

ANALISA PERBANDINGAN PERFORMANSI SKEMA MULTI-LEVEL RED UNTUK DIFFERENTIATED SERVICES DI INTERNET

Dhani Arvianto¹, Hafidudin², Arif Rudiana³

^{1,2}Jurusan Teknik Elektro, Sekolah Tinggi Teknologi Telkom

³Training Center PT Telkom Indonesia

ABSTRAKSI

Perkembangan internet saat ini begitu pesat seiring dengan banyaknya user dan aplikasi-aplikasi yang berjalan di atasnya. Jaringan IP tradisional menawarkan user layanan *best effort*. Dalam layanan *best effort*, semua paket tidak ada yang dibedakan dan akan diberikan perlakuan *forwarding* yang sama. Hal ini dapat menyebabkan adanya beban bagi jaringan yang memiliki *bandwidth* dan *buffer space* yang terbatas, sehingga dapat menghasilkan kongesti. Dengan adanya mekanisme *Quality of Service (QoS)*, jaringan IP menyediakan suatu diskriminasi pada layanan. *Differentiated Service (DiffServ)* merupakan salah satu mekanisme yang digunakan untuk meningkatkan *Quality of Service (QoS)* di jaringan IP. *DiffServ* adalah arsitektur IP *QoS* berdasarkan penandaan pada paket yang memungkinkan paket untuk dapat diprioritaskan sesuai dengan keperluan dari user. *Multi-level Random Early detection (MRED)* kemudian diperkenalkan sebagai suatu skema *AQM* alternatif yang mendukung implementasi dari *DiffServ*.

Dalam Penelitian ini diperkenalkan tiga skema *MRED*, yaitu *RED* dengan *IN/OUT Coupled (RIO-C)*, *RED* dengan *IN/OUT De-Coupled (RIO-D)* dan *Weighted RED (WRED)*. Simulasi menggunakan *NS-2* dilakukan untuk membandingkan kinerja masing-masing skema tersebut. Parameter-parameter performansi yang diujikan antara lain *packet loss*, *throughput* dan *queue delay*.

Hasil simulasi menunjukkan bahwa kinerja *RIO* lebih baik dibandingkan *WRED* dalam hal melindungi paket yang memiliki prioritas yang lebih tinggi pada kondisi load yang dinamis maupun dengan adanya penambahan sumber. *RIO-D* menghasilkan *throughput* terbesar diantara dua skema lainnya untuk tiap skenario namun dilain pihak juga memiliki panjang antrian yang tinggi sehingga *delay* antrian juga tinggi.

Kata kunci: *Kongesti, Differentiated Service, AF-PHB, RIO-C, RIO-D, WRED*

1. PENDAHULUAN

Jaringan IP tradisional menawarkan user layanan *best effort*. Dalam layanan *best effort*, semua paket tidak ada yang dibedakan dan akan diberikan perlakuan *forwarding* yang sama. Meningkatnya penggunaan dan popularitas dari internet maka meningkat pula perkembangan aplikasi-aplikasi baru yang muncul seperti *voice*, *video* dan *web*. Peningkatan ini menyebabkan adanya beban yang signifikan pada jaringan yang memiliki *bandwidth* dan *buffer space* yang terbatas. Hal inilah yang memicu terjadinya kongesti. Adanya *QoS* pada jaringan IP memberikan suatu diskriminasi pada tiap layanan sehingga operator dapat membedakan perlakuan yang diterima pada paket user sesuai dengan yang diinginkan.

Differentiated Service (DiffServ) merupakan suatu arsitektur IP *QoS* yang bekerja berdasarkan penandaan pada paket yang memungkinkan paket untuk diprioritaskan sesuai dengan keperluan user. *Assured Forwarding (AF)* diajukan sebagai salah satu skema yang mendukung implementasi dari *DiffServ*. *AF* menyediakan perlakuan yang berbeda-beda pada trafik dengan membuang lebih banyak paket yang memiliki prioritas yang rendah dibanding paket berprioritas tinggi saat terjadinya kongesti.

Mekanisme *dropping* paket di *router* dalam *AF-PHB (Per-Hop Behaviour)* pada umumnya menggunakan salah satu teknik *Active Queue Management*, yaitu *RED*. *Multi-level RED (MRED)* mengkonfigurasi parameter *RED* yang berbeda untuk tiap *drop precedence*. Salah satu skemanya adalah *RED with IN and OUT (RIO)*, merupakan

manajemen buffer yang membagi paket menjadi *IN profile* dan *OUT profile* dan menandai paket sesuai dengan *policy* yang terdapat di *edge router*. Dimana *IN profile* memiliki probabilitas *dropping* yang lebih rendah dibandingkan *OUT profile* saat terindikasi terjadinya kongesti.

2. LANDASAN TEORI

2.1 Differentiated Service

Differentiated Service (DiffServ) merupakan salah satu metode yang diusulkan oleh IETF dalam menangani masalah *Quality of Service (QoS)* pada jaringan IP di Internet. Arsitektur *DiffServ* memungkinkan adanya suatu mekanisme yang sederhana pada sisi *Core* dan yang lebih kompleks pada sisi *Edge*. *DiffServ* membagi layanan menjadi kelas-kelas yang kemudian diberikan suatu aturan (*Policy*) sesuai yang diinginkan oleh pelanggan.

2.1.1 Arsitektur DiffServ

Arsitektur *DiffServ* didefinisikan dalam RFC 2475 dimana terdiri dari dua komponen utama, yaitu *traffic conditioning* dan *per-hop behaviors*. Arsitektur *DiffServ* memiliki 3 komponen:

1. *Policy* dan *resource manager*
Membuat kebijakan-kebijakan dan mendistribusikannya kepada *DiffServ* router. Sebuah kebijakan menentukan tingkatan layanan mana yang diberikan untuk suatu paket di dalam jaringan. Penugasan ini akan bergantung pada kelakuan dari *flow* sumber tersebut (*average rate-nya* dan *burstiness-nya*).

