

PERBANDINGAN *QUALITY OF SERVICE* ANTARA *ROUTING INFORMATION PROTOCOL (RIP)* DENGAN *OPEN SHORTEST PATH FIRST (OSPF)*

Agus Setiawan, Nina Sevani*

Fakultas Teknik dan Ilmu Komputer Jurusan Teknik Informatika
Universitas Kristen Krida Wacana – Jakarta
*nina.sevani@ukrida.ac.id

Abstrak

Implementasi protokol *routing* dalam jaringan memungkinkan *router* untuk secara dinamis bertukar informasi tentang kondisi jaringan. Tujuan dari penelitian ini adalah untuk menganalisis dan membandingkan kinerja antara RIP (*Routing Information Protocol*) sebagai *distance vector routing protocol* dan OSPF (*Open Shortest Path First*) sebagai protokol *link state routing*. Penelitian ini menggunakan topologi dengan enam *router* dan empat parameter QoS (*Quality of Service*), yaitu *delay*, *jitter*, *packet loss*, dan *throughput*, untuk penilaian kinerja. Perangkat lunak simulator jaringan dan AWK digunakan untuk mendapatkan nilai parameter QoS dan untuk memverifikasi hasil pelaksanaannya. Hasil penelitian menunjukkan bahwa secara keseluruhan OSPF memberikan kinerja yang lebih baik dibandingkan RIP meskipun OSPF tampaknya perlu waktu lebih lama dan *throughput* yang lebih besar untuk proses inisialisasi. Jenis protokol *routing* yang cocok digunakan dalam jaringan tergantung dari jumlah dan spesifikasi dari *router* dan topologi yang digunakan. Pada akhirnya, dapat disimpulkan bahwa lebih baik untuk melakukan simulasi sebelum proses implementasi. Dari simulasi kita dapat mengetahui tentang kinerja protokol *routing* yang digunakan.

Kata Kunci: RIP, OSPF, simulator, kinerja, QoS.

Abstract

Implementation of routing protocol in a network enable router to dynamically exchange information about network condition. The goal of this research is to analyze and compare performance between RIP (Routing Information Protocol) as a distance vector routing protocol and OSPF (Open Shortest Path First) as a link state routing protocol. This research use topology with six router and four parameter of QoS (Quality of Service), that is delay, jitter, packet loss, and throughput, to performance assessment. Network simulator software and AWK used to get the value of the QoS's parameter and to verify the result of the implementation. The result of the research show that for overall, OSPF give better performance than RIP although OSPF seem to need longer time and bigger throughput for the initialization process. The kind of fit routing protocol use in a network is depend on number and spesification of the router and topology used. At the end, can be conclude that its better to doing the simulation before the implementation process. From the simulation we can know about the performance of routing protocol used.

Keywords: RIP, OSPF, simulator, performance, QoS.

1. PENDAHULUAN

1.1 Latar Belakang

Implementasi dan pemanfaatan jaringan komputer dalam kehidupan masyarakat sekarang ini sudah semakin meluas. Hal ini tidak terlepas dari berbagai kemudahan yang ditawarkan oleh jaringan, mulai dari kemudahan berkomunikasi sampai dengan kemungkinan dilakukannya *sharing resources*. Implementasi jaringan sendiri tentunya membutuhkan berbagai alat, seperti *router* dan *switch* serta pemahaman akan berbagai teknik, seperti teknik pengalamatan dengan menggunakan *IP address* dan juga teknik untuk *routing*. *Routing* merupakan proses pengiriman paket data dari *host* asal menuju ke *host* tujuan, yang dilakukan oleh *router*. *Host* merupakan sebutan bagi alat-alat yang terhubung ke jaringan komputer. *Router* sendiri merupakan salah satu alat di jaringan komputer yang dapat menghantarkan paket data antar jaringan berdasarkan pada *IP address*.

Secara umum, sebuah *router* dapat menghantarkan paket data apabila mempunyai informasi yang diperlukan. Seluruh informasi ini dapat diketahui oleh *router* dengan dua cara, yaitu secara statis dan secara dinamis. Informasi yang diperoleh secara dinamis ini dapat diperoleh oleh *router* melalui pertukaran *routing protocol* antar-*router*. Saat ini *routing protocol* yang digunakan dapat dikelompokkan menjadi dua kategori, yaitu *distance vector routing protocol* dan *link state routing protocol*. Kedua kelompok ini sudah tentu mempunyai karakteristik masing-masing, berikut dengan kelebihan dan kekurangannya. *Routing Information Protocol* (RIP) merupakan contoh dari *distance vector routing protocol* sedangkan *Open Shortest Path First* (OSPF) merupakan contoh dari *link state routing protocol*. Penentuan jenis *routing protocol* yang akan digunakan dalam sebuah jaringan, tentunya tergantung pada kondisi, tujuan pembentukan jaringan, serta kebijakan dari pemilik jaringan itu sendiri.

