



KILAT

JURNAL KAJIAN ILMU DAN TEKNOLOGI

*Dine Tiara Kusuma;
M. Yoga Distran Sudirman;
Yessy Fitriani*

*Emillia;
Yuliansyah*

*Rosida Nur Aziza;
Dhizlan Dzhallila*

Ranti Hidayawanti

*Dewi Arianti Wulandari;
Hendra Jatnika;
Yudhy S. Purwanto*

*Rr. Mekar Ageng Kinasti;
Endah Lestari;
Devita Mayasari*

Faisal Piliang

Mauludi Manfaluthy

*Pauzi Hasan;
Peby Wahyu Purnawan*

*Rahmi Amir;
Baginda Oloan Lubis*

*Sabar Hanadwiputra;
Subandri*

*Ndaru Ruseno;
Satria*

PENDEKATAN METODE ALTMAN Z-SCORE DALAM PENENTUAN INSENTIF BONUS PEGAWAI

METODE YURIDIS PENGELOLAAN SAMPAH RUMAH TANGGA DAN SAMPAH SEJENIS DI STT-PLN

METODE KUANTITATIF DENGAN PENDEKATAN KLASIK PADA APLIKASI ANALISIS BUTIR SOAL SEBAGAI MEDIA EVALUASI PENENTUAN SOAL YANG BERKUALITAS

UPAYA TERTIB LISTRIK TERHADAP INSTALATIR KABEL DI DAERAH PADAT PENDUDUK (STUDY KASUS KEC. TAMBORA)

RANCANG BANGUN APLIKASI CLUSTERING DATA MINING MENGGUNAKAN METODE K-MEANS DAN K-MODES

POTENSI PEMANFAATAN LIMBAH PEMBAKARAN BATUBARA (BOTTOM ASH) PADA PLTU SEBAGAI MEDIA TANAM DALAM UPAYA MENGURANGI PENCEMARAN LINGKUNGAN

PEMILIHAN PERANGKAT LUNAK PEMINDAHAN BERKAS DALAM MENINGKATKAN PEMANFAATAN TELEPON PINTAR

PEMANFAATAN RADIASI ENERGI TEGANGAN 150 KV UNTUK LAMPU LED PENERANGAN JALAN

KAJIAN PERBANDINGAN PERFORMANSI ROUTING PROTOCOL RIPNG, OSPFV3 DAN EIGRPV6 PADA JARINGAN IPV6

PERANCANGAN PROGRAM PENGELOLAAN DATA KEUANGAN PASIEN RAWAT JALAN BPJS PADA RUMAH SAKIT GRAHA JUANDA BEKASI

ANALISA DAN IMPLEMENTASI VTP DENGAN ETHERCHANNEL TYPE LACP

PENGEMBANGAN RANCANG BANGUN SISTEM KESISWAAN DENGAN MENGGUNAKAN FRAMEWORK MVC (MODEL VIEW CONTROLLER)

ISSN 2089-1245



9 772089 124519

SEKOLAH TINGGI TEKNIK - PLN (STT-PLN)

KILAT	VOL.7	NO.1	HAL. 1 - 90	APRIL 2018	ISSN 2089 - 1245
-------	-------	------	-------------	------------	------------------

KAJIAN PERBANDINGAN PERFORMANSI ROUTING PROTOCOL RIPNG, OSPFV3 DAN EIGRPV6 PADA JARINGAN IPV6

Pauzi Hasan¹; Peby Wahyu Purnawan²

^{1,2}Fakultas Teknik Universitas Budi Luhur, Jakarta, 12260

Telp: (021) 5853753 ext 255

Email: hasanpauzi@gmail.com

pebywahyupurnawan@budiluhur.ac.id

Abstract

The main purpose of IP next generation version 6 or called IPv6 is to meet the needs of IP addresses for the long term. With the presence of IPv6 it is also necessary routing protocol that supports IPv6 network. Routing protocols work to connect between networks, and choose paths or routes to reach other networks. There are three types of routing protocols that already support IPv6 services. Among other types of distance vector (RIPng), link-state (OSPFv3) and hybrid (EIGRPv6). This final project is designed to know the performance of the three types of routing protocols on IPv6 network. Testing is done by analyzing routing update process, ICMPv6 packet analysis and testing by sending Real-time Transport Protocol (RTP) packet in the form of audio video stream sent via server and accessed through client. In the simulation and analysis results using GNS3 (Graphic Network Simulator 3) and Wireshark found that the OSPFv3 performance testing is better than EIGRPv6 and RIPng when performing routing updates or when link down occurs. Then the data analysis of test results with the delivery of Real-time Transport Protocol package so that it can be concluded that OSPFv3 algorithm better than EIGRPv6 and RIPng in terms of delay, packet loss, throughput and jitter.

Keyword: QoS, Wireshark, RTP, IPv6, Routing Protocol, GNS3, ICMPv6, EIGRPv6, RIPng, OSPFv3

Abstrak

Tujuan utama perkembangan IP next generation versi 6 atau disebut IPv6 adalah untuk memenuhi kebutuhan alamat IP untuk jangka panjang. Dengan hadirnya IPv6 maka dibutuhkan juga routing protocol yang mendukung jaringan IPv6. Routing protocol berfungsi untuk menghubungkan antar jaringan, dan memilih jalur atau rute untuk mencapai jaringan yang lain. Ada tiga jenis routing protocol yang sudah mendukung layanan IPv6. Diantaranya jenis distance vector (RIPng), link-state (OSPFv3) dan hybrid (EIGRPv6). Tugas akhir ini disusun untuk mengetahui performansi dari ketiga jenis routing protocol pada jaringan IPv6. Pengujian dilakukan dengan analisa proses routing update, analisa paket ICMPv6 dan pengujian dengan melakukan pengiriman paket Real-time Transport Protocol (RTP) berupa audio video stream yang dikirim melalui server dan diakses melalui client. Dalam hasil simulasi serta analisis dengan menggunakan GNS3 (Graphic Network Simulator 3) dan Wireshark dapat disimpulkan bahwa pada pengujian performansi OSPFv3 lebih baik dari EIGRPv6 dan RIPng ketika melakukan routing update atau ketika terjadi link down. Kemudian dilakukan analisa data hasil pengujian dengan pengiriman paket Real-time Transport Protocol sehingga dapat disimpulkan bahwa algoritma OSPFv3 lebih baik dari EIGRPv6 dan RIPng dalam hal delay, packet loss, throughput dan jitter.

Kata Kunci: QoS, Wireshark, RTP, IPv6, Routing Protocol, GNS3, ICMPv6, EIGRPv6, RIPng, OSPFv3

1. PENDAHULUAN

Perkembangan teknologi informasi dan telekomunikasi yang berkembang pesat terutama pada teknologi yang menggunakan jaringan internet membuat semakin berkurangnya persediaan IPv4 sehingga disarankan melakukan perpindahan ke IP next generation atau IPv6 yang memiliki kapasitas jauh lebih besar. Akan tetapi perpindahan penggunaan versi IP ini membutuhkan perubahan pula dalam sistem routing. Routing merupakan proses pencarian path atau alur guna memindahkan informasi dari host sumber (source address) ke host tujuan (destinations address) melalui koneksi internetwork. Ada beberapa routing protocol yang sudah mendukung IPv6, diantaranya RIPng, EIGRP dan OSPF. Telah dilakukan sebuah penelitian

mengenai perancangan dan analisis kinerja EIGRP pada jaringan IPv6 (Edi Yusuf, 2016). Penelitian ini bertujuan untuk merancang dan mensimulasikan mengenai routing protocol EIGRP untuk IPv6, menganalisis dan menentukan jalur yang dilewati oleh paket ICMPv6, dalam simulasi ini memakai masing-masing 4 buah router, PC, dan switch. Hal yang diamati dari pengujian ini adalah hasil trace route untuk menentukan jalur yang dilalui oleh paket data dengan melihat dan menentukan nilai metric dengan menggunakan perhitungan rumus kemudian dilakukan dengan mengamati waktu konvergensi dari EIGRP IPv6.