2. *Edge routers*

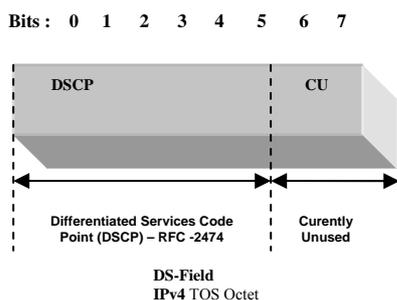
Bertanggung-jawab untuk menandai paket dengan sebuah *code point* sesuai dengan kebijakan yang telah dispesifikasikan sebelumnya oleh administrator jaringan yang merefleksikan level layanan yang diinginkan. Untuk melakukannya *edge router* mengukur parameter input trafik dari setiap *flow*.

3. *Core routers*

Core router bertugas memeriksa paket datang yang sebelumnya telah ditandai dengan *code point* oleh *edge router*. Kemudian *core router* meneruskan (*forwarding*) paket datang sesuai dengan tanda yang telah diberikan (menyediakan reaksi atas tanda yang diberikan *edge router* pada paket).

2.1.2 DiffServ Field

DiffServ menyediakan diferensiasi layanan dengan membagi trafik berdasar kelas-kelas pada *edge router* dengan menggunakan *DSCP field* atau *IP precedence field* untuk mengidentifikasi kelas-kelas layanan. Pada *DiffServ*, trafik dibagi kedalam beberapa kelas dan masing-masing ditangani secara berbeda khususnya ketika jumlah *resource* jaringan terbatas. *Header IPv4* mengandung byte ToS dan



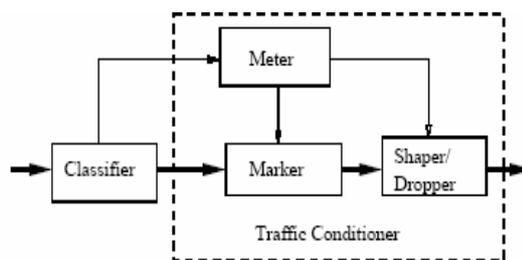
aplikasi dapat men-set 3 bit di sebelah kiri (*IP Precedence*) untuk menunjukkan kelas layanan. Kemudian *DiffServ* menamai ulang byte ToS menjadi *Differentiated Services field (DS field)*, seperti yang ditunjukkan pada Gambar 1. berikut ini.

Gambar 1. Hubungan struktur header IP dan DS field

Arsitektur *DiffServ* menyediakan QoS dengan membagi trafik menjadi beberapa kategori, Menandai tiap paket dengan sebuah kode poin (*code point*) yang mengindikasikan kategorinya, dan menjadwal paket sesuai dengan *code point* tersebut. *DiffServ* dapat mendukung empat kelas trafik, setiap trafiknya memiliki tiga *dropping precedences* (hak di-drop terlebih dahulu) membolehkan perlakuan yang berbeda dari trafik dalam satu kelas. Paket dalam satu kelas trafik diantrikan ke satu antrian fisik RED yang berhubungan, dimana berisi tiga antrian *virtual* (satu untuk tiap *drop precedence*).

2.1.3 Traffic Conditioning

Merupakan mekanisme fungsi kontrol (Gambar 2) yang mengatur suatu trafik pada saat memasuki domain *DiffServ*.



Gambar 2. Komponen dari Traffic Conditioner

Ketika suatu paket tiba di *edge router*, paket akan diperiksa oleh komponen *classifier* untuk menentukan milik kumpulan mana paket tersebut. Komponen *meter* akan meng-*update* semua *variable* yang tersedia kemudian *marker* akan menentukan *code point* yang sesuai dengan *Policy*, kemudian paket diantrikan. *Shaper/Dropper* memberi keputusan akan men-*delay* atau men-*drop* paket sesuai dengan profil yang telah ditentukan sebelumnya.

2.1.4 Per-Hop Behaviors (PHB)

PHB didefinisikan sebagai suatu mekanisme *forwarding* paket yang dilakukan tiap node *DiffServ*. Sebuah node mengalokasikan *resources* dengan *Behaviour Aggregates (BA)* yang berbeda dengan node yang lainnya. Mekanisme alokasi *resource hop-by-hop* ini merupakan dasar dari prinsip *DiffServ*.

PHB digunakan untuk mengidentifikasi perlakuan yang akan diberikan pada sebuah *flow* khusus (atau kumpulan jika *router* adalah *core router*). Pada PHB terdapat proses pengaturan antrian (*queuing*) dan mekanisme *dropping* paket yang dilakukan di setiap hop dalam jaringan.

a. Assured Forwarding (AF) PHB

Assured Forwarding (AF) PHB merupakan suatu metode dari domain *DiffServ* untuk menawarkan jaminan dari level *forwarding* yang berbeda untuk tiap paket IP yang diterima. Objektivitasnya adalah untuk mengirimkan paket sampai di tujuannya, maka dari itu *delay* dan *jitter* tidak sepenting *packet loss*.

Spesifikasi AF yang ada sekarang ini menyediakan pengantaran paket-paket dalam 4 kelas yang masing-masing terdiri dari 3 *drop precedence*. Paket-paket yang berada dalam satu kelas harus di-*forward* terpisah dari paket-paket yang terdapat pada kelas AF yang lainnya.

Dalam tiap kelas AF, paket IP ditandai dengan satu dari tiga kemungkinan nilai *Drop Precedence* yang menentukan pentingnya paket tersebut dalam suatu kelas. Dalam kasus kongesti, paket dengan *drop precedence* yang lebih rendah terlindungi dari kehilangan (*packet lost*) dengan lebih memilih membuang paket dengan *drop precedence* yang lebih tinggi. Level *drop precedence* ini adalah (Tabel 1).

