



**UNIVERSITAS INDONESIA**

**ANALISIS UNJUK KERJA  
WIRELESS MESH NETWORK  
DENGAN ROUTING PROTOCOL OLSR**

**SKRIPSI**

**BAGUS MARDANI  
06 06 04 2342**

**FAKULTAS TEKNIK  
PROGRAM STUDI TEKNIK ELEKTRO  
DEPOK  
DESEMBER 2008**



**UNIVERSITAS INDONESIA**

**ANALISIS UNJUK KERJA  
WIRELESS MESH NETWORK  
DENGAN ROUTING PROTOCOL OLSR**

**SKRIPSI**

**Diajukan sebagai salah satu syarat untuk memperoleh gelar sarjana teknik**

**BAGUS MARDANI  
06 06 04 2342**

**FAKULTAS TEKNIK  
PROGRAM STUDI TEKNIK ELEKTRO  
KEKHUSUSAN TELEKOMUNIKASI  
DEPOK  
DESEMBER 2008**

## **HALAMAN PERNYATAAN ORISINALITAS**

**Tugas akhir ini adalah hasil karya saya sendiri, dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan dengan benar.**

**Nama : BAGUS MARDANI**  
**NPM : 0606042342**  
**Tanda Tangan :**  
**Tanggal : Depok, 10 Desember 2008**

## HALAMAN PENGESAHAN

Tugas akhir ini diajukan oleh :

Nama : BAGUS MARDANI  
NPM : 0606042342  
Program Studi : Teknik Elektro  
Tugas akhir dengan judul : ANALISIS UNJUK KERJA WIRELESS  
MESH NETWORK DENGAN ROUTING  
PROTOCOL OLSR

**Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Elektro, Fakultas Teknik, Universitas Indonesia**

## DEWAN PENGUJI

Pembimbing : Muhammad Salman ST., MIT (.....)

Penguji : Dr. Ir. AAP Ratna, M.Eng. (.....)

Penguji : Ir. Endang Sriningsih, MT. (.....)

Ditetapkan di : Depok

Tanggal : 22 Desember 2008

## **KATA PENGANTAR / UCAPAN TERIMA KASIH**

Puji syukur saya panjatkan kepada Tuhan Yang Maha Esa, karena atas berkat dan rahmat-Nya, saya dapat menyelesaikan tugas akhir ini. Penulisan tugas akhir ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Teknik Jurusan Teknik Elektro pada Fakultas Teknik Universitas Indonesia. Saya menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan tugas akhir ini, sangatlah sulit bagi saya untuk menyelesaikan tugas akhir ini. Oleh karena itu, saya mengucapkan terima kasih kepada:

- (1) Muhammad Salman ST., MIT sebagai dosen pembimbing yang telah menyediakan waktu, tenaga dan pikiran untuk mengarahkan saya dalam penyusunan skripsi ini.
- (2) F. Astha Ekadiyanto ST., Msc yang telah memberikan ide judul serta masukan dalam mengerjakan skripsi ini.
- (3) Papa dan Mama untuk semua pengorbanan dan kesabaran dalam membesarkan saya. Adik-adik, Bonar, Bule dan Adhi untuk ejekan-ejakannya yang telah memotivasi saya untuk lebih fokus dalam mengerjakan skripsi ini.
- (4) Untuk rekan-rekan kelompok, Vebby dan Budi yang telah menemani saya selama pengerjaan skripsi dan telah meluangkan banyak waktu dan tenaga dalam pengambilan data.
- (5) Sahabat-sahabat kampus yang telah memberikan dorongan moral juga material selama pengerjaan skripsi. Rijal untuk kamar kosnya dan musik-musiknya, Taqin untuk hiburannya setiap malam di acara TV, Das buat semangatnya, Yudi dan Rina buat program video conference-nya, Iqbal, Ingot, Marimbun, Citra, Lusi, Estu, Nur, Fadli, Dwi, Dimas, Ari, Dadang dan teman-teman lain yang tidak dapat disebutkan.
- (6) Teman-teman alumni PNJ 2002 yang selalu mengingatkan saya untuk terus semangat. Dee-dee, Eeya, Titi, Panink, Nando, Malih, Bung Jo, Tiara, Mbak Nalu dan teman-teman lain yang tidak dapat disebutkan.

- (7) Teman-teman kosan Citra yang sudah lebih dulu lulus, Bunk chra, Murtopo, Bojonk dan Bunk Ih.
- (8) Sahabat-sahabat di rumah Heri terima kasih sudah menjadi teman curhat yang baik, Agus , Danie Gunawan dll.
- (9) Serta semua pihak yang telah membantu dalam pembuatan dan penyusunan skripsi.

Akhir kata, saya berharap Tuhan Yang Maha Esa berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga tugas akhir ini membawa manfaat bagi pengembangan ilmu.

Depok, 10 Desember 2008

Penulis

Bagus Mardani

## HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

---

Sebagai sivitas akademis Universitas Indonesia, saya yang bertanda tangan di bawah ini:

Nama : BAGUS MARDANI  
NPM : 0606042342  
Program Studi : TELEKOMUNIKASI  
Departemen : Teknik Elektro  
Fakultas : Teknik  
Jenis karya : SKRIPSI

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneklusif** (*Non-exclusive Royalty-FreeRight*) atas karya ilmiah saya yang berjudul:

ANALISIS UNJUK KERJA WIRELESS MESH NETWORK DENGAN ROUTING PROTOCOL OLSR

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan memublikasikan tugas akhir saya tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok  
Pada Tanggal : 10 Desember 2008  
Yang menyatakan

(BAGUS MARDANI)

## ABSTRAK

Nama : BAGUS MARDANI  
Program Studi : Teknik Elektro  
Judul : ANALISIS UNJUK KERJA WIRELESS MESH NETWORK  
DENGAN ROUTING PROTOCOL OLSR

Dengan seiring perkembangan teknologi wireless dalam jaringan komputer maka kebutuhan user akan teknologi dengan kemampuan reliabilitas yang tinggi juga semakin meningkat. Skripsi ini merancang dan membangun sistem *testbed* untuk *wireless mesh network* yang menggunakan *routing protocol* Optimized Link State Routing (OLSR). Bentuk topologi jaringan yang dipakai adalah *infrastructured wireless mesh network* menggunakan beberapa buah wireless router LinkSys yang telah dimodifikasi dengan menggunakan *firmware opensource* bernama OpenWRT. Testbed yang telah dibangun digunakan untuk menguji performansi *routing protocol* OLSR dalam *self-configure* dan *self-healing* serta *performa wireless mesh network* secara keseluruhan serta performansi jaringan secara keseluruhan untuk parameter-parameter seperti *throughput*, *latency* serta *jitter* melalui beberapa skenario pengujian tertentu.

Berdasarkan data pengujian dari performansi *wireless mesh network* seperti kemampuan *self-configure* dan *self-healing* yang lebih dipengaruhi oleh parameter nilai interval *HELLO message*. Sedangkan performansi *wireless mesh network* dalam pemakaian *bandwidth*, *latency*, *throughput* dan *jitter* lebih dipengaruhi akan posisi node pada jalur data dalam sistem *multihop*

Kata kunci :  
*wireless mesh network*, OLSR, OpenWRT, *wireless adhoc*, *multihop*



## ABSTRACT

Name : BAGUS MARDANI  
Study Program: Electrical Engineering  
Title : PERFORMANCE ANALYSIS OF WIRELESS MESH  
NETWORK WITH OLSR ROUTING PROTOCOL

Along with the development of wireless technology in computer communication the needs of networks with a high reliability performance is also increased. This final project designs and builds a testbed system for wireless mesh network with OLSR routing protocol. The network topology use the infrastructured wireless mesh network using some LinkSys wireless router that had been modified with an opensource firmware called OpenWRT. The testbed will be used for testing the OLSR routing protocol performances in self-configure and self-healing also the performance of the whole network with parameters such as network throughput, latency and jitter.

Based on the experiment datas from the testbed scenarios, the performances of wireless mesh network such as self-configure and self-healing ability of the mesh network are influenced by the HELLO message interval parameter. And for the performances such as bandwidth, latency, throughput and jitter are mostly influenced by the position of the node in the data router of the multihop system.

