



Implementasi Active Queue Management pada Jaringan Bottleneck

Misbahul Fajri^{1*}, Anita Ratnasari¹, Yani Prabowo²

¹Teknik Informatika, Universitas Mercu Buana,
Jl. Meruya Selatan, Jakarta 11650, Indonesia

²Sistem Komputer, Universitas Budi Luhur,
Jl. Cileduk Raya, Jakarta 12260, Indonesia

*fajri@mercubuana.ac.id (Corresponding Email only)

Abstrak:

Teknologi *TCP end-to-end congestion control* hanya menangani kemacetan jaringan dari suatu host tetapi tidak pada jaringan atau *intermediate network*, oleh karena itu teknologi *Active Queue Management* (AQM) dikembangkan untuk mengantisipasi kemacetan didalam jaringan yaitu pada router. AQM dapat mengatasi *buffer overflow* pada router dan mencegah terjadinya masalah sinkronisasi pada jaringan yang padat, yaitu dengan mendeteksi sebelum terjadi kemacetan jaringan. AQM RED (*random early detection*) dikembangkan dengan pendekatan heuristik untuk menggantikan kekurangan mekanisme Droptail pada buffer router, sehingga dapat mengantisipasi memburuknya kondisi jaringan yaitu; hilangnya paket yang berlebihan, utilitas link yang rendah (*low throughput*), dan tingginya antrian (*delay*), dengan demikian kemacetan yang parah (*congestion collapse*) tidak akan terjadi. Permasalahan ini menarik untuk dipelajari dampak implementasi AQM yang akan dilakukan dalam penelitian ini. Protokol AQM adalah mekanisme pengendali kemacetan yang diterapkan pada lapisan ke-tiga OSI pada peralatan jaringan router, dengan menggunakan simulator NS2 dibuat topologi jaringan *bottleneck* dan diuji coba dengan skenario trafik. Pada penelitian ini akan diimplementasikan metode AQM RED yang disimulasikan dengan pengiriman paket FTP dengan beberapa skenario trafik. Hasil penelitian menunjukkan RED dapat mengatur queue ke target yang diharapkan 100 dibandingkan Droptail yang mengalami *bufferbloat*, selain itu RED dapat mereduksi *queing delay* yang rendah dibandingkan mekanisme tanpa AQM atau Drop-tail.

Keywords:

RED (*Random Early Detection*);
Drop-tail; *congestion control*;
AQM (*active queue management*);
network traffic;

Riwayat Artikel:

Diserahkan Juli 20 2022
Direvisi September 09 2022
Diterima Februari 01 2023
Dipublikasi 30 Arpil 2023

DOI:

10.22441/incomtech.v13i1.16086

Copyright © 2023 Universitas Mercu Buana.
All right reserved.

1. PENDAHULUAN

Dengan berkembangnya komputer dan layanannya, membuat pemakaian jaringan komputer dan internet semakin ramai dan padat, yang membuat trafik pengiriman paket data menjadi tinggi pada jaringan internet maupun intranet. Penggunaan internet didominasi oleh trafik TCP, yaitu selain aplikasi FTP dan Web juga didukung dengan aplikasi *Video on Demand* (VoD) seperti Youtube dan MSN Video menggunakan TCP untuk paket video melalui HTTP, dan aplikasi Skype menghindari network firewall dengan TCP dengan port HTTP [1]. Bahkan Netflix yang menggunakan port TCP mengkonsumsi 30 persen trafik *downstream* di USA [2], dengan demikian membutuhkan penjaminan layanan-layanan tersebut dalam pengiriman pada jaringan.