RIP menawarkan kemudahan dalam implementasi, baik dari aspek konfigurasi maupun dari aspek biaya yang harus dikeluarkan. Namun seperti halnya karakteristik *distance vector*, implementasi RIP menyebabkan terjadinya waktu untuk *converged* di jaringan menjadi lebih lama sedangkan implementasi OSPF membuat waktu untuk *converged* menjadi lebih cepat, meskipun konfigurasi dan implementasinya dirasakan lebih sulit.

Dengan menggunakan sebuah jaringan yang terdiri dari enam buah *router* yang saling terhubung secara dinamis, akan dicoba untuk mengukur *performance* dari kedua *routing protocol*, RIP, dan OSPF. Pengukuran *performance* ini menggunakan parameter yang merupakan parameter penentu kualitas jaringan, seperti *jitter*, *delay*, dan *packet loss*.

1.2 Rumusan Permasalahan

Berdasarkan uraian pada latar belakang di atas, maka dapat dirumuskan permasalahan seputar pemilihan implementasi *routing protocol*. Permasalahan tersebut adalah “Bagaimana perbandingan *performance* dari penerapan RIP dan OSPF dalam sebuah jaringan sama yang terdiri atas enam buah *router* dilihat dari parameter QoS seperti *jitter*, *delay*, *throughput*, dan *packet loss*?”

1.3 Ruang Lingkup

Beberapa hal yang menjadi ruang lingkup dari penelitian untuk melihat perbandingan *performance* antara RIP dan OSPF, antara lain:

- Rancangan jaringan terdiri atas enam buah *router*.
- Antar-*router* berhubungan menggunakan RIP dan OSPF.

- Menggunakan simulator berupa *network simulator* dan AWK untuk melihat *performance* dari kedua *routing protocol* yang digunakan.
- Menggunakan parameter *Quality of Service* (QoS) berupa *delay*, *jitter*, *packet loss*, dan *throughput* untuk proses perbandingan *performance*.

1.4 Tujuan

Tujuan utama dari penelitian ini adalah untuk melakukan analisis *performance* dari RIP dan OSPF yang digunakan pada sebuah jaringan yang terdiri atas enam *router*. Tujuan lainnya adalah untuk membantu pengelola jaringan dalam melihat efisiensi kerja RIP dan OSPF.

1.5 Manfaat

Manfaat yang dapat diperoleh dari penelitian ini adalah untuk memberikan informasi kepada pengelola jaringan dalam memilih *routing protocol* RIP atau OSPF yang lebih baik digunakan pada jaringan yang terdiri atas enam *router*. Selain bagi pengelola jaringan, manfaat lain yang dapat diperoleh adalah untuk membantu mempermudah pengguna jaringan dalam memahami kelebihan dan kekurangan dalam penggunaan RIP dan OSPF.

1.6 Metodologi Penelitian

Metodologi penelitian yang digunakan untuk melihat perbandingan *performance* antara RIP dan OSPF terdiri atas empat tahap, yaitu:

- 1) Studi pustaka, untuk mengumpulkan berbagai teori yang ada seputar penggunaan RIP dan OSPF.
- 2) Perancangan simulasi berupa penentuan jenis dan jumlah *router*, serta topologi jaringan dan parameter yang akan digunakan.
- 3) Implementasi simulasi pada *software network simulator* dan AWK.
- 4) Pengujian *performance* RIP dan OSPF pada *software network simulator* dan AWK.

2. LANDASAN TEORI

2.1 Topologi Jaringan

Topologi jaringan merupakan suatu konsep menghubungkan satu *device* ke *device* lain. Berdasarkan bentuknya, dikenal dua macam bentuk topologi, yaitu:

- 1) Topologi Fisik adalah struktur jaringan secara fisik (*layout* jaringan yang dapat dilihat secara nyata). Beberapa bentuk topologi fisik adalah:
 - Topologi *Star*
Topologi *star* ini menggunakan sebuah konsentrator untuk menghubungkan semua *device* yang ada pada jaringan. Topologi ini juga disebut sebagai *Hub and Spoke Topology*.
 - Topologi *Bus*
Pada topologi *bus*, satu kabel utama menghubungkan tiap simpul ke saluran tunggal komputer yang mengaksesnya ujung dengan ujung. Masing-masing simpul dihubungkan ke dua simpul lainnya, kecuali komputer di salah satu ujung kabel, yang masing-masing hanya terhubung ke satu simpul lainnya [1].
- 2) Topologi Logik adalah struktur yang menentukan bagaimana beberapa *host* dalam jaringan dapat saling berkomunikasi. Contoh:
 - *Broadcast*

Pada topologi *broadcast*, maka sistem yang dianut adalah *first come first serve*. Dimana setiap *device* pada jaringan mempunyai hak yang sama untuk mengirimkan data ke *device* lainnya. Dengan demikian, bisa terjadi pengiriman data secara bersamaan.

