

Pengendalian Kemacetan Jaringan Melalui *Per-Flow Multipath Routing*

Fransiska Sisilia Mukti¹⁾, Achmad Basuki²⁾, Onny Setyawati³⁾

Abstrak—Kemacetan terjadi ketika jumlah paket yang ditransmisikan melalui jaringan telah mendekati kapasitas penanganan paket jaringan. Hal ini menjadi faktor utama untuk diketahui lebih dini, guna menghindari adanya kegagalan dalam proses routing. Sebagian besar protokol routing yang digunakan saat ini menggunakan algoritma yang menghasilkan jalur tunggal saja, tanpa memperhatikan permintaan trafik yang bersifat fluktuatif. Penggunaan jalur tunggal dapat menyebabkan terjadinya kemacetan yang berdampak kepada pemborosan sumber daya. Penelitian ini bertujuan untuk melakukan pengendalian kemacetan jaringan dengan mengkombinasikan mekanisme *multipath routing* dan *congestion control* pada protokol routing OSPF. Hasil pengujian menunjukkan keunggulan sistem dalam tiga aspek, yaitu kestabilan nilai *throughput* (100%) sehingga tidak ada *packet loss*, pengiriman data 50% lebih cepat, dan utilisasi jaringan yang lebih baik.

Kata Kunci—kemacetan jaringan, *multipath routing*, OSPF, utilisasi bandwidth

I. PENDAHULUAN

JARINGAN INTERNET telah mengubah pola hidup manusia dalam bertukar informasi sehingga menggiring adanya revolusi komunikasi secara signifikan. Pertumbuhan jumlah pengguna dan layanan pada jaringan berbasis IP berdampak kepada peningkatan kualitas *provider* jaringan untuk memenuhi permintaan pengguna. Untuk merespon hal ini, diperlukan adanya optimasi kinerja jaringan melalui adanya rekayasa lalu lintas dengan memanfaatkan sumber daya jaringan secara efisien dan handal [1].

Sebagian besar protokol *routing* yang digunakan saat ini menggunakan algoritma yang menghasilkan jalur tunggal saja. Perhitungan ini dilakukan secara otomatis dengan menggunakan nilai satuan beban (metrik) yang dihitung berdasarkan jarak, *delay*, *bandwidth*, atau kombinasi keduanya. Selain itu, nilai metrik ini ditetapkan tanpa mempertimbangkan permintaan trafik atau beban trafik pada jaringan[2].

¹⁾Fransiska Sisilia Mukti, merupakan mahasiswi program magister Teknik Elektro, Minat Sistem Komunikasi dan Informatika, Universitas Brawijaya, Malang, Indonesia (e-mail: fransiskasisilia.mukti@gmail.com).

²⁾Achmad Basuki, merupakan dosen Teknik Elektro, Universitas Brawijaya, Malang, Indonesia (telp: 0341-551611; fax: 0341-565420; e-mail: abash@ub.ac.id).

³⁾Onny Setyawati, merupakan dosen Teknik Elektro, Universitas Brawijaya, Malang, Indonesia (telp: 0341-551611; fax: 0341-565420; e-mail: onnysetyawati@ub.ac.id).

Open Shortest Path First (OSPF), merupakan protokol *routing* dinamis berbasis *link-state* yang menggunakan algoritma Dijkstra untuk melakukan komputasi jalur terbaiknya berdasarkan jumlah nilai *cost* yang terkecil, dari node sumber ke node tujuan. Hasil perhitungan ini akan menghasilkan *single-path routing*. Jika nilai *cost* yang ditetapkan tidak optimal, maka trafik dalam jaringan bisa saja terpusat pada salah satu jalur dan dapat menyebabkan peningkatan rasio kemacetan dalam jaringan [3].

Kemacetan terjadi ketika jumlah paket yang ditransmisikan melalui jaringan telah mendekati kapasitas penanganan paket yang dapat dilakukan [4]. Penentuan nilai metrik akan mempengaruhi hasil komputasi jalur. Sedangkan penggunaan jalur tunggal dalam proses transmisi data pada siklus trafik yang cenderung tinggi akan menyebabkan terjadinya kemacetan yang berdampak kepada pemborosan sumber daya jaringan [5].

