dengan *time-out*. Ketika proses duplikat tiga ACK diterima, TCP Tahoe akan menggunakan *fast retransmit* dengan menurunkan *CongWin* menjadi 1 dan mulai masuk ke fase *Slow Start* [4].

TCP Reno

TCP Reno merupakan *variant TCP* setelah TCP Tahoe. TCP Reno masih mengimplementasikan mekanisme TCP Tahoe yaitu *slow-start*, *congestion avoidance*, *fast retransmit* dan menambahkan satu mekanisme yang belum digunakan pada TCP sebelumnya yaitu *fast recovery*. *Slow-start* merupakan fase pada TCP ketika mencari tahu tentang kapasitas jaringan yang ada. *Congestion Avoidance* merupakan fase TCP ketika nilai *CWND* akan naik secara linear (bertambah 1) dan ketika terjadi duplikasi ACK maka nilai *ssthreshold* akan diturunkan setengah dari nilai *CWND* dan nilai *CWND* sendiri diturunkan sebesar nilai *ssthreshold*. *i* merupakan fase terjadi retransmisi pada paket yang hilang ketika menerima 3 duplikasi ACK sehingga dilakukan retransmisi pada paket yang hilang. *Fast recovery* merupakan fase ketika terjadi tiga duplikasi ACK dan telah melalui *fast*

retransmission, TCP tidak masuk ke fase *slow-start* tetapi langsung masuk pada fase *congestion avoidance* [5].

TCP Sack

TCP Sack atau *Selective Acknowledgement* merupakan opsi yang memungkinkan penerima untuk memberi tahu pengirim TCP sender kisaran paket tidak berdampak dengan paket yang diterima. Tanpa SACK, penerima hanya dapat memberi tahu pengirim terkait paket yang diterima secara berurutan. Pengiriman menggunakan informasi ini untuk mengirim ulang paket yang telah diseleksi atau hanya paket yang hilang menggunakan *pipe-line variable* untuk menyimpan data di jaringan [5].

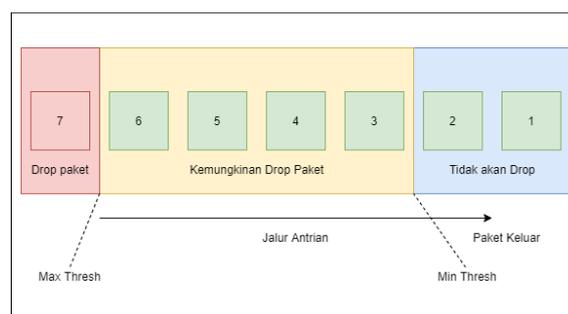
Teori Antrian

Menurut KBBI (Kamus Besar Bahasa Indonesia) antrian merupakan sebuah deretan atau unit yang sedang menunggu giliran untuk dilayani. Antrian pada *router* merupakan deretan atau unit data yang masuk ke dalam ruang penampungan *buffer* yang menunggu giliran untuk diproses dan kemudian ditransmisikan kembali. Manajemen antrian menentukan paket yang akan ditampung pada *buffer*, dibuang atau *drop*, ditandai atau *marked* dan paket yang akan ditransmisikan.

Random Early Detection

Random Early Detection merupakan mekanisme antrian yang dapat melakukan *packet drop* sebelum *buffer* penuh dengan cara menentukan nilai *min thresh* dan *max thresh*. Jika data kurang dari *min thresh*, maka data akan dilayani atau diproses. Jika data diantara *min thresh* dan *max thresh*, maka data akan ditandai dan di *drop* secara acak. Dan jika data melebihi *max thresh* maka data langsung di *drop* [2].

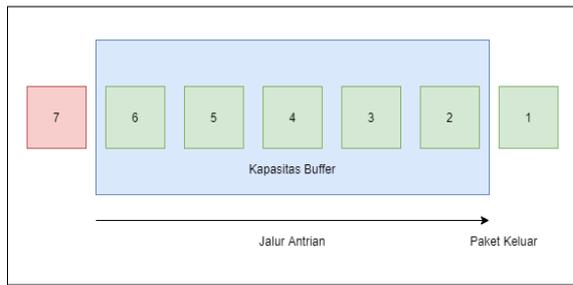
Untuk ilustrasi atau gambaran mengenai model antrian *Random Early Detection* dapat dilihat pada gambar 1.