Keywords :

*wireless mesh network, OLSR, OpenWRT, wireless adhoc, multihop*

## DAFTAR ISI

HALAMAN JUDUL .....	i
HALAMAN PERNYATAAN ORISINALITAS .....	ii
LEMBAR PENGESAHAN .....	iii
KATA PENGANTAR .....	iv
LEMBAR PERSETUJUAN PUBLIKASI KARYA ILMIAH .....	vi
ABSTRAK .....	vii
ABSTRACT .....	viii
DAFTAR ISI.....	ix
DAFTAR GAMBAR .....	xi
DAFTAR TABEL.....	xiii
DAFTAR LAMPIRAN .....	xiv
<b>1. PENDAHULUAN</b> .....	1
1.1 Latar Belakang .....	1
1.2 Perumusan Masalah .....	1
1.3 Tujuan .....	2
1.4 Batasan Masalah .....	2
1.5 Sistematika Penulisan .....	3
<b>2. LANDASAN TEORI</b> .....	4
2.1 <i>Wireless Mesh Network</i> .....	4
2.1.1 <i>Arsitektur Wireless Mesh Network</i> .....	5
2.1.1.1 <i>Client Wireless Mesh Network</i> .....	6
2.1.1.2 <i>Infrastructure Wireless Mesh Network</i> .....	6
2.1.1.3 <i>Hybrid Wireless Mesh Network</i> .....	7
2.1.2 Karakteristik dari <i>Wireless Mesh Network</i> .....	8
2.1.3 Protokol dalam <i>Wireless Mesh Network</i> .....	9
2.1.3.1 <i>Physical Layer Protocol</i> .....	9
2.1.3.2 <i>Medium Acces Control (MAC) Layer Protocol</i> .....	10
2.1.3.3 <i>Network Layer Protocol</i> .....	11
2.1.3.4 <i>Transport Layer Protocol</i> .....	12
2.1.3.5 <i>Application Layer Protocol</i> .....	12
2.2 OLSR ( <i>Optimized Link State Routing</i> ) .....	14
2.2.1 Tahapan Kerja OLSR .....	14
2.2.1.1 <i>Link Sensing</i> .....	14
2.2.1.2 <i>Neighbour Sensing</i> .....	15
2.2.1.3 <i>MPR Selection</i> .....	16
2.2.1.4 <i>Topology Control (TC) Message Diffusion</i> .....	18
2.2.1.5 <i>Routing Calculation</i> .....	18
2.2.2 Format Paket OLSR .....	19
2.2.2.1 <i>Hello Message</i> .....	22
2.2.2.2 <i>TC Message</i> .....	23
2.3 OpenWRT .....	24
<b>3. PERANCANGAN DAN PEMBUATAN JARINGAN</b> .....	26
3.1 Perancangan <i>Testbed</i> .....	26
3.2 Pembangunan <i>Testbed</i> .....	26
3.2.1 Penentuan Topologi Jaringan .....	26
3.3 Spesifikasi Sistem .....	27

3.3.1 Spesifikasi <i>Hardware</i> .....	27
3.3.1.1 Spesifikasi <i>Hardware</i> Untuk <i>Wireless Mesh Router</i> .....	27
3.3.1.2 Spesifikasi <i>Hardware</i> Untuk <i>User/Client</i> .....	29
3.3.2 Spesifikasi <i>Software</i> .....	30
3.3.2.1 Spesifikasi <i>Software</i> Untuk <i>Wireless Mesh Router</i> .....	30
3.3.2.2 Spesifikasi <i>Software</i> Untuk <i>Client/User</i> .....	30
3.3.2.3 Instalasi OpenWRT .....	31
3.3.2.4 Instalasi <i>Package</i> Tambahan Pada OpenWRT ...	34
3.4 Penentuan Lokasi <i>Testbed</i> .....	34
3.5 Konfigurasi Jaringan .....	36
3.5.1 Pengalamatan .....	36
3.5.2 Konfigurasi <i>Routing</i> .....	37
3.5.3 <i>Data Packet Forwarding</i> .....	39
<b>4. PENGUJIAN DAN ANALISA</b> .....	42
4.1 Hasil Rancang Bangun .....	42
4.2 Pengujian Jaringan .....	43
4.2.1 Pengujian <i>Self-Configure</i> .....	43
4.2.2 Pengujian <i>Self-Healing</i> .....	45
4.2.3 Pengujian Penggunaan <i>Bandwidth</i> dan <i>Latency</i> .....	47
4.2.4 Pengujian Sistem <i>Multihop</i> .....	48
4.3 Analisa Sistem .....	50
4.3.1 Analisa Performansi <i>Self Configure</i> .....	50
4.3.2 Analisa Performansi <i>Self Healing</i> .....	52
4.3.3 Analisa Performansi Pemakaian <i>Bandwidth</i> .....	55
4.3.4 Analisa Performansi <i>Latency</i> .....	59
4.3.5 Analisa Performansi <i>End-to-end Throughput</i> .....	63
4.3.6 Analisa Performansi <i>End-to-end Jitter</i> .....	66
<b>5. KESIMPULAN DAN SARAN</b> .....	68
<b>DAFTAR REFERENSI</b> .....	69
<b>DAFTAR PUSTAKA</b> .....	71
<b>LAMPIRAN</b> .....	72

## DAFTAR GAMBAR

Gambar 2.1	Arsitektur <i>Client Wireless Mesh Networking</i> .....	6
Gambar 2.2	<i>Infrastructure Wireless Mesh Networking</i> .....	7
Gambar 2.3	<i>Hybrid Wireless Mesh Networking</i> .....	8
Gambar 2.4	Perbandingan antara FDM dan OFDM .....	10
Gambar 2.5	CSMA/CA dengan RTS/CTS .....	11
Gambar 2.6	Klasifikasi <i>Ad-hoc Routing Protocol</i> .....	12
Gambar 2.7	Perbandingan Sistem <i>Broadcast</i> .....	17
Gambar 2.8	Format Paket OLSR .....	20
Gambar 2.9	Format Paket HELLO .....	22
Gambar 2.10	Format Paket <i>TC Message</i> .....	24
Gambar 3.1	Bentuk Topologi <i>Testbed Wireless Mesh Network</i> .....	27
Gambar 3.2.a	Tampak Depan <i>Wireless Router LinkSYS WRT54GL</i>	28
Gambar 3.2.b	Tampak Belakang <i>Wireless Router LinkSYS WRT54GL</i>	28
Gambar 3.3	Jendela <i>Firmware Upgrade</i> .....	32
Gambar 3.4	Halaman Depan <i>Web Interface OpenWRT</i> .....	33
Gambar 3.5	Tampilan <i>Command Line Interface OpenWRT</i> .....	33
Gambar 3.6	Denah Lokasi Penempatan <i>Wireless Router</i> .....	35
Gambar 3.7	Lokasi-lokasi Penempata <i>Wireless Router</i> .....	35
Gambar 3.8	Jendela <i>Wireless Network Connection</i> .....	41
Gambar 3.9	<i>Web Interface olsrd http-mod-info</i> .....	41
Gambar 4.1	Topologi <i>Testbed</i> Jaringan <i>Wireless Mesh</i> .....	42
Gambar 4.2	Skenario Pengujian <i>Self-Configure</i> .....	44
Gambar 4.3	Skenario Pengujian <i>Self-healing</i> .....	46
Gambar 4.4	Skenario Pengujian Penggunaan <i>Bandwidth</i> .....	47
Gambar 4.5	Skenario Pengujian Performansi Sistem <i>Multihop</i> .....	49
Gambar 4.6	Grafik perbandingan Performansi <i>Self-Configure</i> .....	50
Gambar 4.7	Grafik Perbandingan Performansi <i>Self-Healing</i> .....	53
Gambar 4.8	Grafik Perbandingan Pemakaian <i>Bandwidth</i> Untuk Kondisi Tingkat <i>Interferensi Rendah</i> .....	56

Gambar 4.9	Grafik Perbandingan Pemakaian Bandwidth Untuk Kondisi Tingkat Interferensi Tinggi .....	58
Gambar 4.10	Grafik Perbandingan Pemakaian Bandwidth Berdasarkan Tingkat Interferensi .....	59
Gambar 4.11	Grafik Perbandingan Latency Untuk Kondisi Tingkat Interferensi Rendah.....	60
Gambar 4.12	Grafik Perbandingan Latency Untuk Kondisi Tingkat Interferensi Tinggi .....	61
Gambar 4.13	Grafik Perbandingan Latency Berdasarkan Tingkat Interferensi .....	62
Gambar 4.14	Grafik Perbandingan End-to-End Throughput Sistem Multihop .....	64
Gambar 4.15	Grafik Perbandingan End-to-End Jitter Sistem Multihop .....	65

## DAFTAR TABEL

Tabel 2.1	Tabel <i>Routing</i> OLSR .....	19
Tabel 2.2	Penjelasan <i>Field-field</i> Pada Format Paket OLSR .....	21
Tabel 2.3	Penjelasan <i>Field-field</i> Pada Format Paket HELLO .....	23
Tabel 2.4	Penjelasan <i>Field-field</i> Pada Format Paket TC .....	24
Tabel 3.1	Tabel Pasangan Antara Versi <i>Image</i> OpenWRT Dengan Jenis <i>Router</i> .....	31
Tabel 4.1.a	Tabel Persentase Perbedaan Nilai <i>Self-Configure</i> Untuk Parameter Interval <i>Hello Message</i> .....	71
Tabel 4.1.b	Tabel Persentase Perbedaan Nilai <i>Self-Configure</i> Untuk Parameter Interval <i>TC Message</i> .....	71
Tabel 4.2.a	Tabel Persentase Perbedaan Nilai <i>Self-Healing</i> Untuk Parameter Interval <i>Hello Message</i> .....	74
Tabel 4.2.b	Tabel Persentase Perbedaan Nilai <i>Self-Healing</i> Untuk Parameter Interval <i>TC Message</i> .....	74

