

ANALISIS ALGORITMA ROUND ROBIN, LEAST CONNECTION, DAN RATIO PADA LOAD BALANCING MENGGUNAKAN OPNET MODELER

Husain Nasser¹
husainnassr@gmail.com

Timotius Witono²
timotius@itmaranatha.org

Abstract

Load balancing is a technique to distribute traffic load into two or more lanes to reach balanced connection, optimized traffic performance, maximized data throughput, minimized delay, and avoiding data overload. Load balancing in a computer network used to modulate the balance of bandwidth usage between multiple internet service provider. This research contains Load Balancing performance test on HTTP, FTP, and VoIP. Testing application used is OPNET Modeler 14.5 simulator, and load balancing algorithm which being tested is Least Connection, Round Robin, and Ratio. The test results shows generally Ratio is better than Least Connection and Round Robin. Ratio excels on handling End-to-End Delay and Jitter in VoIP, while Least Connection shows the highest throughput value in all test cases.

Keywords: *load balancing, round robin, least connection, ratio, OPNET modeler*

1. Pendahuluan

Internet Service Provider (ISP) menyediakan layanan *internet*, baik untuk sambungan *internet* lokal maupun internasional. Sambungan lokal menyediakan interkoneksi antar ISP yang terdapat di Indonesia agar dapat saling terhubung satu sama lain, sedangkan sambungan internasional menyediakan *bandwidth* yang dapat digunakan untuk terhubung ke *backbone* internasional. *Bandwidth* adalah besaran yang menunjukkan seberapa banyak data yang dapat dilewatkan dalam koneksi melalui sebuah *network* (Haryadi, 2010).

Kebutuhan *bandwidth internet* semakin hari semakin meningkat, sehingga seringkali dibutuhkan penggunaan layanan *internet* dari beberapa ISP sekaligus. Sebuah perusahaan dapat menambahkan jalur *internet secondary*, jika merasa jalur *internet primary* tidak dapat menampung kebutuhan penggunaan *bandwidth* secara maksimal. Pembagian penggunaan masing-masing jalur *internet* dapat diatur sesuai kebutuhan dan karakteristik pengguna. Penggunaan jalur-jalur *internet* juga dapat dimaksimalkan dengan penerapan teknik *load balancing*.

Load balancing adalah sebuah teknik mendistribusikan beban *traffic* pada dua jalur atau lebih, sehingga didapatkan sambungan yang seimbang, *traffic* yang lebih optimal, *throughput data* maksimal, *delay* minimal, serta tidak terjadi *overload*. *Load balancing* dapat diimplementasikan pada perusahaan yang memiliki minimal dua sambungan internet. Penelitian terkait menyatakan bahwa beban jaringan tidak menjadi lebih ringan dengan adanya *load balance*, akan tetapi *load balance* bertugas sebagai pengatur alokasi beban jaringan (Saputra, 2013). “*Load balancing* algoritma bekerja pada prinsip bahwa situasi di mana beban kerja diberikan, selama waktu kompilasi atau pada *runtime*. Perbandingan menunjukkan bahwa algoritma beban statis *balancing* lebih stabil dibandingkan dengan dinamis dan juga kemudahan untuk memprediksi perilaku statis, tetapi pada saat yang sama algoritma terdistribusi yang dinamis selalu dianggap lebih baik daripada algoritma statis” (Kurniawan, 2011).

2. Dasar Teori

2.1 Algoritma Load Balancing

Load balancing adalah sebuah konsep yang berguna untuk menyeimbangkan beban atau muatan dalam jaringan komputer, makin besar *traffic data* yang ada dalam jaringan komputer

¹ Program Studi SI Teknik Informatika, Fakultas Teknologi Informasi, Universitas Kristen Maranatha, Bandung