Gambar 1. *Random Early Detection*

Droptail

Antrian droptail merupakan bagian dari penjadwalan FIFO (*First In First Out*) dimana data yang datang terlebih dahulu akan keluar terlebih dahulu juga. Manajemen antrian *droptail* menggunakan penjadwalan FIFO ketika ruang antrian *buffer* sudah penuh maka paket akan langsung di *drop* dengan tidak ada perlakuan khusus terhadap paket. Semua paket dianggap sama dan tidak ada prioritas untuk paket tertentu. Ilustrasi atau gambaran mengenai model antrian droptail dapat dilihat pada gambar 2.



Gambar 2. *Droptail*

METODOLOGI

Metode yang dilakukan dalam penelitian ini yaitu *experimental* berbasis pada *network simulator* atau metode simulasi jaringan komputer. Dalam melakukan simulasi pada jaringan sebelumnya perlu dilakukan penetapan skenario simulasi yang sesuai agar kinerja dari varian TCP yang digunakan dapat teruji pada berbagai parameter hingga menghasilkan sebuah nilai *output* dari analisis perbandingan setiap parameter.

Untuk membahas tentang alur dari penelitian yang dilakukan terdapat pada gambar 3 yang merupakan *flowchart* rangkuman jalannya penelitian dari awal hingga didapatkannya nilai *output* dari penelitian ini.



Gambar 3. Alur Penelitian

PERANCANGAN

Pada tahap perancangan dibagi menjadi 3 langkah utama seperti analisa kebutuhan sebagai penentu kebutuhan perangkat keras dan perangkat lunak yang digunakan dalam penelitian ini, Perancangan Simulasi sebagai perencanaan parameter yang digunakan dan Perancangan Topologi sebagai gambaran penyusunan jaringan dalam simulasi.

Analisa Kebutuhan

Pada penelitian ini memiliki 2 aspek kebutuhan yaitu perangkat keras dan perangkat lunak yang digunakan untuk dilakukannya proses simulasi antara lain sebagai berikut:

- 1) Aspek Perangkat Keras
 - a) CPU : Intel Core I5 10400F
 - b) VGA : Radeon RX550
 - c) RAM : 16 GB
 - d) SSD : 1 TB
- 2) Aspek Perangkat Lunak
 - a) Virtual Mechine (Virtual Box)
 - b) Ubuntu 20.04 LTS 64 bit
 - c) Network Simulator 2.35

Perancangan Simulasi

Perancangan simulasi dalam penelitian bertujuan untuk menentukan parameter yang digunakan. Pada dasarnya parameter digunakan sebagai nilai acuan yang berfungsi untuk melakukan proses komputasi ketika simulasi dilakukan. Parameter simulasi dalam penelitian ini dapat dilihat pada tabel 1 untuk metode antrian *Random Early Detection* dan tabel 2 untuk metode antrian *Droptail*.

Tabel 1. Simulasi Antrian *RED*

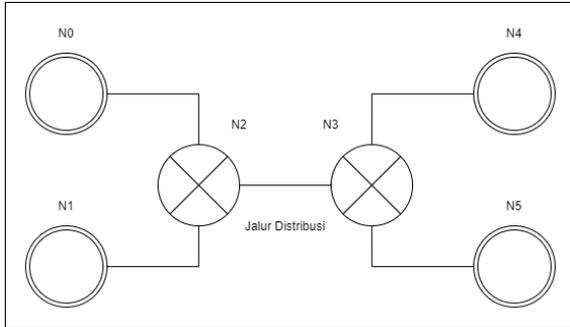
Paramter	Nilai		
Variant TCP	TCP Tahoe	TCP Reno	TCP Sack
Antrian	<i>Random Early Detection</i>		
Link n0-n2, n1-n2	<i>Bandwith: 10 Mbps</i>		
	<i>Delay Propagation: 10 Mbps</i>		
Link n2-n3	<i>Bandwith: 2 Mbps</i>		
	<i>Delay Propagation: 5 Mbps</i>		
Link n3-n4, n3-n5	<i>Bandwith: 5 Mbps</i>		
	<i>Delay Propagation: 10 Mbps</i>		
Nilai Tresh dan Max Tresh	Nilai <i>Min Tresh</i> 5, 10, 15, 20 (4x Pengujian) Nilai <i>Max Tresh</i> 60		
Ukuran Buffer	50 <i>Packet</i>		
Waktu Simulasi	300 Detik		