- **Token**

Teknologi *token ring* distandarkan berdasarkan IEEE 802.5 yang dikembangkan oleh IBM. Jaringan *token ring* umumnya bekerja pada kecepatan 4 Mbps dan 16 Mbps. Metode ini sederhananya adalah sebagai berikut: sebuah tiket (*token*) yang bebas mengalir melalui setiap *node* di jaringan telah berputar – putar. Saat sebuah *node* mengirimkan paket, *node* itu meraih dan meletakkan informasi/*frame* atau pakatnya ke *token* tersebut sehingga tidak dapat digunakan lagi oleh *node* yang lain sampai data mencapai tujuannya. Jika informasi telah sampai, maka *token* dilepaskan lagi oleh *originating station*. *Token* mengalir di jaringan dalam satu arah dan setiap *node* satu persatu [2].

2.2 **Device Pada Jaringan**

Dalam sebuah jaringan komputer, terdapat berbagai macam *device* yang memungkinkan terjadinya komunikasi antar komputer tersebut. Macam-macam *device* yang dipakai secara umum antara lain *switch* dan *router*.

2.2.1 **Router**

Router adalah suatu alat/program yang mengarahkan paket-paket data dalam jaringan hingga mencapai tujuannya. Dalam melakukan tugasnya, *router* menggunakan 'algoritma *routing*' untuk menentukan *route* terbaik dalam proses penghantaran paket data tersebut. 'Rute terbaik' ditentukan oleh berbagai faktor, antara lain '*hop*' (lintasan yang dilalui paket data dari satu *router* ke *router* lain dalam suatu jaringan) [3].

Router merupakan *device layer 3* karena membuat keputusan dalam meneruskan paket data berdasarkan IP *address* tujuan yang terdapat pada paket data yang diterimanya. *Router* memiliki *routing table* yang salah satunya berisi alamat dari *network* yang terhubung dengannya.

2.2.2 **Switch**

Switch adalah sebuah alat yang menyaring dan melewatkan paket yang ada di sebuah LAN. *Switcher* bekerja pada *layer data link (layer 2)* dan terkadang di *Network Layer (layer 3)* berdasarkan referensi *OSI Layer Model* sehingga dapat bekerja untuk paket protokol apapun. LAN yang menggunakan *switch* untuk berkomunikasi di jaringan maka disebut dengan *Switched LAN* atau dalam fisik *ethernet* jaringan disebut dengan *Switched Ethernet LANs*. *Switch* mempunyai fungsi hampir sama seperti *hub*. Perbedaannya adalah *switch* dapat beroperasi dengan mode *full-duplex* dan mampu mengalihkan jalur serta menyaring informasi ke dan dari tujuan yang spesifik, sedangkan *hub* tidak dapat menyaring serta mengalihkan jalur hubungan.

Hampir sama seperti *bridge*, *switch* bekerja atas dasar informasi *MAC address*. Hanya saja *switch* mempunyai kemampuan dan kinerja yang lebih baik dibandingkan dengan *bridge*, karena *switch* selain bekerja secara *software* juga bekerja di atas *hardware*.

Switch menggunakan algoritma *store and forward* dan *cut through* pada saat melakukan pengiriman data. Secara tipikal, *switch* mempunyai beberapa *port* yang menghubungkan beberapa segmen LAN lain dan *port* pada *switch* ini berkecepatan tinggi. Oleh karena itu, *switch* mampu menyediakan media *dedicated* dengan komunikasi

yang bebas dari tumbukan (*collision*) antar perangkat jaringan dan mendukung komunikasi simultan, serta dirancang untuk akses kecepatan tinggi [4].

2.3 *Routing Protocol*

Routing protocol merupakan aturan yang mempertukarkan informasi *routing*. *Routing* digunakan untuk proses pengambilan sebuah paket dari sebuah *device* dan mengirimkannya melalui *network* ke *device* lain pada sebuah *network* [5]. Semua *routing protocol* bertujuan mencari *route* tersingkat untuk mencapai tujuan dan mempunyai cara sendiri dalam proses pengiriman paket [6].

2.3.1 *Routing Information Protocol (RIP)*

RIP merupakan salah satu contoh dari algoritma *Distance vector (DV)*. RIP mengirimkan semua isi *routing table* ke *router* tetangga yang terhubung secara langsung (*directly connected*), secara periodik setiap 30 detik. *Router* yang menerima *routing update* akan meng-*update routing table*-nya dan kemudian mengirimkan *routing update* ke *router* di sampingnya lagi. Proses ini akan terus berulang melalui semua *router* yang ada pada jaringan. Setiap perpindahan 1 *router* maka nilai *hop count* akan bertambah 1. Bila paket data telah melalui 15 *router*, maka paket tersebut akan di-*discard* (dimusnahkan), meskipun mungkin belum mencapai tujuannya, dan *network* tujuan juga akan dianggap *unreachability* (tidak dapat dicapai). RIP menggunakan *hop count* sebagai metrik dengan maksimal *hop count* adalah 15 sebagai upaya agar tidak sampai terjadi *count to infinity* dan *routing loop*.

