

Analisis Bottleneck dan Bufferbloat pada AQM Droptail, RED, dan SFQ di Komunikasi Data TCP Newreno

Moch Ibnu Rian Febriansah*

Teknik Informatika/Universitas Muhammadiyah Malang

ibnufebriansah642@gmail.com*

Abstrak

Antrian atau Queue Active management (AQM) salah satu fungsi dari router yang menyimpan paket-paket yang sebelum di transmisi dalam menangani kejadian congestion network. Ada beberapa jenis AQM dengan mempunyai sifat berbeda dalam menangani terjadinya congestion network. Di jaringan sendiri ada juga yang dapat mengakibatkan terjadinya congestion network atau kemacetan di dalam jaringan salah satunya adalah Bottleneck dan Bufferbloat. Di dalam penelitian ini nantinya akan melakukan pengujian tentang bagaimana AQM Droptail, RED, dan SFQ berjalan pada transmisi TCP NewReno dengan kasus masalah bottleneck dan bufferbloat pada skema pengujian di simulasi NS2. Dari beberapa keadaan bottleneck dan bufferbloat yang dapat mengakibatkan terjadinya gangguan dalam pengiriman paket data atau congestion network. Dalam keadaan ini nantinya peneliti dapat menganalisa dari throughput, paket drop, end to end delay dan Congestion Window (cwnd) variabel pada status TCP yang membatasi jumlah data yang dapat dikirim TCP ke jaringan sebelum menerima ke ACK. Dari hasil parameter ini nantinya akan dianalisa performa terhadap AQM Droptail, RED dan SFQ pada transmisi TCP NewReno. Kontribusi dalam penelitian ini adalah menguji AQM Droptail, RED dan SFQ dalam menangani permasalahan congestion network bottleneck dan bufferbloat dalam komunikasi data TCP NewReno membandingkan dan menganalisis kinerja dari sebuah AQM terhadap skenario yang nantinya akan di ujikan.

Kata kunci: AQM Droptail, AQMRED, AQMSFQ, TCP New Reno, Bottleneck, Bufferbloat

Abstract

Queue or Queue Active Management (AQM) one of the functions of a router that stores packets that are transmitted before handling congestion network events. There are several types of AQM with different properties in handling congestion networks. On the network itself there are also some that can cause the occurrence of network congestion or congestion in the network, one of which is Bottleneck and Bufferbloat. In this study, we will test how AQM Droptail, RED and SFQ run on NewReno's TCP transmission with the case of bottlenecks and bufferbloat problems in the test schemes in NS2 simulations. From several bottlenecks and bufferbloat conditions that can cause disruption in the delivery of data packets or network congestion. In this situation the researcher can later analyze throughput, packet drop, end to end delay and Congestion Window (cwnd) variables on TCP status that limit the amount of data TCP can send to the network before receiving it to the ACK. From the results of these parameters will be analyzed the performance of AQM Droptail, RED and SFQ on NewReno's TCP transmission. The contribution in this study is testing AQM Droptail, RED and SFQ in dealing with congestion network bottleneck and bufferbloat problems in TCP NewReno data communications comparing and analyzing the performance of an AQM for a scenario that will be tested late.

Keywords: AQM Droptail, AQMRED, AQMSFQ, TCP New Reno, Bottleneck, Bufferbloat

1. Pendahuluan.

Antrian atau Queue salah satu fungsi dari router yang menyimpan paket-paket yang sebelum di transmisi. Dalam istilah lainya adalah Sederetan paket data yang masuk kedalam ruang *buffer* yang menunggu giliran untuk diproses dan kemudian ditransmisikan Kembali [1]. Ada beberapa manajemen antrian atau AQM yang dapat diterapkan dalam jaringan, antara lain *Droptail*, *Fair Queuing (FQ)*, *Random Early Drop (RED)* dll. Sebagian besar dalam melakukan komunikasi data di jaringan menggunakan protokol TCP dan UDP. *Transmission Control Protocol (TCP)*, merupakan sebuah protokol yang bersifat *connection oriented* dan *reliable*. Maksud dari

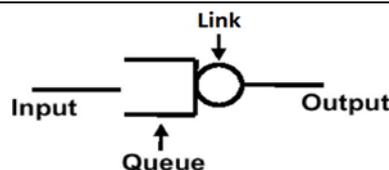
connection oriented yakni adanya negosiasi antara dua *host* untuk membuka sebuah sesi komunikasi sebelum melakukan pertukaran data. Dalam proses komunikasi data diperlukan sebuah *congestion control* yang berfungsi untuk mengatur jalannya komunikasi data. Dan ada beberapa varian TCP dalam pertukaran data yang diantaranya *TCP Tahoe Vegas*, *TCP Reno*, *TCP New Reno* [2]. Pada penelitian sebelumnya oleh Jyoti Chawla dengan judul *Performance Evaluation Of DropTail and Random Early Detection* disini dijelaskan tentang performa dan analisa AQM dengan jenis *DropTail* dan *RED* Pada *TCP RENO* dengan masalah *queue buffering* yang over atau *bufferbloat* pada pengujian simulasi di NS2 dengan parameter Analisa *Throughput*, *Queue loss* dan *end to end* [3]. Pada penelitian lainnya yang di lakukan oleh Buddika Herath tentang *TCP/FTP Queue Size and Packets Drop Performance Simulation Using NS2* dimana pada penelitian ini di jelaskan tentang Analisa paket dorop di AQM *DropTail* pada *TCP Newreno* dengan pengujian menaikkan buffer queue secara bertahap dengan masalah *bottleneck* dengan parameter pengujian yang di Analisa berdasarkan hasil dari *Throughput packetdrop* dan *queue length* [4]. Di penelitian lain yang di lakukan oleh Egba membahas tentang *Comparative Analysis of Three Active Queue Management (AQM) Techniques (RED, SFQ, REM) in Terms of Their Relative Effectiveness in Controlling Congestion in Networks* di penelitian ini membahas Analisa dan kinerja dari beberapa variant AQM (*RED*, *SFQ*, *REM*) untuk menangani *Controlling Congestion* di komukasi data di TCP pada jaringan. Dari hasil pengujian performa AQM *SFQ* baik menangani *paketlost* dan AQM *RED* baik di *delay* dan *queue length* lalu AQM *REM* baik di dalam *throughput* [5]. Dari ketiga penelitian sebelumnya yaitu yang membahas mengenai permasalahan tentang bagaimana AQM berkerja dengan protokol TCP. Dalam segi permasalahan yang terjadi pada ketiga penelitian tersebut adalah kejadian *Bottleneck* dan *Bufferbloat* pada jaringan. Pada *Bottleneck* terjadi ketika peristiwa macetnya proses aliran data (transmisi data) karena perangkat keluaran (output) tidak dapat mengimbangi kinerja perangkat pemrosesan sehingga memperlambat kerja system secara keseluruhan. Pengertian *Bottleneck* secara harfiah dapat diartikan sebagai sumbatan leher botol [6]. Pada penelitian oleh Buddika Herath dan Egba dengan masalah yang di angkat adalah *bottleneck* dengan sekema pengujian mengecilkan bandwidth atau delay antara node pengirim dan penerima paket data dengan *queue size* yang berbeda-beda. Pada *Bufferbloat* adalah sebuah fenomena dalam jaringan packet-switched, di mana kelebihan *buffering paket* menyebabkan latensi tinggi dan variasi dalam penundaan paket (juga dikenal sebagai *jitter*), serta mengurangi throughput keseluruhan jaringan. Latensi terdiri dari tiga jenis penundaan keterlambatan transmisi [7]. Pada penelitian yang dilakukan oleh Jyoti Chawla masalah yang di angkat tentang *bufferbloat* dengan sekema pengujian menaikkan *buffer queue size* yang tinggi dengan pada kekuatan bandwidth yang berbeda-beda.