## DAFTAR LAMPIRAN

Lampiran 1.a	Tabel Performansi Self Configure Tanpa Aplikasi .....	73
Lampiran 1.b	Tabel Performansi Self Configure Dengan Aplikasi .....	75
Lampiran 2.a	Tabel Performansi Self Healing Tanpa Aplikasi .....	77
Lampiran 2.b	Tabel Performansi Self Healing Dengan Aplikasi .....	79
Lampiran 3.a	Pembacaan PRTG Untuk Kondisi Interferensi Rendah ...	81
Lampiran 3.b	Tabel Pembacaan PRTG Untuk Kondisi Interferensi Rendah .....	83
Lampiran 4.a	Pembacaan Latency Untuk Kondisi Interferensi Rendah ..	85
Lampiran 4.b	Tabel Pembacaan Latency Untuk Kondisi Interferensi Rendah .....	87
Lampiran 5.a	Pembacaan PRTG Untuk Interferensi Tinggi.....	89
Lampiran 5.b	Tabel Pembacaan PRTG Untuk Interferensi Tinggi .....	91
Lampiran 6.a	Pembacaan Latency Untuk Kondisi Interferensi Tinggi ...	93
Lampiran 6.b	Tabel Pembacaan Latency Untuk Kondisi Interferensi Tinggi .....	95
Lampiran 7	Performansi End-to-End Throughput .....	97
Lampiran 8	Performansi End-to-End Jitter .....	99

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Keberadaan jaringan *wireless* dengan berbagai jenis teknologi serta aplikasinya semakin berpengaruh terhadap kehidupan masyarakat. Oleh karena itu kebutuhan masyarakat terhadap jaringan yang dapat memberikan pelayanan dengan kualitas pengiriman yang baik serta dapat digunakan kapan saja dan dimana saja pun semakin menuntut perkembangan teknologi *wireless* yang dapat memenuhi kebutuhan tersebut.

Teknologi *wireless mesh networking* dipercaya dapat menjawab kebutuhan tersebut dengan sifatnya yang memiliki jangkauan luas karena menggunakan sistem *multi-hop*, kemampuan *self-configured* dan *self-healing* yang dapat memberikan reliabilitas tinggi, serta kemampuannya untuk berinterkoneksi dengan berbagai jenis teknologi jaringan membuatnya sangat fleksibel.

Selain itu *wireless mesh network* dapat dibangun dengan biaya yang cukup murah karena tidak memerlukan perangkat keras atau perangkat lunak khusus. *wireless mesh network* dapat disusun dengan menggunakan perangkat-perangkat yang telah biasa digunakan dalam jaringan *Wi-fi* sedangkan untuk *firmware router* dapat menggunakan perangkat lunak *opensource* untuk memudahkan dalam memodifikasi perangkat *router* agar dapat bekerja sesuai dengan spesifikasi *wireless mesh network*.

Dengan menggunakan routing protocol OLSR (*Optimized Link State Routing*), yang selalu memperbaharui informasi *routing* secara simultan pada setiap *node* untuk mendapatkan sambungan yang lebih cepat.

### 1.2 Perumusan Masalah

Berdasarkan latar belakang dapat dirumuskan masalah yaitu bagaimana merancang *wireless mesh network* yang cukup ideal untuk dapat digunakan sebagai *testbed* untuk mendapatkan data tentang kinerja *wireless mesh network*. Rumusan masalah dapat diperinci menjadi tiga pertanyaan sebagai berikut :



1. Bagaimana merancang dan membuat *wireless mesh network* dengan menggunakan perangkat-perangkat yang telah biasa digunakan dalam jaringan Wi-Fi ?
2. Bagaimana menentukan skenario pengujian untuk mendapatkan data tentang kinerja dari *wireless mesh network* yang sudah dibangun?

### 1.3 Tujuan

Tujuan dari tugas akhir ini adalah untuk merancang bangun sebuah *wireless mesh network* dengan menggunakan perangkat-perangkat yang biasa digunakan dalam jaringan Wi-fi serta pemakaian perangkat lunak *opensource* sebagai sistem operasinya.. Kemudian jaringan yang sudah dibangun akan diuji dengan menggunakan skenario pengujian tertentu untuk mendapatkan analisa kinerja dari *wireless mesh network* tersebut

### 1.4 Batasan Masalah

Dalam pembangunan jaringan ini, terdapat beberapa pembatasan masalah, antara lain:

1. *Wireless mesh network* yang dibangun dengan bentuk arsitektur *infrastructured wireless mesh network* dengan menggunakan *routing protocol OLSR*.
2. Pembentukan topologi jaringan serta penempatan posisi *router* disesuaikan dengan kebutuhan skenario pengujian.
3. Pemakaian fitur-fitur jaringan *wireless* dibatasi hanya untuk keperluan pengujian seperti SSID, pengaturan *power* transmisi.
4. Data yang akan dianalisa terbatas pada kinerja waktu *self healing*, *self configure* dari jaringan serta penggunaan *bandwidth* pada sistem *multihop* dan juga *latency* dari transmisi serta *end-to-end throughput* dan *jitter*.

## **1.5 SISTEMATIKA PENULISAN**

Pembahasan yang dilakukan pada tugas akhir ini dibagi dalam beberapa tahap, antara lain :

### **BAB 1 PENDAHULUAN**

Bagian ini terdiri dari latar belakang masalah, tujuan, batasan masalah, dan sistematika penulisan.

### **BAB 2 LANDASAN TEORI**

Bagian ini membahas tentang teori *wireless mesh network*, protokol OLSR, dan *firmware* OPENWRT.

### **BAB 3 RANCANG BANGUN WIRELESS MESH NETWORK**

Bagian ini berisi langkah-langkah perancangan dan pembangunan dari *wireless mesh network*.

### **BAB 4 PENGUJIAN DAN ANALISA**

Bagian ini membahas hasil pengujian dari *wireless mesh network* serta analisa dari data hasil pengujian tersebut.

### **BAB 5 KESIMPULAN**

Bagian ini berisi kesimpulan dari keseluruhan tugas akhir.

## BAB II

### LANDASAN TEORI

#### **2.1 *Wireless Mesh Network***

*Wireless mesh network* merupakan suatu bentuk jaringan komunikasi *wireless* yang terbentuk dari susunan *node* radio dimana setidaknya terdapat dua atau lebih jalur komunikasi pada setiap *node*. *Node* pada sebuah *wireless mesh network* dapat berupa sebuah *mesh router* ataupun *mesh client*. Setiap *node* tidak hanya bertindak sebagai sebuah *host* tetapi juga berfungsi sebagai *router* untuk meneruskan paket-paket pengiriman informasi bagi sebuah *node* lain yang mungkin tidak dapat menjangkau tempat yang ingin ditujunya [6] [8].

Karakteristik utama dari *wireless mesh network* adalah kemampuannya dalam mengkonfigurasi dan mengorganisasi dirinya sendiri (*self-configure/self-organize*), atau dengan kata lain mampu membuat dan menjaga konektivitasnya apabila terjadi kerusakan pada salah satu *node*. Kemampuan ini selain membantu para pengguna untuk dapat selalu *on-line* kapan saja dan dimana saja, juga akan membawa keuntungan lain seperti biaya pembuatan yang rendah, kemudahan dalam perawatan jaringan, tingkat *robustness* serta reliabilitas tinggi.

*Node-node* konvensional seperti desktop PC, laptop, PDA dan sebagainya yang telah dilengkapi dengan *wireless network interface card* (NIC) dapat tersambung langsung dengan *wireless mesh routers*. Sedangkan pengguna yang tidak mempunyai *wireless* NIC, tetap dapat terhubung dengan *mesh router* dengan menggunakan bentuk jaringan lain seperti Ethernet. Selain itu dengan menggunakan fungsi *mesh router* sebagai *gateway* atau *bridge*, maka suatu *wireless mesh network* dapat berintegrasi dengan jaringan *wireless* lainnya seperti jaringan *seluler*, Wi-fi, Wimax dan lain sebagainya [5].

*Wireless mesh network* dikembangkan untuk mengantisipasi keterbatasan dan juga meningkatkan performansi dari *wireless adhoc network*, *wireless local area network* (WLAN), dan *wireless metropolitan area network* (WMAN). Dengan berbagai kelebihanannya, *wireless mesh network* dapat digunakan untuk

menyediakan layanan *wireless* untuk berbagai keperluan dan aplikasi baik untuk kepentingan pribadi, area lokal, kampus ataupun area metropolitan.

### 2.1.1 Arsitektur *Wireless Mesh Network*

Seperti telah disebutkan sebelumnya bahwa *wireless mesh network* mempunyai dua jenis *node* yaitu : *mesh router* dan *mesh client*. Untuk lebih meningkatkan fleksibilitas dari *mesh network*, maka suatu *mesh router* juga dapat dilengkapi dengan *multiple wireless interface*. Apabila dibandingkan dengan *wireless router* konvensional, maka *mesh router* dapat memiliki jangkauan area yang sama namun dengan daya transmisi yang jauh lebih rendah melalui komunikasi *multi-hop*.

Walaupun dengan perbedaan tersebut, *wireless router* untuk *mesh* maupun konvensional biasanya tetap dibuat berdasarkan *platform hardware* yang sama. *Mesh router* dapat dibangun menggunakan *dedicated computer systems (embedded systems)*, ataupun juga *general purpose systems (desktop PC dan laptop)*.