Telah diteliti tentang analisis kinerja RIP (routing information protocol) untuk optimalisasi jalur routing (Kadek Chandra Tresna Wijaya, 2012).

Penelitian ini bertujuan untuk mengoptimalkan kinerja router dalam melakukan pertukaran informasi serta menjawab permasalahan didalam komunikasi seperti time-out, data yang dikirimkan lambat atau rusak. pengujian dibagi menjadi 2 tahap, tahap pertama pengujian dilakukan tanpa memutuskan link jaringan yang terhubung, dan yang kedua dengan link terputus. Dalam penelitian dapat disimpulkan RIP mampu mengoptimalkan kinerja jaringan dengan kecepatan waktu konvergensi rata-rata dibawah 30 detik. Telah dilakukan penelitian tentang analisa proses routing RIPv1, RIPv2, dan RIPv6 antara model IPv4 dengan IPv6 pada jaringan data PT. Pertamina (Irsan Fitrah Adhil, 2016). Penelitian ini bertujuan untuk mengetahui serta mengoptimalkan jaringan data pada PT. Pertamina, dimana sebelumnya routing protocol yang digunakan adalah RIPv1 pada IPv4. Perhitungan dari node ke node dilakukan untuk mengetahui kinerja protocol tersebut menggunakan wireshark. Dalam penelitian ini bisa disimpulkan bahwa routing protocol RIPv6 lebih baik dari RIPv1 dan RIPv2, dibuktikan dari nilai delay average, packet loss, dan througput, penambahan perangkat router tidak berpengaruh secara signifikan terhadap kinerja routing protocol.

2. METODE PENELITIAN

2.1 Routing Protocol

Routing adalah suatu protocol yang digunakan untuk mendapatkan rute atau petunjuk dari satu jaringan ke jaringan yang lain, routing merupakan proses dimana suatu router akan memilih jalur atau rute untuk mengirimkan atau meneruskan suatu paket ke jaringan yang dituju. Router menggunakan IP address tujuan untuk mengirimkan paket, dan agar router mengetahui rute mana yang harus digunakan untuk meneruskan paket ke alamat tujuan, router harus belajar atau bertukar informasi sesama router yang saling terhubung untuk mengetahui jalur atau rute yang terbaik. Routing protocol digunakan untuk memfasilitasi pertukaran informasi routing antar router. Dengan routing protocol, router dapat berbagi informasi routing table, yaitu informasi mengenai jaringan lain yang saling terhubung. Ada beberapa routing protocol yang mendukung IPv6, yaitu RIPv6, OSPFv3, EIGRP untuk IPv6 (*Cisco propriarity*), IS-IS for IPv6, BGP IPv6, dan lainnya.

2.2 OSPFv3

Open Shortest Path First (OSPF) adalah *routing protocol* kelas *link-state* yang dikembangkan untuk memperbaiki kinerja dari *routing protocol* RIP. OSPF adalah *routing protocol* yang menggunakan konsep area. Kelebihan dari OSPF dibandingkan dengan RIP adalah kecepatan dalam melakukan konvergensi dan lebih luasnya jaringan yang bisa dijangkau. Pada dasarnya OSPFv3 menggunakan jenis paket yang sama pada OSPFv2. Perbedaan yang paling jelas ialah OSPFv3 mendukung pengalamatan 128-bit. OSPFv2 menggunakan alamat 224.0.0.5 dan 224.0.0.6, OSPFv3 menggunakan alamat *multicast* IPv6 yaitu FF02::5 dan alamat FF02::6 untuk *router DR* (*designated routers*) dan BDR (*Backup DRs*). OSPFv3

menggunakan alamat *link-local* untuk melakukan *advertisements* bukan alamat globalnya. Pada OSPF terdapat beberapa paket LSP (*Link-State Packets*), masing-masing paket dibutuhkan dalam proses *routing* pada OSPF. Berikut paket- paket LSP pada OSPFv3:

- Hello Paket Hello* digunakan untuk memulai dan menjaga keterhubungan informasi dengan *router* OSPF yang lain.
- DBD (*Paket Database Description*) DBD untuk memeriksa dan melakukan sinkronisasi *database* antar *router*.
- LSR (*Link-State Request*) LSR digunakan untuk menarik informasi dari *router* lain.
- LSU (*Link-State Update*) Paket ini digunakan untuk menjawab LSR
- LSAck (*Link-State Acknowledgment*) Paket ini digunakan untuk konfirmasi paket LSU yang diterima oleh *router*.

Masing-masing *router* OSPF menjaga *database* LSA yang diterima dari *router* lain. Ketika LSA dari semua *router* telah diterima maka *router* akan membangun sebuah *local link-state database*. OSPF menggunakan algoritma *Dijkstra's shortest path first* (SPF) untuk membangun sebuah *SPF tree*. *SPF tree* ini yang kemudian digunakan untuk membangun sebuah *routing table* dengan jalur terbaik guna mencapai jaringan yang lain.

Berikut merupakan proses terjadinya *konvergensi* pada *link-state routing protocol*:

- Masing-masing *router* mempelajari koneksinya yang terhubung ke jaringan secara langsung.
- Tiap *router* bertanggung jawab untuk "Hello" ke *router* tetangga yang terhubung langsung.
- Router* membangun *Link-State Packet* (LSP) yang berisi mengenai informasi *link* yang terhubung langsung.
- Masing-masing *router* akan mengirimkan LSP kesemua tetangganya, yang kemudian disimpan pada *database*.
- Tiap *router* menggunakan *database* untuk membangun sebuah peta topologi lengkap dari jaringan dan gambaran jalur atau rute yang dapat digunakan untuk mencapai jaringan tujuan yang ingin dicapai.

2.3 EIGRP

2.3.1 IPv4 EIGRP

EIGRP (*Enhanced Interior Gateway Routing Protocol*) adalah *routing protocol* yang hanya di adopsi oleh *router cisco* atau sering disebut sebagai *proprietary protocol* pada *cisco*. EIGRP ini sangat cocok digunakan untuk *midsize* dan *large company*. EIGRP sering disebut juga *Hybrid routing protocol*, karena pada EIGRPv6 terdapat dua tipe *routing protocol* yang digunakan, yaitu *distance vector* dan *link state*.

EIGRP ini pengembangan dari *routing protocol* IGRP (*distance vector*). Yang juga *proprietary cisco*. Perbandingan antara IGRP dan EIGRP dibagi menjadi beberapa kategori, yaitu *compability mode*, *metric colocation*, *hop count*, *automatic protocol redistribution* dan *route tagging*.

EIGRP dan IGRP dapat dikombinasikan satu sama lain karena EIGRP adalah hanya pengembangan dari IGRP. Dalam perhitungan untuk menentukan jalurnya, EIGRP menggunakan

algoritma DUAL (*Diffusing-Update algorithm*) dalam menentukannya. EIGRP mempunyai 3 tabel dalam menyimpan informasi networknya, yaitu *Neighbor table*, *topology table*, dan *routing table*.