Tabel 2. Simulasi Antrian *Droptail*

Paramter	Nilai		
Variant TCP	TCP Tahoe	TCP Reno	TCP Sack
Antrian	<i>Droptail</i>		
Link n0-n2, n1-n2	<i>Bandwith: 10 Mbps</i>		
	<i>Delay Propagation: 10 Mbps</i>		
Link n2-n3	<i>Bandwith: 2 Mbps</i>		
	<i>Delay Propagation: 5 Mbps</i>		
Link n3-n4, n3-n5	<i>Bandwith: 5 Mbps</i>		
	<i>Delay Propagation: 10 Mbps</i>		
Nilai Tresh dan Max Tresh	-		
Ukuran Buffer	20, 30, 40, 50 <i>Packet</i> (4x Pengujian)		
Waktu Simulasi	300 Detik		

Perancangan Topologi

Perancangan topologi dalam penelitian ini menggunakan jenis topologi *dumb – bell* karena topologi ini pada umumnya dapat digunakan untuk mengamati, mempelajari terkait efek penyempitan *bandwith* yang diatur pada 2 jalur pendistribusian paket yaitu jalur pengiriman paket dan jalur penerimaan paket yang dapat dilihat pada gambar 4.



Gambar 4. Perancangan Topologi

HASIL DAN PEMBAHASAN

Pengujian

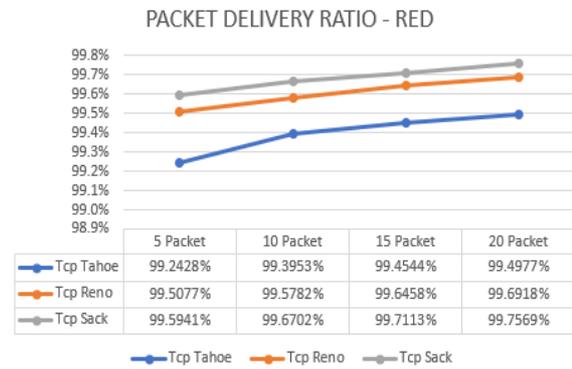
Langkah pengujian dalam penelitian ini dilakukan untuk menguji kinerja dari varian TCP yaitu TCP Tahoe, TCP Reno dan TCP Sack menggunakan antrian *Random Early Detection* dan *Droptail* pada *Network Simulator 2* atau NS2. Pengujian pertama dilakukan dengan antrian *Random Early Detection* pada varian TCP menggunakan skema penambahan pada *min thresh* yaitu 5 paket, 10 paket, 15 paket dan 20 paket untuk selengkapnya terkait parameter pengujian pada antrian RED dapat dilihat pada tabel 4.1. Kemudian pengujian kedua dengan antrian *Droptail* pada varian TCP menggunakan skema penambahan pada *buffer* yaitu dari 20 lalu ke 30 kemudian ke 40 dan 50 untuk selengkapnya terkait parameter pengujian pada antrian *Droptail* dapat dilihat pada tabel 4.2. Setelah melakukan Langkah tersebut kemudian akan menghasilkan file *trace* (.tr) dari hasil simulasi. Kemudian *file* tersebut dianalisa menggunakan *script awk* yang berisi tentang perhitungan parameter *packet delivery ratio*, *throughput*, *end to end delay* dan *packet drop*.

Packet Delivery Ratio

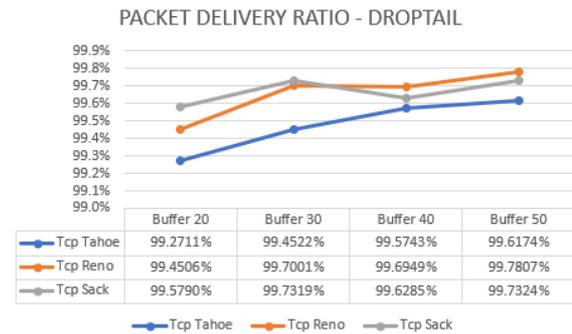
Packet Delivery Ratio merupakan parameter yang digunakan untuk menghitung *presentase* keberhasilan tingkat paket diterima oleh TCP receiver dengan menggunakan perhitungan sebagai berikut:

$$PDR = \frac{\text{Paket Diterima}}{\text{Paket Dikirim}} \times 100\%$$

Sehingga dengan cara perhitungan tersebut dapat diperoleh nilai *packet delivery ratio* dengan antrian *Random Early Detection* pada gambar 5 dan antrian *Droptail* pada gambar 6.