Bedasarkan pada latar belakang tersebut, penulis akan melakukan analisis simulasi performa AQM (*Active Queue Management*) *DropTail*, *RED* dan *SFQ* dengan taransmisi menggunakan *TCP Newreno*. Di dalam penelitian ini nantinya akan melakukan pengujian tentang bagaimana AQM *DropTail*, *RED* dan *SFQ* berjalan pada transmission TCP *NewReno* dengan kasus masalah *bottleneck* dan *bufferbloat* pada skema pengujian di simulasi NS2. Dari beberapa keadaan *bottleneck* dan *bufferbloat* yang dapat mengakibatkan terjadinya gangguan dalam pengiriman paket data atau *congestion network*. Dalam keadaan ini nantinya peneliti dapat menganalisa dari *troughtput*, *paket drop*, *end to end delay* dan *Congestion Window* (cwnd) variabel pada status TCP yang membatasi jumlah data yang dapat dikirim TCP ke jaringan sebelum menerima ke ACK. Dari hasil parameter ini nantinya akan dianalisa performa teradap AQM *DropTail*, *RED* dan *SFQ* pada transmission TCP *NewReno*. Kontribusi dalam penelitian ini adalah menguji AQM *DropTail*, *RED* dan *SFQ* dalam menangani permasalahan *congestion network bottleneck* dan *bufferbloat* dalam komunikasi data TCP *NewReno* membandingkan dan menganalisis kinerja dari sebuah AQM terhadap skenario yang nantinya akan di ujikan.

2. Metode Penelitian

2.1 Model Antrian

Antrian merupakan sebuah deretan atau unit yang sedang menunggu giliran untuk dilayani (Kamus Besar Bahasa Indonesia). Antrian pada router yakni sederetan paket data yang masuk kedalam ruang *buffer* dan menunggu giliran untuk diproses lalu ditransmissikan kembali. Secara umum antrian dapat digambarkan seperti Gambar 1 dibawah ini [1].

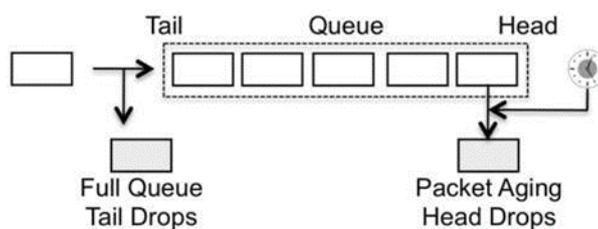


Gambar 1. Antrian

Pada gambar diatas, paket akan dikirimkan dari *node* akan masuk ke dalam sebuah antrian. Ketika status link kosong, maka antrian paling depan akan dilayani oleh link dan kemudian ditransmisikan dan antrian paket yang lain akan ditampung pada *buffer*, manajemen antrian menentukan paket dibuang (*drop*), ditandai (*marked*) dan paket yang mana yang akan ditransmisikan.

2.2 AQM DROPTAIL

Antrian *Droptail* merupakan bagian dari penjadwalan *FIFO* dimana data yang datang terlebih dahulu akan keluar terlebih dahulu juga. Manajemen antrian *droptail* menggunakan penjadwalan *FIFO*, dimana paket yang pertama kali datang maka paket tersebut yang akan di layani akan tetapi ketika buffer antrian sudah mencapai kapasitas maksimum maka paket yang masuk akan dijatuhkan atau di *drop* [3].

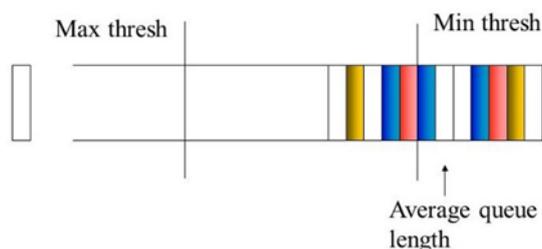


Gambar 2. Proses AQM Droptail

Jadi Droptail pada Gambar 2 akan terus mengabaikan atau menjatuhkan (*drop*) paket data sampai antrian memiliki cukup ruang untuk paket baru atau paket yang akan datang lagi.

2.3 AQM RED

Antrian *Random Early Detection RED* merupakan sebuah model antrian yang terdapat pada *router* yang bertugas untuk memanajemen paket-paket yang terdapat pada ruang antrian (*buffer*) dalam mekanisme antrian *RED* dapat menjatuhkan atau drop paket sebelum congestion terjadi dengan cara menandai atau melakukan drop secara acak. Didalam AQM RED memiliki 2 parameter yaitu α (*alpa*) dengan β (*beta*) untuk menentukan seberapa besar probabilitas [3].