Pertama kali ditemukan adanya kemacetan pada jaringan di internet dengan terjadinya penurunan drastis kecepatan transfer sehingga terjadi kolaps, ini disebabkan karena pengiriman paket yang melebihi kapasitas link dan belum adanya mekanisme yang menangani masalah tersebut, sehingga terjadi *congestion collapse*, yang teori *Congestion Avoidance and Control* [3] diusulkan. *Congestion* akan mengakibatkan penurunan kinerja jaringan diantaranya; pembuangan paket, *low throughput* yang rendah, antrian paket yang tinggi, pemanfaatan kapasitas link yang rendah, dan kemacetan yang tinggi (*congestion collapse*). Oleh karena penanganan *congestion* sangatlah penting karena sangat mempengaruhi terhadap kualitas layanan. Berbagai solusi dan metode ditawarkan dalam algoritma AQM baik dengan mekanisme sederhana maupun yang canggih, tetapi hal tersebut itu mempunyai kekurangan dan kelebihan, serta karakteristik berbeda-beda saat diimplmentasikan pada kondisi jaringan yang dinamis. Hal ini menjadikan penelitian dibidang ini sangat menarik untuk dikembangkan.

Congestion control adalah perancangan mekanisme dan algoritma terhadap kapasitas yang terbatas atau penanganan trafik secara dinamis sumber daya sistem. Solusi sederhana atau statis penanganan trafik dapat dilakukan dengan cara seperti penambahan jumlah memori buffer, memperbesar *bandwidth* link, atau mengganti CPU yang cepat, dimana hal tersebut kurang tepat dan kurang efektif.

Penanganan kemacetan ada macam yaitu *TCP Congestion Control* [4] yang bekerja pada lapisan ke-empat OSI dan *Active Queue Management* (AQM) [5] yang bekerja pada lapisan ketiga OSI, keduanya mempunyai mekanisme berbeda. AQM diimplementasikan pada *intermediate network* yang berfungsi untuk penanganan *Buffer overflow*, dimana penyebab padatnya trafik pada sebuah node dan menjadi masalah dalam pengiriman paket data yang mengakibatkan memburuknya kinerja jaringan [6] yaitu diantaranya pada *buffer overflow* pada router. Sebelum ada AQM penanganan kiriman paket dengan menerapkan first in first out atau yang dikenal dengan mekanisme Droptail. *Random Early Detection* (RED) [7] algoritma adalah AQM yang cukup terkenal dan banyak digunakan pada router, yang dikembangkan untuk mengatasi kekurangan tradisional AQM yaitu metode FIFO/Drop Tail. Skema lain yang dikembangkan oleh peneliti menggunakan pengendali PID dengan optimasi parameternya [8], yang dikemudian diimplemetasikan ke NS2 dengan pengendali PI yang dioptimasi [9].

Manajemen antrian pada umumnya terdiri dari 3 dasar komponen [10], yaitu: *congestion measure* untuk mendeteksi mulai terjadinya kemacetan; *control function* untuk menentukan aksi yang diambil berdasarkan tingkat kemacetan; dan,

feedback mechanism untuk menginformasikan kemacetan ke pengirim agar menyesuaikan kecepatan transmisi untuk mengurangi kemacetan. Klasifikasi *congestion measure* diantaranya berdasarkan *queue based*, dan *control function* diklasifikasikan menjadi 3 yaitu heuristik, teori kendali dan optimisasi [11].

Untuk *robustness* yang pada jangkauan luas kondisi jaringan, sebaiknya memperhatikan 3 parameter jaringan: *network load* (N), *round-trip time* (RTT), dan *link capacity* (C). Dengan demikian, parameter pengendali sebaiknya ditala dengan baik, sehingga AQM dapat dengan cepat mengatur *queue length* ke nilai yang diharapkan dan menjaga kestabilan dengan sedikit osilasi. Nilai parameter yang tepat dapat menghilangkan *disturbance* atau error yang merugikan bagi sistem proses [12]. Dua kriteria kinerja jaringan yang diharapkan diantaranya adalah mendapatkan *queuing delay* yang pendek dan utilitas link yang tinggi. Itu akan berimplikasi pada aplikasi dengan parameter QoS jaringan yang baik, seperti *latency* dan *throughput* [13]. Oleh karena itu, kondisi tersebut merupakan tantangan untuk merancang dan mendapatkan formula AQM yang tepat.