Gambar 5. Packet Delivery Ratio Antrian RED



Gambar 6. Packet Delivery Ratio Antrian Droptail

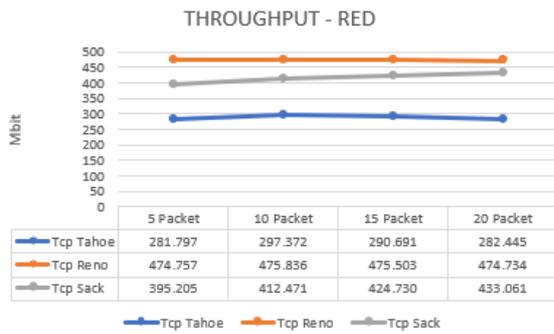
Dari hasil pengujian menggunakan parameter *packet delivery ratio* dapat diperoleh penjelasan bahwa pada antrian *Random Early Detection* TCP Sack sudah mengungguli varian TCP lainnya sejak pengujian menggunakan 5 paket sehingga dapat dihasilkan nilai rata – rata 99.68%. Pada antrian *Droptail* TCP Sack sempat mengalami penurunan pada skema *buffer* 40 dan cenderung berada dibawah TCP Reno namun, hasil rata – rata yang didapat dari antrian *droptail* yaitu TCP Sack lebih unggul dengan nilai 99.67% berbanding 00.01% dengan TCP Reno yang memiliki nilai 99.66%.

Throughput

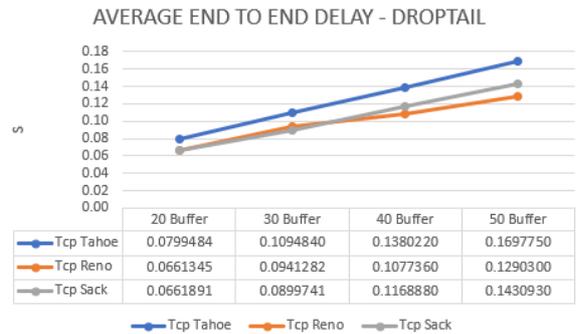
Secara umum, *Throughput* merupakan nilai banyaknya jumlah paket yang diterima oleh TCP *Reciver* dalam waktu simulasi yang sudah ditentukan. Sehingga dapat dilakukan perhitungan dengan menggunakan cara seperti berikut:

$$\text{Throughput} = \frac{\text{Paket Diterima}}{\text{Waktu Pengiriman}} \times \text{Ukuran}$$

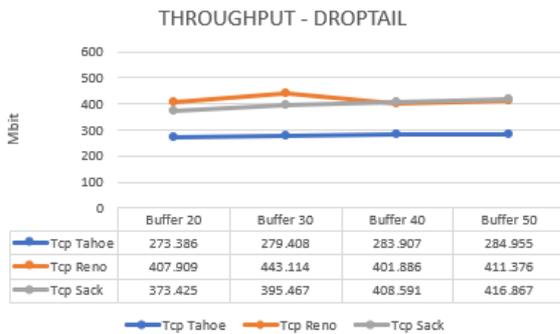
Sehingga dengan cara perhitungan tersebut dapat diperoleh nilai *throughput* yang dihasilkan dari setiap varian TCP yang diujikan dengan antrian *Random Early Detection* pada gambar 7 dan antrian *Droptail* pada gambar 8.



Gambar 7. Throughput Antrian RED



Gambar 10. End to End Delay Antrian Droptail



Gambar 8. Throughput Antrian Droptail

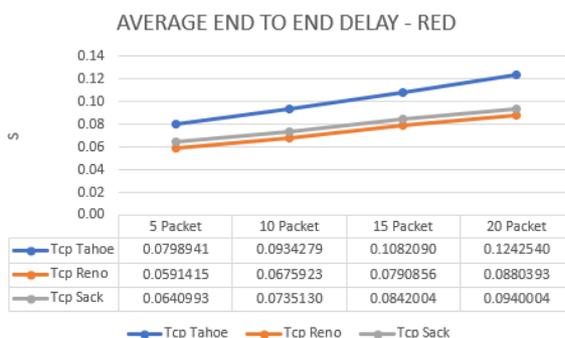
Pada pengujian menggunakan parameter *throughput* dapat diperoleh penjelasan pada antrian *Random Early Detection* TCP Reno cenderung lebih unggul dari varian TCP lainnya sehingga dapat dihasilkan nilai rata – rata *throughput* 475.2075 Mbit. Pada antrian *Droptail* secara sekilas dapat dilihat bahwa TCP Reno juga lebih unggul dari varian TCP lainnya dengan menghasilkan nilai rata – rata 416.071 Mbit.