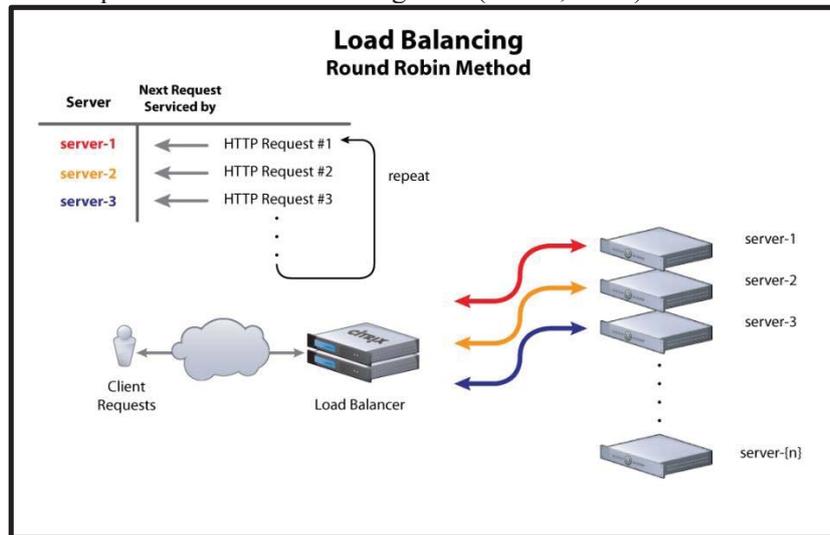
² Program Studi SI Teknik Informatika, Fakultas Teknologi Informasi, Universitas Kristen Maranatha, Bandung

maka semakin berat pengaksesannya. Layanan *load balancing* memungkinkan pengaksesan sumber daya dalam jaringan didistribusikan ke beberapa *host*, agar tidak terpusat, sehingga kinerja jaringan komputer menjadi lebih stabil (Kopparapu, 2002).

Proses pembagian beban pada sistem *load balancing* memiliki teknik dan algoritma tersendiri. Perangkat *load balancing* yang kompleks biasanya menyediakan berbagai teknik dan algoritma pembagian beban ini, tujuannya untuk menyesuaikan pembagian beban dengan karakteristik dari masing-masing pengguna internet. OPNET memungkinkan peneliti untuk membuat simulasi jaringan komputer yang melibatkan berbagai unsur teknologi jaringan yang ada, termasuk *load balancing* (OPNET, 2012).

2.2 Round Robin

Algoritma *Round Robin* merupakan algoritma yang paling sederhana dan paling banyak digunakan oleh perangkat *load balancing*. Algoritma *Round Robin* bekerja dengan cara membagi beban secara bergiliran dan berurutan dari satu *server* ke *server* lainnya. Konsep dasar dari algoritma *Round Robin* ini adalah dengan menggunakan *time sharing*, pada intinya algoritma ini memproses antrian secara bergiliran (Ellrod, 2010).

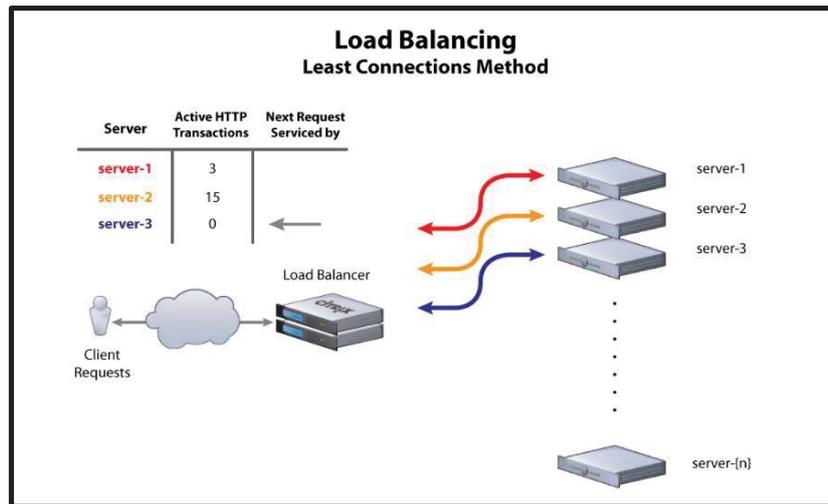


Gambar 1. Proses Round Robin (Ellrod, 2010)

2.3 Least Connection

Algoritma *Least Connection* melakukan pembagian beban berdasarkan banyaknya koneksi yang sedang dilayani oleh sebuah *server*. *Server* dengan koneksi yang paling sedikit akan diberikan beban berikutnya, begitu pula *server* dengan koneksi banyak akan dialihkan bebannya ke *server* lain yang bebannya lebih rendah (Ellrod, 2010).