Pada awalnya teori antrian yang umum digunakan pada jaringan yang diterapkan pada router, yaitu *first-in first-out* (FIFO) antrian [14], dan juga teori lainnya seperti prioritas antrian (PQ), dan rata-adil antrian (WFQ). AQM RED merupakan AQM pertama dikembangkan dan menjadi standar IETF, standar lainnya adalah AQM PIE [15] yang merupakan solusi untuk kondisi *bufferbloat*. Sebuah jaringan dengan algoritma RED mendeteksi kemacetan dan mengukur beban lalu lintas dalam jaringan dengan menggunakan antrian ukuran rata-rata. Kinerja AQM RED berdasarkan [16] adalah sebagai berikut; pertama, berdasarkan kepadatan trafik, parameternya perlu disetting untuk mendapatkan *throughput* yang baik. Kedua, probabilitas paket yang didrop menaik seiring kenaikan waktu delay. Ketiga, paket didrop ketika rata-rata probabilitas lebih besar dari ambang batas atas. Keempat, utilisasi link efektif untuk ukuran buffer yang kecil. Kelima, terjadi delay pada queue size yang besar.

Permasalahan bagaimana pengaruh trafik terhadap AQM RED sehingga diketahui kinerjanya, dan bagaimana perbandingan AQM RED terhadap apabila tidak diterapkannya AQM yaitu dengan TCP standar.

Pada penelitian ini dilakukan analisa unjuk kerja metode RED dengan mensimulasi jaringan dengan beberapa skenario seperti dengan adanya trafik FTP dan HTTP, sehingga dapat ditunjukkan karakteristik RED pada kondisi jaringan yang dinamis. Pengujian dilakukan menggunakan simulator NS2 dengan topologi jaringan *bottleneck* yang mekanisme AQM nya menggunakan metode RED, hasil simulasi dibandingkan dengan tanpa AQM atau mekanisme Droptail, skenario yang akan dilakukan adalah menggunakan berbagai layanan seperti, FTP dan HTTP. Dengan berbagai skenario trafik jaringan dapat dianalisa jumlah antrian pada *buffer* router, banyaknya paket yang dibuang, dan *delay* serta karakteristik lainnya.

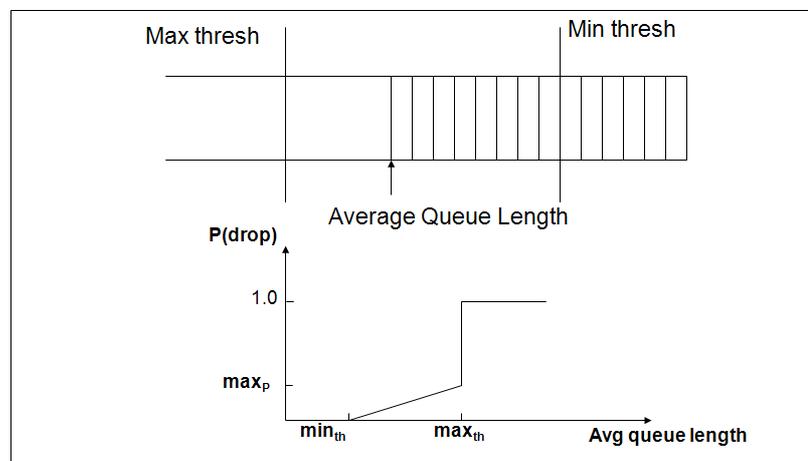
2. METODE

2.1 Mekanisme RED

Sebuah jaringan dengan algoritma RED mendeteksi kemacetan dan mengukur beban lalu lintas dalam jaringan dengan menggunakan antrian ukuran rata-rata. Bila ukuran antrian rata-rata lebih kecil dari *minth* batas minimum, tidak ada paket

dijatuhkan. Setelah ukuran antrian rata-rata melebihi batas minimum, router menganggap jaringan dalam kemacetan dan secara acak menjatuhkan paket-paket yang tiba dengan probabilitas penurunan diberikan. Probabilitas bahwa sebuah paket tiba di antrian RED dijatuhkan tergantung pada panjang antrian rata-rata, waktu yang telah berlalu sejak paket terakhir dijatuhkan, dan Drop maksimum max_p parameter probabilitas. Jika ukuran antrian rata-rata lebih besar dari max_{th} ambang batas maksimum, semua paket tiba akan dibuang. Menjatuhkan semua paket masuk dapat menyebabkan sinkronisasi global, yang biasanya diikuti dengan periode berkelanjutan utilitas link yang rendah.