Tabel 1. DSCP Assured Forwarding PHB [2]

Drop precedence	Class 1	Class 2	Class 3	Class 4
Low	(AF11)	(AF21)	(AF31)	(AF41)
	001010	010010	011010	100010
Medium	(AF12)	(AF22)	(AF32)	(AF42)
	001100	010100	011100	100100
High	(AF13)	(AF23)	(AF33)	(AF43)
	001110	010110	011110	100110

2.2 Active Queue Management (AQM)

Drop tail merupakan teknik tradisional yang mengatur panjang antrian di router dengan cara memberi panjang maksimum (dalam paket) untuk setiap antrian, menerima semua paket sampai panjang antrian maksimum penuh kemudian membuang paket (*drop*) yang datang berikutnya sampai panjang antrian turun karena sudah dikirimkan. Metode ini memang sederhana tetapi memiliki banyak kelemahan karena dapat mengakibatkan *lockout*, *full queue*, *global synchronization* dan diikuti dengan *link utilization* yang rendah dan mengurangi *throughput*.

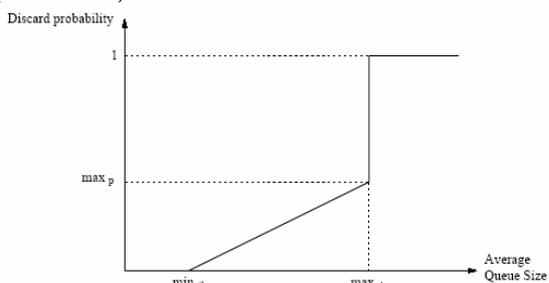
Active Queue Management (AQM) merupakan pendekatan proaktif untuk menangani kongesti di router. AQM akan memberikan sinyal kongesti lebih awal sebelum terjadinya kongesti sehingga source bisa bereaksi lebih cepat terhadap *feedback* dari jaringan

Tujuan utama dari AQM yaitu untuk memberikan informasi tentang adanya kongesti kepada sumber TCP sehingga sumber dapat menyesuaikan kecepatan pengiriman paket ke jaringan sehingga jaringan dapat terpakai dengan efektif dan tetap menjaga panjang antrian untuk menjamin delay yang rendah.

2.3 Random Early Detection (RED)

Salah satu bentuk AQM yang diperkenalkan oleh *Internet Engineering Task Force (IETF)* untuk aplikasi jaringan adalah *Random Early Detection (RED)*. RED merupakan teknik manajemen antrian aktif (AQM) yang digunakan untuk jaringan IP berskala besar yang akan diterapkan pada *IP Next Generation Network*.

RED menggunakan suatu paket *drop profile* untuk mengontrol keagresifan proses pembuangan paket. *Drop profile* mendefinisikan suatu tingkat probabilitas *dropping* paket melalui tingkat antrian. Algoritma RED menggunakan rata-rata panjang antrian untuk menentukan kapan paket ditandai (Gambar 3.).



Gambar 3. Grafik Probabilitas RED

RED akan melakukan *dropping* suatu paket yang datang dengan menghitung probabilitas *dropping*-nya secara dinamis berdasarkan nilai rata-rata antriannya. Nilai antrian rata-ratanya adalah:

$$avg(t) = (1-\alpha)avg(t-1) + \alpha.q, \text{ dengan } \alpha = 0.02 \quad (2.1)$$

- Jika antrian rata-rata < batas minimum (*minthresh*), $avg < min_{th}$, maka RED akan melewatkan paket yang datang ke *bottleneck link* (diantrikan).
- Jika antrian rata-rata berada diantara batas minimum (*minthresh*) dan batas maksimum (*maxthresh*), $min_{th} < avg < max_{th}$, maka RED akan membuang paket yang datang ke *bottleneck link* berdasarkan probabilitas p_a

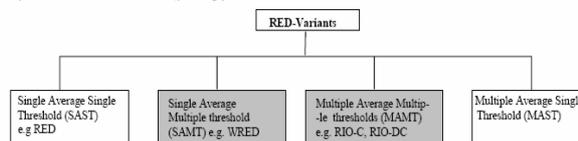
$$p_a = \frac{\max p * (avg - \min_{th})}{(\max_{th} - \min_{th})} \quad (2.2)$$

- Dengan $\max p = 0.1$ merupakan batas maksimum probabilitas *dropping*.
- Jika antrian rata-rata > batas maksimum (*minthresh*), $avg > max_{th}$, maka RED akan membuang paket yang datang ke *bottleneck link*.

Namun RED memiliki banyak kelemahan antara lain yaitu performansinya sangat bergantung pada banyaknya aliran TCP dan sulit dalam konfigurasi parameternya.

2.4 Multilevel RED (MRED)

MRED merupakan istilah umum yang digunakan untuk mendeskripsikan suatu skema dimana untuk paket dengan *drop precedence* yang berbeda, perlu untuk dihitung secara independen. Hal ini diperoleh dengan mempertahankan *multiple set* dari *threshold* RED, satu untuk tiap *drop precedence*, dan rata-rata antrian yang digunakan dalam keputusan drop dapat dihitung menggunakan sejumlah skema yang berbeda.

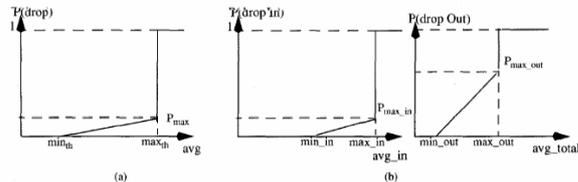


Gambar 4. Varian dari RED [13]

2.4.1 RED With In And Out (RIO)

RIO menggunakan mekanisme yang sama dengan RED tapi dikonfigurasi dengan dua set parameter, satu untuk paket masuk (IN profile) dan satu untuk paket keluar (OUT profile). Pada setiap kedatangan paket, router mengecek apakah paket tersebut ditandai sebagai IN atau OUT. Jika merupakan sebuah paket IN, router menghitung *avg_in* (rata-rata antrian untuk paket IN) dan jika merupakan paket OUT, router menghitung *avg_total*, ukuran rata-rata total antrian untuk semua (keduanya IN dan OUT) paket yang datang. Probabilitas *dropping* dari sebuah paket IN tergantung pada *avg_in*, dan probabilitas *dropping* dari sebuah paket OUT tergantung pada *avg_total*.