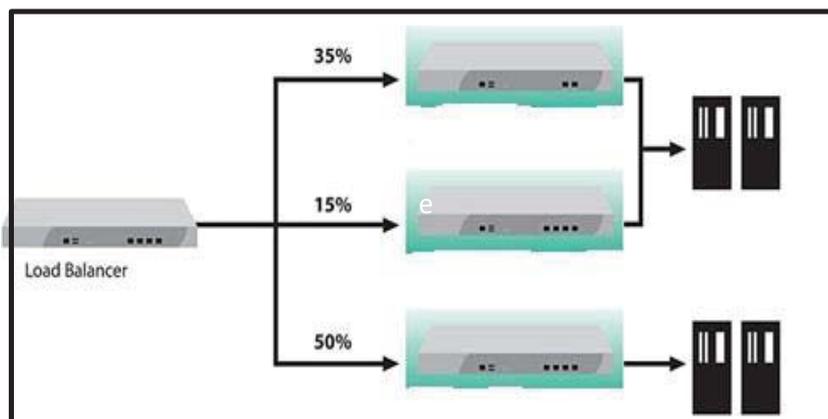
Penjadwalan ini termasuk salah satu algoritma penjadwalan dinamik, karena memerlukan perhitungan koneksi aktif untuk masing-masing *real server* secara dinamik. Metode penjadwalan ini baik digunakan untuk melancarkan pendistribusian ketika *request* yang datang sangat banyak. Sebagai contoh terdapat dua *Service-HTTP* yaitu *Service HTTP-1* (terdapat 3 *active HTTP transaction*) dan *Service HTTP 2* (terdapat 1 *active HTTP transaction*), maka *Service HTTP-2* akan menerima request selanjutnya dikarenakan $\text{Service HTTP-1} > \text{Service HTTP-2}$ pada nilai transaksi aktifnya.



Gambar 2. Proses Least Connections (Ellrod, 2010)

2.4 Ratio

Ratio menggunakan parameter *ratio* pada masing-masing *server* di dalam sistem *load balancing*. Parameter *ratio* ini akan menjadi landasan pembagian beban pada *server-server* yang terlibat. *Server* dengan *ratio* terbesar diberi beban besar, sebaliknya *server* *ratio* kecil akan lebih sedikit diberi beban (DELL inc, 2015).



Gambar 3. Proses Ratio (DELL inc, 2015)

2.5 Pengukuran Kinerja Jaringan

Quality of Service (QoS) merupakan kemampuan jaringan untuk menyediakan *service* yang lebih baik untuk *traffic* tertentu dari berbagai macam teknologi seperti jaringan IP, *frame relay*, ATM dan SDH. Elemen QoS tergantung dari informasi yang ditransmisikan (*voice*, *data* atau *video*) (Anton, 1997) (Haryadi, 2010). Beberapa faktor yang mempengaruhi QoS pada jaringan IP:

1. Throughput

Throughput adalah jumlah *bit* yang diterima dengan sukses per detik melalui sebuah *sistem* atau media komunikasi (kemampuan sebenarnya suatu jaringan dalam melakukan pengiriman data). *Throughput* dapat diukur setelah transmisi *data* dilakukan, karena suatu sistem akan menambah *delay* yang disebabkan *processor limitations*, *buffering inefficients*, *error transmission*, *traffic loads* dan lain-lain. Aspek utama *throughput* berkisar pada ketersediaan *bandwidth* yang cukup untuk menjalankan aplikasi.

2. Jitter

Jitter adalah variansi *delay*, yaitu perbedaan selang waktu kedatangan antar paket di terminal tujuan. Untuk mengatasi *jitter* maka paket *data* yang datang dikumpulkan dulu dalam *jitter buffer* selama waktu yang telah ditentukan sampai paket dapat diterima pada sisi penerima dengan urutan yang benar.