Pada Gambar 3 adalah operasional metode RED saat menangani paket antrian, terlihat grafik paket yang akan didrop sesuai dengan parameter maksimum dan minimum panjang antrian yang telah ditentukan sebelumnya.



Gambar 1. Grafik Antrian pada RED

Algoritma RED dapat dijabarkan sebagai berikut:

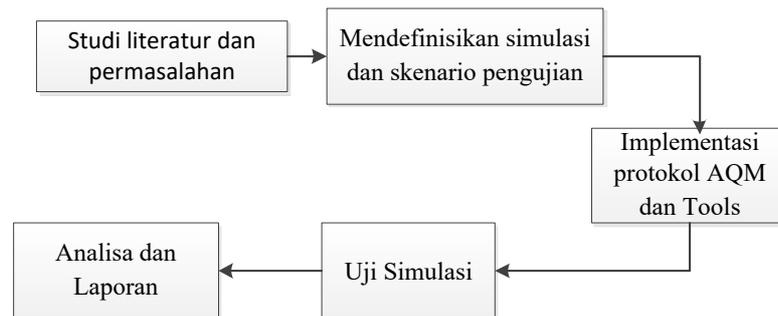
- Jika $ql < min_{th}$
Maka paket dilewatkan/diteruskan
- Jika $ql > max_{th}$
Maka buang semua paket
- Jika $ql > min_{th}$ dan $ql < max_{th}$
Maka paket didrop berdasarkan probabilitas p_d

Dimana,

- ql = Rata-rata panjang antrian
- p_d = Probabilitas drop

2.1 Tahapan Pengujian

Dalam pelaksanaan riset yang terlihat pada Gambar 2, kami akan melakukan berbagai kegiatan yang merupakan jalan riset. Riset diawali dengan mendefinisikan data simulasi. Selanjutnya selanjutnya dibuat skenario trafik yang akan diuji. Kemudian dilakukan pengujian dan evaluasi dengan membandingkan AQM Drop-tail.



Gambar 2. Diagram Flow Penelitian

Studi literatur dan permasalahan

Pada tahapan awal ini melakukan literatur review terkait metode AQM RED dan permasalahannya, sehingga dapat ditentukan langkah berikutnya untuk melakukan penelitian.

Mendefisikan simulasi skenario pengujian

Pada tahap ini akan disiapkan 4 buah senario yaitu terhadap layanan TCP/UDP dan beban trafik diantaranya yang pertama dengan menjalankan 30 aliran trafik FTP selama 80 detik, yang kedua dengan dengan penambahan 10 aliran FTP pada skenario pertama pada detik 80 dan 100, serta disimulasikan selama 160 detik, skenario ketiga mengganti 10 aliran FTP dengan UDP pada skenario kedua, skenario terakhir dengan mengubah nilai parameter q_weight . Diharapkan karakteristik respon AQM RED terhadap skenario-skenario tersebut ditunjukkan pada hasil simulasi dan dibandingkan dengan mekanisme Droptail.

Pada tahapan ini modul AQM RED dipastikan sudah ada dalam simulator dan berjalan baik, serta membuat topologi dan skenario skenario yang telah ditentukan. Script atau tools untuk mengolah data disiapkan untuk menganalisa karakteristik jaringan tersebut.