Multipath routing (MP) merupakan pendekatan alternatif yang dapat digunakan untuk menyelesaikan permasalahan yang ditimbulkan dari penggunaan *single-path routing* (SP). Peningkatan performa jaringan dengan MP dicapai melalui penggunaan beberapa jalur secara bersama selama proses transmisi data berlangsung. Pendekatan ini membuat penggunaan sumber daya jauh lebih efektif dan meningkatkan *throughput* jaringan [1].

Penggabungan antara teknik *multipath routing* dan pengendalian kemacetan jaringan merupakan salah satu contoh optimasi *cross-layer*, yang menawarkan keuntungan peningkatan performa melalui pemisahan layer. *Routing* menentukan jalur mana yang seharusnya digunakan, sedangkan *congestion control* (CC) menentukan seberapa banyak trafik yang dapat dilewatkan pada setiap jalur [6].

Salah satu metode rekayasa lalu lintas melalui MP adalah *Equal Cost Multi Path* (ECMP). ECMP memungkinkan adanya penggunaan jalur dengan nilai metrik yang sama dari node sumber ke node tujuan. Trafik akan dibagi secara merata ke seluruh jalur yang terdefinisi oleh ECMP, dengan tujuan menghindari kemacetan dan meningkatkan kecepatan dalam proses pengiriman data [7].

Penelitian ini bertujuan untuk melakukan pengendalian kemacetan jaringan melalui rekayasa lalu lintas (*traffic engineering*) dengan mengkombinasikan mekanisme CC+MP pada protokol *routing* berbasis *link-state Open Shortest Path First* (OSPF). Kemacetan

diidentifikasi berdasarkan penggunaan sumber daya pada *router* kemudian melakukan modifikasi nilai *cost* untuk membentuk beberapa jalur (*multi-path*). Dijkstra menjalankan tugasnya untuk melakukan komputasi jalur terbaik, dan ECMP mendistribusikan trafik melalui beberapa jalur. Analisa performa jaringan dilakukan dengan melakukan perbandingan nilai *throughput* dan *latency* antara jalur tunggal pada OSPF konvensional dengan implementasi *multipath-routing* (ECMP) pada OSPF.

II. OPEN SHORTEST PATH FIRST (OSPF)

OSPF merupakan salah satu protokol *routing* berbasis teknologi *Shortest Path First* (SPF) atau *link-state*. Masing – masing *router* memiliki *link-state database* yang identik, yang mendeskripsikan topologi kondisi jaringan pada sebuah, yaitu informasi mengenai topologi *Autonomous System* (AS), *interface* yang tersedia, *neighbor router*, dan informasi lain berkaitan dengan jalur pendistribusian data.

OSPF merupakan protokol *routing* yang menggunakan konsep hirarki *routing*, artinya OSPF membagi-bagi jaringan menjadi beberapa tingkatan. Tingkatan-tingkatan ini diwujudkan dengan menggunakan sistem pengelompokan area. Konsep ini menjadikan sistem penyebaran informasinya menjadi lebih teratur dan tersegmentasi. Efek dari keteraturan distribusi *routing* ini adalah jaringan yang penggunaan *bandwidth*-nya lebih efisien, lebih cepat mencapai konvergensi, dan lebih presisi dalam menentukan rute-rute terbaik menuju ke sebuah lokasi. Hal ini membuat protokol *routing* OSPF menjadi sangat cocok untuk terus dikembangkan menjadi jaringan berskala besar[8].

A. Algoritma Dijkstra

Algoritma Dijkstra merupakan salah satu jenis algoritma *Shortest Path Computation* (SPF) yang digunakan untuk mencari lintasan terpendek pada sebuah graf berarah dengan menggunakan nilai satuan bobot (*cost*). Cara kerja algoritma Dijkstra memakai strategi *greedy*, dimana pada setiap langkah dipilih sisi dengan bobot terkecil yang menghubungkan sebuah simpul yang sudah terpilih dengan simpul lain yang belum terpilih [9].