Neighbor table adalah tabel yang menyimpan list tentang *router-router* tetangganya. Setiap ada *router* baru yang dipasang, *address* dan *interface* dicatat di tabel ini. DUAL mengambil informasi dari "*neighbor tabel*" dan "*topology table*" untuk melakukan kalkulasi "*lowest cost routes to each destination*". Setelah melakukan kalkulasi akan dibuat "*successor route*" yang disimpan di tabel ini. *Routing table* berfungsi menyimpan rute terbaik untuk ke tujuan. Informasi tersebut diambil dari "*topology table*". Untuk mudahnya dapat dilihat pada Gambar 1.



Gambar.1 Alur Tabel pada EIGRP

Jumlah di mana EIGRP mengirimkan paket *hello* disebut *hello Interval*, dan dapat disesuaikan per antarmuka dengan perintah `ip hello-interval eigrp`. Waktu tunggu adalah jumlah waktu yang *router* butuhkan untuk mempertimbangkan *router* tetangga hidup tanpa menerima paket halo. Waktu tunggu biasanya tiga kali *hello Interval*, secara default, 15 detik dan 180 detik. Dan dapat terus menyesuaikan waktu dengan perintah `ip hold-time eigrp`.

2.3.2 IPv6 EIGRP

Pada IPv6 EIGRP semua fitur dan konfigurasi hampir sama seperti pada IPv4. Hanya sedikit perbedaan pada IPv6, yaitu keadaan default jaringan berada dalam posisi "shutdown", filter routing menggunakan perintah "*distribute-list prefix-list*", pada IPv6 tidak ada konsep *classfull*, dan membutuhkan *router-id*.

2.4 RIP

RIP adalah sebuah *routing protocol* jenis *distance-vector*. RIP mengirimkan *routing table* yang lengkap ke semua *interface* yang aktif setiap 30 detik. RIP hanya menggunakan jumlah *hop* untuk menentukan cara terbaik ke sebuah *network remote*, tetapi RIP secara default memiliki sebuah nilai jumlah *hop* maksimum yang diizinkan yaitu 15, yang berarti nilai 16 dianggap tidak terjangkau (*unreachable*). RIP bekerja dengan baik di *network-network* yang kecil, tetapi RIP tidak efisien pada *network-network* besar atau pada *network-network* yang memiliki banyak *router* terpasang.

2.4.1 RIP versi 1

Menggunakan *classful routing*. Pembaruan *routing* periodik tidak membawa *subnet* informasi, dukungan kurang untuk *subnet mask* panjang variabel (VLSM). Keterbatasan ini tidak memungkinkan untuk memiliki ukuran yang berbeda *subnet* yang sama dalam kelas jaringan. Dengan kata lain, semua *subnet* dalam jaringan kelas harus memiliki ukuran yang sama. Juga tidak ada dukungan untuk autentikasi *router*, membuat RIP rentan terhadap berbagai versi RIP attacks. RIP versi 1 hanya ada jumlah hop 16 (0-15). Jika ada lebih dari 16 hop antara dua *router* itu gagal untuk mengirim paket data ke alamat tujuan.

2.4.2 RIP versi 2

Karena kekurangan dari spesifikasi asli RIP, maka RIP versi 2 (RIPv2) di ciptakan, kemampuan yang di miliki untuk membawa informasi subnet, sehingga mendukung *classless inter-domain routing* (CIDR). Untuk menjaga kompatibilitas ke belakang, jumlah hop limit 15 tetap. RIPv2 memiliki fasilitas untuk sepenuhnya *interoperate* dengan spesifikasi awal jika semua protokol bidang harus Zero dalam pesan RIPv1 yang benar ditentukan. Dalam upaya untuk menghindari beban yang tidak perlu di host yang tidak berpartisipasi dalam *routing*, *multicast RIPv2 routing table* seluruh untuk semua *router* berdekatan di alamat 224.0.0.9, sebagai lawan RIPv1 yang menggunakan siaran. Pengalamatan *unicast* masih diperbolehkan untuk aplikasi khusus.

2.4.3 RIPng

Routing Information Protocol Next Generation adalah *routing protocol* yang berdasarkan *routing protocol* RIP di IPv4 yang sudah mendukung IPv6. RIPng ini digunakan untuk *internal routing protocol* dan menggunakan protokol UDP sebagai *transport*. Tidak seperti RIPv1/v2 yang berjalan pada port UDP 520, RIPng ini menggunakan port 521 sebagai komunikasi antar RIPng.

Metode yang dipakai RIPng adalah *distance vector* (vektor jarak), yaitu:

- Jarak *local network* dihitung 0
- Kemudian mencari *neighbour* sekitar dan dihitung jaraknya dan *cost*.
- Dibandingkan jarak dan *cost* antar *neighbour*.
- Dilakukan perhitungan secara *continue*.
- Menggunakan algoritma *Ballman-Ford*.

Command pada RIPng Header berisi:

- Request*, meminta daftar *routing table* pada RIPng yang lain.
- Response*, membalas *request* dari RIPng yang lain dan memberikan daftar *routing*.

Protokol RIPng ini memiliki beberapa kelemahan

- Hanya bisa sampai 15 HOP.
- Lambat dalam memproses *routing*, dikarena melakukan pengecekan terus menerus.
- Bersifat *Classful*.

RIPng menggunakan *timer*, prosedur, dan tipe *message* yang sama dengan RIPv2. Misalnya, RIPv2 menggunakan *update timer* 30 detik yang telah ditambahi sedikit untuk mencegah sinkronisasi, periode *timeout* 180-detik, dan timer untuk *garbage-collection* 120 detik, dan *holddown*

timer 180 detik. RIPng juga menggunakan *metric hop-count*, dengan 16 menunjukkan nilai *unreachable*. Dan juga menggunakan *Request* dan *Response messages* dengan cara yang sama seperti RIPv2. Serta pesan *Request* dan *Response* dikirim secara *multicast* dengan sedikit pengecualian untuk unicast yang digunakan RIPv1 dan v2. *Address multicast Ipv6* yang digunakan RIPng adalah FF02::9.

RIPng menentukan *address next-hop* dengan cara yang sama seperti RIPv2. Dengan kata lain, *address next-hop non-zero* yang valid menentukan router *next-hop* selain dari pengirim dari *message Response* dan *address next-hop 0:0:0:0:0:0:0:0* menentukan pengirim dari *message Response* itu sendiri sebagai *address next-hop*. Bedanya, RIPng menentukan bahwa *address next-hop* pada *entri route spesial* kemudian mengelompokkan semua entri route yang menggunakan *address next-hop* setelahnya.

2.5 Quality of Service (QoS)

Quality of Service adalah kemampuan untuk memberikan prioritas yang berbeda untuk berbagai aplikasi, pengguna, atau aliran data, atau untuk menjamin tingkat kinerja pada aliran data. QoS bertujuan untuk menyediakan kualitas layanan yang berbeda-beda untuk beragam kebutuhan akan layanan di dalam jaringan IP, sebagai contoh untuk menyediakan bandwidth yang khusus, menurunkan hilangnya paket-paket, menurunkan waktu tunda dan variasi waktu tunda di dalam proses transmisinya. Parameter-parameter dalam QoS antara lain: *Throughput*, *delay*, *jitter*, *packet loss*.