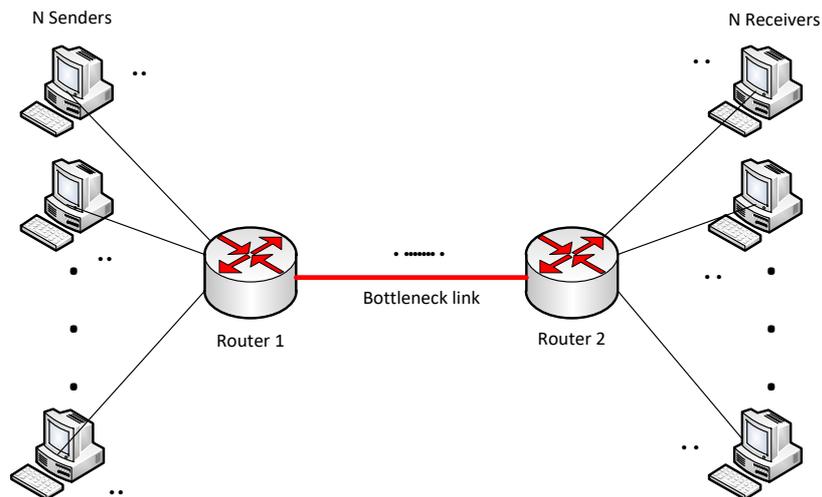
Uji simulasi

Penentuan topologi jaringan dengan link bottleneck serta konfigurasi jaringan didefinisikan pada tahapan awal. Metode *Modeling and Simulation cycle* yang diterapkan dan dimodelkan pada Simulator NS2 dengan kondisi sebagai berikut:

- Diagram/Topologi jaringan dirancang sesuai dengan Gambar 3.
- Dua buah Router yang terhubung ke host client atau server, Router 1 diimplementasikan algoritma RED

- Link 10 Mbps antara klien/server dan Router, dan Link bottleneck diantara kedua router sebesar 10 Mbps.
- 30 buah Workstation dengan layanan FTP klien
- 30 buah Server dengan FTP Application Server.

Perancangan topologi jaringan *bottleneck* di buat sesuai pada Gambar 3. Pada simulator jaringan NS2, jumlah client dan server disesuaikan dengan skenario yang akan dijalankan



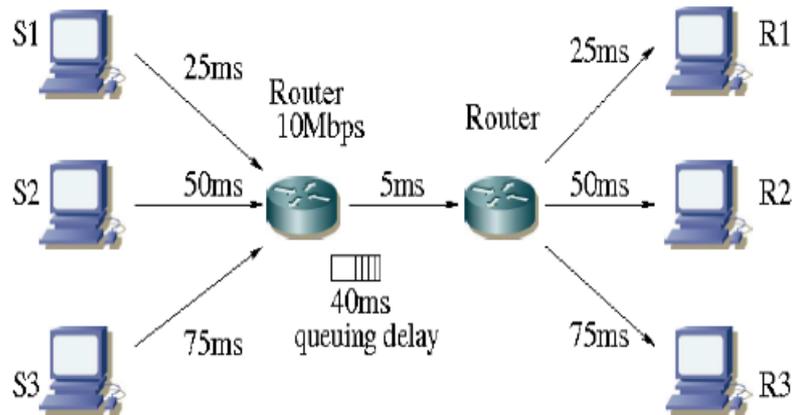
Gambar 3. Topologi Jaringan *Bottleneck*

Analisa

AQM RED dan Droptail dijalankan simulasinya sesuai usulan rancangan tahapan sebelumnya. Empat buah skenario yang diuji terhadap mekanisme kedua mekanisme dibandingkan dan dianalisa kinerja dan karakteristik responnya. Data mentah hasil simulasi selanjutnya diolah dengan script untuk menghitung karakteristik dan indikator jaringan.