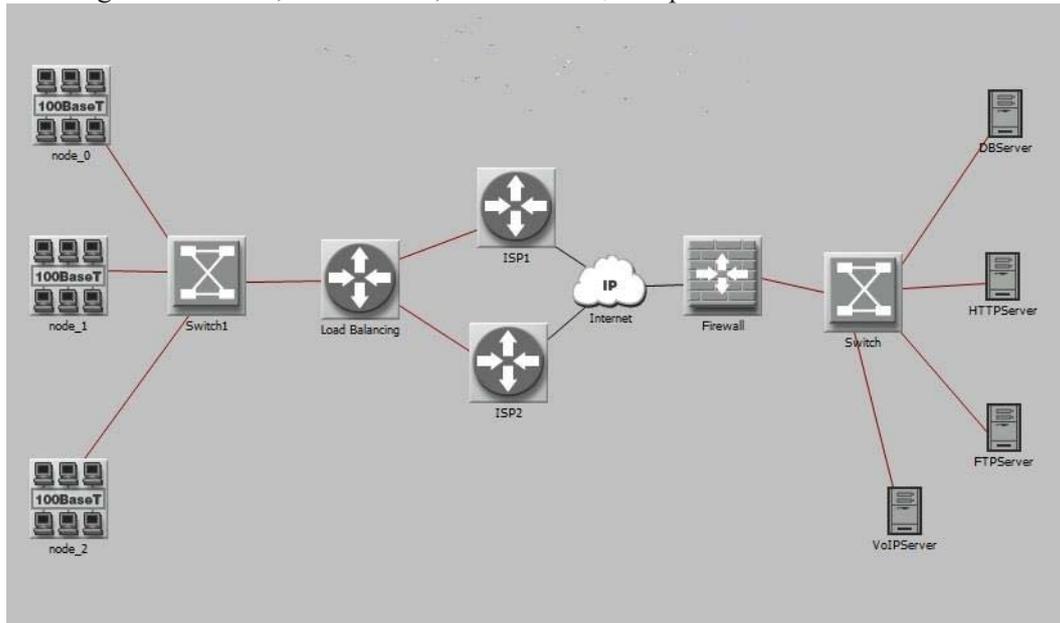
3. Delay

Delay adalah waktu tunda paket yang diakibatkan oleh proses transmisi dari satu titik lain yang menjadi tujuannya. Ada dua penyebab terjadinya *delay* dalam jaringan, yaitu *Fixed Delay* dan *Variable Delay*.

3. Analisis dan Perancangan Jaringan

3.1 Topologi Jaringan

Simulasi jaringan *load balancing* menggunakan 3 buah *workstation* yang terhubung ke *switch*, kemudian *switch* terhubung ke *router*. Sebelum terhubung ke *internet*, jaringan terhubung ke ISP-1 dan ISP-2, lalu terhubung ke *internet* dan melewati *firewall*. Secara keseluruhan jaringan akan memampukan seluruh *workstation (node client)* untuk dapat terhubung ke DB *server*, VoIP *server*, HTTP *server*, ataupun FTP *server*.



Gambar 4. Topologi Jaringan Load Balancing

Berikut adalah komponen jaringan dari Gambar 4:

- Workstation*, merupakan komputer PC yang digunakan sebagai *client*. *Workstation* bertugas untuk mengakses aplikasi-aplikasi yang terdapat pada masing-masing *server*.
- Router*, perangkat ini berfungsi sebagai penghubung antar jaringan-jaringan yang berbeda. *Router* bertugas sebagai jembatan dua buah ISP untuk mengatur jaringan komputer. ISP-1 dengan kapasitas *bandwidth* 4 Mbps dan ISP-2 dengan kapasitas *bandwidth* 1 Mbps.
- Internet*, merupakan jaringan komputer publik.
- Firewall*, adalah perangkat yang berfungsi untuk memfilter paket yang masuk ke dalam jaringan tertentu, *firewall* dapat memilah-milah jenis paket dan tugasnya mengamankan suatu jaringan dari paket-paket yang tidak diinginkan.
- Switch*, merupakan penghubung beberapa perangkat untuk membentuk jaringan kecil atau *Local Area Network (LAN)*.
- Server*, yang berfungsi sebagai perangkat untuk penyedia layanan dalam jaringan komputer.

3.2 Simulasi Skenario

9 buah skenario jaringan yang berbeda digunakan dalam lingkungan simulasi. Skenario 1, 2 dan 3 digunakan untuk mengukur pengaruh algoritma *round robin* terhadap jaringan *load balancing*. Skenario 4, 5 dan 6 mengukur pengaruh algoritma *least connection* terhadap jaringan *load balancing*. Skenario 7, 8 dan 9 mengukur pengaruh *ratio* terhadap jaringan *load balancing*.