Pemilihan parameter terhadap paket OUT ($min_in, max_in, Pmax_in$), dan ($min_out, max_out, Pmax_out$) di RIO diciptakan dengan secara hati-hati. Seperti diilustrasikan di dua gambar dibawah ini (Gambar 5.), sebuah router RIO lebih agresif dalam dropping paket OUT.



Gambar 5. (a) Algoritma RED dan (b) Algoritma RIO [17]

Pertama, router men-drop paket OUT lebih dulu dibanding paket IN; hal ini dilakukan dengan memilih min_out yang lebih kecil dari min_in . Kedua, didalam fase *congestion avoidance*, router men-drop paket OUT dengan probabilitas yang lebih tinggi dengan mengatur $Pmax_out$ lebih besar dari $Pmax_in$ ($Pmax_out > Pmax_in$). Ketiga, router melalui fase *congestion control* untuk paket OUT lebih dahulu dibandingkan untuk paket IN dengan memilih $max_out < max_in$. Intinya, RIO men-drop paket out terlebih dahulu ketika router mendeteksi kongesti yang baru saja terjadi., dan men-drop semua paket OUT jika kongesti tidak berhenti. Hanya sebagai pilihan terakhir, terjadi ketika router dipenuhi oleh paket IN, router men-drop paket IN, dengan harapan untuk mengontrol kongesti. Didalam jaringan yang telah ditentukan secara baik peristiwa ini tidak akan terjadi.

```

For each packet arrival
  if it is an In packet
    calculate the average In queue size avg_in;
    calculate the average queue size avg_total;

  If it is an In packet
    if min_in < avg_in < max_in
      calculate probability P_in;
      with probability P_in, drop this packet;
    else if max_in < avg_in
      drop this packet.
  if this is an Out packet
    if min_out < avg_total < max_out
      calculate probability P_out;
      with probability P_out, drop this packet;
    else if max_out < avg_total
      drop this packet.
    
```

Gambar 6. Algoritma RIO

Pilihan untuk menggunakan avg_total , ukuran rata-rata total untuk menentukan probabilitas *dropping* dari sebuah paket OUT, adalah tidak terlalu kentara. Tidak seperti paket IN, dimana jaringan dapat secara benar menetapkannya, paket OUT mewakili trafik yang oportunistik, dan tidak ada indikasi yang jelas mengenai jumlah paket OUT yang pantas. Jika kita telah menggunakan rata-rata antrian paket OUT untuk mengontrol dropping dari paket OUT, hal ini tidak akan dapat menutupi kasus dimana antrian total berkembang akibat kedatangan paket IN. Dengan menggunakan avg_total , ukuran rata-rata total antrian, router dapat mempertahankan panjang antrian agar tetap pendek dan *throughput* yang tinggi bagaimanapun jenis campuran trafiknya.

Mungkin saja salah satunya dapat memperoleh kontrol yang lebih responsif dari paket OUT dengan mengubah parameter dropping untuk bergantung pada kedua ukuran rata-rata antrian in (avg_in) dan ukuran rata-rata total antrian (avg_total).

Terdapat beberapa skema sebagai bagian dari varian RED (MRED) yang diajukan untuk penghitungan rata-rata antrian sesuai dengan drop precedence yang telah ditetapkan sebagai profil dari paket. Varian tersebut adalah RIO-C, RIO-D, dan WRED.

a. RIO-C (Coupled)

Probabilitas *dropping* paket dengan prioritas rendah (dinamakan “paket out-of-profile”) didasari oleh ukuran panjang rata-rata dari semua antrian *virtua*. Sebaliknya, probabilitas *dropping* paket dengan prioritas yang tinggi (“in-profile”) didasari hanya oleh berat panjang rata-rata dari antrian *virtual*-nya sendiri.

b. RIO-D (De-coupled)

Sebaliknya, dalam RIO-D (*de-coupled*), probabilitas *dropping* tiap paket didasari oleh ukuran dari antrian *virtual*-nya.

2.4.2 WRED (Weighted RED)

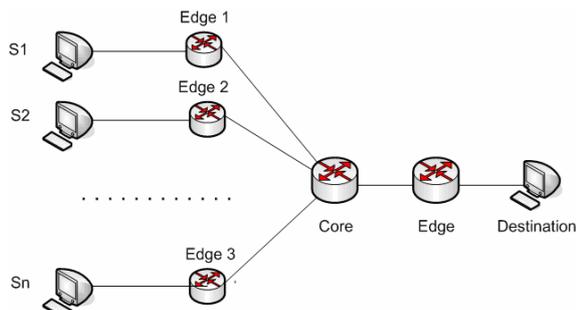
Probabilitas *dropping* berdasarkan pada panjang tunggal antrian. WRED Mengijinkan kita untuk menggunakan beberapa profil *dropping* RED untuk tipe-tipe trafik yang berbeda. Kemampuan untuk menetapkan profil *dropping* yang berbeda untuk antrian yang berbeda, atau untuk tipe-tipe trafik yang berbeda dalam antrian yang sama, menyediakan suatu dinamika kontrol yang lebih baik dibandingkan RED klasik.

Sebagai contoh: asumsikan bahwa manajemen memori antrian membolehkan kita untuk menetapkan sebuah 2 layer *drop precedence* dalam suatu antrian. Hal ini mengijinkan kita untuk menggunakan profil *dropping* RED yang kurang agresif untuk sebagian tipe paket dan profil drop RED yang lebih agresif untuk tipe paket yang lain yang diberikan level kongesti yang sama.