3. HASIL DAN PEMBAHASAN

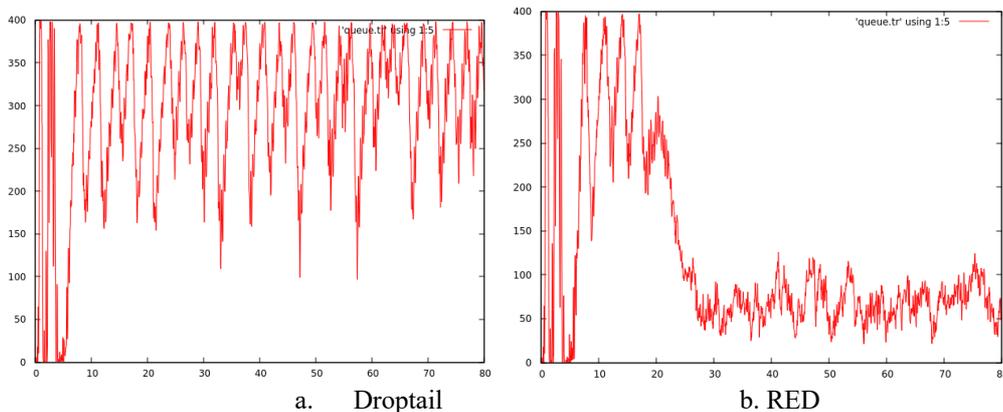
Pada bab ini memperlihatkan beberapa skenario simulasi yang dilakukan pada simulator jaringan NS-2 (release 2.25). Pengujian dengan melakukan pengirim paket dari sejumlah host pengirim ke sejumlah host penerima melalui link bottleneck yang ditunjukkan pada Gambar 4. Fenomena kemacetan jaringan terjadi pada router pertama, dan diatasi dengan mengimplementasikan AQM untuk mengatur *queue length* router sebesar 100 paket, *buffer size* pada router tersebut disetting 400 paket. Waktu *propagation* pada link bottleneck sebesar 5 ms dengan *bandwidth* sebesar 10 Mbps dimana dengan besar paket sebesar 500 bytes akan dapat mengirim paket maksimum 2500 paket per detik, nilai ini didapat dengan perhitungan besar *bandwidth* dibagi dengan besar paket dalam bit. Dengan kondisi tersebut diharapkan *equilibrium queueing delay* sebesar 40 ms, nilai ini didapat dengan *queue length* 100 dibagi dengan jumlah paket maksimum yang dikirim yaitu sebesar 2500 paket per detik.



Gambar 4. Topologi simulasi jaringan

Ada tiga tipe host pengirim yang masing-masing mempunyai delay propagasi terlihat pada Gambar 4, setiap tipe host mempunyai 10 host yang menggunakan protokol TCP/Reno dalam pengiriman paket. Dengan parameter RED disetting $\text{minth} = 50$, $\text{maxth} = 300$, $\text{wq} = 0.1$.

Skenario pengujian pertama adalah dengan mengalirkan flow TCP dari host pengirim ke host penerima, yang dibandingkan dengan tanpa AQM atau mekanisme Droptail yang ditunjukkan pada Gambar 5a. Kinerja RED diperlihatkan pada Gambar 5b yaitu perubahan *queue length* terhadap waktu dimana RED mendekati target 100 pada 27 detik, sedangkan Droptail yang ditunjukkan Gambar 8a mengalami *bufferbloat* yaitu terjadinya kapasitas penuh sehingga terjadinya fluktuasi karena *synchronization problem*.



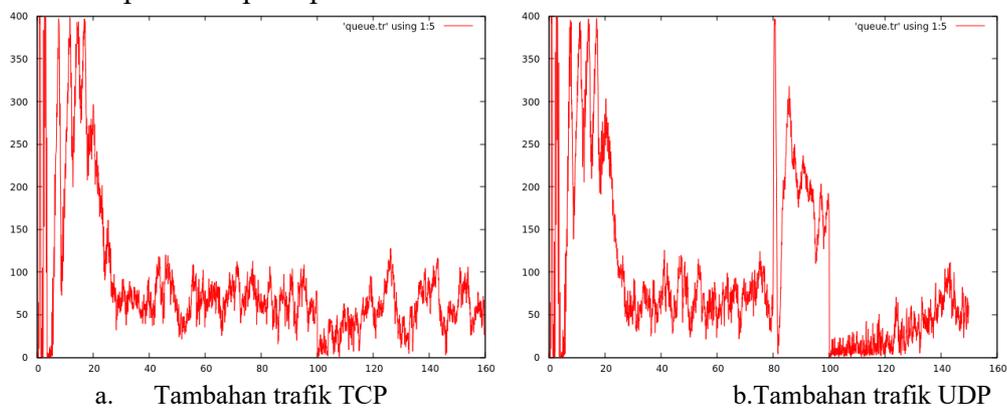
Gambar 5. Perubahan queue length AQM Droptail dan AQM RED

Rata-rata queue RED adalah 118 paket sedangkan Droptail 292 pada Tabel 1, sehingga AQM RED terbukti dapat mengendalikan queue length dengan target yang diinginkan dimana dalam penelitian ini targetnya 100 paket. Rata2 queuing delay pada buffer sebesar 52 detik untuk RED yang mendekati titik equilibrium sebesar 40 detik, dibandingkan droptail yang sebesar 128 detik. Rasio Packet loss pada RED lebih tinggi dibandingkan dengan DropTail yaitu 1.05 dan 1.12 yang ditunjukkan pada Tabel 1.

