Implementasi Load Balancing Metode ECMP, NTH dan PCC dengan Empat Link Internet Menggunakan Mikrotik

M Saripuddin Adnan[1], Syariful Ikhwan[2], Yosy Rahmawati[3]
Progam Studi DIII , Fakultas Teknik Telekomunikasi dan Elektro
Institut Teknologi Telkom Purwokerto
Jl. DI Panjaitan No 128 Purwokerto 53147
15201011@st3telkom.ac.id[1] syariful@ittelkom-pwt.ac.id[2] yosy@ittelkom-pwt.ac.id[3]

Abstrak – Untuk meningkatkan performa internet tanpa batasan waktu maka dibutuhkan solusi untuk menjamin ketersedia koneksi internet. Beberapa solusi untuk menjamin koneksi internet yaitu *bandwidth* yang besar unutk kestabilan *traffic* data dan menggunakan beberapa *link*/*gateway* ke internet, dengan menggunakan beberapa *link* maka koneksi tetap tersedia setiap saat (*high availability*). Menggunakan beberapa *link* dibutuhkan teknik *load balancing* untuk menbagi *traffic* data ke beberapa *link* tersebut, sehingga *bandwidth* semua *link* dapat digunakan. Penelitian ini menerapkan teknik *failover* dan teknik *load balancing* metode ECMP, NTH dan PCC menggunakan empat *link* (*bandwidth* 100Mbps/*link*) pada MikroTik. Metodologi penelitian yaitu menguji teknik *failover* dan *load balancing* dengan men*dowloadfile* pada rentang ukuran 50-500 Mbps. Untuk analisa kinerja QoS berdasarkan parameter: *delay*, *jitterpacket loss* dan *throughput*. Hasil *Quality of Servicefailover* dan *load balancing* yang diterapkan jika dibandingkan yang terbaik berdasar parameter *delay, jitter, packet loss* dan *throughput*. Pameter *delay* ada Nth dengan nilai *delay* rata-rata terendah. Parameter *Jitter* ada Nth dengan nilai *jitter* rata-rata terrendah. Parameter *packetloss* ada ECMP dengan nilai *packet loss* yang kecil. Parameter *throughput* ada Nth dengan nilai *throughput* dengan rata-rata tertinggi.

Kata kunci: Internet, Failover, Load Balancing, ECMP, NTH, PCC, QoS

Abstrack – To improve internet performance without time constraints it takes a solution to ensure the availability of internet connection. Some solutions to ensure internet connection is a large bandwidth for the stability of data traffic and using multiple links / gateway s to the internet, using multiple links then the connection remains available at all times (high availability). Using multiple links requires load balancing techniques to share data traffic across multiple links, so that all link bandwidth can be used. This research applies failover technique and load balancing technique of ECMP, NTH and PCC method using four links (bandwidth 100Mbps / link) on MikroTik. The research methodology is to test failover and load balancing techniques by downloading files in the 50-500 Mbps size range. For QoS performance analysis based on parameters: delay, jitter packet loss and throughput. The results of Quality of Service failover and load balancing are applied when compared to the best based on delay, jitter, packet loss and throughput parameters. Pameter delay is Nth with the lowest average delay value. The Jitter parameter is Nth with the lowest average jitter value. Packet loss parameter exists ECMP with small packet loss value. The parameters of throughput are Nth with the highest average throughput.

Keywords: Internet, Failover, Load Balancing, ECMP, NTH, PCC, QoS

I. PENDAHULUAN

Perkembangan dan kemajuan teknologi informasi sangatlah cepat, kebutuhan akan aksesibilitas data yang cepat dan stabil telah mendorong penggunaan teknologi informasi disegala bidang, baik politik, ekonomi, kesehatan, hiburan dan terutama dibidang pendidikan, dimana teknologi informasi menjadi pendukung utama untuk meningkatkan mutu dan performa suatu bidang. Kemampuan komunikasi yang tidak mengenal batasan waktu melalui teknologi internet, merupakan syarat utama disetiap bidang untuk dapat terus meningkatkan efisiensi kinerja. Untuk meningkatkan performa internet tanpa batasan waktu, dibutuhkan solusi untuk menjamin ketersedia koneksi internet secara terus menerus. Solusi untuk stabilitas koneksi internet yaitu *bandwidth* yang memadai untuk kestabilan *traffic* data dan menggunakan beberapa *link* menuju ke internet. Menggunakan beberapa *link*/jalur internet maka koneksi internet tetap tersedia setiap saat (*high availability*)[1]. Dengan menggunakan beberapa *link* internet dibutuhkan teknik *load balancing* untuk menbagi *traffic* data internet pada beberapa *link* tersebut, sehingga *bandwidth* digunakan secara keseluruhan pada semua *link*.