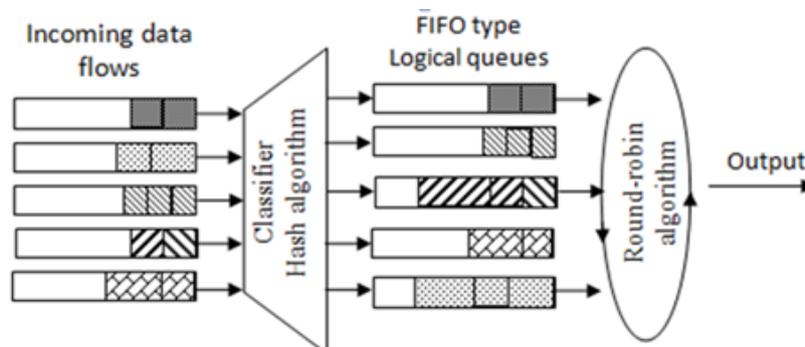
Gambar 3. Proses AQM RED

Dari Gambar 3, menunjukkan jika paket data di bawah zona α maka diterima dan di transmisikan. Jika berada di antara α dan β maka paket data akan di drop secara acak atau random sedangkan paket data di dalam zona β akan di drop atau di buang.

2.4 AQM SFQ

SFQ (Stochastic Fair Queuing) adalah suatu mekanisme antrian congestion control dengan membuat beberapa pembagian atau sekumpulan flow antrian yang menggunakan proses hash. Dari pembagian 128 paket dan ada 1024 sub-stream yang tersedia dan jika terdapat lebih

dari kapasitasnya maka akan dilewati. Kemudian algoritma round-robin akan mulai mendistribusikan bandwidth yang tersedia untuk ke semua sub-stream yang ada di AQM SFQ. Pada Gambar 4 yang fungsi utamanya menyembaingkan *Flow trafik* jika link telah benar-benar penuh. SFQ dapat digunakan untuk TCP maupun UDP [5].



Gambar 4. Proses AQM SFQ

2.5 Congestion network

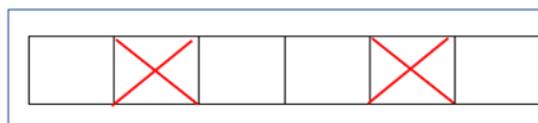
Congestion adalah Diterjemahkan dari bahasa Inggris-Kemacetan jaringan dalam jaringan data dan teori antrian adalah berkurangnya kualitas layanan yang terjadi ketika node atau tautan jaringan membawa lebih banyak data daripada yang bisa ditangani. Efek yang sering terjadi adalah penundaan antrian, kehilangan paket atau pemblokiran koneksi baru. Tanda-tanda dari kehilangan paket atau (*buffer overflow pada router*) dan *Delay besar / Long delay* (antrian pada *buffer router*) Penyebab terjadinya congestion adalah terjadinya Packet data yang dikirim melebihi kapasitas, antrian yang terbatas dan terjadi kebuntuan antara beberapa paket yang ditransmisikan. Efek dari *Congestion* dapat mengakibatkan terjadinya Lambatnya proses pengiriman transmisi paket data, adanya pengiriman packet data yang drop, proses penyimpanan pada *buffer* yang berlebihan sehingga dapat mengakibatkan peralatan jaringan (*hang/down*) dan *Deadlock* [8].

2.6 Transmission Control Protocol (TCP)

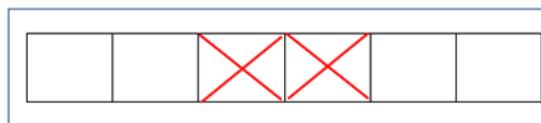
Transmission Control Protocol (TCP), merupakan sebuah protokol yang bersifat *connection oriented* dan *reliable*. Maksud dari *connection oriented* adalah adanya negosiasi antara dua *host* yang saling membuka sebuah sesi komunikasi sebelum melakukan pertukaran data. *Reliable* yakni data dikirim melalui jaringan secara terurut. Dengan adanya penomoran pada setiap data yang akan dikirim, diharapkan paket yang diterima dijawab dengan *acknowledgment*. *Acknowledgment* merupakan sebuah indikasi bahwa data yang dikirim telah diterima kembali oleh sender dan ada beberapa fase dalam tcp di dalam jaringan yaitu fase slow start congestion avoidance, fast retransmit dan recovery [9].

2.7 TCP NewReno

TCP New Reno adalah pengembangan dari TCP Reno cuma perbedaan dari pengembangan di TCP New reno berada pada fase Fase fast recovery dimana TCP New Reno mampu menangani multiple paket error atau paket drop. Pada fase ini TCP New Reno melakukan sedikit pengembangan atau pembaruan, yaitu ketika TCP Reno dan TCP New Reno mengalami single paket drop maka kedua TCP ini akan bertahan di fase retransmit dan akan di lanjutkan ke fase recovery, dan berbeda ketika TCP mengalami multi paket drop atau dua multi paket drop TCP Reno akan masuk ke fase slow start sedangkan TCP New Reno akan tetap di fase Fast recovery [10].



Gambar 5. Paket Single Paket Drop



Gambar 6. Paket Multi Paket Drop

Jika didalam paket terjadi multi paket drop (dua paket drop atau paket error) maka TCP Reno akan gagal dan memulai dari fase slowstart berbeda dengan TCP New Reno yang bertahan di fase fast recovery karena dapat menangani multi paket drop. Didalam menangani multi paket drop atau error TCP New Reno hanya mampu menangani 2 paket drop error atau paket drop yang berderet, Jika paket drop yang berderet melebihi dari 2, maka akan terjadi timeout dan akan masuk ke fase slowstart, seperti pada Gambar 5 dan Gambar 6.