Kontribusi algoritma Dijkstra dalam dunia komputer secara khusus dalam dunia *routing*, digunakan oleh OSPF. Masing – masing *router* dalam OSPF akan secara otomatis mengatur nilai *cost* berdasarkan alokasi *bandwidth* yang tersedia pada masing- masing *interface*, atau dapat juga diatur secara manual oleh *administrator* jaringan. Nilai *cost* dalam sebuah *interface* memiliki nilai berbanding terbalik dengan nilai *bandwidth* pada *interface* tersebut. Nilai *bandwidth* tertinggi mengindikasikan nilai *cost* terkecil. Persamaan (1) menunjukkan rumus yang digunakan untuk menghitung nilai *cost* secara otomatis [10].

$$\text{Cost} = \frac{\text{Reference_Bandwidth}}{\text{Interface_Bandwidth}} \quad (1)$$

di mana, $\text{Reference_Bandwidth} = 10^8$ in bps

B. Equal Cost Multi Path (ECMP)

Protokol *link-state* seperti OSPF dan IS-IS merupakan protokol *routing* yang berbasis algoritma *Shortest Path First* (SPF) untuk menghitung *single shortest path* (jalur terpendek tunggal) dari *node* sumber ke *node* tujuan. ECMP merupakan sebuah teknik yang memungkinkan penggunaan beberapa jalur yang memiliki nilai *cost* yang sama dalam perutean. Fitur ini bukan hanya akan membantu mendistribusikan lalu lintas lebih merata, tetapi juga sebagai metode proteksi atau keamanan dalam jaringan. Dengan menggunakan ECMP, jalur yang memiliki nilai *cost* yang sama akan dipasang ke dalam table *load balancing* di lapisan *forwarding* pada *router*. Pada saat *router* mendeteksi adanya kegagalan *link*, trafik akan didistribusikan secara otomatis kepada jalur kedua dengan tanpa adanya kehilangan trafik yang cukup signifikan.

ECMP tidak memerlukan konfigurasi secara khusus, sebab algoritma SPF akan menghitung secara otomatis jalur-jalur dengan nilai *cost metric* yang sama, untuk kemudian melakukan *advertising* kepada layer *forwarding* pada *router*. Jumlah jalur ECMP bergantung kepada algoritma *load balancing* yang mendukung. Umumnya, dikonfigurasi antara 1 sampai 8 dan 16 jalur, bergantung kepada nilai maksimum dari jalur yang ada [7].

III. METODOLOGI PENELITIAN

A. Pengamatan Tingkah Laku Jaringan

Kasus yang sering terjadi berkaitan dengan kemacetan jaringan adalah kemungkinan terjadinya konflik dalam proses *routing*, beberapa trafik dari sumber yang berbeda namun memiliki tujuan yang sama, sehingga menyebabkan *overutilization* pada sebagian *path* pada jaringan. Hasil pengamatan tingkah laku jaringan OSPF menghasilkan kesimpulan sebagai berikut.

- OSPF menggunakan nilai *cost metric* sebagai satu-satunya variabel penentu jalur pendistribusian data
- Jalur dengan nilai *cost metric* terkecil dianggap sebagai jalur terbaik
- Paket data dengan tujuan yang sama akan mengalami penumpukan pada salah satu jalur dengan nilai *cost metric* terkecil
- Penumpukan data pada salah satu jalur menyebabkan kemacetan jaringan.

B. Perancangan Sistem

Konsep sistem yang dibangun berpedoman kepada kesimpulan penyelesaian masalah berikut ini.