Berdasarkan jurnal berjudul "Implementasi *ProxyServer* Dan *Load balancing* Menggunakan

Metode *Per Connection Classifier* (PCC) Berbasis Mikrotik (Studi Kasus : Shmily.net)" menyatakan dalam penggunaan *load balancing* ketika salah satu *linkdown* maka akan otomatis di-*backup* oleh koneksi yang lain[2]. Dan berdasarkan jurnal berjudul "Analisa *Performance* Jaringan Komputer Dengan Mekanisme *Load Balancing+Failover*" menyatakan *load balancing* yang digunakan pada Mikrotik terdiri dari ECMP, NTH dan PCC. ECMP menggunakan algoritma *equal cost*, NTH menggunakan prinsip penjadwalan (*round robin*) dan PCC menggunakan proses *hashing* dan *comparing* dalam menentukan *traffic* data yang perlu diteruskan[3].

II. KAJIAN PUSTAKA

A. *Failover*

Failover merupakan teknik yang memungkinkan dialihkannya beberapa*traffic* dari satu *link* yang mengalami masalah ke *link* yang tidak bermasalah[3]. Konsep *failover* terdiri dari 2 macam, yaitu :

a) *Load balancing* + *Failover* merupakan teknik yang menggunakan dua *link* atau lebih secara bersamaan. Apabila ada salah satu *link* yang putus atau bermasalah maka semua *traffic* akan dialihkan ke *link* yang tidak bermasalah.

b) *Only Failover* yaituketika terdapat kondisi tidak menggunakan teknik *load balancing*, namun memiliki beberapa *link*, *router* bisa menerapkan teknik *failover* saja.

B. *Equal Cost Multi Path* (ECMP)

Equal Cost Multi Path (ECMP) adalah pemilihan jalur secara bergantian pada *gateway* dan merupakan salah satu metode *load balancing* yang paling mudah dan paling cepat untuk melakukan *load balancing* atau membagi beban *traffic* menuju internet[4].

a) *Round Robin* adalah teknik pemilihan *gateway* secara acak (*roundrobin*) yang akan digunakan oleh *router* untuk mengirim paket data sampai ketujuan.

b) *Per address pair load balancing yaitu* acuan untuk menentukan *gateway* atau *link* internet yang digunakan. Acuan yang digunakakan yaitu src-*address* dan *dst-address*.

c) *Flushing routing table* setiap 10 menit. Flushing atau pembersihan tabel *routing* ini bisa menyebabkan *traffic* berpindah dari *link* akses internet satu ke yang lain.

C. NTH

Nth *load balancing* merupakan teknik *load balancing* yang membentuk suatu deret tertentu (Nth), yang nantinya digunakan sebagai suatu sistem antrian di dalam *firewall mangle*. Nth diimplementasikan dalam suatu deret yang terdiri dari *every* dan *packet* yang akan direalisasikan dalam suatu deret integer. Pada metode *load balancing* ini, paket data yang masuk akan ditandai sebagai suatu variabel n dalam tipe data integer. Nilai interger pada *every* merupakan jumlah kelompok yang ingin dibentuk, jadi jika beban ingin dibagi ke dalam 2 kelompok, maka *every* akan bernilai 2. Sedangkan pada *packet*, nilai integer berupa urutan antrian yang dimulai dari angka 1, 2, 3 dan seterusnya atau paket yang masuk akan dikenal sebagai paket 1, 2, 3 dan seterusnya. Dengan aturan yang ada, jalur yang telah ditandai sebagai nth ini akan digabungkan, atau total *bandwidth* pada keluaran merupakan penjumlahan dari masing-masing *bandwidth* pada 2 koneksi[5]. Di dalam Nth terdapat variabel yang harus dimengerti, yaitu: *Every, Packet* dan *Counter*.