Tabel 1. Kinerja Perbandingan skenario pertama

Kinerja	Satuan	RED	Droptail
Average queue	Paket	118	292
Standar deviasi queue	Paket	99.48	90
Average queuing delay	ms	52	128
Packet loss	Paket	1924	2069
	Rasio	1.05	1.12

Pengujian kedua dengan menambahkan trafik TCP dan UDP pada detik 80 sampai dengan 100 pada skenario pertama AQM RED dan simulasi dijalankan selama 160 detik. Terlihat pada Gambar 6a dengan penambahan 10 aliran TCP, queue length tetap terjaga antara detik ke 80 dan 100. Sedangkan penambahan 10 aliran UDP mengalami lonjakan buffer size pada detik ke 80 100 ditunjukkan pada Gambar 6b. Ini menandakan bahwa AQM ini masih kurang untuk menghandel trafik unrenspovise seperti paket UDP.



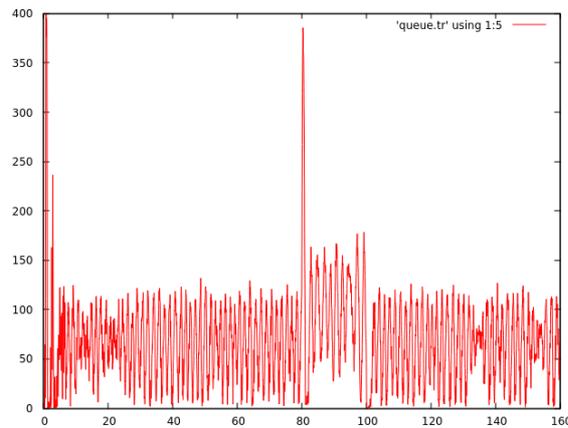
Gambar 6. RED dengan tambahan Trafik TCP dan UDP pada detik 80 sampai 100.

Pada kedua skenario TCP dan UDP pada RED terlihat pada Tabel 2 pengaturan dengan pada kondisi paket TCP lebih baik dibandingkan UDP, yaitu dengan kinerja rata-rata queue, standar deviasi, rata queueing delay dan packet loss TCP lebih kecil dibandingkan dengan UDP.

Tabel 2. Kinerja Perbandingan skenario kedua

Kinerja	Satuan	TCP	UDP
Average queue	Paket	86	98
Standar deviasi queue	Paket	77.6	92.44
Average queuing delay	ms	38	43
Packet loss	Paket	2677	3337
	Rasio	0.72	0.91

Simulasi terakhir atau ketiga ini dengan skenario perubahan setting nilai weghting queue q_weight sebesar 0.002 dengan skenario kedua UDP yang hasil uji cobanya ditunjukkan pada Gambar 7. Perubahan nilai q_weight dari 5.599E-6 pada skenario pertaman disetting ke 0.002 terjadi osilasi perubahanan queue lenght pada buffer yang mengakibatkan kinerja pengiriman paket data tidak stabil.



Gambar 7. RED dengan $q_weight=0.002$

Dari keseluruhan skenario pengujian dapat disimpulkan bahwa RED sudah dapat memperbaiki trafik pada target yang diinginkan dan juga merupakan solusi terhadap kondisi tanpa menggunakan AQM yaitu dengan Droptail.