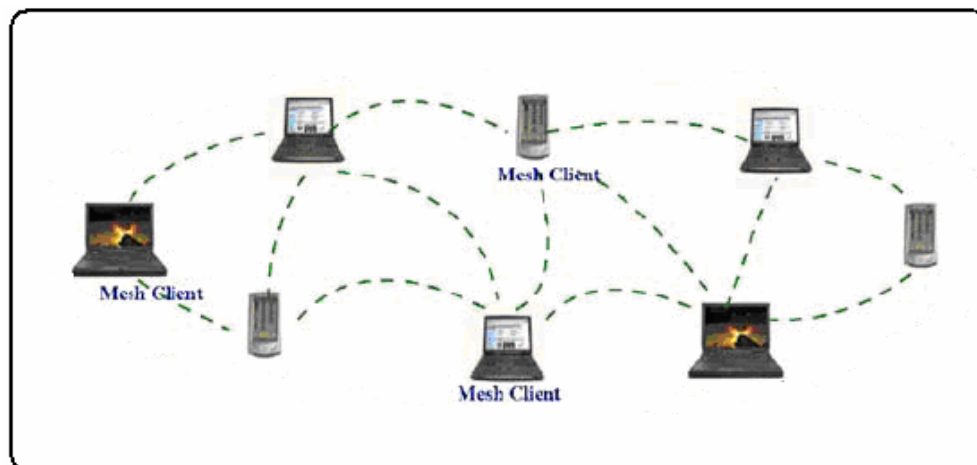
Untuk *mesh client* juga mempunyai fungsi tertentu pada suatu *mesh networking*, dimana suatu *mesh client* juga mempunyai fungsi *router*. Namun *node* ini tidak mempunyai fungsi *gateway* ataupun *bridge* seperti yang ada pada *mesh router*. Oleh karena itu biasanya *mesh client* cukup dilengkapi dengan satu *wireless interface* sehingga *platform hardware* dan *software* yang dipakaipun menjadi lebih sederhana dibandingkan dengan *mesh router*. *Mesh client* dapat terdiri dari *desktop PC*, *laptop*, *PDA*, *RFIP reader* dan lain sebagainya [1].

Pemakaian kedua tipe *node* inilah yang mengklasifikan arsitektur *wireless mesh network* yang dapat dibagi menjadi tiga bagian [5], yaitu :

1. *Client wireless mesh network*,
2. *Infrastructure wireless mesh network*,
3. *Hybrid wireless mesh network*.

### 2.1.1.1 *Client Wireless Mesh Network*

*Client wireless mesh network* menyediakan jaringan *peer-to-peer* antara *node mesh client*. Pada tipe arsitektur ini, jaringan terbentuk dari sekumpulan *node mesh client* yang dapat melakukan fungsi *routing* dan konfigurasi serta menyediakan aplikasi *end-user* pada pengguna jaringan seperti yang terlihat pada Gambar 2.1. Tipe arsitektur ini tidak memerlukan *mesh router*, oleh karena itu tingkat mobilitasnya menjadi lebih tinggi bila dibandingkan dengan tipe arsitektur *infrastructure wireless mesh network*. Pada *client wireless mesh network*, paket yang dikirimkan ke suatu *node* tujuan akan melalui serangkaian lompatan (*hops*) melalui beberapa *node* untuk mencapai tujuan. Oleh karena itu, apabila dibandingkan dengan tipe *infrastructure wireless mesh network*, maka *node mesh client* pada tipe ini memerlukan kebutuhan akan aplikasi *end-user* yang lebih tinggi karena harus memiliki kemampuan *routing* serta konfigurasi sendiri.

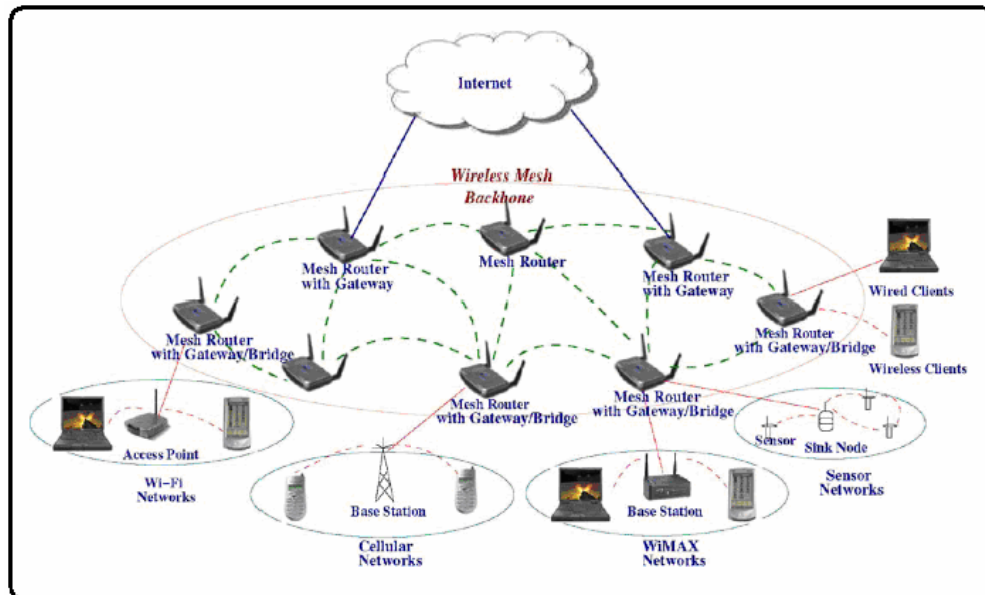


Gambar 2.1 . Arsitektur *Client Wireless Mesh Networking* [5].

### 2.1.1.2 *Infrastructure Wireless Mesh Networking*

Pada tipe arsitektur ini terdapat beberapa *mesh router* yang membentuk sebuah infrastruktur bagi *client-client* yang terhubung dengannya. Seperti terlihat pada Gambar 2.2 dimana *wireless mesh network* dapat terkoneksi dengan *mesh client* ataupun dengan berbagai jenis teknologi *wireless* lainnya yang dalam hal ini bertindak sebagai *client* dari jaringan infrastruktur yang dibentuk oleh *mesh router*. Dengan kemampuan sebagai *gateway*, *mesh router* juga dapat

dihubungkan ke internet. *Mesh router* harus mempunyai kemampuan untuk mengkonfigurasi serta memperbaiki hubungan antara *router* secara mandiri, sehingga hubungan tidak sampai terputus.

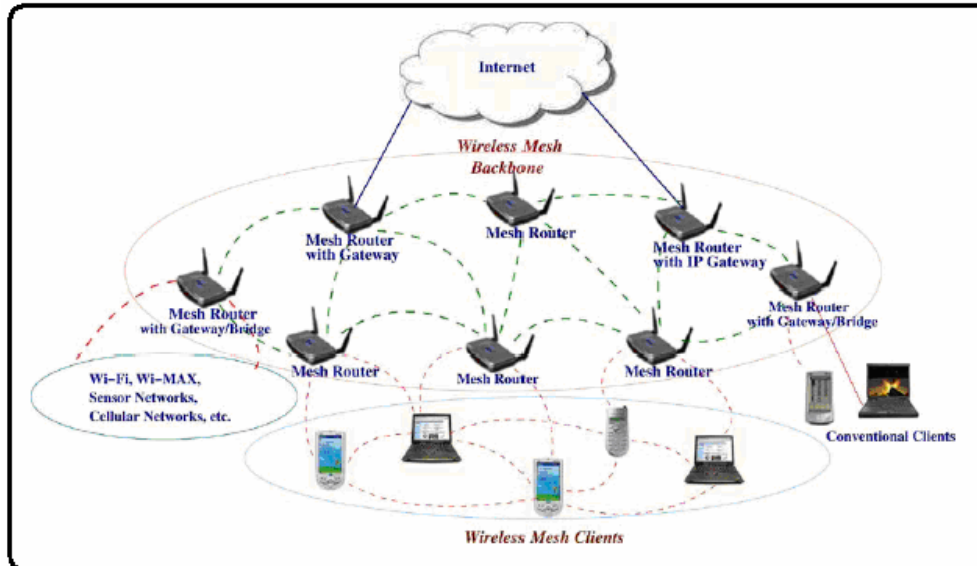


Gambar 2.2 *Infrastructure Wireless Mesh Networking* [5].

Untuk *client* konvensional yang memiliki teknologi yang sama dengan *wireless mesh network*, mereka dapat terhubung langsung dengan *mesh router*. Sedangkan untuk *client* yang menggunakan teknologi *wireless* selain *wireless mesh network* maka mereka dapat terhubung dengan *mesh router* melalui sebuah *base station* yang terhubung dengan *mesh router* melalui kabel.

### 2.1.1.3 *Hybrid Wireless Mesh Networking*

Tipe ini merupakan bentuk gabungan dari dua tipe arsitektur lainnya yaitu *Client* dan *Infrasructure wireless mesh network* seperti yang terlihat pada Gambar 2.3. *Mesh client* dapat terhubung pada jaringan melalui *mesh router* sekaligus tetap berhubungan langsung dengan *mesh client* lainnya. Selain itu jaringan *infrastructure* dari *mesh router* juga dapat terhubung dengan jaringan teknologi *wireless* lainnya.



Gambar 2.3 Hybrid Wireless Mesh Network [5].

### 2.1.2 Karakteristik dari Wireless Mesh Network

*Wireless mesh network* memiliki beberapa karakteristik umum yang sangat mempengaruhi kinerjanya, seperti berikut.