End to End Delay

End to End Delay merupakan waktu yang diperlukan sebuah paket dalam pendistribusiannya dari TCP *sender* hingga dapat diterima oleh TCP *receiver* sebagai tujuannya. Dengan melakukan cara perhitungan seperti berikut:

$$Delay = \frac{Waktu\ diterima - waktu\ dikirim}{Jumlah\ paket\ dikirim}$$

Hasil yang diperoleh pada parameter *End to End Delay* dapat dilihat pada gambar 9 untuk antrian *Random Early Detection* dan gambar 10 untuk antrian *Droptail*.



Gambar 9. End to End Delay Antrian RED

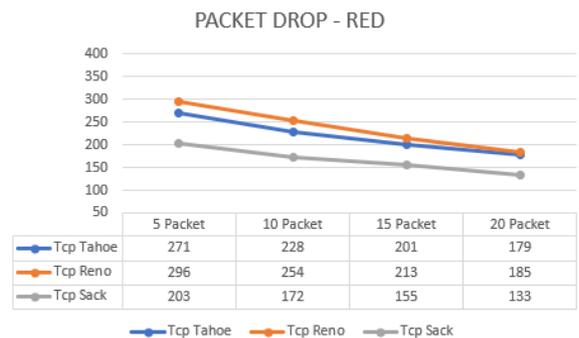
Pada gambar 9 dan 10 merupakan hasil pengujian *End to End Delay* untuk membuktikan varian TCP yang memiliki nilai terendah dalam pengatasan delay pada pendistribusian paket. Dapat dilihat pada antrian *Random Early Detection* TCP Reno memiliki nilai delay yang rendah dan cenderung konsisten dari pengujian 5 paket hingga 20 paket sehingga dapat diperoleh nilai rata – rata *End to End Delay* yang dihasilkan yaitu 0.073464675 *second*. Kemudian, pada antrian *Droptail* dapat dianalisa TCP Reno tetap mempertahankan tingkat konsistensi dalam penanganan *delay* yang baik sehingga menghasilkan nilai rata – rata *end to end delay* 0.099257175 *second*.

Packet Drop

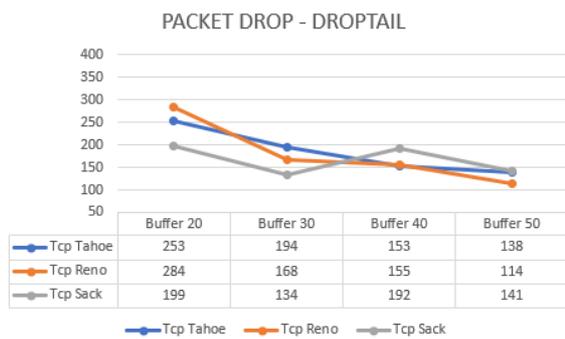
Packet Drop merupakan paket yang hilang ketika proses pengiriman atau pendistribusian paket yang dilakukan oleh TCP *sender* ke TCP *receiver*. Jumlah paket yang hilang tersebut dapat dihitung dengan menggunakan cara seperti berikut:

$$Packet\ Drop = Paket\ dikirim - paket\ diterima$$

Hasil yang diperoleh pada paramter *Packet Drop* dapat dilihat pada gambar 11 untuk antrian *Random Early Detection* dan 12 untuk antrian *Droptail*.



Gambar 11. Packet Drop Antrian RED



Gambar 12. *Packet Drop* Antrian *Droptail*

Packet Drop juga menjadi salah satu parameter penting dalam penentu kualitas jaringan yang dimiliki varian TCP karena faktor jumlah paket yang hilang saat dalam proses pengiriman merupakan tolak ukur sebuah varian TCP dapat mengatur antrian dengan baik. Pada hasil penelitian ini menampilkan pengujian jumlah *packet drop* dari varian TCP yang diujikan, TCP Sack pada antrian *Random Early Detection* cenderung lebih unggul dalam penanganan *packet drop* karena memiliki jumlah rata – rata *packet drop* yang lebih rendah yaitu 165.75 paket dan pada model antrian *Droptail* TCP Sack juga tetap konsisten menjaga kualitas buffer antriannya sehingga jumlah rata - rata *packet drop* yang didapat lebih rendah dari varian TCP lainnya yaitu 166.5. Sehingga dari kedua model antrian TCP Sack memiliki kinerja yang lebih baik dalam parameter manajemen *packet drop* dengan jumlah yang lebih rendah dibanding varian TCP lainnya.