4. KESIMPULAN

Pada penelitian ini mempresentasikan perancangan AQM untuk kendali kemacetan pada router, dengan simulasi pengirim paket TCP dari host pengirim ke host penerima melalui topologi jaringan bottleneck. Mekanisme pengaturan kemacetan dengan mengimplementasikan AQM RED dapat mengantisipasi kelemahan Droptail yaitu terjadi *buffer bloat* dan *synchronisasi problem*, serta dapat mengatur target *queue length* yang diharapkan, dimana pada penelitian ini rata-rata queuing delay RED mendekati titik *equilibrium* 40 ms. Walaupun demikian AQM RED perlu disetting parameter dengan tepat agar dapat kinerja yang baik

REFERENSI

- [1] G. Miranda, D. F. Macedo, and J. M. Marquez-Barja, "Estimating Video on Demand QoE from Network QoS through ICMP Probes," *IEEE Trans. Netw. Serv. Manag.*, 2021.
- [2] A. Lad, S. Butala, and P. Bide, "A comparative analysis of over-the-top platforms: Amazon Prime Video and Netflix," in *International Conference on Communication and Intelligent Systems*, 2019, pp. 283–299.
- [3] M. Fajri, "Studi Network Congestion Dengan TCP Tahoe," *J. Ilm. FIFO*, vol. VIII/2/November/2021, 2021.
- [4] Y. Lu, X. Fan, and L. Qian, "Dynamic ECN marking threshold algorithm for TCP congestion control in data center networks," *Comput. Commun.*, vol. 129, pp. 197–208, 2018.
- [5] O. Lamrabet, N. El Fezazi, F. El Haoussi, E. H. Tissir, and T. Alvarez, "Congestion Control in TCP/IP Routers Based on Sampled-Data Systems Theory," *J. Control. Autom. Electr. Syst.*, vol. 31, no. 3, pp. 588–596, 2020.
- [6] Y. N. Reddy and P. V. S. Srinivas, "A Routing Delay Predication Based on Packet Loss and Explicit Delay Acknowledgement for Congestion Control in MANET," *Int. J. Commun. Networks Inf. Secur.*, vol. 10, no. 3, p. 447, 2018.
- [7] C. W. Feng, L. F. Huang, C. Xu, and Y. C. Chang, "Congestion Control Scheme Performance Analysis Based on Nonlinear RED," *IEEE Syst. J.*, vol. 11, no. 4, pp. 2247–2254, 2017.
- [8] M. Fajri and K. Ramli, "Optimizing PID TCP/AQM using nelder-mead simplex

- approach,” in *Proceedings of the 3rd International Conference on Communication and Information Processing*, 2017, pp. 292–295.
- [9] M. Fajri and K. Ramli, “Design of Network Traffic Congestion Controller with PI AQM Based on ITAE Index,” *Int. J. Electron. Telecommun.*, pp. 715–721, 2020.
- [10] G. Abbas, S. Member, Z. Halim, and Z. H. Abbas, “Fairness-Driven Queue Management : A Survey and Taxonomy,” vol. 18, no. 1, pp. 324–367, 2016.
- [11] R. Adams, “Active Queue Management : A Survey,” vol. 15, no. 3, pp. 1425–1476, 2013.
- [12] M. Mohebbi and M. Hashemi, “Designing a 2-Degree of Freedom Model of an Unbalanced Engine and Reducing its Vibrations by Active Control,” *Int. J. Technol.*, vol. 8, no. 5, pp. 858–866, 2017.
- [13] P. Rukmani and R. Ganesan, “Enhanced Low Latency Queuing Algorithm for Real Time Applications in Wireless Networks,” *Int. J. Technol.*, vol. 7, no. 4, pp. 663–672, 2016.
- [14] Y. Irawan and N. Surantha, “Performance Evaluation of Queue Algorithms for Video-on-Demand Application,” in *2020 International Conference on Information Management and Technology (ICIMTech)*, 2020, pp. 966–971.
- [15] S. D. Patil and M. P. Tahiliani, “Minstrel PIE: Curtailing queue delay in unresponsive traffic environments,” *Comput. Commun.*, vol. 139, pp. 16–31, 2019.
- [16] D. Kumhar, “Performance Analysis of AQM Algorithms in Network Congestion Control,” *Int. J. Adv. Res. Comput. Sci.*, vol. 8, no. 3, 2017.