2.3.2 *Open Shortest Path First (OSPF)*

OSPF merupakan *link-state routing protocol (LS)* yang berdasarkan pada *open standard*. Setiap *router* yang dikonfigurasi dengan menggunakan *link-state routing protocol* akan mengirimkan dua macam paket, yaitu:

- **LSR** (*Link-State Refreshment*) yang dikirimkan secara periodik ke *router* yang berada di sekitarnya untuk mengetahui apakah *router* tersebut masih aktif dan masih terbentuk *link* ke *router* tersebut.
- **LSA** (*Link-State Advertisement*) atau *routing update*, yang dikirimkan hanya pada saat ada perubahan pada jaringan dan pada saat awal (inisialisasi).

Pada awal OSPF *router* dihidupkan, maka *router* akan mengirimkan LSA secara *multicast*. *Router* yang menerima LSA, akan menyalin informasi yang dibawanya, kemudian meneruskan (*forward*) LSA tersebut. Informasi yang diperoleh dari LSA akan disimpan pada *topological (link-state) database*. Berdasarkan LSA dan juga *topological database* yang berisi semua *route* yang ada pada jaringan tersebut, maka setiap *router* akan menjalankan SPF algoritma dan membentuk *SPF Tree*. Berdasarkan *SPF Tree* inilah, *router* akan menentukan *best route* ke setiap *network* yang ada. *Best route* merupakan *route* terbaik hasil perhitungan algoritma dengan metrik, yaitu *route* dengan *cost* terkecil. *Routing update* atau LSR hanya dikirimkan bila terdapat perubahan pada jaringan, sehingga disebut sebagai *triggered update*. LSA akan dikirimkan ke semua *router* yang ada pada jaringan, bukan hanya ke *neighbour router* saja, dan hanya berisi sebagian dari *routing table*, yaitu bagian yang mengalami perubahan saja.

2.4 *Quality of Service (QoS)*

Pengukuran *performance* merupakan salah satu upaya dalam peningkatan efisiensi dan efektifitas kerja suatu jaringan guna meningkatkan produktifitas kerja pada jaringan. Pengukuran *performance* dapat dilakukan menggunakan beberapa parameter yang termasuk dalam QoS, seperti [5]:

2.4.1 Throughput

Throughput adalah kecepatan (*rate*) data transfer efektif yang diukur dalam bps. Header dalam paket data mengurangi nilai ini. Throughput dapat dihitung dengan melihat jumlah paket yang datang terhadap yang dikirim. Rumus berikut dapat digunakan untuk menghitung throughput:

$$S = \frac{\text{Jumlah paket sukses} \times \text{waktu transmisi paket}}{\text{lama pengamatan}} \dots\dots\dots(1)$$

Waktu transmisi paket (*t trans*) terdiri dari dua komponen, yaitu waktu paket (*t paket*) dan *delay* propagasi (*t prop*), yang dihitung dengan rumus:

$$t \text{ transmisi} = t \text{ paket} + t \text{ prop} \dots\dots\dots(2)$$

2.4.2 Jitter

Jitter didefinisikan sebagai variasi *delay* yang diakibatkan oleh panjang *queue* dalam suatu pengolahan data dan *reassemble* paket-paket data di akhir pengiriman akibat kegagalan sebelumnya. Secara umum jitter merupakan masalah dalam *slow speed links*. Diharapkan bahwa peningkatan QoS dengan mekanisme *priority buffer*, *bandwidth reservation* (RSVP, MPLS, dan lain-lain) serta *high speed connections* dapat mereduksi masalah jitter di masa yang akan datang. Dari Tabel 1 terlihat bahwa jitter di antara titik awal dan akhir komunikasi seharusnya kurang dari 150 ms sedangkan untuk *wireless* kurang dari 5 ms.

Tabel 1. Jitter

Kategori Degradasi	Peak Jitter (ms)
Sangat bagus	0
Bagus	75
Sedang	125
Jelek	225

2.4.3 Packet Loss

Tabel 2 menunjukkan keterangan tentang kategori dari *Packet Loss*.

Tabel 2. Packet Loss

Kategori Degradasi	Packet Loss (%)
Sangat bagus	0
Bagus	3
Sedang	15
Jelek	

2.4.4 Delay

Delay adalah waktu tunda saat paket yang diakibatkan oleh proses transmisi dari satu titik lain yang menjadi tujuannya. Pada Tabel 3 adalah rekomendasi ITU-T G.114 untuk *delay*.