Tabel 1.
Skenario Load Balancing

Skenario	Algoritma Load Balancing	Besar Data Uji	Keterangan
Skenario 1	Round Robin	1 Mbps	Metric Pengukuran: <ul style="list-style-type: none"> • Throughput • Delay • Jitter
Skenario 2		5 Mbps	
Skenario 3		7 Mbps	
Skenario 4	Least Connection	1 Mbps	Data Uji: Transaksi workstation dengan <ul style="list-style-type: none"> • HTTP Server • DB Server • FTP Server • VoIP Server
Skenario 5		5 Mbps	
Skenario 6		7 Mbps	
Skenario 7	Ratio	1 Mbps	Waktu Pengukuran: 90 menit (diukur setelah jaringan mencapai titik jenuh)
Skenario 8		5 Mbps	
Skenario 9		7 Mbps	

3.3 Simulasi Jaringan Data

Penelitian mengukur pengaruh transaksi data HTTP, VoIP, FTP dan DB terhadap jaringan *load balancing*. Pengukuran dilakukan dalam berbagai skenario yang berbeda berdasarkan algoritma *round robin*, *least connection* dan *ratio*. Parameter perbandingan yang diukur adalah:

1. *Throughput*
Throughput yang diukur adalah *throughput* masuk keseluruhan
2. *Delay*
Delay yang diukur berdasarkan komunikasi antara VoIP_src dengan VoIP_dst
3. *Jitter*
Jitter yang diukur berdasarkan komunikasi antara VoIP_src dengan VoIP_dst

4. Pengujian

4.1 Perbandingan Keseluruhan Load Balancing

Berikut adalah tabel keseluruhan dari simulasi algoritma *Round Robin*, *Least Connection* dan *Ratio* dengan rata-rata uji data 1 Mbps, 5 Mbps dan 7 Mbps. Setiap percobaan menghasilkan pengukuran *throughput*, *end-to-end delay* dan *jitter*.

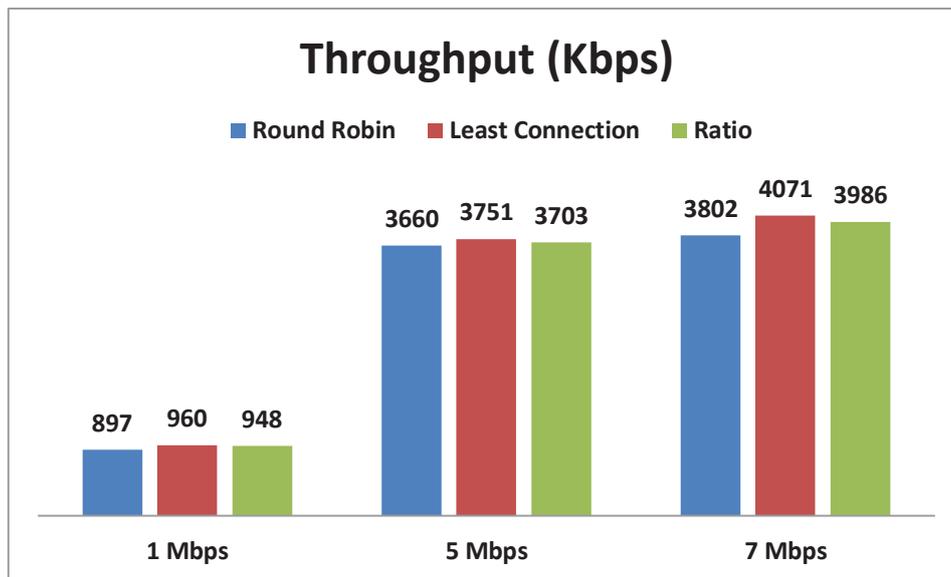
Tabel 2.
Hasil Simulasi Perbandingan Rata-Rata Keseluruhan Jaringan Load Balancing

Faktor Pengukuran	Average Data 1 Mbps		
	Round Robin	Least Connection	Ratio
<i>Throughput (Kbps)</i>	897	960	948
<i>End-to-End Delay (ms) – VoIP</i>	738	336	329
<i>Jitter (ms) – VoIP</i>	0.041	0.031	0.020
Faktor Pengukuran	Average Data 5 Mbps		
	Round Robin	Least Connection	Ratio
<i>Throughput (Kbps)</i>	3660	3751	3703
<i>End-to-End Delay (ms) – VoIP</i>	371	406	426
<i>Jitter (ms) – VoIP</i>	0.006	0.047	0.002