- Kepadatan jalur mengindikasikan tingkat kemacetan jaringan.
- Tingkat kemacetan jaringan didasarkan pada nilai utilisasi *bandwidth* pada setiap jalur.
- Pengubahan nilai *cost metric* dilakukan secara dinamis berdasarkan tingkat kemacetan jaringan

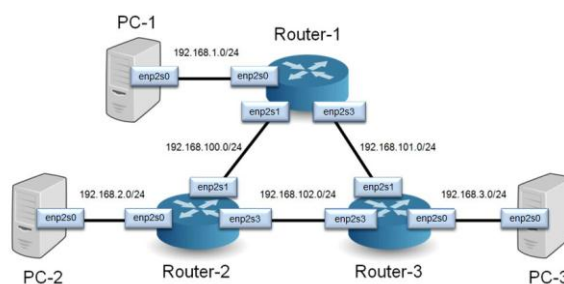
- d. Jalur terbaik merupakan jalur dengan tingkat kemacetan rendah namun memiliki waktu pendistribusian data tercepat

Kemacetan diidentifikasi apabila jumlah trafik yang akan dilewatkan melalui suatu *interface* telah mencapai 80% dari kapasitas *bandwidth* yang tersedia. Untuk mencegah terjadinya kemacetan pada jaringan maka sistem akan memberikan limitasi *bandwidth link* antar *router* sebesar 10Mbps. Limitasi *interface* dilakukan menggunakan aturan *traffic control* dengan metode antrian *Hierarchical Token Buckets (HTB)*.

Pada saat jumlah trafik yang akan dikirimkan melalui *interface* tersebut mencapai batasan limitasi tersebut, maka *router* akan mendeteksinya sebagai suatu perintah untuk mengalihkan jalur pengiriman data berdasarkan list jalur yang telah didefinisikan oleh ECMP. ECMP akan melakukan pembagian beban jaringan (*load balancing*) melalui penggunaan beberapa jalur dengan nilai *cost metric* yang sama dari *node* sumber ke *node* tujuan.

C. Simulasi

Untuk melakukan evaluasi performansi terhadap usulan sistem, maka dibangun sebuah lingkungan uji coba (*testbed*) protokol *routing* OSPF dengan menggunakan Quagga. Sistem dibuat dengan menghubungkan 3 buah PC yang dibangun sebagai PC-Router, yang ketiganya masing-masing terhubung dengan 1 buah PC-Client, dengan topologi jaringan yang tertera pada Gambar 1.



Gbr. 1. Topologi Jaringan Simulasi OSPF

Setiap *link* yang menghubungkan antar *node* menggunakan *Ethernet* dengan kapasitas sebesar

TABEL I
PARAMETER UJI COBA

Parameter	Value
Testbed	Quagga
Routing protocol	OSPF
Media	Ethernet
Link Capacity	100Mbps
Threshold	10Mbps
Time	50s, 100s, 300s, 600s
Number of flows	4
Size of flows	4Mbps
Traffic type	TCP, UDP
Tool	Iperf

100Mbps, sedangkan nilai *cost metric* untuk *Ethernet* adalah sebesar 10 (nilai *cost* terdefinisi secara otomatis oleh OSPF). Tabel I menunjukkan parameter yang digunakan untuk simulasi penelitian.

Simulasi dilakukan dengan mengirimkan 4 aliran trafik sekaligus secara bersamaan dari PC-1 ke PC-3, dengan interval waktu mulai dari 50s, 100s, 300s hingga 600s. Setiap *link* antar *router* diberikan *threshold* sebesar 10Mbps, sehingga jika terdapat 4 aliran trafik @4Mbps (total trafik = 4x4Mbps = 16Mbps) yang dikirimkan melalui jalur tersebut, akan menimbulkan kemacetan jaringan, karena nilai *input traffic* (16Mbps) > *output capacity* (10Mbps). Trafik dikirimkan dengan menggunakan *tool iperf* untuk protokol UDP untuk menghasilkan evaluasi nilai *packet loss* dan *jitter* pada sisi *client*, dan menggunakan *iperf3* untuk protokol TCP untuk menghasilkan evaluasi nilai *throughput* pada sisi *router*.