3. PERANCANGAN MODEL SIMULASI

3.1 Topologi Jaringan

Adapun pemodelan sistem secara umum pada Penelitian ini dapat dimodelkan seperti gambar di bawah ini.



Gambar 7. Desain sistem secara umum [19]

3.2 Konfigurasi Sistem

Terdapat node sumber sebanyak-n, dan tiap node-nya membangkitkan aliran TCP. Keterangan yang lebih jelasnya:

Semua link berkapasitas 10 Mbps dan delay 5 ms kecuali link antara *core node* dan *edge node* yang terhubung ke node sumber. Antrian terjadi hanya pada *bottleneck link*, dalam kasus ini *bottleneck* dialami pada link antara *core node* dengan *edge node* yang terhubung dengan node tujuan. Batas antrian adalah 100 paket. Suatu skema manajemen antrian, yaitu *Multi-level RED (MRED)* digunakan pada *core node* tersebut. *Policy* yang diterapkan di edge router adalah TSW2CM, dimana akan menggunakan dua tingkatan prioritas (*drop precedence*). Tujuan dari penggunaan parameter tersebut tidak untuk mendapatkan kinerja yang optimal, tetapi adalah untuk menciptakan kondisi yang memungkinkan untuk mempelajari efek dari *DiffServ* dalam mengurangi probabilitas *loss* dan akibatnya terhadap performansi jaringan (*delay, throughput*).

Dalam mendesain sistem, digunakan beberapa asumsi yang akan digunakan dalam melakukan simulasi. Berikut adalah asumsi-asumsi yang dipakai dalam perancangan sistem:

- TCP yang digunakan adalah TCP SACK pada sisi pengirim (sumber), dan TCP Sink pada sisi penerima.
- Aplikasi yang dipakai adalah FTP, yang merupakan aplikasi yang sering dipakai untuk pengiriman data dari internet kepada pengguna internet untuk mengirimkan paket data.
- Aplikasi CBR dibangkitkan dan untuk menghasilkan kongesti di *bottleneck link*.
- Selalu ada data yang dikirim hingga waktu simulasi yang ditentukan selesai, yaitu 50 detik
- Analisis didasarkan pada manajemen *buffer* pada *single bottleneck link*
- Antrian yang digunakan untuk *bottleneck link* adalah *Multi-level RED (MRED)*
- Parameter performansi berupa *throughput, queue delay, dan packet loss*.

4. ANALISIS HASIL SIMULASI

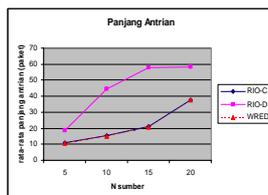
4.1 Analisis Performansi

4.1.1 Analisis Pengaruh Perubahan Sumber

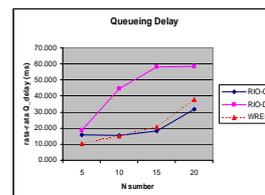
Tabel 2. Data *Throughput* Terhadap Perubahan N sumber

N Sumber	Throughput	Index Fairness	MRED
5	401.8228	0.988631006	RIO-C
	494.4914	0.996781866	RIO-D
	355.2026	0.976466518	WRED
10	82.980	0.915239996	RIO-C
	98.213	0.966146606	RIO-D
	78.500	0.885150038	WRED
15	9.418	0.761299412	RIO-C
	11.162	0.806860396	RIO-D
	7.013	0.794008736	WRED
20	3.631	0.855971324	RIO-C
	4.736	0.792927202	RIO-D
	2.939	0.757034173	WRED

Pada tabel 2 diatas dapat dilihat bahwa skema RIO-D memiliki rata rata *throughput* yang lebih besar dari RIO-C dan WRED untuk N kecil karena memelihara panjang antrian yang besar sehingga selalu ada paket yang dikirimkan sehingga *throughput* besar. Tetapi saat penambahan sumber panjang antrian yang besar tidak mampu ditampung *buffer* sehingga terjadi *buffer overflow* yang akan merugikan performansi. WRED memiliki rata rata *throughput* yang paling rendah diantara ketiga skema MRED.



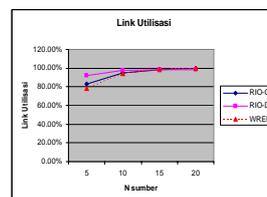
Gambar 8. Panjang Antrian Skenario 1.1



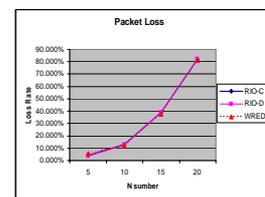
Gambar 9. Queue delay Skenario 1.1

Pada gambar 8 dapat dilihat RIO-C dan WRED mengatur panjang antrian tetap stabil, tetapi untuk jumlah sumber 20 ukuran antrian meningkat. Panjang antrian RIO-D meningkat sesuai dengan jumlah penambahan sampai 20 sumber dan stabil untuk jumlah sumber 15 dan 20, tetapi rata-rata ukuran antriannya masih lebih tinggi dibanding skema lainnya. Dari tabel 4.2 jumlah antrian dari RIO-D mencapai 58 saat sumber berjumlah 15 dan 20, tertinggi diantara kedua skema lainnya.

Dapat dilihat pada gambar 9 bahwa *queue delay* untuk RIO-C walaupun masih dapat menjaga antrian pada batasnya sehingga *queue delay* cukup kecil, tetapi kestabilan panjang antrian dikorbankan dan juga *loss* yang terus bertambah. Untuk WRED tidak terlalu berbeda dengan RIO-C. Pada RIO-D terjadi peningkatan ukuran antrian untuk setiap penambahan jumlah sumber yang menyebabkan adanya peningkatan juga pada *queue delay* hingga mencapai 49 ms saat jumlah sumber 15 dan 20.