Tabel 3. *Delay*

<i>Range in milisecond</i>	<i>Description</i>
0 – 150 msec	<i>Acceptable for most user application</i>
150 – 400 msec	<i>Acceptable provided that administrators are aware of the transmission time and it's impact on transmission quality of user application</i>
> 400 msec	<i>Unacceptable for general network planning purpose, it is recoqnized that in some exceptional cases this limit will be exceeded.</i>

3. PERANCANGAN

3.1 Identifikasi Kebutuhan

Kebutuhan akan implementasi jaringan yang dapat berbeda-beda di setiap lokasi membuat perbedaan pada jenis dan jumlah alat yang dibutuhkan. Perbedaan jenis dan jumlah alat ini pula yang pada akhirnya membuat terjadinya perbedaan pada teknik serta metode untuk mendapatkan informasi yang diperlukan oleh *router*, sebagai salah satu alat untuk proses *routing* di jaringan. Penggunaan banyak *router* membuat *network administrator* memilih untuk menggunakan *routing protocol* sebagai cara bagi *router* untuk mendapatkan informasi untuk melakukan *routing*.

RIP sebagai contoh *distance vector routing protocol* cenderung lebih sederhana dalam hal konfigurasi dan *resource router* yang digunakan untuk implementasi meskipun hanya dapat digunakan pada jaringan yang terdiri dari maksimal 15 *router*. OSPF sebagai contoh *link state routing protocol* memungkinkan untuk lebih cepat mencapai kondisi *converged* dibanding RIP. *Converged* adalah kondisi dimana seluruh *router* di jaringan sudah mempunyai informasi yang sama tentang kondisi jaringan secara keseluruhan. OSPF juga memungkinkan untuk digunakan pada jaringan yang terdiri dari puluhan *router*. Dalam implementasinya OSPF lebih kompleks serta butuh *resource router* yang lebih besar dari RIP [7].

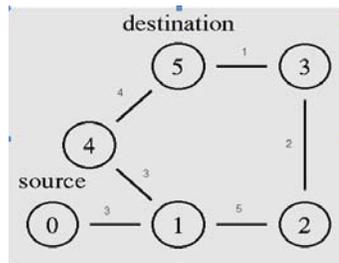
Dalam kondisi jaringan yang sama, yaitu terdiri atas enam *router*, dapat dilihat *performance* dari RIP dan OSPF berdasarkan pada empat parameter QoS, yaitu *jitter*, *delay*, *throughput*, dan *packet loss*. Untuk menunjukkan dan menilai *performance* dari RIP dan OSPF tersebut akan digunakan *software network simulator* [8] VMWare dan AWK sebagai program data *compiler*. AWK merupakan program berbasis *perl* yang berfungsi mengolah data dari setiap *output* yang ada. *Output* yang dimaksud merupakan *output* dari program simulator agar dapat di-*compile*.

3.2 Tahapan Perancangan

Berikut ini adalah tahapan perancangan yang dilakukan pada penelitian ini:

- 1) Instalasi *software network simulator* [8], yaitu VMWare, pada komputer yang digunakan sebagai media simulasi.
- 2) Instalasi Linux Ubuntu 8.04 pada VMWare.
- 3) Instalasi program *network simulator* pada Linux Ubuntu 8.04. Program *network simulator* ini dapat diunduh gratis pada alamat <http://sourceforge.net/projects/nsnam/>.
- 4) Instalasi AWK pada Linux Ubuntu 8.04. AWK merupakan program data *compiler* berbasis *perl* untuk mengolah data dari setiap *output* yang ada. AWK dapat diunduh dari <http://linux.wareseeker.com/download/gawk-3.1.7.rar/29166ba3b4>.
- 5) Penentuan topologi rancangan, yang terdiri dari 6 buah *router* seperti pada Gambar 1. Dimana *node 0* dapat disebut sebagai *router 0*, yang merupakan *router* asal dan *node*

5 atau *router* 5 merupakan *router* tujuan dalam simulasi *routing*. *Node* 1, 2, 3, 4, menggambarkan *router* 1, 2, 3, 4. Garis penghubung setiap *node* merupakan *path* yang menggambarkan *link* atau *path* penghubung dari setiap *router*. Setiap *path* atau *link* ini mempunyai nilai yang menggambarkan *cost* dari *path* tersebut. *Cost* merupakan metrik untuk OSPF yang diperlukan untuk menghitung dan menentukan *best route*. Topologi yang sama yang akan digunakan untuk konfigurasi dengan RIP dan OSPF.