2.8 Bottleneck

Bottleneck pada Gambar 7 adalah peristiwa macetnya proses aliran data (transmisi data) karena sebab-sebab tertentu. Biasanya disebabkan perbedaan antara kecepatan kerja suatu komponen dengan kecepatan jaringannya dan lain-lain. Bottleneck juga dapat terjadi karena perangkat output yang tidak mengimbangi kinerja perangkat sehingga dapat mengakibatkan memperlambat proses kinerja system secara keseluruhan. Pengertian Bottleneck secara umum dapat diartikan sebagai sumbatan atau penyempitan yang menyerupai leher pada botol [7].



Gambar 7. Pengertian Bottleneck

2.9 Bufferbloat

Bufferbloat adalah penyebab terjadinya latensi tinggi di dalam jaringan packet-switch yang di sebabkan oleh buffering paket yang berlebihan. Bufferbloat juga dapat menyebabkan variasi penundaan paket atau dapat mengurangi throughput jaringan secara keseluruhan. Kita ambil contoh ketika sebuah router atau switch dikonfigurasi dengan menggunakan buffer yang terlalu tinggi atau terlalu besar maka jeda waktu tunggu paket akan lama sehingga dapat terjadinya congestion network bahkan dengan kecepatan bandwidth yang tinggi dapat secara praktis tidak dapat digunakan untuk anyak aplikasi interaktif seperti voice over IP (VoIP), game online, dan bahkan surfing web biasa [6].

2.10 QoS (Quality of Service)

QoS (*Quality of Service*) pada sebuah jaringan merupakan parameter dalam sebuah jaringan sebagai metode pengukuran untuk mengukur seberapa baik suatu jaringan yang merupakan suatu usaha untuk mendefinisikan karakteristik dan sifat dari suatu servis pada jaringan. QoS digunakan untuk mengukur sekumpulan atribut kinerja yang telah dispesifikasikan dan diasosiasikan dengan suatu servis [11]. Di harapkan dengan adanya nantinya QoS dapat mengukur tingkat efisien dari sebuah servis-servis yang ada dalam jaringan seperti AQM dan komunikasi data TCP/FTP. pada QoS di penelitian ini saya menggunakan parameter *throughput*, *end to delay* dan *paket drop* sebagai sarana mengukur seberapa baik pengujian yang telah di buat [12]. Serta menggunakan Congestion Window sbagai monitoring paket data pada jaringan (cwnd), variabel status TCP yang membatasi jumlah data yang dapat dikirim TCP ke jaringan sebelum menerima ACK [13]. cendela Penerima (cwnd) adalah variabel yang menampilkan jumlah data yang dapat diterima sisi tujuan.

2.11 Network Simulator – 2

Pada tahun 1995 pembangunan *Network Simulator* didukung oleh DARPA (*Defense Advanced research Project Agency*). *Network simulator* (NS2) adalah alat simulasi jaringan yang

bersifat *open source* yang banyak digunakan dalam mempelajari struktur dinamik dari jaringan komunikasi data. Simulasi dari jaringan *nirkabel* dan wired protokol (seperti algoritma routing, TCP, dan UDP) serta dapat menggunakan AQM di NS2 dapat menyelesaikan dengan baik dengan simulator ini [8].

2.12 AWK

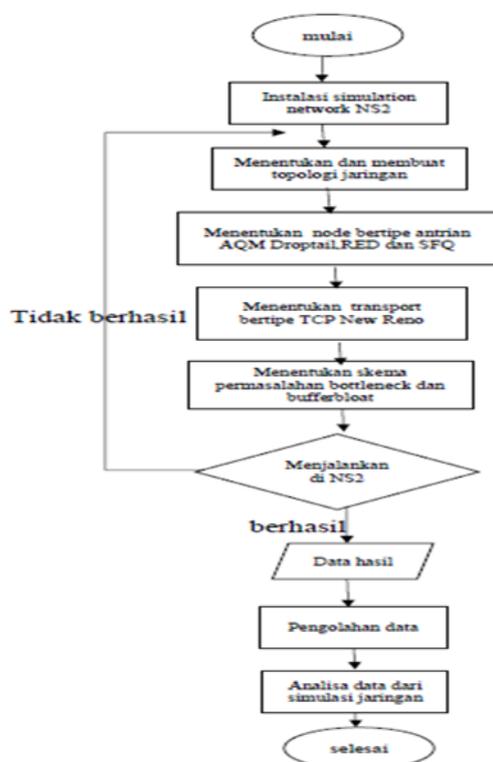
AWK adalah sebuah singkatan dari pembuat algoritma pengurai ini. AWK diambil dari inisial ketiga pembuatnya yaitu "*Aho, Weinberger, and Kernighan*". Awk paling sering digunakan untuk mescan dan memproses sebuah pola. Pencarian pada sebuah file yang cocok dengan pola yang dibuat lalu menyaring pencarian tersebut kedalam sebuah file baru. Cara kerja awk adalah dengan mengambil sebuah data berbasis text yang nantinya data tersebut di cocokkan per baris atau label sesuai action yang di perintahkan[14]. Xgraph adalah alat yang digunakan untuk menampilkan hasil Analisa dan menampilkan grafik atau plot grafik [14].

3.1 Skenario Simulasi

Skenario simulasi yang digunakan pada analisis kinerja AQM terhadap congestion network bottleneck dan bufferbloat dengan membuat beberapa skenario pengujian yang dapat terjadi congestion network untuk membandingkan antara varian kinerja AQM. Untuk membandingkan knerja performa kinerja AQM Droptail, RED dan SFQ terhadap komunikasi data TCP New Reno maka dilakukan tahap skenario perancangan simulasi jaringan dengan parameter yang telah d tentukan. Dengan ketetapan parameter pada Tabel 1. Topologi untuk menunjang scenario pengujian ini mengunkan toplogi dumb-bell dengan menentukan skenario paket data dan buffer yang akan di gunakan Gambar 9. Setelah melakukan beberapa skenario pengujian maka peroses selanjutnya adalah melakukan pengumpulan data dan analisis. Untuk mengambil data yang diperlukan, digunakan program tcl script untuk mendapatkan *file trace* yang dibuat oleh NS2. Kemudian untuk mengetahui nilai dari QoS dengan cara menginstall AWK untuk menggolah data mentah dari NS2.

3.2 Analisis sitem Metodologi Alur penelitian

Berikut ini adalah alur penjabaran dari penelitian tugas akhir ini yang rancangan alurnya seperti Gambar 8 di bawah ini.