IV. HASIL DAN PEMBAHASAN

Untuk melakukan komparasi hasil penelitian dengan performansi dari standar OSPF, dua set simulasi *routing* dijalankan. Simulasi pertama, menggunakan OSPF konvensional *single-path routing*, dan simulasi kedua, menjalankan OSPF yang telah diintegrasikan dengan mekanisme CC+MP. Kedua simulasi ini dijalankan dengan menggunakan komponen yang sama, topologi jaringan yang sama, dan parameter konfigurasi jaringan yang sama.

A. End-to-End Communications

Komparasi dilakukan dengan melakukan perhitungan *end-to-end communications* terhadap nilai *jitter*, *packet loss*, dan *transfer file* pada sisi *client*. Tabel II menunjukkan perbandingan hasil rata-rata pengukuran 4 flow trafik pada protokol UDP antara OSPF *single-path* dan OSPF *multi-path*.

Jumlah transfer file selama proses pengiriman data dipengaruhi oleh waktu dan nilai *throughput*. Semakin tinggi nilai *throughput* yang dapat dihasilkan, maka jumlah data yang dapat dikirim semakin besar. Jika nilai *throughput* yang dihasilkan tidak sepadan dengan jumlah trafik flow yang dikirimkan, menandakan bahwa terjadi *packet loss* di dalamnya.

Berdasarkan hasil pengukuran protokol UDP yang tertera pada Tabel II, terlihat bahwa pada OSPF *single-path*, terdapat penurunan nilai *throughput* hingga 2.43Mbps, yang mengakibatkan adanya *packet loss* selama pengiriman data hingga mencapai 39%. Adanya penurunan nilai *throughput* menunjukkan bahwa jalur tunggal yang digunakan untuk pengiriman data mengalami kemacetan, dan OSPF konvensional tidak memiliki mekanisme khusus untuk melakukan pengendalian terhadap kemacetan tersebut.

Sebaliknya, nilai *throughput* yang dihasilkan pada OSPF *Multi-Path* adalah konstan pada nilai 4.00Mbps, sesuai dengan jumlah trafik flow yang dikirimkan. Pengiriman data berhasil dilakukan tanpa adanya *packet loss*, karena data dikirimkan melalui dua jalur sekaligus, yaitu melalui eth1 dan eth2 pada Router-1 secara bersamaan. Router melakukan pembagian beban jaringan secara merata, sehingga pengiriman data terjadi lebih cepat. Hal ini dibuktikan pada nilai

jitter pada *multi-path* lebih kecil dibandingkan *single-path*.

B. Router Communications

Setiap *link* yang menghubungkan antar *router* diberikan *threshold* sebesar 10Mbps, dengan demikian trafik yang dilewatkan tidak dapat melebihi nilai tersebut. Nilai *threshold* ini diterapkan pada masing-masing simulasi, baik untuk *single-path* maupun *multi-path*.

Selanjutnya dilakukan pengamatan terhadap jumlah total beban pada setiap *interface router* baik yang masuk ataupun trafik yang keluar. Hal ini dilakukan dengan mengirimkan 4 flow trafik dari PC-1 menuju ke PC-3, dengan besar masing-masing 4Mbps melalui protokol TCP menggunakan *tool iperf3*. Percobaan dilakukan dengan interval waktu 50s, 100s, 300s dan 600s (lihat detail parameter pada Tabel I).

Gambar 2 menunjukkan beban jaringan yang dipantau Router-3 pada OSPF *Single-Path*, sedangkan Gambar 3 menunjukkan beban jaringan Router-3 pada OSPF *Multi-Path*, dengan masing-masing waktu pengiriman trafik selama 600s. Eth-1 merupakan *interface* yang menghubungkan ke Router-1, eth-2 merupakan *interface* yang menghubungkan ke Router-2, eth-3 merupakan *interface* yang menghubungkan ke PC-3.