Gambar 9. Link Utilization Skenario 1.1



Gambar 10. Loss rate Skenario 1.1

Pada gambar 9 dapat dilihat bahwa skema RIO pada semua perubahan N mempunyai *link utilization* yang tinggi diatas 90%. Sedangkan pada WRED memiliki *link utilization* paling rendah dibanding RIO-C dan RIO-D. Pada RIO-C *link utilization* yang tetap tinggi dicapai dengan menyesuaikan *aggregate input* dengan kapasitas link dengan penggunaan antrian yang efisien. Sedangkan pada RIO-D memiliki *link utilization* tinggi dengan memelihara panjang antrian yang tinggi sehingga

selalu ada paket yang dikirimkan sehingga *link utilization* tinggi diikuti dengan *queuing delay* yang tinggi pula sedangkan WRED karena harus menjaga panjang antrian tetap rendah tetapi kondisi antrian beresilasi tinggi sehingga sering mengalami kekosongan sehingga memiliki *link utilization* yang lebih rendah dibanding AQM yang lain.

Pada gambar 10 *loss rate* untuk semua skema MRED meningkat akibat adanya penambahan sumber. Peningkatan *loss rate* yang signifikan terlihat pada penambahan sumber N sebesar 20. Hal ini terjadi akibat banyaknya paket yang diterima sehingga adanya penumpukan di antrian yang menyebabkan banyaknya *dropping*.

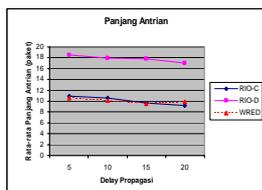
Dari tabel 3 diatas dapat dilihat RIO-C mampu melindungi paket hijau sehingga tidak ada paket yang di-drop, sebaliknya untuk paket dengan *drop precedence* yang lebih tinggi (*OUT profile*) mengalami *dropping* di antrian, yaitu sebesar 7%. Hal ini disebabkan paket hijau (*IN profile*) yang memiliki *drop precedence* yang lebih rendah diproteksi oleh skema RIO-C. Hal yang berbeda terjadi saat menggunakan skema WRED, pada skema ini paket hijau tidak terlindungi sehingga sekitar 0.03 % dari paket yang diterima di-drop.

Tabel 3. Packet Dropping Skema MRED

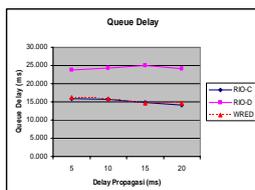
Skema	Marking Policy	Total Paket Diterima di Core	Paket Droppin g	% Paket Droppin g	Force Droppin g (%)	Early Droppin g (%)	Pengam atan
RIO-C	Green (10)	14788	0	0	0	0	Paket Hijau terlindun gi
	Yellow (11)	24118	1714	7.11	681 (2.82)	1033 (4.28)	
RIO-D	Green (10)	14056	43	0.31	43 (0.31)	0	Paket hijau tidak terlindun gi
	Yellow (11)	28112	1399	4.98	191 (0.68)	1208 (4.3)	
WRE D	Green (10)	15048	4	0.03	0	4 (0.03)	Paket hijau tidak terlindun gi
	Yellow (11)	22287	2041	9.16	1141 (5.12)	900 (4.04)	

4.1.2 Analisis Perubahan Delay Propagasi Pada Bottleneck Link

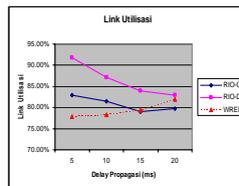
Dari gambar dibawah ini dapat dilihat bahwa perubahan delay propagasi tidak telalu mempengaruhi performansi dari ketiga skema MRED. Ukuran dari *loss rate* tetap stabil antara 3%-5%. *Throughput* dari ketiga skema MRED juga tidak megalami perubahan yang signifikan. Untuk delay antrian itu sendiri RIO-D tetap tinggi, tertinggi diantara ketiga skema MRED, mencapai 24 ms. Hal ini diakibatkan RIO-D terus mempertahankan rata-rata panjang antriannya yang juga terbesar diantara dua skema lainnya, hingga mencapai sekitar 18.



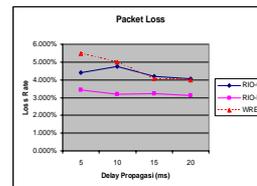
Gambar 11. Panjang Antrian Skenario 1.2



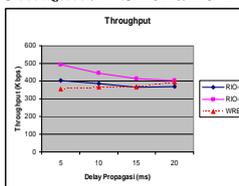
Gambar 12. Queue delay Skenario 1.2



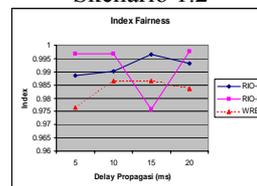
Gambar 13. Link Utilization Skenario 1.2



Gambar 14. Loss rate Skenario 1.2



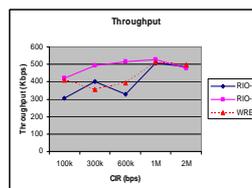
Gambar 15. Throughput Skenario 1.2



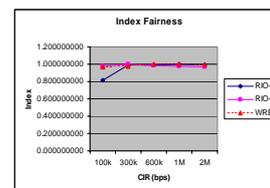
Gambar 16. Index fairness Skenario 1.2

Dari gambar 11 memperlihatkan bahwa ukuran antrian untuk setiap skema MRED tidak mengalami perubahan. Skema-skema tersebut tetap mempertahankan ukuran antriannya Rata-rata panjang antrian RIO-C diatas panjang antrian dua skema lainnya, hal ini yang menyebabkan delay antrian yang besar di router. Keseluruhan delay propagasi tidak terlalu mempengaruhi dari rata-rata panjang antrian tipa skema MRED. Dari gambar diatas dilihat bahwa perubahan pada parameter delay propagasi tidak terlalu mempengaruhi dari performansi ketiga skema MRED. Untuk setiap parameter *link utilization* dan *throughput* terjadi sedikit penurunan karena adanya perubahan delay propagasi.