Faktor Pengukuran	Average Data 7 Mbps		
	Round Robin	Least Connection	Ratio
Throughput (Kbps)	3802	4071	3986
End-to-End Delay (ms) – VoIP	415	463	423
Jitter (ms) – VoIP	0.016	0.015	0.014

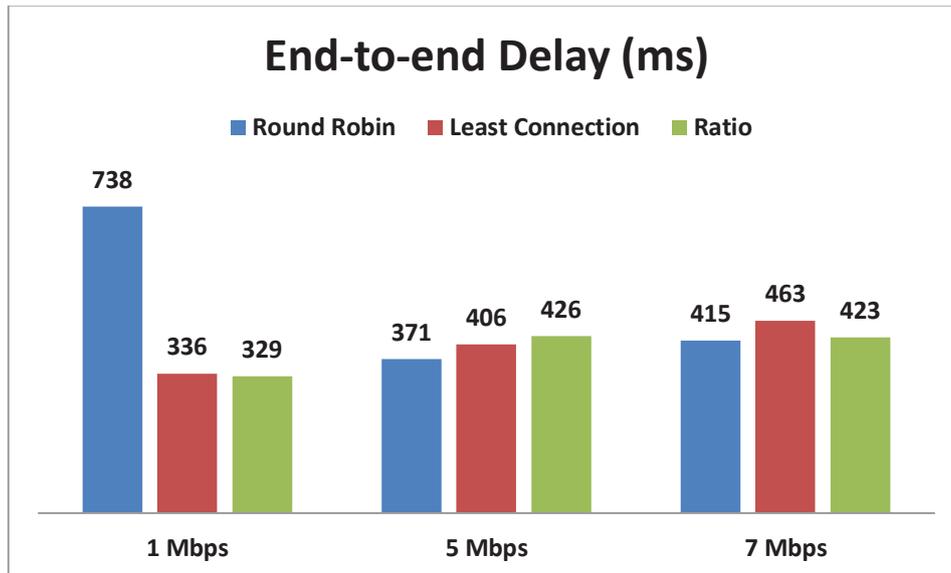
Berdasarkan Tabel 2, dimana semua skenario dibandingkan, maka hasil yang di peroleh sebagai berikut:

- 1) Algoritma *Load Balancing* terbaik untuk skenario uji data 1 Mbps:
 - a) *Throughput* : *Least Connection*
 - b) *End-to-end Delay* : *Ratio*
 - c) *Jitter* : *Ratio*
- 2) Algoritma *Load Balancing* terbaik untuk skenario uji data 5 Mbps:
 - a) *Throughput* : *Least Connection*
 - b) *End-to-end Delay* : *Round Robin*
 - c) *Jitter* : *Ratio*
- 3) Algoritma *Load Balancing* terbaik untuk skenario uji data 7 Mbps:
 - a) *Throughput* : *Least Connection*
 - b) *End-to-end Delay* : *Round Robin*
 - c) *Jitter* : *Ratio*



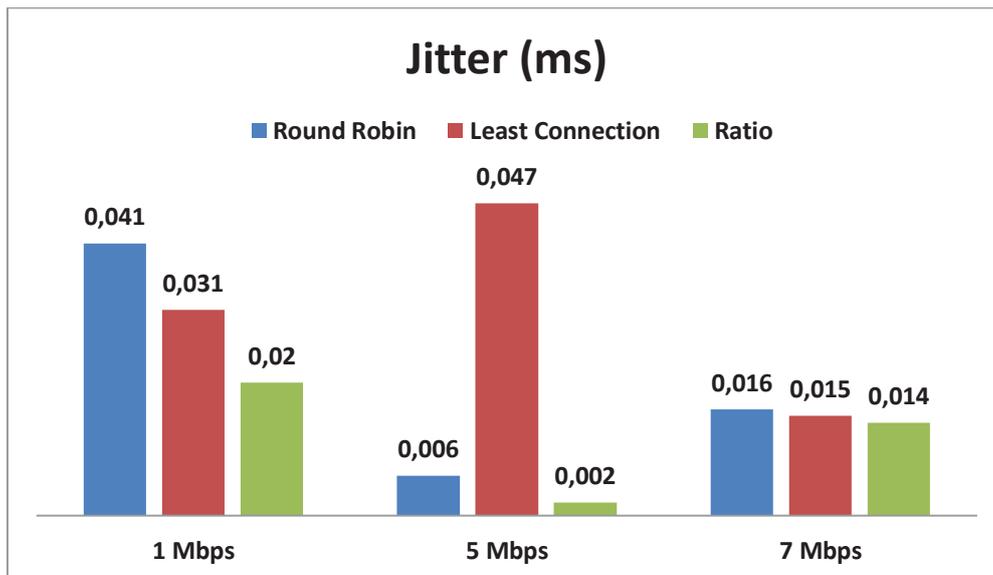
Gambar 5. Grafik *throughput* keseluruhan