2.5.1 Throughput

Throughput adalah kecepatan rata-rata data yang diterima oleh suatu node dalam selang waktu pengamatan tertentu. Karena sejumlah faktor, *Throughput* biasanya tidak sesuai dengan bandwidth yang ditentukan dalam implementasi lapisan fisik seperti Ethernet. Banyak faktor yang mempengaruhi *Throughput*. Diantara faktor-faktor tersebut jumlah lalu lintas, jenis lalu lintas, dan jumlah perangkat jaringan ditemui pada jaringan yang diukur. GPON menggunakan TDMA sebagai teknik *multiple access upstream* dengan data rate sebesar 1.2 Gbps dan menggunakan *broadcast* kearah *downstream* dengan data rate sebesar 2.5 Gbps. Model paketisasi data menggunakan GEM (GPON *Encapsulation Methode*) atau ATM cell untuk membawa layanan TDM dan *packet based*. GPON jadi memiliki efisiensi *bandwidth* yang lebih baik dari BPON (70 %), yaitu 93 %.

Tabel.1 Tingkat Kualitas *Throughput*

Kategori <i>Throughput</i>	<i>Throughput</i>	Indeks
Sangat bagus	100%	4
Bagus	75%	3
Sedang	50%	2
Jelek	< 25%	1

2.5.2 Delay

Delay adalah waktu tunda suatu paket yang diakibatkan oleh proses transmisi dari suatu *node* ke *node* lain yang menjadi tujuannya. Penundaan dapat menyebabkan masalah yang signifikan pada QoS aplikasi pada suara dan video, dan aplikasi seperti SNA dan transmisi faks yang gagal dalam kondisi penundaan yang berlebihan. Beberapa aplikasi dapat mengimbangi sejumlah kecil penundaan tetapi jumlah yang telah terlampaui, QoS menjadi terganggu. Sebagai contoh, *gateway VoIP* dan telepon menyediakan beberapa penyangga lokal untuk mengkompensasi delay jaringan. *Delay* terdiri dari *delay* tetap ataupun *delay* variabel. Contoh *delay* tetap adalah:

- Aplikasi berbasis *delay*, *Delay* yang disebabkan oleh waktu yang diperlukan untuk proses pembuatan paket pada sisi sumber informasi atau pengirim.
- Transmisi data, (*delay* antrian) atas media jaringan fisik pada setiap jaringan
- Propagasi *delay* waktu proses dari seluruh jaringan berdasarkan jarak transmisi

Contoh *delay* variabel adalah:

- Ingress queuing delay*, untuk trafik yang memasuki node jaringan
- Contention*, hubungan dengan trafik lainnya pada setiap node jaringan
- Egress queuing delay*, untuk lalu lintas keluar node jaringan

Tabel.2 Tingkat Kualitas *Delay*

Kategori Latensi	Besar <i>Delay</i>	Indeks
Sangat Bagus	< 150% ms	4
Bagus	150 s/d 300 ms	3
Sedang	300 s/d 450 ms	2
Jelek	> 450 ms	1

2.5.3 Jitter

Jitter adalah ukuran variasi delay antar paket yang berturut-turut untuk arus trafik tertentu. *Jitter* memiliki efek pada real-time, aplikasi yang mempunyai delay-sensitif seperti suara dan video. aplikasi real-time ini mengharapkan untuk menerima paket pada tingkat yang konstan dengan delay tetap antara paket yang berturut-turut. Sebagai tingkat kedatangan bervariasi, *jitter* berdampak pada kinerja aplikasi. Jumlah minimal sebuah *jitter* dapat diterima, tetapi meningkatnya *jitter* dapat menyebabkan aplikasi tidak bisa digunakan. Semua jaringan memiliki beberapa *jitter* karena variabilitas dalam delay dimiliki oleh setiap node jaringan sebagai paket antrian. Namun, selama *jitter* dapat dibatasi, QoS dapat dipertahankan.

Tabel.3 Tingkat kualitas *jitter*

Kategori penilaian	<i>Jitter</i>
Baik	0-25 ms
Bisa diterima	25-50 ms
Tidak bisa diterima	> 50 ms

2.5.4 Packet Loss

Packet loss didefinisikan sebagai kegagalan transmisi paket mencapai tujuannya. Kegagalan paket tersebut mencapai tujuan dapat disebabkan oleh beberapa kemungkinan antara lain:

- Terjadinya *overload* trafik di dalam jaringan
- Tabrakan (*congestion*) dalam jaringan
- Error* yang terjadi pada media fisik
- Kegagalan yang terjadi pada sisi penerima, antara lain dapat disebabkan karena *overflow* yang terjadi pada *buffer*
- Di dalam Implementasi jaringan IP (*Internet Protocol*), nilai *packet loss* ini diharapkan mempunyai nilai yang minimum.

Tabel.4 Tingkat kualitas *packet loss*

Kategori penilaian	<i>packet loss</i>
Baik	0-5 %
Masih dapat diterima	5-25%
Tidak dapat diterima	>25%

2.6 Metode Pengujian

2.6.1 Metode Pengujian Pemilihan Jalur oleh Routing Protocol

Pengujian pada penelitian ini dilakukan dengan menentukan *trace route* yang akan dilewati paket data dari masing-masing *routing protocol*. *Routing table* menyimpan informasi rute-rute atau jalur untuk mencapai jaringan yang lain, dan *routing table* juga menyimpan *metric* dari rute-rute yang ada. Untuk melihat *routing table* dari *router* dapat dilakukan dengan mengetikkan perintah pada CLI *command line interface* "*show ip route*" untuk *router cisco*, atau perintah "*show ipv6 route*" untuk menampilkan informasi *routing table* IPv6 pada *router*. pada masing-masing *router* dilakukan perintah untuk menampilkan *routing table* baik dalam topologi RIPng, topologi EIGRPv6 atau topologi OSPFv3.

```

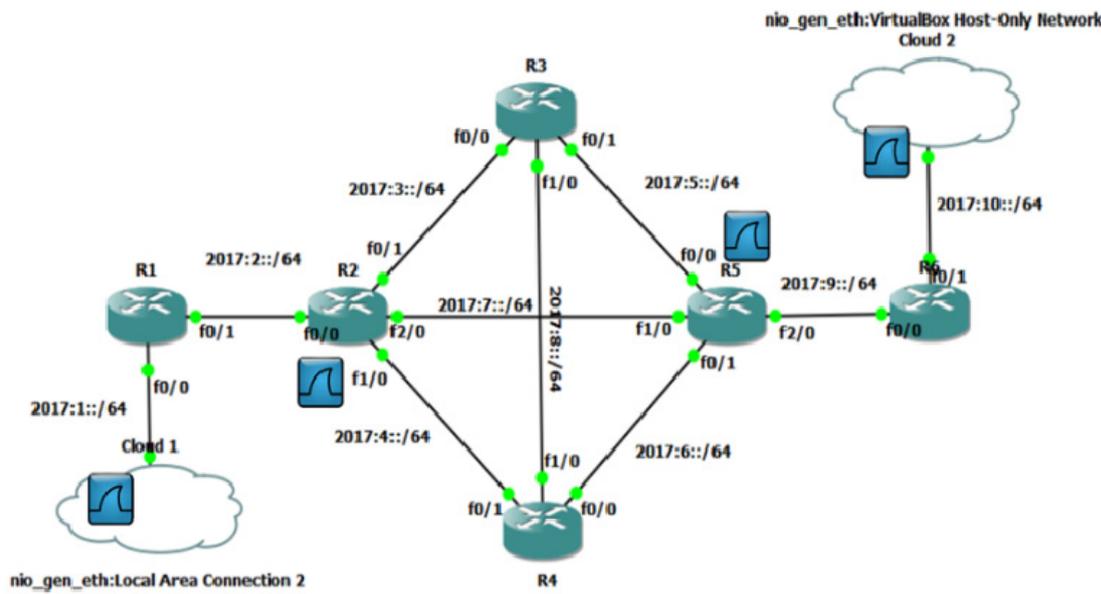
R2#show ipv6 route
IPv6 Routing Table - 0 entries
Codes: C - Connected, I - Local, S - Static, R - RIP, B - BGP
U - Per-user Static route, M - MIPv6
D - ISIS D1, D2 - ISIS L2, IA - ISIS Interarea, IS - ISIS summary
O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
D - EIGRP, EX - EIGRP External
D 2017:1::/64 [90/307200]
  via FE80::C004:AFF:FE0C:1, FastEthernet0/0
E 2017:2::/64 [0/0]
  via 11, FastEthernet0/0
E 2017:3::/64 [0/0]
  via 11, FastEthernet0/0
E 2017:4::/64 [0/0]
  via 11, FastEthernet0/1
E 2017:5::/64 [0/0]
  via 11, FastEthernet0/1
E 2017:6::/64 [0/0]
  via 11, FastEthernet0/1
E 2017:7::/64 [0/0]
  via 11, FastEthernet0/0
E 2017:8::/64 [0/0]
  via 11, FastEthernet0/0
E 2017:9::/64 [0/0]
  via FE80::C004:AFF:FE0C:10, FastEthernet2/0
D 2017:10::/64 [90/284160]
  via FE80::C004:AFF:FE0C:10, FastEthernet2/0
E 2017:11::/64 [0/0]
  via 11, FastEthernet2/0
E 2017:12::/64 [0/0]
  via 11, FastEthernet2/0
E 2017:13::/64 [0/0]
  via FE80::C004:AFF:FE0C:1, FastEthernet1/0
D 2017:14::/64 [90/307200]
  via FE80::C004:AFF:FE0C:10, FastEthernet2/0
E 2017:15::/64 [0/0]
  via FE80::C004:AFF:FE0C:10, FastEthernet2/0
D 2017:16::/64 [90/284720]
  via FE80::C004:AFF:FE0C:10, FastEthernet2/0
E 2017:17::/64 [0/0]
  via 11, Null0
  