4.1.3 Analisis Perubahan Parameter Policy (CIR)



Gambar 17. Throughput Skenario 1.3



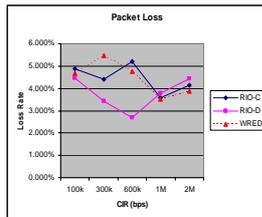
Gambar 18. Index Fairness Skenario 1.3

Dari gambar 17. diatas untuk skema RIO-D dan WRED terjadi suatu peningkatan *throughput* ketika CIR diset sebesar 600 Kbps. RIO-D memiliki *throughput* terbesar dibanding RIO-C dan WRED. Penurunan *throughput* terjadi saat perubahan CIR menjadi 2 Mps.

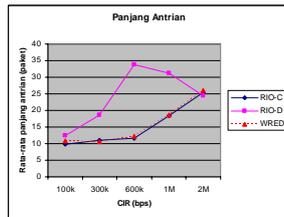
Tabel 4 memperlihatkan untuk perubahan CIR menjadi 100 Kbps dan 300 Kbps, setiap skema MRED masih dapat melindungi paket hijau (*low precedence* - 10) dari *dropping* paket tetapi ketika CIR diset menjadi 600 Kbps dan seterusnya paket hijau menjadi tidak terlindungi. Semakin besar CIR semakin banyak pula paket yang dibuang, terutama paket yang berprioritas rendah (paket hijau).

Tabel 4. Data *Packet Dropping* Terhadap Perubahan Parameter CIR

Skema	Dropping	Marking Policy	CIR				
			100k	300k	600k	1M	2M
RIO-C	Drop Paket	Green (10)	0	0	30	260	1710
		Yellow (11)	2016	1714	1867	1256	12
	Force Dropping	Green (10)	0	0	22	167	1070
RIO-D	Drop Paket	Green (10)	803	681	1510	1089	12
		Yellow (11)	0	0	8	93	640
	E-Dropping	Green (10)	1213	1033	357	67	0
WRED	Drop Paket	Green (10)	11	43	388	1623	1830
		Yellow (11)	1749	1399	766	5	0
	Force Dropping	Green (10)	11	43	234	1074	1262
E-Dropping	Drop Paket	Green (10)	354	191	68	5	0
		Yellow (11)	0	0	154	549	568
	Force Dropping	Green (10)	1395	1208	698	0	0
WRED	Drop Paket	Green (10)	0	4	85	361	1616
		Yellow (11)	1836	2041	1754	1128	8
	Force Dropping	Green (10)	0	0	77	189	943
E-Dropping	Drop Paket	Green (10)	429	1141	1136	932	7
		Yellow (11)	0	4	8	172	673
E-Dropping	Drop Paket	Green (10)	1407	900	618	196	1
		Yellow (11)					



Gambar 19. *Packet Loss* Skenario 1.3



Gambar 20. Panjang antrian Skenario 1.3

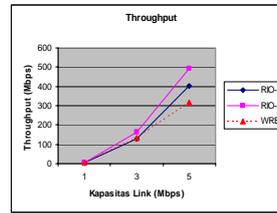
Dari gambar 19 dapat dilihat pengaruh perubahan CIR terhadap *loss rate* untuk skema RIO-D. Terjadi penurunan *loss rate* ketika CIR di-set sebesar antara 100 Kbps – 600 Kbps, tapi kemudian terjadi peningkatan saat CIR di-set lebih besar dari 600 Kbps. Lain halnya dengan skema RIO-C dan WRED, penurunan mulai terjadi pada saat CIR sebesar 300 Kbps dan kembali terjadi peningkatan saat CIR sebesar 1Mbps.

Pada gambar 20 terlihat terjadi peningkatan panjang antrian terbesar untuk skema RIO-C. Puncaknya terjadi saat CIR sebesar 600 Kbps kemudian terjadi penurunan ketika CIR lebih besar dari 600 Kbps. RIO-C dan WRED memiliki rata-rata panjang antrian yang besarnya hampir sama ketika terjadi perubahan CIR.

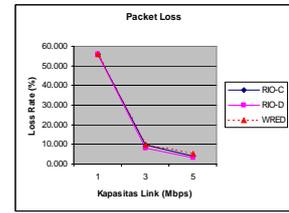
4.1.4 Analisis Perubahan Kapasitas *Bottleneck Link*

Dari gambar dibawah ini dapat dilihat peningkatan *throughput* karena adanya perubahan kapasitas *bottleneck link* yang semakin membesar. RIO-D menghasilkan *throughput* dan *index fairness* yang paling baik diantara dua skema MRED lainnya. Kapasitas yang terbatas, hanya 1 Mbps, menyebabkan adanya jumlah antrian yang besar di router sehingga menyebabkan delay antrian menjadi

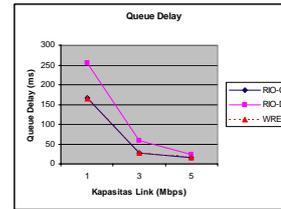
sangat besar antara 160 ms untuk skema RIO-C dan WRED serta hingga mencapai 254 ms untuk skema RIO-D. *Loss rate* yang dihasilkan oleh ketiga skema MRED juga besar, sebesar 55 %.



Gambar 21. *Throughput* Skenario 1.4



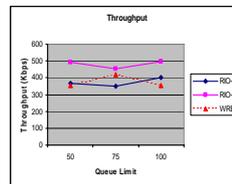
Gambar 22. *Packet Loss* Skenario 1.4



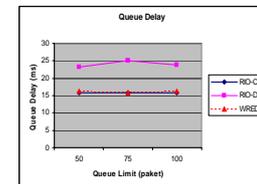
Gambar 23. *Delay Antrian* Skenario 1.4

Kapasitas yang membesar menyebabkan antrian mengecil yang diikuti oleh delay antrian yang kecil pula, dapat dilihat pada gambar 23 Kapasitas yang besar juga menghasilkan *packet loss* yang kecil, seperti yang dapat dilihat pada gambar 22 diatas.