Gambar 8. Alur Metode Penelitian

3.3 Pengolahan Data

Setelah Script di jalankan maka di peroleh sebuah output yang di sebut file.tr. Dari data yang di hasilkan merupakan data mentah sehingga perlunya di olah atau di filter dengan menggunakan script .AWK [8]. dimana nantinya script. AWK digunakan untuk mencari nilai rata-rata throughput, rata-rata byte percongestion window, rata-rata delay dan paket lost atau paket drop. Dan bisa juga di buat trafik dari simulasi jaringan tadi dengan menggunakan x-graph dengan format data “.xg” sebagai output dari simulasi jaringan yang telah di buat.

3.4 Analisa Data dari simulasi jaringan

Dari hasil beberapa data skenario pengujian yang diperoleh, penulis nantinya akan melakukan analisis sehingga dapat menarik sebuah kesimpulan dari penelitian ini. dari hasil proses data mentah NS-2 dan hasil yang telah di proses untuk mencari nilai QoS oleh software file AWK sehingga dapat memperoleh hasil dan kesimpulan dari Analisa tugas akhir ini.

3.5 Parameter Simulasi

Pada penelitian ini, penulis menggunakan simulator dalam melakukan pengujian. Simulator yang digunakan adalah *Network Simulator2* (NS2). Parameter yang digunakan dalam pengujian dan penelitian ini seperti pada Tabel 1.

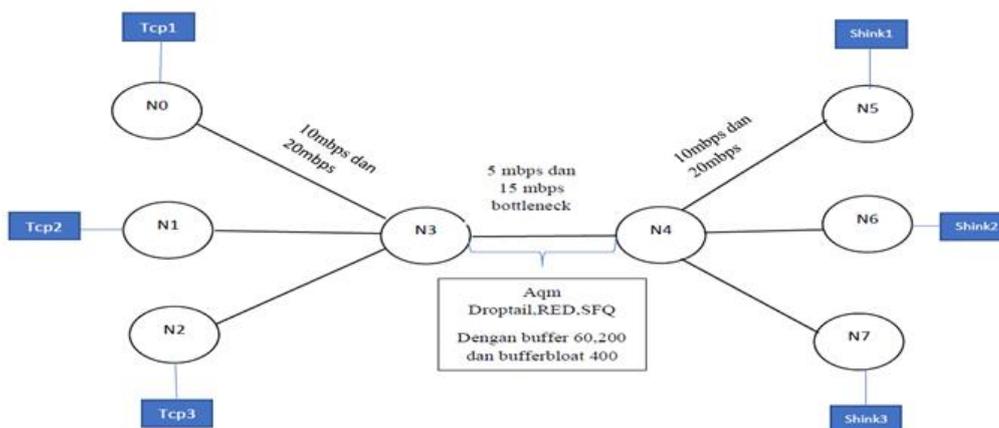
| Parameter simulasi | Nilai parameter | | |
|--|---|---|---|
| | Droptail | RED | SFQ |
| Link Node n0-n3, n1-n3,n2,n3 dan n4-n5,n4-n6,n4-n7 | Bandwidth 10,20 Mbps Delay Propagation 10 ms | Bandwidth 10,20 Mbps Delay Propagation 10 ms | Bandwidth 10,20 Mbps Delay Propagation 10 ms |
| Link Node n3-n4 Bottleneck link | Bandwidth 5,15 Mbps Delay Propagation 10 ms | Bandwidth 5,15 Mbps Delay Propagation 10 ms | Bandwidth 5,15 Mbps Delay Propagation 10 ms |
| Protokol Transport | Tcp New Reno | Tcp New Reno | Tcp New Reno |
| Durasi simulasi | 60 menit | 60 menit | 60 menit |
| Buffer size | 60,200,400 | 60,200,400 | 60,200,400 |
| Jumlah paket data | 1000 | 1000 | 1000 |

Tabel 1. Parameter Simulasi

3.6 Skenario Pengujian

Pada skenario pengujian ini, menggunakan Topologi dibawah ini merupakan topologi sederhana yang bernama *dumb-bell*. Topologi ini pada umumnya digunakan untuk mengamati, mempelajari efek penyempitan bandwidth dimana contohnya seperti gambar di bawah ini [9]. Pada skenario pengujian ini, menggunakan akan menjalankan 3 trafik TCP secara bersamaan mulai detik ke 0.1 hingga berakhir pada 60 menit. Trafik TCP1 berasal dari n0 untuk trafik TCP2 berasal dari n1 dan TCP3 berasal dari node n2. Node n3 dan n4 dalam hal ini bertindak sebagai persntara node yang menghubungkan node lainnya yang dan menjalankan mekanisme pengujian di bottleneck. Pada node n3 dan n4 akan diaplikasikan dengan model 3 AQM / antrian tadi yaitu *Droptail*, *RED* dan *SFQ* dengan menambahkan pengujian dari tingkat nilai *buffer* 60, maximal 200 sampai *buffer* 400 yang memungkinkan terjadinya *bufferbloat* dengan kecepatan bandwidth 10mbps,20mbps dan bottleneck 5 mbps,15 mbps. Dengan skenario pengujian seperti topologi pada Gambar 9 di bawah ini.

Pada skenario pengujian di atas menunjukan jalannya jaringan komunikasi data dari node n0, n1, n2 dan akan menuju ke node n5, n6 dan n7 serta sebagai TCP shinknya. Melalui perantara penghubung 2 yaitu node n3 dan n4 sebagai perantara antara node pengirim paket data. Di dalam penelitian ini ada 18 skenario pengujian dengan membagi setiap 3 model AQM mempunyai 6 skenario yang berbeda berdasarkan dari pengujian parameter.



Gambar 9. Skenario Pengujian

3.7 Hasil Simulasi

Setelah menjalankan simulasi sesuai dengan skenario yang telah dibuat, maka didapatkan hasil grafik congestion window pada AQM Droptail, RED dan SFQ serta hasil dari Analisa throughput paket drop dan end to delay Tabel 2, Tabel 3, Tabel 4, Tabel 5, Tabel 6, dan Tabel 7 berikut ini.