```

Gambar.2 Routing table IPv6 pada R2

Perintah "*trace route*" atau "*tracert*" digunakan untuk mengetahui jalur yang akan dilalui paket data. *Tracert* menggunakan protokol ICMP (*Internet Control Messaging Protocol*), protokol ini bekerja dengan mengirimkan ICMP *echo request* ke alamat tujuan. Rute yang dilalui dan ditampilkan adalah daftar *interface* router yang digunakan pada jalur antara host dan tujuan. Pada pengujian dilakukan perintah *tracert* dari *cloud 2* ke *cloud 1*, hal ini dilakukan untuk mengetahui jalur atau *router* mana saja yang dilalui oleh paket untuk menuju *cloud 1*.

2.6.2 Metode Pengujian Update Routing Table dan Kecepatan Waktu

Capture pada paket data, untuk mengetahui paket dari masing-masing *routing protocol*, proses update *routing table* pada *routing protocol*, dan kecepatan masing-masing *routing protocol* dalam melakukan *update table*. Berikut gambar 3 adalah gambar titik-titik pengujian dengan *capture* paket data pada jaringan.



Gambar.3 pengujian dengan capture paket data

Pengujian yang dilakukan adalah dengan melakukan pengiriman paket berupa audio video stream local area network sebagai trafik dari server (cloud 1) menuju client (cloud 2), lalu dilakukan capture pada paket data yang melintas menggunakan wireshark. Capture dilakukan pada interface local area connection 2 dan virtualbox host-only networ Lalu untuk menguji jalur yang dipilih routing table dilakukan dengan trace route pada server menuju client yaitu dengan perintah "tracert 2017:10::2". Dengan perintah trace route dapat diketahui interface-interface dan router yang dilalui paket untuk mencapai tujuan.k.

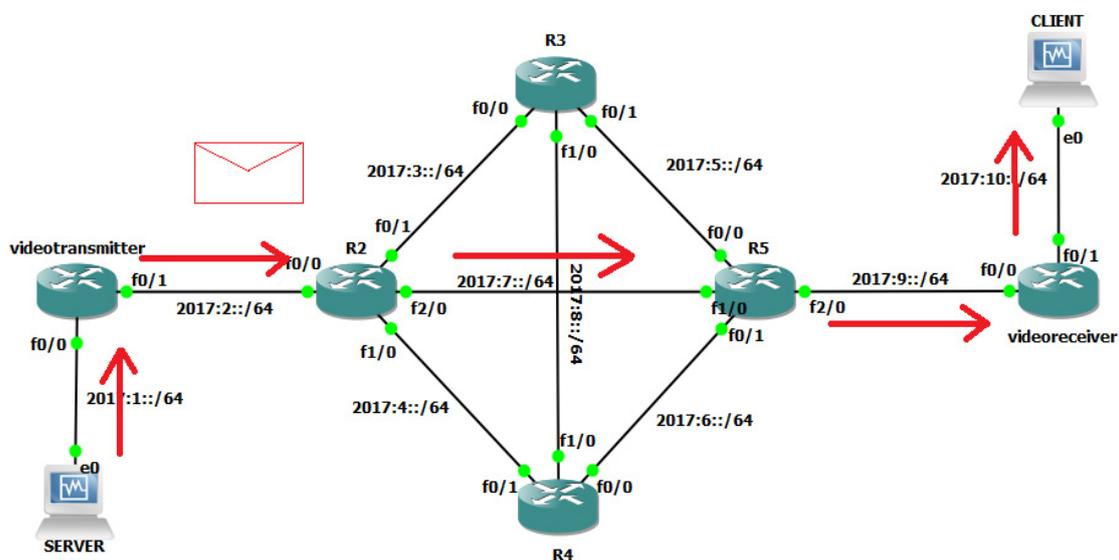
3 HASIL DAN PERANCANGAN

3.1 Pengujian Pemilihan Rute Jaringan

Setelah semua router, server dan client dikonfigurasi. Topologi jaringan sudah konvergensi. Lalu dilakukan analisa routing table pada masing-masing router. Dari routing table terlihat jaringan-jaringan yang bisa dijangkau dan jalur-jalur yang dipilih oleh tiap routing protocol (OSPFv3, EIGRPv6 dan RIPng).

Lalu untuk menguji jalur yang dipilih routing table dilakukan dengan trace route pada server menuju client yaitu dengan perintah "tracert 2017:10::2". Dengan perintah trace route dapat diketahui interface-interface dan router yang dilalui paket untuk mencapai tujuan.

Pada masing – masing routing protocol, memiliki cara sendiri dalam memilih jalur untuk membangun suatu jaringan. Pada semua topologi routing protocol media sebagai trafik yang dikirim pada masing – masing link adalah sama.



Gambar.4 Pemilihan jalur

Dapat dilihat pada gambar 3 dan 4 pemilihan topologi menggunakan topologi mesh, semua perangkat router saling terhubung satu sama lain. Sehingga paket data yang dikirim akan mencari jalur tercepat untuk mencapai tujuan. Yaitu dari videotransmitter menuju R2 kemudian ke R5 dan terakhir melalui videoreceiver.