1. *Multi-hop wireless network.*

Karakteristik ini berguna untuk meningkatkan area jangkauan dari jaringan tanpa harus mengorbankan kapasitas kanal. Selain itu berguna pula untuk menyediakan bentuk layanan *Non Line-of-Sight* (NLOS).

2. Kemampuan *self-forming*, *self-healing*, *self-organizing* serta mendukung *ad-hoc networking*.

Karakteristik ini menambah performansi dari *wireless mesh network* karena membawa sifat fleksibel dalam jaringan, pembuatan dan konfigurasi yang mudah, serta *fault tolerance*.

3. Tingkat mobilitas tergantung dari jenis *node*.

*Mesh router* biasanya memiliki tingkat mobilitas yang lebih rendah dibanding dengan sebuah *mesh client* yang dapat bersifat *fixed* maupun *mobile*.

4. Dapat mengakses ke berbagai jenis teknologi jaringan lainnya.

Dalam *wireless mesh network* sebuah jaringan infrastruktur dari *mesh router* dapat terhubung baik ke *mesh client*, internet maupun dengan berbagai jenis teknologi jaringan lainnya.

5. Dependensi terhadap pemakaian daya tergantung dari jenis *node*.

*Mesh router* tidak memiliki suatu batasan tertentu dalam pemakaian daya karena biasanya bersifat fixed dan terhubung langsung dengan sumber daya. Sedangkan bagi *mesh client* kemungkinan memerlukan suatu protokol tertentu untuk mengatur pemakaian daya karena seringkali *mesh client* bersifat *mobile*.

### **2.1.3 Protokol dalam *Wireless Mesh Network***

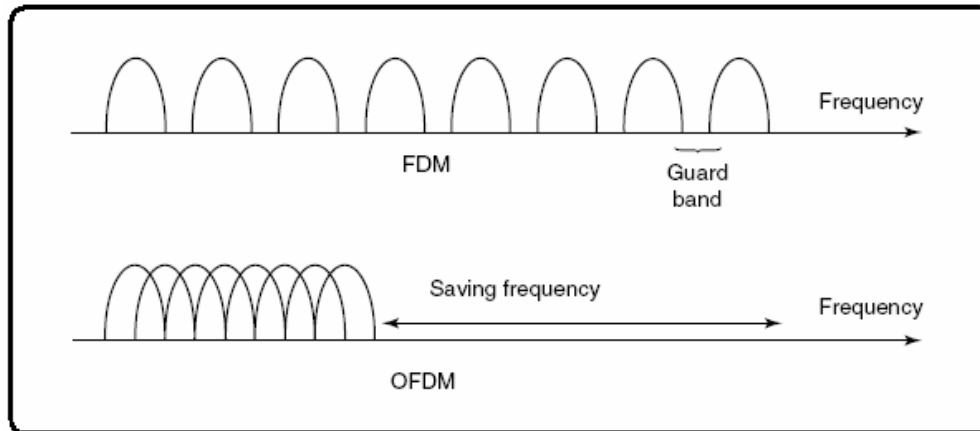
Protokol-protokol yang digunakan dalam suatu sistem *wireless mesh network* dapat dibagi menjadi beberapa bagian menurut lapisan (*layer*), mulai dari *physical layer* hingga *application layer*.

#### **2.1.3.1 *Physical Layer Protocol***

Jenis *protocol* yang umum digunakan untuk *physical layer* dalam sebuah system *wireless mesh network* saat ini adalah *Orthogonal Frequency Division Multiplexing* (OFDM), yang memungkinkan pengiriman data berkecepatan tinggi dalam lingkungan yang *mobile*. OFDM merupakan perkembangan dari sistem *Frequency Division Multiplexing* (FDM) yang dapat mengirimkan berbagai sinyal secara simultan dengan cara membagi-bagi mereka ke dalam beberapa band frekuensi yang berbeda (*subcarrier*). Dalam FDM terdapat *guard band* yang berfungsi untuk mengurangi interferensi antar frekuensi yang berbeda, hal ini menyebabkan adanya *bandwidth* yang terbuang percuma.

Sedangkan OFDM menerapkan sistem yang lebih efisien dimana OFDM tidak lagi menggunakan *guard band* untuk mengurangi interferensi melainkan dengan memodulasi sinyal secara *orthogonal* seperti terlihat pada gambar 2.4. Dengan cara ini OFDM dapat menyediakan lebih banyak ruang pada *bandwidth* untuk dipakai [9].





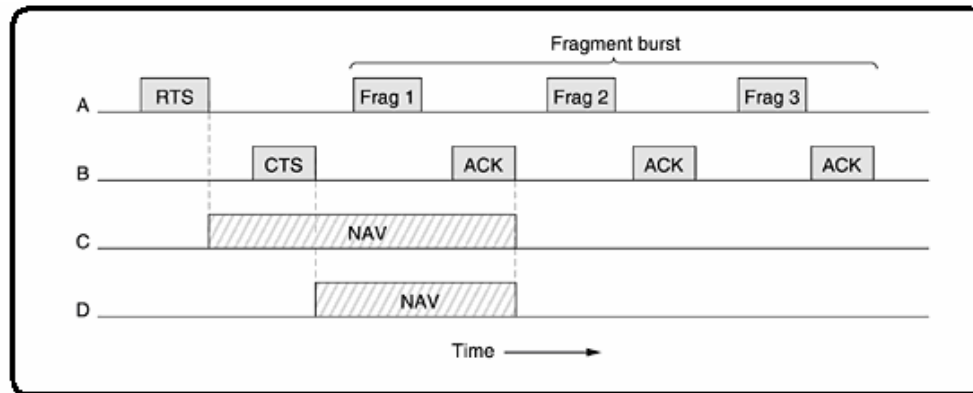
Gambar 2.4 Perbandingan Antara FDM dan OFDM [9].

OFDM menggunakan *Fast Fourier Transform* (FFT) yang dan *Inverse FFT* untuk mengubah serial data ke dalam bentuk *multiple channel*. OFDM mempunyai 256 *subchannel* (*carrier*), dan sinyal asli akan dibagi-bagi menjadi 256 *subcarrier* dan ditransmisikan secara paralel [9].

Namun dalam perkembangannya, lebih banyak lagi protokol yang dapat digunakan pada *physical layer* dari sebuah sistem *wireless mesh network* seperti *Code Division Multiple Access* (CDMA), *Ultra Wideband Access* (UWB) ataupun *Multiple Input Multiple Output* (MIMO) yang dapat digunakan sesuai dengan kebutuhan [5].

### 2.1.3.2 Medium Access Control (MAC) Layer Protocol

Protokol yang dikembangkan untuk digunakan pada *MAC Layer* untuk sebuah sistem *wireless mesh networking* adalah *protocol* MAC berbasis IEEE 802.11 yaitu *Carrier Sense Multiple Access with Collision Avoidance* (CSMA/CA) dengan paket control RTS/CTS (*Ready to Send/Clear to Send*). *Frame* yang akan dikirimkan membawa informasi waktu perkiraan untuk *frame* tersebut agar dapat sampai pada tujuan. Waktu perkiraan inilah yang akan dipakai oleh *node* yang lain dalam menentukan waktu minimum pemakaian jalur atau *network location vector* (NAV) [3] [5]. Gambar 2.5 menunjukkan gambaran dari sistem kerja dari CSMA dengan menggunakan RTS/CTS.



Gambar 2.5 CSMA/CA Dengan RTS/CTS [10].

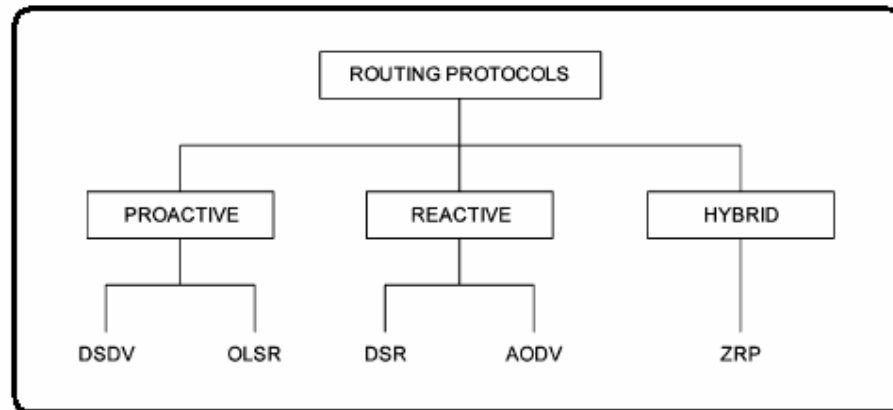
Dengan menggunakan paket kontrol RTS/CTS, saat pengirim akan mengirimkan *frame* maka ia akan mengirimkan paket RTS yang berisi informasi waktu yang dibutuhkan untuk mengirimkan data dan sinyal *Acknowledgement* (ACK). Penerima yang menerima paket RTS tersebut akan mengirimkan paket CTS sebagai balasan yang menandakan bahwa pengirim dapat mulai mengirimkan data. *Node* lain yang mengetahui adanya sinyal RTS/CTS ini dapat menunggu hingga waktu pengiriman selesai untuk menghindari adanya *collision* [10].