Tabel 2. Qos pengujian 1 bandwith 10 mbps dan 60 buffer

| Nama AQM | Troughput | | | Paket drop | | | End to delay | | |
|----------|-------------------|-------------------|-------------------|------------|------|------|-------------------|-------------------|-----------------|
| | Tcp1 | Tcp2 | Tcp3 | Tcp1 | Tcp2 | Tcp3 | Tcp1 | Tcp2 | Tcp3 |
| Droptail | 146.449 Mbit | 76.6641 Mbit | 58.0893 Mbit | 64 | 19 | 16 | 0.092429 seconds | 0.104585 seconds | 38.9882 seconds |
| | Avg 0.292897 Mbps | Avg 0.153328 Mbps | Avg 0.116179 Mbps | Drop:99 | | | | | |
| RED | 130.508 Mbit | 81.8771 Mbit | 65.6509 Mbit | 78 | 54 | 26 | 0.0524716 seconds | 0.0551333 seconds | 37.0869 seconds |
| | Avg 0.261016 Mbps | Avg 0.163754 Mbps | Avg 0.131302 Mbps | Drop:158 | | | | | |
| SFQ | 121.978 Mbit | 82.1866 Mbit | 69.9039 Mbit | 46 | 45 | 42 | 0.0500906 seconds | 0.0559535 seconds | 37.951 seconds |
| | Avg 0.243957 Mbps | Avg 0.164373 Mbps | Avg 0.139808 Mbps | Drop:133 | | | | | |

Tabel 3. Qos pengujian 2 bandwith 10 mbps dan 200 buffer

| Nama AQM | Troughput | | | Paket drop | | | End to delay | | |
|----------|-------------------|-------------------|-------------------|------------|------|------|-------------------|-------------------|-----------------|
| | Tcp1 | Tcp2 | Tcp3 | Tcp1 | Tcp2 | Tcp3 | Tcp1 | Tcp2 | Tcp3 |
| Droptail | 132.198 Mbit | 82.131 Mbit | 62.1359 Mbit | 166 | 103 | 99 | 0.179182 seconds | 0.238344 seconds | 35.9011 seconds |
| | Avg 0.264396 Mbps | Avg 0.164262 Mbps | Avg 0.124272 Mbps | Drop: 368 | | | | | |
| RED | 120.891 Mbit | 81.0599 Mbit | 76.8784 Mbit | 31 | 26 | 19 | 0.062744 seconds | 0.0649404 seconds | 37.2121 seconds |
| | Avg 0.241783 Mbps | Avg 0.16212 Mbps | Avg 0.153757 Mbps | Drop: 76 | | | | | |
| SFQ | 121.978 Mbit | 82.1866 Mbit | 69.9039 Mbit | 46 | 45 | 42 | 0.0500906 seconds | 0.0559535 seconds | 37.951 seconds |
| | Avg 0.243957 Mbps | Avg 0.164373 Mbps | Avg 0.139808 Mbps | drop:133 | | | | | |

Tabel 4. Qos pengujian 3 bandwidth 10 mbps dan 400 buffer

| Nama AQM | Troughput | | | Paket drop | | | End to delay | | |
|----------|-------------------------|-------------------------|-------------------------|------------|------|------|----------------------|----------------------|--------------------|
| | Tcp1 | Tcp2 | Tcp3 | Tcp1 | Tcp2 | Tcp3 | Tcp1 | Tcp2 | Tcp3 |
| Droptail | 109.029 Mbit | 104.879 Mbit | 62.3105 Mbit | 340 | 222 | 205 | 0.264069 seconds | 0.328241 seconds | 35.3815 seconds |
| | Avg 0.218058 Mbps | Avg 0.209759 Mbps | Avg 0.124621 Mbps | Drop: 767 | | | | | |
| RED | 137.022 Mbit | 83.2974 Mbit | 60.0571 Mbit | 22 | 19 | 20 | 0.0687181 seconds | 0.070794 seconds | 36.9126 seconds |
| | Avg 0.274045 Mbps | Avg 0.166595 Mbps | Avg 0.120114 Mbps | Drop: 61 | | | | | |
| SFQ | 121.978 Mbit | 82.1866 Mbit | 69.9039 Mbit | 46 | 45 | 42 | 0.0500906 seconds | 0.0559535 seconds | 37.951 seconds |
| | Avg 0.243957 Mbps | Avg 0.164373 Mbps | Avg 0.139808 Mbps | Drop:133 | | | | | |

Tabel 5. Qos pengujian 4 bandwidth 20 mbps dan 60 buffer

| Nama AQM | Troughput | | | Paket drop | | | End to delay | | |
|----------|-------------------------|-------------------------|-------------------------|------------|------|------|----------------------|----------------------|--------------------|
| | Tcp1 | Tcp2 | Tcp3 | Tcp1 | Tcp2 | Tcp3 | Tcp1 | Tcp2 | Tcp3 |
| Droptail | 280.67 Mbit | 270.252 Mbit | 172.871 Mbit | 112 | 53 | 34 | 0.0448283 seconds | 0.0473921 seconds | 36.8511 seconds |
| | Avg 0.56134 Mbps | Avg 0.540504 Mbps | Avg 0.345742 Mbps | Drop: 119 | | | | | |
| RED | 257.826 Mbit | 250.05 Mbit | 210.727 Mbit | 120 | 53 | 16 | 0.0352015 seconds | 0.035394 seconds | 38.2838 seconds |
| | Avg 0.515652 Mbps | Avg 0.500101 Mbps | Avg 0.421453 Mbps | Drop: 189 | | | | | |
| SFQ | 295.785 Mbit | 227.794 Mbit | 202.03 Mbit | 46 | 29 | 27 | 0.0341219 seconds | 0.0346748 seconds | 38.4275 seconds |
| | Avg 0.59157 Mbps | Avg 0.59157 Mbps | 0.404061 Mbps | Drop:102 | | | | | |