D. *Per Connection Classifier* (PCC)

Per Connection classifier (PCC) merupakan metode yang mengelompokkan *traffic* koneksi ke *router* menjadi beberapa kelompok. Pengelompokan ini bisa dibedakan berdasarkan *src-address*, *dst-address*, *src-port* dan atau *dst-port*. Sehingga dengan pengelompokan koneksi, *traffic* menggunakan atau dilewatkan dari semua link yang ada. *Router* akan menyimpan informasi tentang *linkgateway* yang dilewati data ditiap *traffic* koneksi, sehingga pada *packet-packet* selanjutnya yang masih berkaitan dengan *packet* data sebelumnya akan dilewatkan pada jalur *gateway* yang sama juga[6].

E. *QUALITY OF SERVICE*

Quality of Service (QoS) merupakan metode pengukuran seberapa baik jaringan dan merupakan suatu usaha untuk mendefinisikan karakteristik dan sifat dari satu servis. Parameter QoS yang diamati pada penelitian ini adalah parameter *delay*, *throughput*, *Packet loss*, dan *jitter*[7].

a) *Delay* merupakan waktu yang dibutuhkan data untuk menempuh jarak dari asal ke tujuan. *Delay* dapat dipengaruhi oleh jarak, media fisik, kongesti atau juga waktu proses yang lama.

Tabel 2.4 Standarisasi *Delay*[8]

Kategori	*Delay* (ms)	Index
Sangat Bagus	< 150	4
Bagus	150 - 300	3
Sedang	300 - 450	2
Jelek	>450	1

b) *Jitter* diakibatkan oleh variasi dalam panjang antrian, dalam waktu pengolahan data, dan juga dalam waktu penggabungan ulang paket-paket. *Jitter* lazimnya disebut variasi *delay*, berhubungan erat dengan *latency*.

Tabel 2.5 Standarisasi *Jitter*[8]

Kategori	Peak *Jitter* (ms)	Index
Sangat Bagus	0	4
Bagus	0 - 75	3
Sedang	75 - 125	2

Jelek	125 - 255	1

c) *Packet loss* merupakan *packet* yang rusak atau hilang selama proses pengirimin *packet* berlansung. *Packet* yang hilang dapat terjadi karena *collision* dan *congestion* pada jaringan.

Tabel 2.6 Standarisasi *Packet loss*[8]

Kategori	*Packet loss* (%)	Index
Sangat Bagus	0	4
Bagus	3	3
Sedang	15	2
Jelek	25	1

d) *Throughput* merupakan kecepatan (rate) transfer data efektif, yang diukur dalam bps.

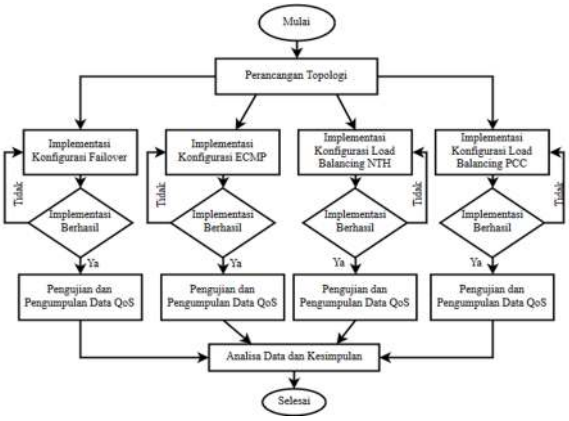
Tabel 2.7 Standarisasi *Throughput*[8]

Kategori	*Throughput (*Mbps*)*	Index
Terbaik	> 2.1	4
Lebih Baik	1,2 – 2,1	3
Baik	0,7 – 1,2	2
Cukup Baik	0,338 – 0,7	1
Buruk	0 – 0,338	0

III. METODE

Perancangan jaringan diterapkan teknik *load balancing* dengan metode *Equal Cost Multi Path* (ECMP), NTH, *Per Connection Classifier* (PCC) dan *Failover (recursive gateway)* menggunakan empat *link* internet. Dimana setiap *link* internet memiliki *bandwidth* 100 Mbps, total *bandwidth* menjadi 400 Mbps.