Gambar 1. Rancangan topologi jaringan

6) Konfigurasi *routing protocol* pada *network simulator*. Langkah-langkah konfigurasi, baik pada RIP maupun OSPF, adalah sebagai berikut:

a) Pembuatan *router*, yang dilakukan dengan *coding* berikut ini:

```

set n0 [$ns node]          set n3 [$ns node]
set n1 [$ns node]          set n4 [$ns node]
set n2 [$ns node]          set n5 [$ns node]
    
```

b) Pembuatan agar rancangan tidak berubah pada saat dijalankan, yang dilakukan dengan *coding* berikut ini:

```

$ns duplex-link-op $n0 $n1 orient right
$ns duplex-link-op $n1 $n2 orient right
$ns duplex-link-op $n2 $n3 orient up
$ns duplex-link-op $n1 $n4 orient up-left
$ns duplex-link-op $n3 $n5 orient left
$ns duplex-link-op $n4 $n5 orient right-up
    
```

c) Pembuatan jalur antar-*router*, yang dilakukan dengan *coding* berikut:

```

$ns duplex-link $n0 $n1 1Mb 10ms DropTail
$ns duplex-link $n1 $n2 1Mb 10ms DropTail
$ns duplex-link $n2 $n3 1Mb 10ms DropTail
$ns duplex-link $n1 $n4 1Mb 10ms DropTail
$ns duplex-link $n3 $n5 1Mb 10ms DropTail
$ns duplex-link $n4 $n5 1Mb 10ms DropTail
    
```

d) Khusus penggunaan OSPF sebagai *routing protocol*, langkah selanjutnya adalah menambahkan nilai atau *cost* untuk setiap jalur yang telah dibentuk sebelumnya. *Coding* berikut merupakan cuplikan program untuk menentukan *cost* dari setiap jalur sesuai dengan topologi pada Gambar 1.

```

$ns cost $n0 $n1 3
$ns cost $n1 $n4 3
$ns cost $n1 $n2 5
$ns cost $n4 $n5 4
$ns cost $n2 $n3 2
$ns cost $n3 $n5 1
    
```

Pada penggunaan RIP sebagai *routing protocol* tidak diperlukan langkah berupa penentuan nilai atau *cost* untuk setiap jalur yang ada.

e) Identifikasi *routing protocol* yang digunakan. Untuk penggunaan RIP yang merupakan kategori *distance vector routing protocol*, ditambahkan *coding* sebagai berikut: `$ns rtproto DV`

Untuk penggunaan OSPF yang merupakan kategori *link-state routing protocol*, ditambahkan *coding* sebagai berikut: `$ns rtproto LS`

4. HASIL DAN EVALUASI

Tahapan selanjutnya adalah tahap implementasi untuk mengetahui hasil rancangan yang dibuat. Hasil rancangan ini kemudian akan diuji untuk mengetahui *performance* dari RIP dan OSPF dilihat dari beberapa jenis parameter QoS.

4.1 Implementasi

Berikut adalah tahapan implementasi dari hasil rancangan yang dibuat:

- 1) Membuat *source code* untuk *network simulator* dan AWK. Dalam *source code* yang dibuat juga berisi *coding* untuk menghitung *throughput*, *jitter*, *delay*, dan *packet loss*. Cuplikan *coding* untuk perhitungan *throughput* dapat dilihat pada Gambar 2.

```

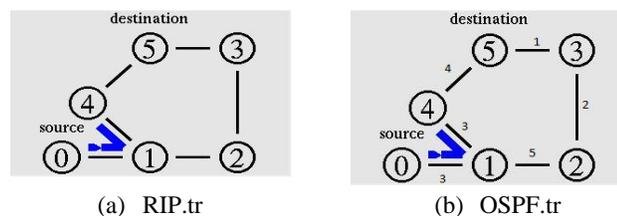
throughput1.awk
BEGIN {
    for (i in send) {
        send[i] = 0
    }
    for (i in rcv) {
        rcv[i] = 0
    }
    tx = 0
    drop = 0
    pkt_loss = 0
    pkt_rcvd = 0
    time_obsrv = 15
}

```

Gambar 2. Cuplikan *coding* perhitungan *throughput*

Pada cuplikan *coding* di atas, pertama kali akan dideklarasikan perulangan untuk `send[i]` sebagai variabel perulangan untuk pengiriman data, kemudian `rcv[i]` adalah variabel perulangan untuk penerimaan data. Variabel `tx` adalah variabel penampung data yang akan dikirimkan, sedangkan variabel `drop` adalah banyaknya data yang diterima. Variabel `pkt_loss` sebagai variabel penampung untuk data yang hilang dan variabel `pkt_rcvd` sebagai variabel penampung untuk paket yang diterima sedangkan variabel `time_obsrv` adalah waktu yang dibutuhkan untuk menerima suatu data.