Tabel 6. Qos pengujian 5 bandwidth 20 mbps dan 200 buffer

| Nama AQM | Troughput | | | Paket drop | | | End to delay | | |
|----------|-------------------------|-------------------------|-------------------------|------------|------|------|----------------------|----------------------|--------------------|
| | Tcp1 | Tcp2 | Tcp3 | Tcp1 | Tcp2 | Tcp3 | Tcp1 | Tcp2 | Tcp3 |
| Droptail | 301.593 Mbit | 234.768 Mbit | 206.037 Mbit | 163 | 126 | 125 | 0.0849957 seconds | 0.0913695 seconds | 39.8326 seconds |
| | Avg 0.603187 Mbps | Avg 0.469537 Mbps | Avg 0.412075 Mbps | Drop: 414 | | | | | |
| RED | 330.904 Mbit | 249.765 Mbit | 212.853 Mbit | 44 | 18 | 13 | 0.036746 seconds | 0.0368991 seconds | 38.9263 seconds |
| | Avg 0.661807 Mbps | Avg 0.499529 Mbps | Avg 0.425706 Mbps | Drop: 75 | | | | | |
| SFQ | 295.785 Mbit | 227.794 Mbit | 202.03 Mbit | 46 | 29 | 27 | 0.0341219 seconds | 0.0346748 seconds | 38.4275 seconds |
| | Avg 0.59157 Mbps | Avg 0.455588 Mbps | 0.404061 Mbps | Drop:102 | | | | | |

Tabel 7. Qos pengujian6 bandwith 20 mbps dan 400 buffer

| Nama AQM | Troughput | | | Paket drop | | | End to delay | | |
|----------|-------------------|-------------------|-------------------|------------|------|------|-------------------|-------------------|-----------------|
| | Tcp1 | Tcp2 | Tcp3 | Tcp1 | Tcp2 | Tcp3 | Tcp1 | Tcp2 | Tcp3 |
| Droptail | 350.407 Mbit | 210.647 Mbit | 243.52 Mbit | 96 | 246 | 281 | 0.104924 seconds | 0.143677 seconds | 36.7423 seconds |
| | Avg 0.700814 Mbps | Avg 0.421295 Mbps | Avg 0.48704 Mbps | Drop: 623 | | | | | |
| RED | 352.311 Mbit | 244.948 Mbit | 202.737 Mbit | 60 | 18 | 14 | 0.038226 seconds | 0.0383627 seconds | 37.8393 seconds |
| | Avg 0.704622 Mbps | Avg 0.489897 Mbps | Avg 0.405473 Mbps | Drop: 92 | | | | | |
| SFQ | 295.785 Mbit | 227.794 Mbit | 202.03 Mbit | 46 | 29 | 27 | 0.0341219 seconds | 0.0346748 seconds | 38.4275 seconds |
| | Avg 0.59157 Mbps | Avg 0.455588 Mbps | Avg 0.404061 Mbps | Drop: 102 | | | | | |

Keterangan tabel: Pada Tcp 1 adalah proses pengiriman ke 1 dari node n0 ke node n5 untuk Tcp 2 memproses pengiriman ke 2 dari node n1 ke node n6 dan Tcp 3 memproses pengiriman ke 3 dari node n2 ke node n7

4 Kesimpulan

Berdasarkan hasil dari pengujian dan analisis yang telah dilakukan, kesimpulan yang dapat diambil sebagai berikut:

1. Perbandingan Data Hasil di AQM Droptail, RED dan SFQ pada perubahan bandwidth dan buffer di pengujian bandwidth 10 mbps dan 60 buffer. dapat kita lihat nilai yang tinggi throughput yang tinggi berada di AQM Droptail dan RED sedangkan SFQ mempunyai throughput yang lebih kecil. Dalam segi menangani paket drop atau loss dalam pengujian ini Droptail dan SFQ mempunyai nilai loss yang lebih kecil sedangkan RED mempunyai nilai drop yang besar. Untuk end to delay dalam pengujian ini AQM RED dan SFQ mempunyai sedikit lebih cepat dari pada AQM Droptail.
2. Perbandingan Data Hasil di AQM Droptail, RED dan SFQ pada perubahan bandwidth dan buffer di pengujian bandwidth 10 mbps dan 200 buffer. dapat kita lihat nilai yang tinggi throughput yang tinggi berada di AQM Droptail dan SFQ sedangkan RED mempunyai throughput yang sedikit lebih kecil. Dalam segi menangani paket drop atau loss dalam pengujian ini RED dan SFQ mempunyai nilai loss yang lebih kecil sedangkan Droptail mempunyai nilai drop yang besar. Untuk end to delay dalam pengujian ini AQM Droptail dan SFQ mempunyai sedikit lebih cepat dari pada AQM RED.
3. Perbandingan Data Hasil di AQM Droptail, RED dan SFQ pada perubahan bandwidth dan buffer di pengujian bandwidth 10 mbps dan 400 buffer. dapat kita lihat nilai yang tinggi throughput yang tinggi berada di AQM Droptail dan RED sedangkan SFQ mempunyai throughput yang sedikit lebih kecil tetapi dari hasil tersebut tidak jauh berbeda nilai throughputnya. Dalam segi menangani paket drop atau loss dalam pengujian ini RED dan SFQ mempunyai nilai loss yang lebih kecil sedangkan Droptail mempunyai nilai drop yang besar. Untuk end to delay dalam pengujian ini AQM SFQ dan RED mempunyai sedikit lebih cepat dari pada AQM Droptail tetapi dari hasil tersebut tidak jauh berbeda nilai end to delay.
4. Perbandingan Data Hasil di AQM Droptail, RED dan SFQ pada perubahan bandwidth dan buffer di pengujian bandwidth 20 mbps dan 60 buffer. dapat kita lihat nilai yang tinggi throughput yang tinggi berada di AQM Droptail dan RED sedangkan SFQ mempunyai throughput yang sedikit lebih kecil tetapi dari hasil tersebut tidak jauh berbeda nilai throughputnya. Dalam segi menangani paket drop atau loss dalam pengujian ini Droptail dan SFQ mempunyai nilai loss yang lebih kecil sedangkan RED mempunyai nilai drop yang besar. Untuk end to delay dalam pengujian ini AQM Droptail dan SFQ mempunyai sedikit lebih cepat dari pada AQM RED tetapi dari hasil tersebut tidak jauh berbeda nilai end to delay.
5. Perbandingan Data Hasil di AQM Droptail, RED dan SFQ pada perubahan bandwidth dan buffer di pengujian bandwidth 20 mbps dan 200 buffer. dapat kita lihat nilai yang tinggi throughput yang tinggi berada di AQM Droptail dan RED sedangkan SFQ mempunyai throughput yang sedikit lebih kecil tetapi dari hasil tersebut tidak jauh berbeda nilai throughputnya. Dalam segi menangani paket drop atau loss dalam pengujian ini RED dan SFQ