Berikut adalah alur perancangan umum penelitian, pada Gambar 3.1 *Flowchart* merupakan langkah-langkah yang akan dilakukan selama penelitian dan digambarkan langkah-langkah secara umum.

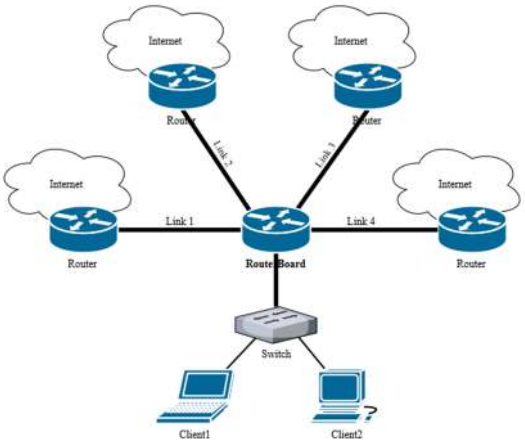


Gambar 3.1*Flowchart* Perancangan Umum

Gambar 3.1 Menunjuk alur dari pengerjaan penelitian ini, dimana penjelasan secara detail setiap alur sebagai berikut

A. Perancangan Topologi

Desain topologi yang digunakan yaitu menggunakan empat *link* internet, *mikrotik* , *switch* dan beberapa *client*. Bentuk topologi yang digunakan seperti pada gambar 3.2 di bawah.



Gambar 3.2 Topologi Implementasi Jaringan

B. Konfigurasi Jaringan

Untuk konfigurasi pada penelitian ini akan menggunakan konfigurasi, yaitu: konfigurasi *failover* dan *load balancing* ECMP, NTH dan PCC.

C. Implementasi

Pada tahap implementasi akan dilakukan implementasi secara *real* pada perangkat MikroTik.

D. Pengujian Jaringan

Pada tahan pengujian akan dilakukan pengujian secara bergilir dari konfigurasi yaitu *failover* dan *load balancing*. Pengujian parameter QoS dilakukan dengan cara *clientdownloadfile* di internet dan men-*capturetrafficdownload* menggunakan *wireshark*. Untuk *file* yang *download* sesuai dengan tabel 3.2. *Filedownload*

Tabel 3.2 *FileDownload*

No	*LinkDownload*	*Size* (MB)	*RealSize* (MB)
1	bit.ly/*file*50MB	50	54
2	bit.ly/*file*100MB	100	100.60
3	bit.ly/*file*150MB	150	166.33
4	bit.ly/*file*200MB	200	212.43
5	bit.ly/*file*250MB	250	260.50
6	bit.ly/*file*300MB	300	315.00
7	bit.ly/*file*350MB	350	354.86
8	bit.ly/*file*400MB	400	404.64
9	bit.ly/*file*450MB	450	463.46

10	bit.ly/*file*500MB	500	507.07

E. Pengumpulan Data dan Analisa QoS

Tahap akhir yaitu mengumpul data dan analisa untuk mendapatkan kesimpulan dari penelitian. Parameter-parameter data dan analisa QoS yaitu: *Throughput, Delay, Jitter* dan *Packet loss*.

IV. PEMBAHSAN DAN HASIL

Implementasi penelitian menggunakan empat *link* internet dengan masing-masing *bandwidthup to* 100 Mbps untuk *Download* dan 25 Mbps untuk *Upload*, dengan total *bandwidthdownload* 400 Mbps dan *upload* 100 Mbps. Pengujian QoS (*Quality of Service*), parameter QoS yang digunakan yaitu *throughput*, *delay*, *jitter* dan *packet loss*.