Pada OSPF *Single-Path*, terlihat hanya eth-1 yang

TABEL II
HASIL SIMULASI PROTOKOL UDP

Parameter	Time	Single-Path	Multi-Path
Transfer File (KBytes)	50s	14.3825	23.8
	100s	29.275	47.7
	300s	87.15	143
	600s	174.25	286
Throughput (Mbps/sec)	50s	2.4275	4
	100s	2.4275	4
	300s	2.43	4
	600s	2.4275	4
Jitter (ms)	50s	1.024	0.3625
	100s	0.74975	0.611
	300s	0.913	0.487
	600s	0.86375	0.44325
Packet Loss (%)	50s	37.75	0
	100s	38.475	0
	300s	39	0
	600s	39	0

menerima trafik, dengan nilai rata-rata trafik yang diterima sebesar 9.771528 Mbps, sehingga jumlah trafik yang ditransmisikan pada eth-3 memiliki nilai yang hampir sama dengan jumlah trafik yang diterima pada eth-1 dan eth-2. Jika jumlah trafik yang dikirimkan adalah 16Mbps (4 flow trafik @4Mbps), dan trafik yang ditransmisikan hanyalah sebesar 9.7Mbps, maka terdapat adanya *loss transmission* sebesar 39% selama pengiriman berlangsung.

Sedangkan pada Gambar 3, terlihat bahwa eth-1 dan eth-2 pada OSPF *Multi-Path* memiliki beban

jaringan yang relatif sama, yaitu rata-rata trafik yang diterima sebesar 8.387443 Mbps, sehingga trafik yang masuk akan ditransmisikan secara penuh melalui eth-3. Hal ini menunjukkan bahwa dengan menggunakan OSPF *Multipath* tidak terdapat *loss transmission* selama pengiriman data.

V. KESIMPULAN

Penggabungan antara teknik *multipath routing* dan pengendalian kemacetan jaringan pada OSPF, terbukti efektif dalam mengurangi rasio kemacetan dan meningkatkan performansi jaringan. Hal ini dicapai dengan mengendalikan kemacetan jaringan melalui pengalihan jalur dan pemerataan beban pada jaringan.

Setiap *link* yang menghubungkan antar *router* diberikan nilai *threshold*. Kemacetan diidentifikasi apabila jumlah trafik yang dikirim melewati nilai *threshold*, maka *router* langsung menerapkan mekanisme *multipath*. ECMP membantu mendefinisikan beberapa jalur yang memiliki nilai *cost* yang sama, sedangkan Dijkstra berperan dalam melakukan komputasi jalur yang terpendek.

Pengujian dilakukan melalui komparasi performa antara OSPF konvensional dan OSPF *Multipath*, berdasarkan parameter nilai *throughput*, *jitter* dan *packet loss* pada interval waktu 50s – 600s. Pada OSPF konvensional, terjadi penurunan *throughput* hingga 50% (2.4Mbps), sehingga menyebabkan terjadinya *packet loss* hingga 39%. Sedangkan pada OSPF *Multipath*, nilai *throughput* yang dihasilkan adalah sesuai dengan pengujian flow trafik yang dikirim (4Mbps) sehingga tidak terjadi adanya *packet loss* (0%). Dengan nilai *throughput* yang konstan, menjadikan OSPF *Multipath* memiliki waktu pendistribusian data yang lebih cepat, dengan nilai *jitter* mencapai 0.3ms, atau sepertiga dari waktu yang dihasilkan pada OSPF konvensional (1.02ms).

VI. DAFTAR PUSTAKA

- [1] R. Singh and A. Yadav, "Loop Free Multipath Routing Algorithm," *arXiv:1601.01245*, 2016.
- [2] H. Abrahamsson, B. Ahlgren, J. Alonso, A. Andersson, and P. Kreuger, "A multi-path routing algorithm for IP networks based on flow optimisation," *From QoS Provisioning to QoS Charg.*, pp. 135–144, 2002.
- [3] Y. Nakahodo, T. Naito, and E. Oki, "Implementation of Smart-OSPF in Hybrid Software-Defined Network," in *IEEE Proceedings of IC-NIDC 2014*, 2014, pp. 0–4.
- [4] W. Stallings, "Congestion in Data Networks." Prentice Hall, p. Chapter 13, 2003.
- [5] L. Lan, L. Li, and C. Jianya, "A Multipath Routing Algorithm Based on OSPF Routing Protocol," *2012 Eighth Int. Conf. Semant. Knowl. Grids*, pp. 269–272, 2012.
- [6] P. Key, L. Massoulié, and D. Towsley, "Multipath Routing, Congestion Control, and Dynamic Load Balancing," no. 2, pp. 1341–1344, 2007.