3.1.1 Pengujian Routing Update

Pada penelitian routing ini, dilihat dari router yang dilewati oleh paket data dari server menuju client di GNS3. Untuk melihat jalur yang dilewati menggunakan command tracert seperti yang sudah dilakukan pada Gambar 3. Lama routing update ketika salah satu router yang menjadi jalur aktif diputus, yaitu perubahan table yang dilewati. Dengan router yang diputus adalah router R2 interface fastethernet 2/1, karena semua routing protocol memilih jalur yang sama sebagai jalur aktif seperti yang ditunjukkan pada gambar 3 dan 4.

Ketiga routing protocol menggunakan jalur yang sama untuk dijadikan jalur aktif pengiriman paket data. Ketika salah satu interface yang

menjadi jalur aktif paket data diputus maka paket data yang dikirim akan mencari jalur alternatif lain untuk mencapai tujuan yaitu menuju client. Pada pengujian routing update pada routing protocol RIPng memiliki hasil rata-rata 44.2 ms saat tidak ada pemutusan jalur. Namun pada saat terjadinya link down RIPng sangat lambat dalam proses update routing, bahkan bisa dibilang buruk karena mendapatkan pesan "destination has unreachable" dan "request time out" yang berarti tidak bisa mengenali interface alternatif. Sehingga dibutuhkan pengecekan terus menerus untuk mendaftarkan interface alternatif yang akan menjadi jalur alternatifnya. Pada pengujian EIGRPv6 dalam melakukan routing update. EIGRPv6 mampu membaca interface tetangga tanpa perlu melakukan pemindaian terus menerus akan tetapi EIGRPv6 memiliki nilai latency yang lebih tinggi dari RIPng yaitu memiliki rata-rata 117 ms. Sedangkan pada pengujian routing update pada OSPFv3 lebih baik dalam melakukan routing update dan nilai latency nya stabil dengan rata-rata 42.2 ms. Hal tersebut

karena OSPFv3 mempunyai *neighbor table* yang menyimpan list tentang router-router tetangganya.

3.1.2 Pengujian Paket ICMPv6

Pada pengujian ini dilakukan pengiriman ICMPv6 (*Internet Control Message Protocol IPv6*). Adalah protokol jaringan yang berfungsi untuk memberikan pesan-pesan ke dalam sebuah jaringan, mulai dari pesan *error*, pesan diterima, dan *connection lost*. Pengambilan data ICMPv6 berupa parameter *delay* dan *jitter* saja. Paket ICMPv6 tidak ada *packet loss* karna pengujian dilakukan dengan 10 buat paket yang dikirim dan paket tersebut terkirim semua tanpa adanya paket yang hilang. Seperti yang ditunjukkan pada Gambar 5.

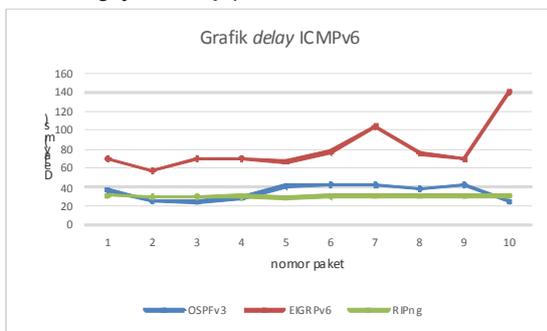
```
C:\Users\JONI>ping 2017:10::2 -n 10
Pinging 2017:10::2 with 32 bytes of data:
Reply from 2017:10::2: time=398ms
Reply from 2017:10::2: time=514ms
Reply from 2017:10::2: time=371ms
Reply from 2017:10::2: time=423ms
Reply from 2017:10::2: time=395ms
Reply from 2017:10::2: time=337ms
Reply from 2017:10::2: time=379ms
Reply from 2017:10::2: time=290ms
Reply from 2017:10::2: time=290ms
Reply from 2017:10::2: time=358ms

Ping statistics for 2017:10::2:
    Packets: Sent = 10, Received = 10, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 290ms, Maximum = 514ms, Average = 375ms
```

Gambar.5 Paket ICMPv6

Tidak adanya paket loss pada paket ICMPv6 membuktikan bahwa konfigurasi perangkat *router* dan *host* pada topologi yang dirancang sudah saling terhubung satu sama lain. Lalu diamati *delay* dan *jitter* paket tersebut.

- Pengujian *delay* paket ICMPv6



Gambar.6 Grafik delay ICMPv6

Pada Tabel.2 untuk mencari rata-rata *delay* digunakan rumus sebagai acuan dalam mengetahui berapa besarnya rata-rata *delay* dari total paket yang dikirim pada masing-masing *routing protocol*, rumus yang digunakan untuk ketiga metode *routing* tersebut adalah sama, yaitu:

$$\text{Delay rata-rata} = \frac{\text{Total delay}}{\text{total paket yang diterima}} \dots (1)$$

Perhitungan menggunakan rumus diambil 1 *sample* metode *routing* saja untuk dilakukan perhitungan dengan rumus pada metode *routing* OSPFv3 yaitu :

$$\text{Delay Rate-rata} = \frac{34814}{10} = 34.814 \text{ ms}$$

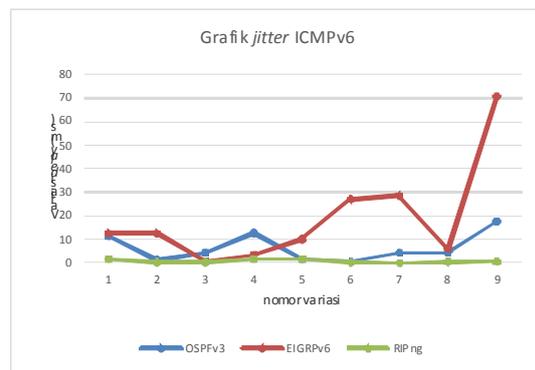
Berdasarkan perhitungan didapatkan rata-rata *delay* dari ketiga metode *routing* yaitu 34.814 ms untuk OSPFv3, 80.158 ms untuk EIGRPv6 dan 30.471 ms untuk RIPng. Sedangkan untuk mencari *delay* dari 1 paket ICMPv6 di *wireshark* digunakan rumus.

$$\text{Delay} = \text{time paket diterima} - \text{time paket dikirim} \dots (2)$$

Perhitungan terhadap *delay* OSPFv3 paket no. 1

$$\text{Delay} = 5.724226 - 5.692564 = 0.031662 \text{ s}$$

- Pengujian *jitter* paket ICMPv6



Gambar.7 Grafik jitter ICMPv6

Kemudian untuk mengetahui *jitter* pada paket ICMPv6 digunakan rumus yang berlaku bagi ketiga metode *routing* (OSPFv3, EIGRPv6 dan RIPng), yaitu:

$$\text{Jitter} = \frac{\text{Total variasi delay}}{\text{Total paket yang diterima}-1} \dots (3)$$

Perhitungan pencarian *jitter* menggunakan rumus hanya dilakukan sebagai *sample* terhadap 1 *routing protocol* OSPFv3 yaitu didapatkan:

$$\text{Jitter} = \frac{30.97}{10-1} = \frac{30.97}{9} = 6.88$$

Setelah dilakukan pencarian dengan menggunakan rumus *jitter* didapatkan hasil *jitter* dari pengujian dengan pengiriman paket ICMPv6 pada ketiga metode *routing* yaitu sebesar 6.33 ms untuk OSPFv3, 19.012 ms untuk EIGRPv6 dan RIPng sebesar 0.66 ms. Pada pengujian *jitter* paket ICMPv6 nilai *jitter* pada RIPng memiliki nilai yang paling kecil dibandingkan OSPFv3 dan EIGRPv6.