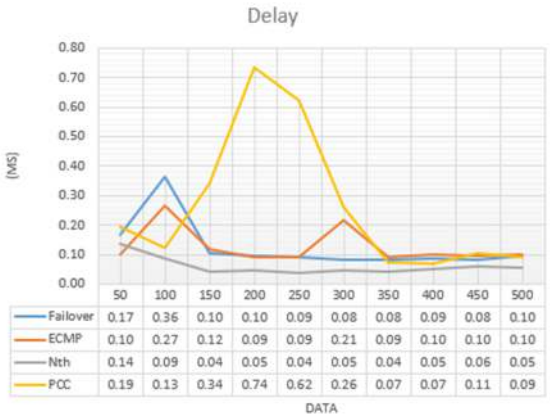
A. *Delay*

Tabel 4.1 Rata-rata *Delay*

Size (MB)	*Failover*	ECMP	Nth	PCC
50	0.7	0.10	0.14	0.19
100	0.36	0.27	0.09	0.13
150	0.10	0.12	0.04	0.34
200	0.10	0.09	0.05	.74
250	0.09	0.09	0.04	0.62
300	0.08	0.21	0.05	0.26
350	0.08	0.09	0.04	0.07
400	0.09	0.10	0.05	0.07
450	0.08	0.10	0.06	0.11
500	0.10	0.10	0.05	0.09

Gambar 4.23 menunjukan perbandingan *delay* empat metode, ketika melakukan *downloadfile* dengan ukuran 50 MB sampai 500 MB. Grafik rata-rata *delay* untuk metode *failover* terjadi perubahan signifikan pada ukuran *file* 100 MB. Hasil dari analisa *capturewireshark* menunjukan tidak stabilnya koneksi *client* dengan *serverdownload*, sedangkan untuk ukuran *file* yang lain tidak terjadi perubahan *delay* yang signifikan. Metode ECMP pada grafik rata-rata *delay* terjadi perubahan yang signifikan pada ukuran *file* 100 MB dan 300 MB yang disebabkan karena tidak stabilnya koneksi *client* dengan *serverdownload*. Tidak stabilnya koneksi yaitu ketika transmisi *packet* antara *client* dengan *server* dalam waktu tertentu tidak ada *packet* yang terkirim/ diterima, yang mengakibatkan terjadinya *delay* yang panjang antara *packet* satu dengan yang lain.

Metode PCC pada grafik rata-rata *delay* mengalami penurunan yang signifikan pada ukuran *file* 150 MB, 200 MB, 250 MB dan 300 MB. Hal ini disebabkan karena tidak stabil koneksi pada rentang ukuran *file* tersebut sehingga membutuhkan waktu lebih lama untuk *downloadfile* tersebut. Metode Nth pada grafik rata-rata *delay* terlihat tidak terjadi perubahan yang signifikan pada grafik karena pada metode Nth diterapkan teknik *mark-packet*. Nth membuat antrian *packet* untuk semua *link* yang tersedia, sehingga transmisi *packet* pengirim ke penerima lebih stabil karena menggunakan semua *link* dan mengurangi *delay* setiap *packet*.



Gambar 4.1 Grafik Rata-rata *Delay*

Dari semua metode yang digunakan dapat disimpulkan pada beberapa ukuran *filedownload* terjadi penurunan koneksi pada metode *Failover*, ECMP dan PCC dan untuk *delay* dengan index paling bagus yaitu ECMP. Ukuran *file* lebih kecil tidak menentukan *delay* lebih bagus, dapat dilihat pada grafik rata-rata *delayfile* dengan ukuran lebih besar memiliki *delay* yang stabil, berarti pada implementasi ini *delay* lebih dipengaruhi konektivitas *clientserver* dan metode yang digunakan.

B. *Jitter*

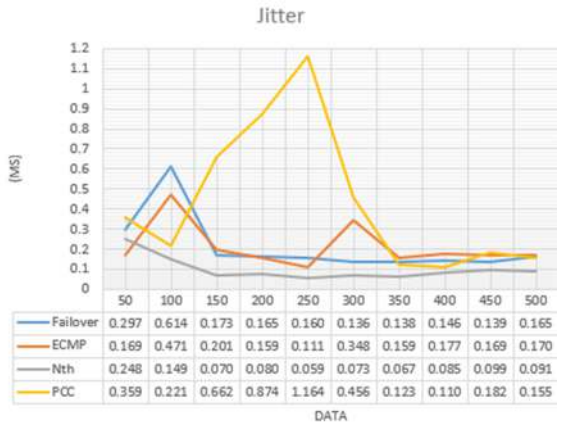
Jitter adalah variasi dari *delay*, variasi *delay* merupakan variasi kedatangan *packet* satu dengan *packet* lain.