- 2) Menjalankan *source code* yang dibuat pada terminal Linux. Proses eksekusi *source code* dilakukan dengan cara mengetik `ns nama_file.tcl`. Gambar 3(a) merupakan cuplikan hasil eksekusi dari menjalankan *source code* RIP dan Gambar 3(b) merupakan cuplikan hasil eksekusi *source code* OSPF. Hasil yang didapat dari menjalankan *source code* ini adalah RIP.tr dan OSPF.tr. Kedua *file* tersebut yang nantinya akan diolah dengan menggunakan AWK agar bisa dianalisis kinerja dari simulasi yang dirancang.



Gambar 3. Hasil eksekusi *source code*

- (b) Eksekusi *file* AWK dengan *output* yang dihasilkan dari *source code network simulator*, dengan cara mengetik perintah `awk -f nama_file.awk nama_file.tr` pada terminal linux. Gambar 4 merupakan cuplikan hasil yang didapat dari eksekusi *file* awk, yang merupakan perhitungan/analisis *output* RIP.tr dan OSPF.tr. Gambar 4(a) merupakan cuplikan hasil perhitungan *packetloss* sedangkan Gambar 4(b) merupakan cuplikan hasil perhitungan *delay*.

```

test1@test1-desktp:~/Desktop$ awk -f packetloss1.awk a.tr
Packet sent           = 612
Packet dropped        = 11
==> Packet loss data  = 1.79739 %

test1@test1-desktp:~/Desktop$ awk -f packetloss1.awk b.tr
Packet sent           = 612
Packet dropped        = 12
==> Packet loss data  = 1.96078 %

test1@test1-desktp:~/Desktop$ awk -f delay1.awk a.tr
==> Average delay data = 0.0674769 s
                        = 67.4769 ms

test1@test1-desktp:~/Desktop$ awk -f delay1.awk b.tr
==> Average delay data = 0.0616557 s
                        = 61.6557 ms
    
```

(a) Packetloss AWK

(b) Delay AWK

Gambar 4. Tampilan hasil perhitungan *output file*

4.2 Hasil Perhitungan

Analisis dilakukan terhadap hasil perhitungan pada empat jenis parameter QoS, yaitu *throughput*, *delay*, *jitter*, dan *packetloss*. Perhitungan dilakukan terhadap *path* yang menghubungkan *node* asal (*node* 0) dengan *node* tujuan (*node* 5). Hasil perhitungan untuk setiap parameter yang digunakan dapat dilihat pada serangkaian tabel dibawah. Tabel 4 merupakan hasil perhitungan yang diperoleh dari eksekusi *file output* untuk parameter *throughput*.

Tabel 4. Hasil perhitungan *throughput* RIP.tr dan OSPF.tr

JALUR	Node Source – Node Destination	RIP	OSPF
JALUR 1	node 0 – node 1	228.48 kbps	228.48 kbps
	node 1 – node 4	176.96 kbps	208.693 kbps
	node 4 – node 5	175.467 kbps	207.947 kbps
JALUR 2	node 1 – node 2	47.4133 kbps	15.3067 kbps
	node 2 – node 3	47.4133 kbps	15.3067 kbps
	node 3 – node 5	47.4133 kbps	15.3067 kbps
	Rata-rata	Rata-rata	Rata-rata
	node 0 – node 5	222.88 kbps	223.253 kbps

Dari hasil eksekusi *output file* RIP.tr dan OSPF.tr untuk empat parameter QoS yang digunakan dapat dilihat bahwa pada OSPF mempunyai nilai *packetloss* yang sama dengan RIP. Karena pada pengiriman data dengan OSPF sangat dipengaruhi oleh *cost* dari setiap *path* yang digunakan. Dimana semakin tinggi *cost* akan semakin cepat paket dihantarkan. Namun sebaliknya nilai *throughput* yang besar pada OSPF membuat *delay* pada OSPF menjadi lebih kecil daripada RIP, karena semakin besar *throughput* semakin cepat paket data dihantarkan. Tabel 5 menunjukkan hasil perhitungan terhadap parameter *packetloss*. Tabel 6 menunjukkan hasil perhitungan terhadap parameter *delay*.

Tabel 5. Hasil perhitungan *packetloss* RIP.tr dan OSPF.tr

JALUR	Node Source – Node Destination	RIP	OSPF
JALUR 1	node 0 – Node 1	1.79739 %	1.96078 %
	node 1 – node 4	21.797 %	7.16667 %
	node 4 – node 5	0.425532 %	0.5386 %
JALUR 2	node 1 – node 2	78.8686 %	93.1667 %
	node 2 – node 3	0 %	0 %
	node 3 – node 5	0 %	0 %
	Rata-rata	Rata-rata	Rata-rata
	node 0 – node 5	2.77778 %	2.77778 %