KESIMPULAN

Dari hasil penelitian yang telah dilakukan dapat diperoleh kesimpulan sebagai berikut:

1. Penerapan varian TCP seperti TCP Tahoe, TCP Reno dan TCP Sack dengan model antrian *Random Early Detection* dan *Droptail* yang diujikan menggunakan parameter *Packet Delivery Ratio*, *Throughput*, *End to End Delay* dan *Packet Drop* telah berhasil dilakukan. Proses penerepan telah melalui beberapa Langkah mulai dari instalasi *Network Simulator 2* atau bisa disebut NS 2, Selanjutnya melakukan konfigurasi *script* dari varian TCP yang akan diujikan beserta manajemen antrian yang digunakan dengan menentukan *min thresh* hingga *max thresh* pada antrian *Random Early Detection* dan kapasitas *buffer* pada antrian *Droptail*. Setelah ketentuan simulasi berhasil dilaksanakan selanjutnya, melakukan konfigurasi *script AWK* dengan menghitung atau melakukan *trace* untuk mendapatkan nilai dari setiap parameter yang diujikan.
2. Setelah melakukan pengujian maka diperoleh hasil dari parameter *packet delivery ratio*, TCP Sack kinerja lebih baik dibanding varian TCP lainnya yang diujikan pada model antrian *Random Early Detection* TCP Sack memperoleh rata – rata 99.68% dan pada model antrian *Droptail* 99.67%. Selanjutnya untuk hasil dari parameter *Throughput* TCP Reno memiliki kinerja yang lebih baik dengan nilai rata – rata 475.2075 Mbit pada antrian *Random Early Detection* dan 416.071 Mbit pada antrian *Droptail*. Kemudian untuk model parameter *End to End Delay*, TCP Reno kembali memiliki kinerja yang lebih baik dengan tingkat rata – rata *delay* yang rendah yaitu 0.073464675 *second* pada antrian *Random Early Detection* dan 0.099257175 *second* pada antrian

Droptail. Untuk parameter yang terakhir yaitu *Packet Drop*, TCP Sack memiliki kinerja yang lebih baik dengan jumlah paket yang dibuang rata – rata 165.75 paket pada antrian *Random Early Detection* dan 166.5 pada antrian *Droptail*.

3. Berdasarkan hasil parameter pengujian TCP Sack memiliki keunggulan atau kinerja lebih baik pada *Packet Delivery Ratio* dan *Packet Drop*. Selanjutnya TCP Reno juga memiliki kinerja yang lebih baik pada parameter *Throughput* dan *End to End Delay* karena pada dasarnya TCP Reno merupakan varian TCP yang dikembangkan setelah TCP Tahoe.

DAFTAR PUSTAKA

- [1] J. F. Kurose and K. W. Ross, *Computer networking: a top-down approach*, 6th ed. Boston: Pearson, 2013.
- [2] G. W. Pamungkas, W. Yahya, and H. Nurwarsito, 'Analisis Perbandingan Kinerja TCP Vegas Dan TCP New Reno Menggunakan Antrian Random Early Detection Dan Droptail', p. 10.
- [3] D. Oleh, 'ANALISIS PERBANDINGAN UNJUK KERJA TCP RENO DAN TCP VEGAS PADA JARINGAN KABEL', p. 62, 2016.
- [4] D. Oleh and A. C. Gumilang, 'PROGRAM STUDI TEKNIK INFORMATIKA JURUSAN TEKNIK INFORMATIKA FAKULTAS SAINS DAN TEKNOLOGI UNIVERSITAS SANATA DHARMA YOGYAKARTA', p. 73, 2017.
- [5] H. Kaur and D. G. Singh, 'TCP Congestion Control and Its Variants', p. 10.

BIOGRAFI PENULIS



Naufal Faiz Alfarizi

Mahasiswa Program Studi Magister Informatika dari UIN Sunan Kalijaga yang aktif dalam bidang konsentrasi jaringan computer dan Big Data Analyst.



Bambang Sugiantoro

Dosen Program Studi Informatika dari UIN Sunan Kalijaga yang aktif sebagai tenaga pengajar dalam bidang jaringan computer dan merupakan ketua program studi Magister Informatika UIN Sunan Kalijaga.