Tabel 4.2 *Jitter*

Size (MB)	*Failover*	ECMP	Nth	PCC
50	0.297	0.169	0.248	0.359
100	0.614	0.471	0.149	0.221
150	0.173	0.201	0.070	0.662
200	0.165	0.159	0.080	0.874
250	0.160	0.111	0.059	1.164
300	0.136	0.348	0.073	0.456
350	0.138	0.159	0.067	0.123
400	0.146	0.177	0.085	0.110

450	0.139	0.169	0.099	0.182
500	0.165	0.170	0.091	0.155

Gambar 4.2 Grafik *Jitter* dapat dilihat bahwa grafik *jitter* hampir sama dengan grafik *delay* pada Gambar 4.23, namun terdapat perbedaan pada metode PCC dengan ukuran *file* 200 MB dan 250 MB. Dimana pada Gambar 4.23 Rata-rata *Delay* menunjukan grafik yang lebih tinggi pada *file* 200 MB dari pada 250 MB, sedangkan pada *jitter* terjadi sebaliknya, dilihat dari hasil *capturepacket* ini disebabkan karena besarnya *delaypacket* satu dengan yang lain sehingga didapatkan variasi *delay* yang besar pada *file* 250 MB. Untuk metode yang lain, tidak terlihat perbedaan yang signifikan pada grafik *jitter* maupun *delay*.

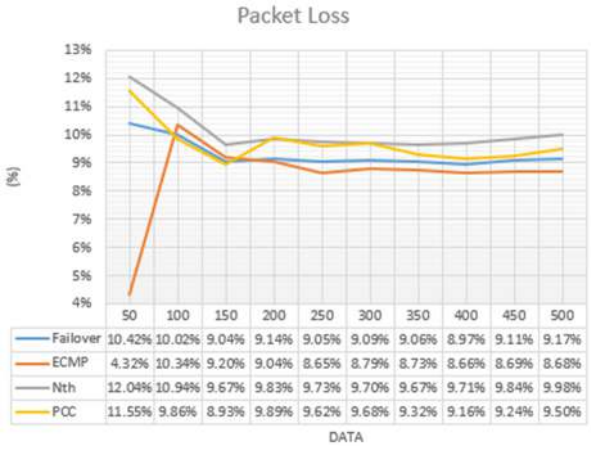


Gambar 4.2 Grafik *Jitter*

C. *Packet loss*

Tabel 4.3 *Packet loss*

Size (MB)	*Failover*	ECMP	Nth	PCC
50	10.42%	4.32%	12.04%	11.55%
100	10.02%	10.34%	10.94%	9.86%
150	9.04%	9.20%	9.67%	8.93%
200	9.14%	9.04%	9.83%	9.89%
250	9.05%	8.65%	9.73%	9.62%
300	9.09%	8.79%	9.70%	9.68%
350	9.06%	8.73%	9.67%	9.32%
400	8.97%	8.66%	9.71%	9.16%
450	9.11%	8.69%	9.84%	9.24%
500	9.17%	8.68%	9.98%	9.50%



Gambar 4. 3 Grafik *Packet loss*

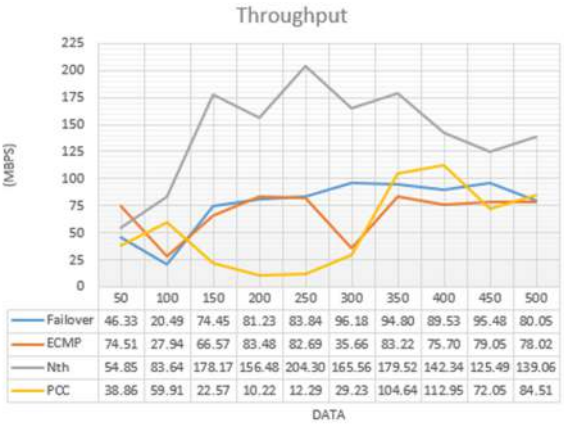
Gambar 4.25 Grafik *Packet loss* merupakan nilai dari parameter *packet loss* hasil olah data pada *capturepacketwireshark*. Dilihat pada grafik terdapat perbedaan grafik yang signifikan pada *file* ukuran 50 MB untuk metode ECMP. Hasil dari *capturepacketfile* 50 MB terdapat beberapa IP address untuk *downloadfile* ini yang menunjukan perbedaan *serverdownload*. Untuk metode ECMP koneksi *client* dengan *server* memilik IP address 74.125.10.43, sedangkan untuk metode *Failover*, NTH dan PCC koneksi *client* dengan *server* memiliki IP address 118.98.111.208. Dengan perbedaan pada IP address tentu ini berpengaruh karena terjadinya perbedaan jalur/*link* koneksi *client* dengan *server*.