3.2 Pengujian performa dengan pengiriman paket RTP (Audio Video Stream)

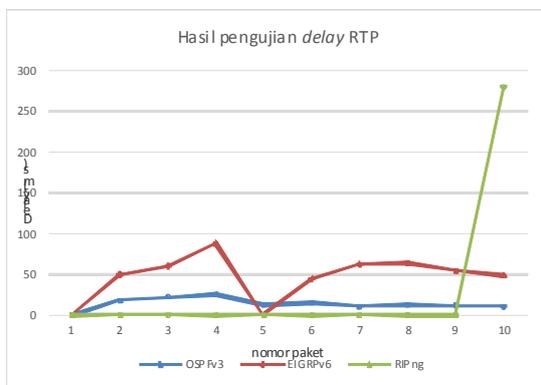
Pada pengujian ini digunakan perhitungan data menggunakan RTP. paket data RTP didapat dari pengiriman audio dan video yang dikirim dari server menuju *client*. Karena pada *network analyzer* paket data yang ditangkap berupa UDP, maka diubah terlebih dahulu atau *decode* menjadi paket RTP, guna mendapatkan hasil parameter QoS. Penelitian dilakukan dengan 5 kali pengujian. Pada

saat mencapai 8000 *packet* dan 8000 *displayed* (100%) yang di *capture network analyzer wireshark*, lalu dilakukan pemberhentian *capture* pada masing-masing *routing protocol*. Kemudian didapat hasil rata-rata *delay*, *packet loss*, *throughput* dan *jitter*. Dari pengukuran berdasarkan analisis data dari *software network analyzer wireshark* didapatkan statistik.

Pada pengujian ini digunakan perhitungan data menggunakan RTP. paket data RTP didapat dari pengiriman audio dan video yang dikirim dari server menuju *client*. Karena pada *network analyzer* paket data yang ditangkap berupa UDP, maka diubah terlebih dahulu atau *decode* menjadi paket RTP, guna mendapatkan hasil parameter QoS. Penelitian dilakukan dengan 5 kali pengujian. Pada saat mencapai 8000 *packet* dan 8000 *displayed* (100%) yang di *capture network analyzer wireshark*, lalu dilakukan pemberhentian *capture* pada masing-masing *routing protocol*. Kemudian didapat hasil rata-rata *delay*, *packet loss*, *throughput* dan *jitter*. Dari pengukuran berdasarkan analisis data dari *software network analyzer wireshark* didapatkan statistik.

3.2.1 Pengukuran Delay Paket RTP

Delay adalah waktu tunda suatu paket yang diakibatkan oleh proses transmisi dari suatu *node* ke *node* lain yang menjadi tujuannya. Pengukuran *delay* ini diambil *sample* diperoleh 10 paket pertama dari 8000 paket yang dikirim. Lalu didapatkan rata-rata *delay* pada masing-masing pengujian yang telah diamati.



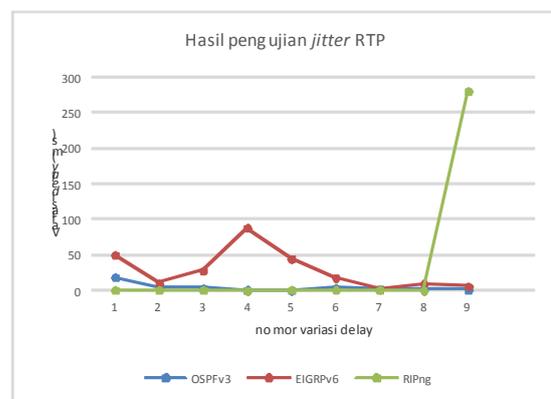
Gambar.8 Hasil pengujian delay RTP

Hasil pengujian *delay* dari paket RTP yang diambil 10 paket pertama dengan menggunakan *network analyzer wireshark*, diperoleh nilai rata-rata yaitu 14.16 ms untuk OSPFv3, 47.237 ms untuk EIGRPv6 dan 28.22 ms untuk RIPng. Dari Gambar.8 terlihat perbedaan nilai rata-rata *delay* dari 10 paket pertama yang dikirim pada masing-masing *routing protocol* sangat sedikit karena ketiganya di implementasikan pada jaringan dengan topologi yang sama yaitu *mesh*. Nilai rata-rata *delay* dari 10 paket pada RIPng lebih kecil dibandingkan OSPFv3 dan EIGRPv6. Namun pada *routing protocol* RIPng dengan nomor paket 10 mengalami kenaikan sebesar 280.20 ms. Dalam parameter *delay* kenaikan nilai yang sangat tinggi dari paket sebelumnya dapat mengganggu visual yang

ditampilkan saat paket data dikirim. Serta dapat menyebabkan kehilangan paket atau *packet loss*. Sedangkan nilai *delay* dari 10 paket pertama yang dikirim pada OSPFv3 dan EIGRPv6 cenderung stabil. Perlu diketahui bahwa nilai *delay* tidak *absolute* karena dipengaruhi banyak faktor dan kondisi jaringan pada saat pengiriman paket data.

3.2.2 Pengukuran Jitter Paket RTP

Jitter adalah suatu parameter yang menunjukkan variasi *delay* antar paket dalam pengiriman yang sama. Jaringan yang baik adalah jaringan yang memiliki nilai *jitter* yang kecil. *Jitter* memiliki efek pada *real-time*, aplikasi yang mempunyai delay sensitif seperti suara dan video. *Jitter* dapat menyebabkan *packet loss* terutama pada kecepatan transmisi yang tinggi. Pengujian *jitter* ini diperoleh dari 10 paket pertama dari 8000 paket yang diamati.



Gambar.9 Hasil pengujian jitter RTP

Nilai *jitter* yang didapat memiliki nilai yang fluktuatif pada tiap-tiap *routing protocol*. Dengan *routing protocol* RIPng memiliki nilai rata-rata *jitter* yang lebih besar yaitu 31.47 ms dibandingkan *jitter* dengan EIGRPv6 sebesar 28.41 ms dan OSPFv3 sebesar 5.56 ms. Secara umum nilai *jitter* meningkat dengan nilai besaran yang fluktuatif. Hal ini terjadi karena semakin banyak data yang dikirim maka semakin besar kemungkinan terjadinya tabrakan (*congestion*) pada jaringan. Sama seperti pengujian *delay*, dalam *jitter* EIGRPv6 dan RIPng memiliki nilai yang lebih besar dibandingkan metode *routing* OSPFv3. Kemudian nilai rata-rata *jitter* untuk metode *routing* OSPFv3 lebih kecil dari EIGRPv6 dan RIPng, hal ini menunjukkan kestabilan dalam pengiriman paket data pada metode *routing* OSPFv3. Karena metode *routing* OSPFv3 mampu beradaptasi dengan kegagalan jaringan.

3.2.3 Pengukuran Packet Loss Paket RTP

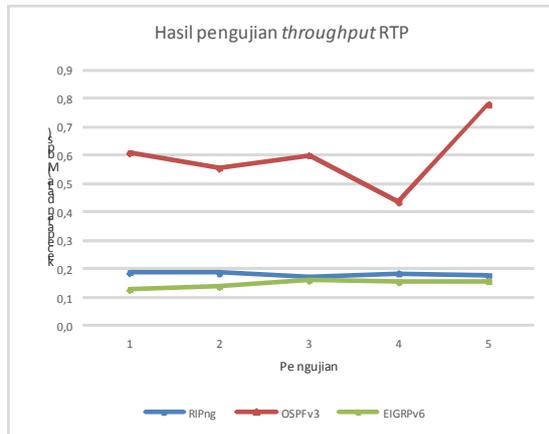
Packet loss didefinisikan sebagai kegagalan transmisi paket dari sumber mencapai tujuannya. Pada pengujian *packet loss* ini dilakukan dengan 5 kali pengujian dan pengiriman paket data sebanyak 8000 paket yang ditangkap oleh *wireshark* pada masing-masing *routing protocol*.