### 2.1.3.3 Network Layer Protocol

Protokol yang dipakai dalam *network layer* pada suatu *wireless mesh network* banyak mengadaptasi *routing protocol* yang digunakan dalam *wireless adhoc network* yang seperti terlihat pada Gambar 2.6 dibagi menjadi tipe *proactive*, *reactive* dan *hybrid* [2]. Jenis *proactive routing protocol* mengirimkan informasi seperti keterangan *node* tetangga, rute dan lain-lain secara *broadcast* dalam periode tertentu. Hal ini memungkinkan waktu *set-up* yang cepat, namun meningkatkan penggunaan *overhead*. Contoh dari tipe *protocol* ini adalah *Destination Sequenced Distance Vector* (DSDV) dan *Optimized Link State Routing* (OLSR).

Pada tipe *reactive routing protocol* rute dibangun berdasarkan apabila adanya permintaan, sehingga mengurangi pemakaian *overhead* pemilihan rute namun menimbulkan *delay* yang cukup besar pada saat pengiriman *frame* pertama. Contoh dari *routing protocol* ini adalah *Dynamic Source Routing* (DSR)

dan *Ad-hoc On-demand Distance Vector* (AODV). Sedangkan untuk tipe *hybrid routing protocol* merupakan gabungan dari kedua tipe diatas. Contoh dari *routing protocol* ini adalah *Zone Routing Protocol* (ZRP).



Gambar 2.6 Klasifikasi *Ad-hoc Routing Protocol* [2].

#### 2.1.3.4 *Transport Layer Protocol*

Protokol TCP yang selama ini diadaptasi untuk *wireless mesh network* walaupun tetap digunakan namun tidak memberikan hasil yang maksimal. Kekurangan dari TCP dalam *wireless mesh network* karena pada TCP yang standard tidak membedakan *loss* yang diakibatkan oleh *congestion* maupun *non-congestion*, sehingga bila terjadi *non-congestion loss* maka *throughput* jaringanpun menurun drastis. Selain itu apabila terjadi perbaikan jalur setelah mengalami kerusakan, protokol TCP konvensional tidak dapat diperbaiki dengan cepat. Untuk itu menutupi kekurangan-kekurangan ini, sedang dikembangkan *protocol-protocol* berbasis TCP yang dapat dipakai untuk *wireless mesh network* [5].

#### 2.1.3.5 *Application Layer Protocol*

Hampir semua dari *protocol-protocol* yang bekerja pada *layer 7* pada OSI *layer* dapat digunakan dengan baik pada *wireless mesh network*. Terdapat tiga fungsi utama aplikasi yang dapat digunakan dalam suatu *wireless mesh network* yaitu akses internet, penyimpanan informasi terdistribusi, pertukaran informasi antara jaringan teknologi *wireless* lainnya [5].

## 2.2 OLSR (*Optimized Link State Routing*)

OLSR (*Optimized Link State Protocol*) merupakan salah satu jenis dari *proactive routing protocol* yang biasa digunakan dalam jaringan *ad hoc*. Protokol ini melakukan pertukaran pesan secara periodik dalam rangka menjaga informasi topologi jaringan yang ada pada setiap node [6] [11].

Protokol OLSR mewarisi sifat kestabilan dari *link state algorithm*. Berdasarkan sifat proaktifnya, protokol ini dapat menyediakan rute dengan segera apabila dibutuhkan. Dalam sebuah *link state protocol* yang murni, setiap node tetangga akan dideklarasikan dan dibanjiri dengan paket informasi yang akan memenuhi seluruh jaringan. OLSR merupakan sebuah optimasi dari *link state protocol* yang biasa digunakan dalam *mobile adhoc network* (MANET).

Langkah pertama dari optimasi tersebut adalah mengurangi ukuran dari paket kontrol, daripada membanjiri paket kontrol tersebut pada setiap jalur, OLSR lebih memilih sejumlah jalur dengan *node* tetangga yang disebut dengan *multipoint relay selector*. Langkah kedua, OLSR meminimalisir pembanjiran paket kontrol pada jaringan dengan menggunakan MPR untuk menghantarkan paket-paket tersebut. Teknik ini akan mengurangi secara signifikan jumlah dari transmisi ulang yang akan membanjiri jaringan dengan prosedur *broadcast*.

Protokol OLSR dirancang untuk dapat bekerja pada kondisi yang terdistribusi atau selalu bergerak serta tidak memerlukan adanya pengaturan secara terpusat. Selain itu OLSR juga tidak memerlukan transmisi yang bagus dalam mengirimkan paket-paket kontrolnya. Setiap *node* mengirimkan paket kontrolnya masing-masing secara periodik sehingga dapat mentoleransi terjadinya *loss* dari beberapa paket pada saat-saat tertentu akibat dari tubrukan data ataupun akibat gangguan transmisi lainnya. Setiap paket kontrol yang dikirimkan akan diberikan *sequence number* (nomor urut) yang dapat menandakan tingkat baru tidak paket tersebut.

OLSR menggunakan *multihop routing* dimana setiap *node* menggunakan informasi *routing* terbaru yang ada pada *node* tersebut dalam mengantarkan sebuah paket informasi. Sehingga, walaupun sebuah *node* bergerak ataupun berpindah tempat maka pesan yang dikirimkan padanya akan tetap dapat diterima [12].

## 2.2.1 Tahapan Kerja OLSR

Secara umum langkah-langka kerja dalam OLSR dapat diurutkan sebagai berikut [12] :

1. *Link Sensing* (Pendeteksian hubungan).

Link Sensing dilakukan dengan mengirimkan pesan HELLO secara periodik dan berkesinambungan. Hasil dari *link sensing* adalah *local link set* yang menyimpan informasi hubungan antara *interface* yang ada pada *node* tersebut dengan *node-node* tetangga.

2. *Neighbour detection* (pendeteksian node tetangga).

Node pengirim pesan HELLO akan menerima informasi alamat-alamat dari *node-node* tetangganya beserta *link status*-nya.

3. *MPR selection* (pemilihan MPR).

Melalui pesan HELLO *node* utama akan menentukan sejumlah *node* tetangga untuk dipilih sebagai *multipoint relay* (MPR) yang bertugas meneruskan paket-paket kontrol ke dalam jaringan.

4. Pengiriman TC (*Topology Control*) *Messages*.

TC *Messages* dikirimkan untuk memberikan informasi *routing* kepada setiap *node* yang ada pada jaringan yang akan digunakan untuk penentuan jalur.

5. *Route calculation* (penghitungan jalur).

Berdasarkan informasi rute yang didapat dari paket-paket kontrol seperti HELLO dan TC maka setiap *node* akan memiliki *routing table* yang berisi informasi rute yang dapat dilalui untuk dipakai mengirimkan data ke *node-node* lainnya yang ada pada jaringan.

### 2.2.1.1 *Link Sensing*

Setiap *node* pada jaringan dengan protokol OLSR harus mengetahui jenis hubungan yang dimiliki dengan *node-node* tetangganya. Jenis-jenis hubungan inilah yang kemudian digunakan untuk menentukan kedudukan *node-node* tetangga terhadap *node* tersebut. Proses pendeteksian hubungan dengan *node-node* tetangga tersebut dinamakan *Link Sensing* [14].

*Link sensing* (pendeteksian hubungan) dikerjakan dengan pengiriman pesan HELLO secara periodik melalui *wireless interface* yang digunakan dalam *node* tersebut. Bila *node* tersebut menggunakan lebih dari satu *interface*, maka setiap *interface* akan mengirimkan HELLO *message* yang berbeda-beda.

Hasil pemrosesan data yang didapat dari HELLO *message* yang diterima oleh setiap *node* akan menghasilkan “*local link set*” yang berisi informasi tentang hubungan antara *local interface* (*interface* pada *node* tersebut) dengan *remote interface* (*interface* pada *node* tetangga). Informasi yang terdapat pada *link set* tersebut antara lain adalah :

- L\_local\_iface\_addr,
- L\_neighbor\_iface,
- L\_SYM\_time,
- L\_ASYM\_time,
- L\_time.

L\_local\_iface\_addr merupakan alamat *interface* pada *node* tersebut, L\_neighbor\_iface adalah alamat *interface* pada *node* tetangga, L\_SYM\_time adalah waktu yang dibutuhkan hingga sebuah hubungan dianggap *simetris*, L\_ASYM\_time merupakan waktu yang dibutuhkan hingga *interface* tetangga dianggap terdeteksi, dan L\_time merupakan waktu yang dibutuhkan hingga informasi ini dianggap kadaluarsa.

L\_SYM\_time digunakan untuk menentukan Link Type yang merupakan status hubungan dengan *node-node* tetangga. Jika L\_SYM\_time tidak kadaluarsa, maka hubungan akan bersifat *simetris*. Sedangkan bila L\_SYM\_time telah kadaluarsa maka hubungan dengan *node* tetangga akan bersifat *asimetris*. Dan apabila L\_SYM\_time dan L\_ASYM\_time keduanya telah kadaluarsa maka hubungan dengan tetangga dinyatakan hilang [12].

### **2.2.1.2 Neighbour Sensing**

Setiap *node* pada jaringan juga harus mendeteksi *node-node* tetangga yang ada pada daerah jangkauannya. Untuk melakukan hal tersebut, setiap *node* akan mengirimkan paket pesan HELLO secara *broadcast* dalam periode waktu tertentu. Paket HELLO berisi informasi tentang *node-node* tetangga serta *link status*.