Pada grafik juga dapat dilihat *packet loss* rata-rata paling tinggi terjadi dimetode NTH ini sesuai dengan teknik *mark-packet,* dimana semua *packet* dibuatkan antrian sehingga antara *packet* sehingga menggunakan semua *link*, namun dengan begitu juga besar kemungkinan terjadi collision pada *packet*. Meskipun begitu *packet loss* untuk semua metode dapat dikatakan rata-rata stabil karena tidak banyak perubahan signifikan pada setiap ukuran *file* downlad.

D. *Throughput*

Tabel 4.4 *Throughput*

Size (MB)	*Failover*	ECMP	Nth	PCC
50	46.33	74.51	54.85	38.86
100	20.49	27.94	83.64	59.91
150	74.45	66.57	178.17	22.57
200	81.23	83.48	156.48	10.22
250	83.84	82.69	204.30	12.29
300	96.18	35.66	165.56	29.23
350	94.80	83.22	179.52	104.64
400	89.53	75.70	142.34	112.95
450	95.48	79.05	125.49	72.05
500	80.05	78.02	139.06	84.51



Gambar 4. 4 Grafik *Throughput*

Gambar 4.26 merupakan grafik hasil *capture* untuk parameter *throughput* . Dilihat pada grafik tidak stabilnya nilai *throughput* disetiap rentang ukuran *file*. Metode *failover* pada grafik menunjukan *throughput* yang stabil, namun terdapat penurunan pada ukuran *file* 100 MB. Dari hasil *capture* pada *file*100 MB terjadi penurunan *throughput* karena tidak stabilnya koneksi *client* dengan *server* sehingga menyebabkan lamanya proses *download*. Metode ECMP pada grafik terjadi perubahan yang signifikan pada ukuran *file* 100 MB dan 300 MB, dari hasil *capture* pada *file* 100 MB koneksi *client* dengan *server* tidak stabil, sedangkan pada *file* 300 MB koneksi stabil, akan tetapi *delaypacket* cukup besar sehingga mengakibat *throughput* menurun. Selain itu pada metode ECMP terdapat algoritma *round robin* sehinga mengakibat perpindahan *gateway* / *link* yang digunakan setiap 10 menit.

Metode PCC pada grafik menunjukan *throughput* tidak stabil setiap rentang *filedownload*. Hasil dari *capture* setiap *packet* pada metode PCC tidak stabilnya koneksi, transmisi *packet* tidak stabil dalam satuan waktu dapat dilihat pada Graph I/O *wireshark*. Tidak stabilnya diakibat karena klasifikasi koneksi berdasarkan *both-addresses-and-ports*. Klasifikasi ini dalam satu koneksi *filedownload* akan dipecah berdasarkan address dan port, sehingga menggunakan semua *link* yang tersedia akan tetapi dengan dipecahnya koneksi berakibat *delay* yang besar dan lamanya waktu *download*. Metode Nth pada grafik menunjukan hasil *throughput* yang jauh berbeda dangan metode yang lain yaitu lebih baik dari semua metode. Tingginya *throughput* pada Nth terjadi karena penggunaan antrian *mark-packet* dimana semua *link* yang tersedia digunakan untuk transmisi *packet*, sehingga tidak terjadi penumpukan *packet* hanya pada satu *link*, meskipun tidak stabilnya grafik yang dapat diakibat koneksi *client* dengan *server*.