Berdasarkan kelima pengujian didapatkan nilai *packet loss* EIGRP sedikit lebih baik dari RIPng dengan nilai rata-rata *packet loss* sebesar 47.80%. Dan pada pengujian OSPFv3 memiliki nilai *packet*

loss yang lebih baik dibanding RIPng dan EIGRPv6, yaitu 22.77%

3.2.3 Pengukuran Throughput Paket RTP

Throughput adalah parameter yang menunjukkan jumlah bit rata-rata data yang dapat ditransfer dari satu *node* ke *node* yang lain perdetiknya. Pengujian *throughput* dengan paket RTP diamati dari 5 kali pengujian dan 8000 paket yang ditangkap oleh *network analyzer wireshark*.



Gambar.10 Hasil pengujian throughput RTP

Pada Gambar.10 dapat dilihat bahwa nilai *throughput* pada *routing protocol* OSPFv3 memiliki rata-rata nilai atau kecepatan akses data sebesar 0.595 Mbit/s, lalu pada *routing protocol* RIPng memiliki nilai rata-rata 0.182 Mbit/s, sedangkan pada EIGRPv6 memiliki nilai rata-rata yang lebih kecil dibanding RIPng dan OSPFv3 yang hanya 0.149 Mbit/s.

3.3 Analisa Quality of Service (QoS)

Pada pengujian layanan RTP (*real-time transport protocol*) yaitu dengan mengirimkan video secara stream pada masing-masing *routing protocol* (OSPFv3, EIGRPv6 dan RIPng) yang sudah dilakukan. Dapat disimpulkan bahwa *routing protocol* jenis *link-state* yaitu OSPFv3 yang lebih optimal dalam menampilkan visual yang dikirimkan, dibandingkan dengan EIGRPv6 dan RIPng. Dibuktikan dengan nilai QoS yang sudah diuji. Nilai QoS pada OSPFv3 memiliki nilai yang lebih baik dari EIGRPv6 dan RIPng. Seperti terlihat pada Tabel.5.

Tabel.5 Quality of Service

Rata-rata	OSPFv3	EIGRPv6	RIPng
Delay (ms)	14.16	47.237	28.22
Jitter (ms)	5.56	28.41	31.47
Packet Loss (%)	9.65	50.44	70.08
Throughput Mbit/s	0.595	0.149	0.182

4. KESIMPULAN

Dari hasil pengujian simulasi dan analisis yang sudah dilakukan, dapat diambil kesimpulan ialah sebagai berikut:

1. *Routing protocol* RIPng tidak cocok diimplementasikan pada *network* IPv6 berskala besar. Karena berdasarkan pengujian *routing update*, RIPng sangat lambat dalam merespon *interface* R3 atau R4 dalam perubahan topologi. Sementara yang lebih baik dalam pengujian *routing update* ini adalah OSPFv3 dan mampu dengan cepat membaca *interface* R3 atau R4 jika ada perubahan topologi. OSPFv3 memiliki nilai rata-rata *latency* sebesar 52.4 ms, lebih stabil dibandingkan dengan EIGRPv6 mencapai 160.2 ms dan RIPng mengalami kegagalan update sejumlah 6 paket dari 10 kali pengulangan.
2. Dalam pengujian paket ICMPv6 RIPng memiliki hasil rata-rata sekitar 30.471 ms untuk *delay* dan 0.66 ms untuk *jitter*. Hasil tersebut membuat *routing* RIPng lebih baik dari *routing* OSPFv3 34.814 ms untuk *delay*, 6.33 ms untuk *jitter* dan EIGRPv6 yang memiliki hasil 80.158 ms *delay*, 80.158 ms untuk *jitter*.
3. Pada percobaan *delay* paket RTP, OSPFv3 membutuhkan waktu *delay* sebesar 14.16 ms, daripada EIGRPv6 dengan *delay* 47.237 ms dan RIPng dengan 28.22 ms pada pengukuran 10 paket pertama yang ditangkap menggunakan *wireshark*. Hal ini membuat *routing protocol* OSPFv3 unggul dalam hal *delay*.
4. Untuk pengujian *jitter* paket RTP, OSPFv3 masih memiliki *jitter* lebih baik sekitar 5.56 ms daripada EIGRPv6 dengan *jitter* 28.41 ms. Sedangkan RIPng memiliki nilai *jitter* tertinggi yaitu 31.47 ms.
5. Hasil pengujian *packet loss* paket RTP dari lima kali pengujian pada RIPng memiliki hasil yang paling tinggi yaitu sekitar 69.06% paket yang hilang. Sedangkan pada EIGRPv6 memiliki hasil sekitar 48.15%. Lalu pada OSPFv3 mendapatkan hasil lebih baik dari RIPng dan EIGRPv6. Jika dirata-rata hanya 13.03% paket yang hilang.
6. Untuk nilai *Throughput* paket RTP pada OSPFv3 masih memiliki hasil yang baik 0.595 Mbit/s dari EIGRPv6 dengan 0.149 Mbit/s dan RIPng 0.182 Mbit/s
7. Berdasarkan kesimpulan point nomor 3 sampai dengan point nomor 6. Dapat disimpulkan bahwa performansi *routing protocol* OSPFv3 pada jaringan IPv6 dengan pengujian *Real-time Transport Protocol* yang telah dilakukan memiliki nilai QoS lebih baik dari EIGRPv6 dan RIPng.

DAFTAR PUSTAKA

- [1]. Budiyanto, Setyo “Analisa Kinerja Protokol IS-IS dan OSPF v3 untuk Layanan Video Streaming”, Jurnal Tugas Akhir, Teknik Elektro Fakultas Teknik Elektro Universitas Mercu Buana, Jakarta 2014.
- [2]. Dwi Aryanta, Arsyad Ramadhan Darlis, Dimas Priyambodho, “Analisis Kinerja EIGRP dan OSPF Pada Topologi Ring dan Mesh”, Jurnal ELKOMIKA, Teknik Elektro Institut Teknologi Nasional Bandung, Volume 2, Nomor 1, Juni 2014.
- [3]. Edi Yusuf, Dwi Aryanta, Lita Lidyawati, “Perancangan dan Analisis Kinerja EIGRP pada Jaringan IPv6”, Jurnal Online Institut Teknologi Nasional Volume 4, Nomor 2, Juli 2016.
- [4]. Irsan Fitrah Adhil, Linna Oktaviana Sari, “Analisa Proses Routing RIPv1, RIPv2 dan RIPng Antara Model IPv4 Dengan IPv6 Pada Jaringan Data PT. Pertamina RU II Dumai”, Jurnal FTEKNIK Elektro Universitas Riau, Volume 3 No 2, Oktober 2016.
- [5]. Iwan Sofana, Cisco CCNA-CCNP ROUTING DAN SWITCHING, Januari 2017, Informatika Bandung, 2017.
- [6]. Rahmadita, Ayu, Dwi, “Analisa Aplikasi VoIP Pada Jaringan Berbasis MPLS” Jurnal Tugas Akhir, Program Studi Teknik Telekomunikasi, Politeknik Elektronika Negeri Surabaya Institut Teknologi Sepuluh Nopember, Surabaya 2014.