Tabel 6. Hasil perhitungan *delay* RIP.tr dan OSPF.tr

JALUR	Node Source – Node Destination	RIP	OSPF
JALUR 1	node 0 – Node 1	67.4769 ms	61.6557 ms
	node 1 – node 4	18.3126 ms	18.3068 ms
	node 4 – node 5	18.3118 ms	18.3056 ms
JALUR 2	node 1 – node 2	18.3249 ms	18.32 ms
	node 2 – node 3	18.3204 ms	18.32 ms
	node 3 – node 5	18.32 ms	18.3317 ms
	Rata-rata	Rata-rata	Rata-rata
	node 0 – node 5	107.798 ms	99.2489 ms

Untuk nilai *jitter*, OSPF secara rata-rata mempunyai nilai yang lebih kecil karena adanya perbedaan *delay* yang lebih kecil pula untuk setiap paket yang diterima oleh *node* tujuan. Meskipun demikian perbedaan *jitter* yang tampak tidak terlalu mencolok. Tabel 7 menunjukkan hasil perhitungan nilai *jitter* untuk RIP.tr dan OSPF.tr.

Tabel 7. Hasil perhitungan *jitter* RIP.tr dan OSPF.tr

JALUR	Node Source – Node Destination	RIP	OSPF
JALUR 1	node 0 – Node 1	8.67711 ms	8.40773 ms
	node 1 – node 4	0.000204255 ms	0.00114901 ms
	node 4 – node 5	15.2756 ms	16.9299 ms
JALUR 2	node 1 – node 2	0.00188976 ms	1.5164e-13 ms
	node 2 – node 3	0.000755906 ms	2.59955e-13 ms
	node 3 – node 5	19.632 ms	24.1287 ms
	Rata-rata	Rata-rata	Rata-rata
	node 0 – node 5	8.74278 ms	8.41466 ms

Dari seluruh percobaan terhadap empat parameter QoS, yaitu *throughput*, *delay*, *jitter*, *packet loss* yang dilakukan terhadap topologi yang sama, dapat diketahui bahwa kinerja dari *routing protocol* OSPF lebih baik dibandingkan dengan kinerja dari *routing protocol* RIP. Hal itu dapat dibuktikan dari adanya perbedaan hasil perhitungan *throughput*, *delay*, dan *jitter*, sedangkan untuk *packet loss* tidak terdapat perbedaan. Perbedaan tersebut diketahui dari rata-rata setiap *performance* pada *routing protocol* RIP dan OSPF. Rata-rata tersebut diambil dari pengiriman paket dari *node* awal, yaitu *node* 0 sampai *node destination* atau *node* 5.

5. KESIMPULAN

Beberapa kesimpulan yang dapat diambil terkait dengan percobaan analisis *performance* dari RIP dan OSPF adalah:

- *Delay RIP* lebih besar dibandingkan dengan *OSPF*, dikarenakan untuk *OSPF* algoritma yang dipakai adalah *dijkstra shortest path*. Algoritma tersebut sangat kompleks namun dapat menemukan jalan terpendek yang bebas hambatan, sehingga memperkecil *delay* yang dibutuhkan untuk mengirim suatu paket.
- *Throughput RIP* lebih kecil dibandingkan dengan *OSPF*, dikarenakan untuk *RIP* terdapat jumlah paket yang sukses yang diterima pada *node* tujuan lebih kecil.
- *Jitter RIP* lebih besar dibandingkan dengan *OSPF*, dikarenakan lama waktu yang dibutuhkan untuk paket data yang dikirimkan menuju *node* tujuan.

- *Packetloss* antara RIP sama besarnya dengan *OSPF* dikarenakan untuk keduanya paket data yang dikirimkan dan di terima sama besarnya.

REFERENSI

- [1] Tim Penelitian Dan Pengembangan Wahana Komputer, "*Konsep Jaringan Komputer Dan Pengembangan*", Edisi pertama, Salemba Infotek, Jakarta, 2003.
- [2] <http://phitoosh.blogspot.com/2007/08/dasar-jaringan.html>
- [3] Tanenbaum, Andrew S , "*Jaringan Komputer Edisi Bahasa Indonesia*", Edisi ketiga, Jakarta, 1997.
- [4] <http://fariethermawan.wordpress.com/2009/10/09/perangkat-keras-dalam-jaringan/>
- [5] Soujanya, B., Sitamahalakshmi, T., and Divakar, C.H., "*Study of Routing Protocols in Mobile Ad-Hoc Network*", International Journal of Engineering Science and Technology, Volume 3, Number 4, April 2011.
- [6] <http://dhitapriillia.blogspot.com/2010/01/pengertian-man-wan-internet-routing.html>
- [7] Ayub, N., et al., "*Performance Analysis of OSPF and EIGRP Routing Protocols with Respect to The Converged*", European Journal of Scientific Research, Volume 6, Number 3, 2011.
- [8] Acharjee, U. K., Ahmed, A., and Rafique, S., "*Experimental Analysis of Ad-Hoc Routing Protocol Using Network Simulator*", Volume 1, Number 2, December 2007