V. PENUTUP

A. KESIMPULAN

Metode *Failover* tidak bisa menggunakan *link* secara bersamaan dalam satu waktu, atau terdapat *primarylink* dan backup *link*. Sehingga tidak menggunakan semua *bandwidth* yang tersedia dalam satu waktu, tapi hanya menggunakan *bandwidth* dari satu *link* saja. Metode ECMP (*Equal Cost Multi Path*) dapat menggunakan semua *link* bersamaan dalam satu waktu, tetapi tidak menggunakan semua *bandwidthlink* yang tersedia, karena dalam satu waktu hanya bisa menggunakan satu *gateway* untuk setiap koneksi. Metode Nth berdasarkan antrian (*queue* N-th) *mark-packet*, mampu menggunakan semua *link* dalam satu waktu. Meskipun metode Nth juga menggunakan satu *gateway* untuk setiap koneksi, akan tetapi dengan *mark-packet* Nth tetap bisa mentransmisikan setiap *packet* pada *gateway*/*link* yang berbeda sehingga menggunakan semua *bandwitdh* yang tersedia. Metode PCC (*per-connection classifier*) yang menggunakan *mark-connection* atau membuat klasifikasi koneksi berdasarkan *both-addresses-and-ports*. Sehingga PCC menggunakan semua *link* dan *bandwidth* yang tersedia. Hasil *Quality of Service* empat metode yang diterapkan dibandingkan yang terbaik berdasar parameter *delay*, *jitter*, *packetloss*dan *throughput*. Pameter *delay* ada Nth dengan nilai *delay* yang rata-rata terendah. Parameter *Jitter* ada Nth dengan nilai *jitter* yang rata-rata rendah. Parameter *packetloss* ada ECMP dengan nilai *packetloss* yang kecil. Parameter *throughput* ada Nth dengan nilai *throughput* dengan rata-rata tertinggi.

B. SARAN

Untuk penelitian selanjutnya yang berkaitan dengan failover dan load balancing, bisa lebih secara detail pada setiap metode karena dalam satu metode saja terdapat banyak klasifikasi koneksi. Menggabungkan setiap metode load balancing sesuai dengan kriteria dan kebutuhan. Menambah/mengurangi jumlah *link* yang digunakan ataupun *bandwidth* setiap *link*.

DAFTAR PUSTAKA

[1] F. Wahyu Santoso, "ANALISIS KINERJA LOAD BALANCINGMENGGUNAKAN METODE PEER CONNECTION CLASSIFIER (PCC) PADA PONDOK PESANTREN AL-JAILANI YOGYAKARTA," UNIVERSITAS ISLAM NEGERI SUNAN KALIJAGA, 2014.

[2] F. J. Bhayangkara and I. Riadi, "Implemetasi Proxy Server dan Load Balancing Menggunakan Metode Per Connection Classifier (PCC) Berbasis Mikrotik (Studi kasus : Shmily.net)," *Sarj. Tek. Inform.*, vol. 2, pp. 1206–1217, 2014.

[3] R. Ramandito, S. Suning Kusumawardhani, and S. Sumaryono, "ANALISA PERFORMANCE JARINGAN KOMPUTER DENGAN MEKANISME LOAD BALANCING-FAILOVER," *Anal. Perform. Jar. Komput. DENGAN Mek. LOAD Balanc.*, vol. 3, no. 4, pp. 177–181, 2010.

[4] R. Towidjojo, *Mikrotik Kung Fu : Kitab 4*. Jasakom, 2016.

[5] P. Ardiansyah, "IMPLEMENTASI TERHADAP METODE Nth LOAD BALANCING DAN METODE PCC LOAD BALANCING PADA 2 ISP BERBEDA DI MIKROTIK ROUTER OS," *Univ. Dian Nuswantoro Semarang*, 2012.

[6] M. Hafizh, "LOAD BALANCING DENGAN METODE PER CONNECTION CLASSIFIER (PCC) MENGGUNAKAN PROXY SERVER SEBAGAI CACHING," JAKARTA, 2011.

[7] E. Abraham Andrian Matualaga, Jusak, and P. Susanto, "ANALISIS KARAKTERISTIK LALU LINTAS DATA INTERNET: APLIKASI WEB VIDEO STREAMING," *J. Control Netw. Syst.*, vol. 3, no. 1, pp. 1–10, 2014.

[8] ETSI, "Telecommunication and Internet Protocol Harmonization Over Network (TIPHON); General Aspects of Quality of Service (Qos)," 1999.