Dalam setiap *node* akan menyimpan informasi tentang *node-node* tetangga tersebut dalam “*neighbor set*” . *Neighbor set* tersebut berisi informasi sebagai berikut :

- N\_neighbor\_main\_addr,
- N\_status,
- N\_willingness.

N\_neighbor\_main\_addr merupakan alamat dari *node* tetangga, N\_status adalah informasi yang menunjukkan apakah hubungan dengan *node* tetangga tersebut bersifat SYM atau NOT\_SYM. N\_willingness merupakan tingkat kesediaan *node* tersebut untuk meneruskan paket untuk kepentingan *node* lain yang ditunjukkan dalam bentuk integer 0-7.

Selain *neighbor set* yang digunakan untuk menyimpan informasi tentang tetangga 1-hop, setiap *node* juga akan menyimpan informasi *node-node* tetangga 2-hop nya. Informasi tersebut disimpan dalam *2-hop neighbor set* yang berisi informasi antara lain :

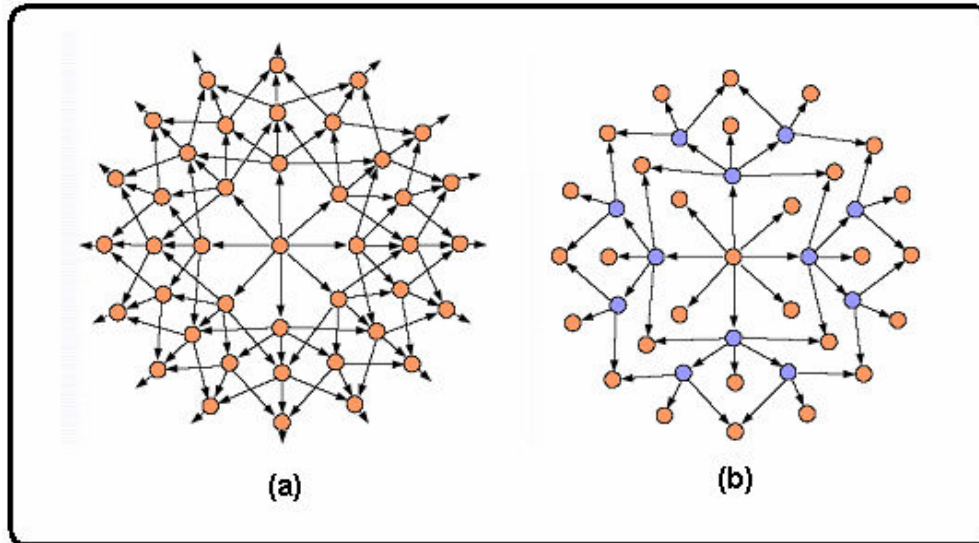
- N\_neighbor\_main\_addr,
- N\_2hop\_addr,
- N\_time.

N\_neighbor\_main\_addr merupakan informasi alamat dari *node* tetangga 1-hop, N\_2hop\_addr merupakan alamat dari *node* tetangga 2-hop yang bersifat simetris dengan tetangga 1-hop nya. N\_time menunjukkan waktu yang dibutuhkan hingga informasi yang ada pada set tersebut akan kadaluarsa dan harus dihilangkan [12].

### **2.2.1.3 MPR Selection**

Tujuan dari penggunaan *Multipoint Relay* (MPR) adalah meminimalisir penggunaan *overhead* yang pesan broadcast pada jaringan dengan cara mengurangi retransmisi (pentransmisi ulang) pada daerah yang sama [14]. Setiap *node* pada jaringan akan memilih sejumlah *node* tetangga 1-hop nya yang bersifat simetris yang akan melakukan transmisi ulang pesan-pesannya. Sejumlah *node* tetangga tersebutlah yang disebut dengan MPR. Setiap *node* tetangga yang tidak terpilih menjadi MPR tetap akan menerima dan memproses pesan *broadcast* namun tidak akan meneruskan atau mengirimkan kembali pesan-pesan tersebut.

Pemilihan *node-node* untuk dijadikan MPR selain harus bersifat simetris juga harus sedemikian rupa dapat menjangkau sejumlah *node* tetangga *2-hop*. Makin sedikit jumlah MPR maka makin sedikit penggunaan *control traffic overhead* yang digunakan dalam *routing protocol*. Perbandingan kinerja pengiriman paket untuk OLSR dan *link state protocol* pada umumnya digambarkan pada Gambar 2.7.



Gambar 2.7 Perbandingan Sistem Broadcast (a) General Broadcasting (b) MPR Broadcasting [13].

Setiap *node* akan menyimpan informasi tentang *node-node* tetangga yang telah dipilihnya sebagai MPR dalam sebuah “*MPR set*” yang berisi alamat-alamat *node* MPR tersebut. Selain itu setiap *node* juga akan menyimpan informasi tentang siapa-siapa saja yang telah memilihnya sebagai MPR. Informasi tersebut disimpan dalam “*MPR Selector Set*” yang berisi informasi antara lain :

- *MS\_main\_addr*,
- *MS\_time*.

*MS\_main\_addr* merupakan alamat-alamat *node* tetangga yang telah memilihnya sebagai MPR. *MS\_time* berisi informasi waktu yang dibutuhkan hingga informasi tersebut kadaluarsa dan harus dihilangkan [12].



#### 2.2.1.4 Topology Control (TC) Messages Diffusion

Pendeteksian hubungan serta pendeteksian *node-node* tetangga dari protokol OLSR pada dasarnya menyediakan informasi daftar tetangga yang dapat berkomunikasi secara langsung, dan dikombinasikan dengan mekanisme *broadcast* dengan menggunakan MPR informasi topologi dapat dikirimkan ke seluruh jaringan.

Rute dibentuk dari *advertised link* dan hubungan dengan *node-node* tetangga. Setiap *node* setidaknya harus mempunyai informasi tentang hubungan antara dirinya sendiri dengan *node-node* yang ada pada *MPR-selector set* nya dalam rangka mendapatkan informasi routing yang baik.

Pesan TC dikirimkan untuk menyediakan informasi *link-state* bagi setiap *node* pada jaringan yang dapat digunakan untuk penentuan jalur yang dapat digunakan. Setiap *node* akan menyimpan informasi yang didapat dari pesan TC ini dalam “Topology Set” yang berisi informasi antara lain:

- T\_dest\_addr,
- T\_last\_addr,
- T\_seq,
- T\_time.

T\_dest\_addr merupakan alamat dari sebuah node yang dapat dicapai dalam 1 *hop* dari *node* dengan alamat yang pada T\_last\_addr. Sehingga T\_last\_addr merupakan MPR dari *node* yang ada pada T\_dest\_addr. T\_seq adalah nomor urut dan T\_time menunjukkan batas waktu kadaluarsa dari informasi tersebut.

#### 2.2.1.5 Routing Calculation

Dengan menggunakan informasi *link state* yang didapatkan dari pertukaran pesan secara periodik dan juga disertai dengan konfigurasi *interface* dari setiap *node* maka *routing table* dari setiap node dapat dikalkulasi.

Setiap *node* memiliki *routing table* yang dapat dipakai untuk jalur data menuju *node-node* lain dalam jaringan. *Routing* tersebut dibuat berdasarkan informasi dalam *local link information base* (*local link set*, *neighbour set*, *2-hop neighbour set*, *MPR set*), serta informasi pada *topology set*. Oleh karena itu, apabila terjadi perubahan pada *set-set* tersebut maka *routing table* akan

dikalkulasi ulang untuk meng-*update* informasi tentang setiap tujuan dalam jaringan.

Adapun informasi yang disimpan dalam suatu *routing table* adalah seperti terlihat pada tabel 2.1.

Tabel 2.1 Tabel *Routing* OLSR [12]

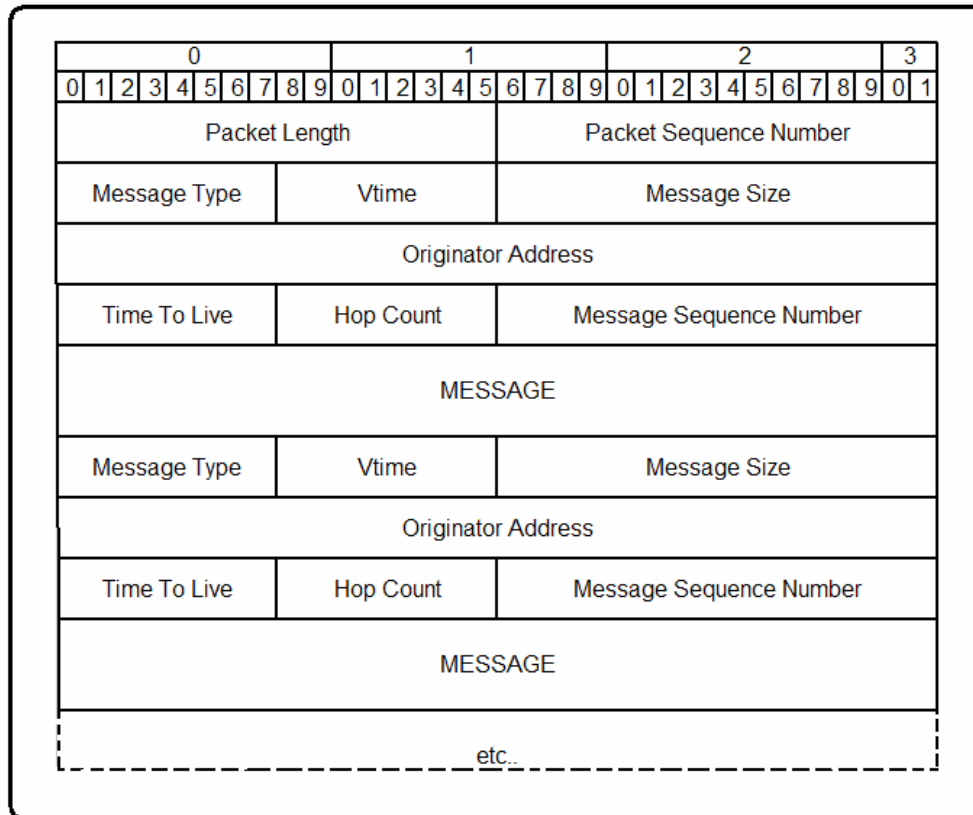
1	<b>R_dest_addr</b>	<b>R_next_addr</b>	<b>R_dist</b>	<b>R_iface_addr</b>
2	<b>R_dest_addr</b>	<b>R_next_addr</b>	<b>R_dist</b>	<b>R_iface_addr</b>
3	.....	.....	.....	.....