[7] A. Lappetel, "Equal Cost Multipath Routing in IP Networks," 2011.

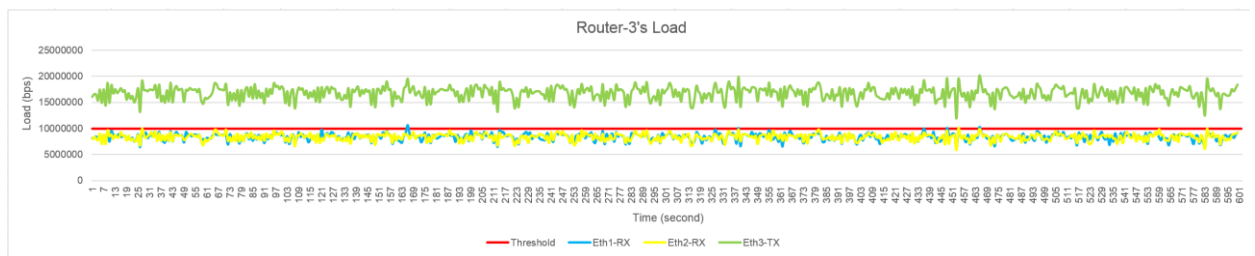
[8] T. M. Thomas, *OSPF network design solutions*, vol. 10. Cisco Press, 2003.

[9] G. A. Nugroho, A. S. Lamaida, and Y. Ahsana, "Analisis Algoritma Pencarian Rute Terpendek Dengan Algoritma Dijkstra dan Bellman - Ford," *Tek. Inform. ITB*, 2006.

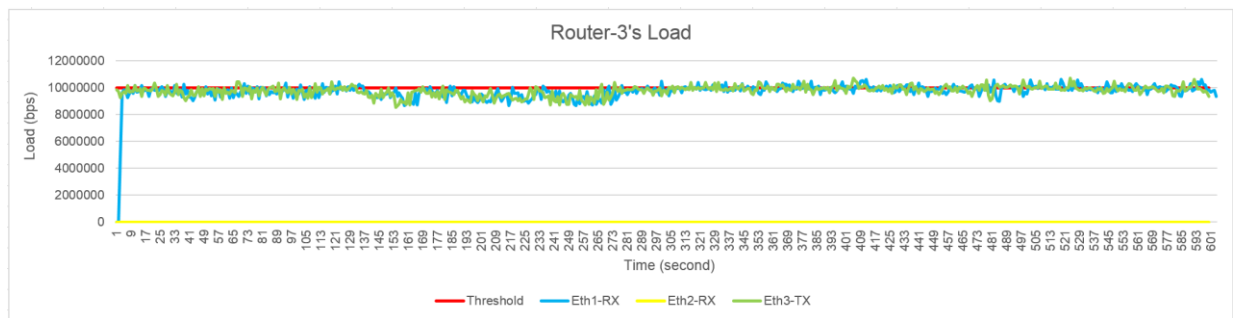
[10] R. Solichin and M. Y. L. T. Oktoviana, "Implementasi Algoritma Dijkstra dalam Pencarian Lintasan Terpendek Lokasi Rumah Sakit, Hotel dan Terminal Kota Malang," pp. 1-7, 2013.

[11] M. A. Muslim, "Analisis Codec dan Payload pada Micronet dan CISCO Pada Jaringan VPN - MPLS," vol. XII, no. 2, pp. 109-121, 2007.

[12] R. Susitaival and S. Aalto, "Adaptive load balancing with OSPF," no. January 2004, pp. 1-10, 2004.



Gbr. 3. Beban Jaringan pada Router-3 (MPATH)



Gbr. 2. Beban Jaringan pada Router-3 (SPATH)