Gambar 5 menggambarkan bahwa *Least Connection* menghasilkan *throughput* paling tinggi dibandingkan *Ratio* dan *Round Robin*. *Round Robin* menghasilkan nilai paling rendah, sehingga *Round Robin* tidak disarankan untuk kebutuhan koneksi yang membutuhkan *throughput* tinggi.



Gambar 6. Grafik delay keseluruhan

Gambar 6 menggambarkan *Ratio* paling efisien dalam hal *delay* untuk skenario 1 Mbps, akan tetapi *Round Robin* lebih kecil dan paling efisien pada skenario 5 Mbps dan 7 Mbps. Hanya saja pada skenario 1 Mbps, *Round Robin* ternyata memiliki *delay* paling besar, sehingga stabilitasnya tidak cukup baik. *Ratio* disarankan sebagai solusi jika diperlukan koneksi dengan *delay* yang efisien dan stabil.



Gambar 7. Grafik jitter keseluruhan

Gambar 7 memperlihatkan dengan mutlak bahwa *Ratio* memiliki variansi *delay* (*jitter*) yang paling kecil, sehingga dapat dikatakan paling stabil dalam hal penanganan *jitter*. *Ratio* direkomendasikan menjadi solusi untuk koneksi yang memerlukan *jitter* yang rendah (*delay* stabil).

5. Kesimpulan

Kesimpulan dari penelitian yang telah dilakukan adalah sebagai berikut:

1. OPNET Modeler dapat melakukan pemodelan dan simulasi jaringan *load balancing*, sehingga dapat dihasilkan data perbandingan kinerja dari berbagai algoritma *load balancing*. Hasil pengukuran kinerja algoritma *load balancing* secara umum menunjukkan bahwa *Ratio* lebih baik dibandingkan dengan *Least Connection* atau *Round Robin*, hal ini terlihat dari angka pengukuran parameter *End-to-end Delay* dan *Jitter* yang paling baik ditemukan pada *Ratio*.
2. Pemilihan algoritma *load balancing* untuk implementasi di kasus nyata:

- a) *Ratio* baik digunakan untuk jaringan *internet* yang memerlukan *jitter* dan *end-to-end delay* yang paling efisien.
- b) *Least Connection* baik digunakan untuk jaringan *internet* yang memerlukan *throughput* yang paling maksimal.

Daftar Pustaka

- Anton, W. (1997). *Jaringan Komputer*. Bandung: Informatika.
- DELL inc. (2015). *SRA Load Balancing and High Availability*. Retrieved 2015, from <http://www.sonicwall.com: http://www.sonicwall.com/de/de/products/326.html>
- Ellrod, C. (2010). *Load Balancing – Round Robin*. Retrieved 2015, from <http://blogs.citrix.com: http://blogs.citrix.com/2010/09/03/load-balancing-round-robin/>
- Haryadi, S. (2010). *Quality of Service (QoS) Dan Pengukurannya*.
- Kopparapu, C. (2002). *Load Balancing Servers, Firewalls, and Caches*. John Wiley & Sons, Inc.
- Kurniawan, H. (2011). ANALISIS KINERJA BEBERAPA ALGORITMA LOAD BALANCING . *Jurnal Seminar Nasional Informatika UPN*.
- OPNET. (2012, Juni 5). (OPNET) Retrieved September 2, 2014, from www.opnet.com: https://support.riverbed.com/bin/support/static/doc/opnet/17.5.A/online/modeler_17.5_PL5/Tutorials/wwhelp/wwhimpl/common/html/wwhelp.htm#context=tutorials&file=tut_com_intro.html
- Saputra, A. R. (2013). ANALISA PERBANDINGAN METODE LOAD BALANCING PEER CONNECTION CLASSIFIER (PCC) DENGAN NTH PADA ROUTER MIKROTIK. *Jurnal Skripsi Teknologi Informatika UIN*.