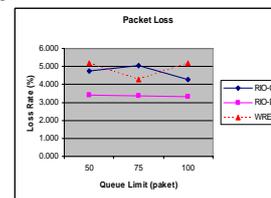
4.1.5 Analisis Perubahan Batas Antrian *Bottleneck Link*



Gambar 24. *Throughput* Skenario 1.5



Gambar 25. *Delay Antrian* Skenario 1.5



Gambar 26. *Packet Loss* Skenario 1.5

Dari gambar dapat dilihat bahwa RIO-D memiliki *loss rate* yang terendah diantara kedua skema lainnya, yaitu sekitar 3,5 % dan terus memperahkannya saat batas antrian diubah. Dari gambar 24 dapat dilihat RIO-D memiliki besar *throughput* yang lebih tinggi dibandingkan RIO-C dan WRED yaitu rata-rata mencapai 450–490 Kbps. Tetapi untuk jumlah antrian, RIO-D memiliki jumlah antrian yang besar hingga mencapai 18 yang menyebabkan *delay* antriannya juga tinggi hingga 25 ms, terbesar diantara dua skema lainnya.

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari simulasi dan analisis yang telah dilakukan, dapat diambil kesimpulan sebagai berikut:

1. Peningkatan jumlah sumber sebesar N (5,10,15,20) menghasilkan penurunan *throughput* dan *indeks fairness* untuk semua skema MRED. Nilai *throughput* yang terbesar dialami pada skema RIO-D sebesar 494 Kbps. Untuk jumlah sumber yang sama WRED memiliki *throughput* terkecil yaitu 355 Kbps.
2. RIO-D menghasilkan *throughput* yang lebih besar dibandingkan RIO-C dan WRED ketika terjadi penambahan sumber. Namun disisi yang lain RIO-D juga menghasilkan panjang antrian yang lebih besar yaitu mencapai 60 sehingga *queue delay* juga besar, tertinggi hingga 49 ms. Penambahan sumber menghasilkan jumlah antrian yang semakin besar. Untuk panjang antrian yang lebih besar akan menghasilkan *throughput* yang lebih besar tetapi *queue delay* juga bertambah besar.
3. Untuk jumlah sumber yang sama RIO-C mampu untuk melindungi paket *low precedence* (hijau) lebih baik dibandingkan dengan skema yang lainnya dengan loss rate adalah 0%. WRED memiliki loss rate untuk paket hijau sebesar 0.03%, sementara RIO-D sebesar 0.31%.
4. Terjadi juga peningkatan *loss rate* paket data secara keseluruhan untuk semua skema sebesar 3%-5% ketika N=5 dan terus naik hingga 80% saat N=20.
5. *Delay* propagasi tidak terlalu berpengaruh pada performansi antrian secara keseluruhan. Besar *throughput*, *loss* paket dan *delay* dari ketiga skema tetap stabil.
6. Saat parameter CIR ditingkatkan (600 Kbps), RIO-D menghasilkan *loss* paket terendah (2.695%) dibanding skema yang lain namun meningkat lagi ketika CIR > 600 Kbps.
7. Semakin tinggi CIR semakin banyak paket yang dikategorikan sebagai paket *high priority* sehingga semua paket di-*marking low precedence* (hijau) dan tidak ada lagi paket kuning.
8. RIO-C memiliki kinerja yang paling baik diantara ketiga skema MRED. Hal ini ditandai dengan *throughput* yang tinggi tetapi juga tetap menjaga jumlah antrian dan *delay* antrian agar tetap kecil.

PUSTAKA

- [1] Agustina, Elisha, "Analisa Performansi Jaringan TCP Reno dan TCP Sack Menggunakan Pengontrol Random Early Detection (RED)", Jurusan Teknik Elektro, STT Telkom, Bandung, 2005
- [2] Andreozzi, S., "Diffserv Simulation Using The Network Simulator: Requirements, Issues and Solutions", University of Pisa, 2001
- [3] Arimbawa, AA Gde Alit "Analisis Performansi AQM Routers Yang Mendukung

Aliran TCP Dengan Menggunakan Random Exponential Marking (REM)", Jurusan Teknik Informatika, STT Telkom, Bandung, 2006

- [4] Altman, E., "A Stateless Approach for Improving TCP Performance Using Diffserv", submitted
- [5] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., Weiss, W., "An Architecture for Differentiated Services", RFC 2475, December 1998
- [6] Clark, D., and Fang, W., "Explicit Allocation of Best Effort Packet Delivery Service", *ACM Transaction on Networking*, Vol.6, No.4, August 1998, pp 362-373
- [7] Elloumi, O., Snodder, and Pauwels, K., "Usefulness of Three Drop Precedence in Assured Forwarding Service", IETF Draft, <draft-elloumi-diffserv-threestwo-00.txt>, July 1999
- [8] Fall, K., and Varadhan, K., "The ns Manual", available at <http://www.isi.edu/nam/ns/>
- [9] Sally Ford "RED: Discussions of Setting Parameters", November 1997
- [10] Fang, W., Seddigh, N., and Nandy, B., "A Time Sliding Window Three Colour Marker (TSWTCM)", RFC 2859, <http://rfc.sunsite.dk/rfc/rfc2859.html>, June 2000
- [11] Goyal, M., Duressi, A., Jain, R., and Lei, C., "Effect of Drop Precedences in Assured Forwarding", Internet Draft, draft-goyal-diffserv-spstdy-02.pdf, July 1999
- [12] Ibanez, J., Nichols, K., "Premilinary Simulation Evaluation of an Assured Service", Internet Draft, <draft-ibanez-diffserv-assured-eval-00.txt>