- mempunyai nilai loss yang lebih kecil sedangkan Droptail mempunyai nilai drop yang besar. Untuk end to delay dalam pengujian ini AQM RED dan SFQ mempunyai sedikit lebih cepat dari pada AQM Droptail tetapi dari hasil tersebut tidak jauh berbeda nilai end to delay.
6. Perbandingan Data Hasil di AQM Droptail, RED dan SFQ pada perubahan bandwidth dan buffer di pengujian bandwidth 20 mbps dan 200 buffer dan buffer di pengujian bandwidth 20 mbps dan 400 buffer. dapat kita lihat nilai yang tinggi throughput yang tinggi berada di AQM Droptail dan RED sedangkan SFQ mempunyai throughput yang sedikit lebih kecil tetapi dari hasil tersebut tidak jauh berbeda nilai throughputnya. Dalam segi menangani paket drop atau loss dalam pengujian ini RED dan SFQ mempunyai nilai loss yang lebih kecil sedangkan Droptail mempunyai nilai drop yang besar. Untuk end to delay dalam pengujian ini AQM RED dan SFQ mempunyai sedikit lebih cepat dari pada AQM Droptail tetapi dari hasil tersebut tidak jauh berbeda nilai end to delay
 7. Dari perbandingan hasil di 6 percobaan terhadap AQM Droptail RED dan SFQ dengan menggunakan bandwidth 10 dan 20 dengan link bottleneck 5 dan 15 bandwidth serta juga menggunakan variasi bufferbloat dari 60 200 dan 400 buffer dengan menggunakan TCP NewReno. Dalam pengujian ini kejadian bottleneck dan bufferbloat. Yang berpengaruh terhadap jaringan tetapi kenaikan nilai dan penurunan yang signifikan dari QoS pada AQM di TCP NewReno adalah congestion network bottleneck sedangkan pada bufferbloat juga berpengaruh tetapi tidak signifikan di bottleneck.
 8. Dalam penelitian ini juga ada pengaruh lain yaitu bahwa AQM SFQ tidak berpengaruh oleh perubahan bandwidth tetapi berpengaruh di pada perubahan bottleneck. Kenapa demikian Karena ketika ada sebuah paket pertama masuk dan kemudian ada paket kedua yang masuk secara hampir bersamaan masuk maka paket kedua akan dua akan dikirimkan atau di sisipkan paket ke paket pertama dengan pembagian substream-substream yang ada tanpa adanya antrian menunggu giliran seperti droptail dan ketika melebihi kapasitas buffer maka paket tersebut akan di drop.

4.2 Saran

Saran yang diberikan penulis untuk pengembangan penelitian selanjutnya yakni :

1. Melakukan pengujian lebih lanjut mengenai kapasitas *bandwidth* yang terpakai pada AQM Droptail RED dan SFQ terhadap variasi TCP yang lain
2. Melakukan pengujian lebih lanjut mengenai tentang pengaruh congestion network apa saja yang berpengaruh pada jaringan

Referensi

- [1] U. R. Pujeri, V. Palaniswamy, P. Ramanathan, and R. Pujeri, "Comparative analysis and comparison of various AQM algorithm for high speed," *Indian J. Sci. Technol.*, vol. 8, no. 35, 2015
- [2] Chaudhary, Pooja Kumar, Sachin, "A Review of Comparative Analysis of TCP Variants for Congestion Control in Network," *International Journal of Computer Applications.*, 2017
- [3] Chawla, Jyoti, Kumari, Shailja, "Performance Evaluation Of DropTail and Random Early Detection,"., 2016
- [4] Herath, Buddika, Engineering, Faculty, "TCP / FTP Queue Size and Packets Drop Performance Simulation Using NS2,"., 2015
- [5] Ifeanyichukwu, Egba, "Comparative Analysis of Three Active Queue Management (AQM) Techniques (RED, SFQ, REM) in Terms of Their Relative Effectiveness in Controlling Congestion in Networks,"., 2018
- [6] Kumar, Deepak. Verma, Chaman, "A Resolute Approach to Bufferbloat Crises: A Experimental Study using Network Simulator-2,"., 2016
- [7] Saini, Himanshi, "Impact of Location of Bottleneck Link on TCP- Tahoe Performance,"., 2017
- [8] Wu, Wei. Du, Wencai. Ruan, Guolong, "Network congestion control methods and theory," *International Journal of Grid and Utility Computing.*, 2015
- [9] Abed, Ghassan A. Ismail, Mahamod. Jumari, Kasmiran, "Exploration and evaluation of traditional TCP congestion control techniques,". *Journal of King Saud University - Computer and Information Sciences.*, 2012
- [10] Lar, Saleem-ullah. Liao, Xiaofeng. Guo, Songtao, "Modeling TCP NewReno Slow Start and Congestion- Avoidance using Simulation Approach,". *Journal of Computer Science.*, 2011

-
- [11] Healy, Maurice, "Development of the Smart QoS Monitors to Enhance the Performance of the NS2 Network Simulator,".,2010
- [12] Fakultas, Hasanul Fahmi, " Analisis Qos (Quality of Service) Pengukuran Delay, Jitter, Packet Lost Dan Throughput Untuk Mendapatkan Kualitas Kerja Radio Streaming Yang Baik Analysis Qos (Quality of Service) Measurement of Delay , Jitter, Packet Lost and Throughput To Get Good Qualiy,".,2018
- [13] Dukupati, Nandita.Refice, Tiziana.Cheng, Yuchung.Chu, "An argument for increasing TCP's initial congestion window ,",2010
- [14] Patel, Radhika.Patel, Nehal.Patel, Sandip, "Information and Communicatio Technology for Competitive Strategies,".'2019