R\_dest\_addr menunjukkan alamat *node* yang dapat dituju sedangkan R\_dist merupakan jarak atau jumlah *hop* yang harus dilalui untuk mencapai *node* tujuan tersebut. R\_next\_addr merupakan alamat node dari hop berikutnya yang dari rute untuk menuju alamat tujuan. R\_iface\_addr merupakan alamat *interface* pada *node* sumber yang dapat dipakai untuk menghubungi *node* pada R\_next\_addr.

Setiap *entries* yang disimpan dalam *routing table* untuk setiap *node* tujuan yang ada pada jaringan dimana telah terdapat rute untuk menuju *node* tersebut. Setiap tujuan yang rutenya putus atau tidak memiliki informasi yang lengkap tidak akan dimasukkan dalam *routing table*.

### 2.2.2 Format Paket OLSR

Gambar 2.8 menggambarkan format paket pada protokol OLSR, paket tersebut dapat dibagi menjadi tiga bagian utama yaitu : *Packet Header*, *Message Header* dan *Message*.



Gambar 2.8 Format Paket OLSR [12].

Penjelasan mengenai *field-field* yang ada pada format paket OLSR dapat terlihat pada tabel 2.2.

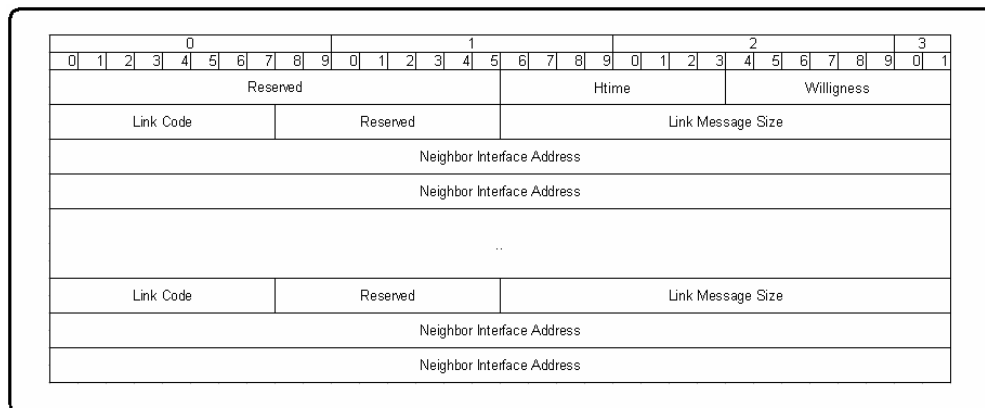
Tabel 2.2 Penjelasan Field-field Pada Format Paket OLSR

No	Nama Field	Deskripsi	Detail
<b>A OLSR Packet Header</b>			
1	Packet Length	Panjang dari paket OLSR dalam byte	
2	Packet Sequence Number	Merupakan nomor urut dari pengiriman paket, Packet Sequence Number harus ditambah dengan nilai 1 setiap terjadi pentransmisi-an paket yang baru.	
<b>B OLSR Message Header</b>			
1	Message Type	Menampilkan jenis pesan dari paket tersebut.	HELLO_Messages = 1 TC_Messages = 2 MID_Messages = 3 HNA_Messages = 4
2	Vtime	Field ini mengindikasikan berapa lama waktu yang dibutuhkan untuk menentukan validasi suatu paket oleh suatu node sejak diterimanya paket tersebut. Vtime dapat diperkirakan dengan menggunakan persamaan sebagai berikut : $\text{validity time} = C \cdot (1 + a/16)^{2^b} \text{ [in seconds]}$ dimana: C = nilai dari scaling factor (1/16 detik) a = nilai integer dari empat bit pertama dari Vtime b = nilai integer dari empat bit terakhir dari Vtime	
3	Message Size	Menunjukkan ukuran dari pesan tersebut dimulai, dinyatakan dalam byte. Ukuran dari pesan dimulai dari awal field Message Type hingga bertemu awal dari field Message Type lainnya.	
4	Originator Address	Field ini berisi alamat utama dari node pengirim yang mengirimkan pesan tersebut untuk pertama kalinya.	
5	Time To Live	Field ini berisi jumlah maksimum dari hop yang dapat dilalui pada saat transmisi. Untuk setiap pentransmisi-an ulang, nilai dari field ini harus dikurangi dengan 1. Node yang menerima paket dengan nilai Time To Live sama dengan 0 atau 1 maka node tersebut tidak dapat meneruskan transmisi pesan tersebut	
6	Hop Count	Field ini berisi informasi jumlah hop yang sudah dilewati oleh pesan tersebut. Nilai dari field ini harus dijumlahkan dengan nilai 1 setiap akan melakukan pentransmisi-an ulang.	
7	Message Sequence Number	Setiap terjadi pengiriman pesan, node pengirim awal akan memberikan nomor urut tertentu pada setiap pesan yang akan dikirimkan, Fungsi dari pemberian nomor urut ini adalah untuk memastikan tidak terjadinya pentransmisi-an ulang lebih dari satu kali oleh node manapun	

### 2.2.2.1 HELLO Messages

Setiap *node* pada harus mendeteksi *node-node* tetangga yang ada pada daerah jangkauannya. Untuk melakukan hal tersebut, setiap *node* akan mengirimkan paket HELLO message secara *broadcast* dalam periode waktu tertentu. Paket HELLO berisi informasi tentang node-node tetangga serta link status.

Adapun format dari pesan HELLO adalah terlihat pada Gambar 2.9.



Gambar 2.9 Format Paket HELLO [12].

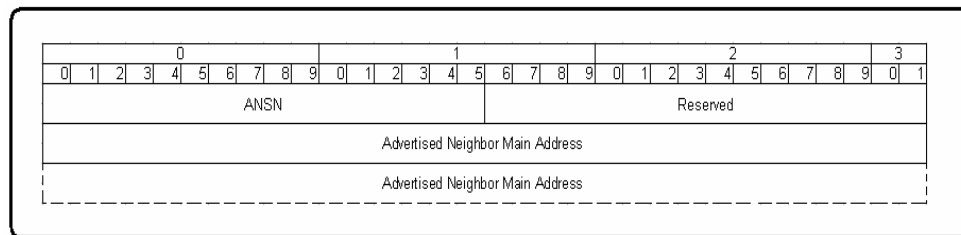
Penjelasan mengenai fungsi-fungsi dari setiap *field* yang ada pada format paket HELLO terlihat pada Tabel 2.3.

Tabel 2.3 Penjelasan *Field-field* Pada Format Paket HELLO

No	Nama Field	Deskripsi	Detail
1	Reserved	Field ini harus berisi "000000000000" agar dapat bekerja pada spesifikasi format paket ini	
2	Htime	Field ini berisi informasi tentang interval waktu yang digunakan antara pengiriman paket HELLO satu dengan pengiriman paket HELLO selanjutnya. Untuk menentukan interval waktu tersebut digunakan persamaan sebagai berikut : Hello emission interval = $C \cdot (1 + a/16)^b \cdot 2^b$ [in seconds] dimana: C = nilai dari scaling factor (1/16 detik) a = nilai integer dari empat bit pertama dari Vtime b = nilai integer dari empat bit terakhir dari Vtime	
3	Willingness	Field ini menunjukkan kesediaan suatu node untuk meneruskan paket yang diterimanya atau tidak.	WILL_NEVER = 0 WILL_LOW = 1 WILL_DEFAULT = 3 WILL_HIGH = 6 WILL_ALWAYS = 7
4	LinkCode	Field ini menyatakan informasi tentang link antara interface dari node pengirim dengan interface node-node tetangganya serta berisi pula informasi tentang status dari node tetangga tersebut	Link Types : UNSPEC_LINK = 0 ASYM_LINK = 1 SYM_LINK = 2 LOST_LINK = 3 Neighbour Types : NOT_NEIGH = 0 SYM_NEIGH = 1 MPR_NEIGH = 2
5	Link Message Size	Field ini menyatakan ukuran dari pesan HELLO (dalam byte) yang diukur dari awal field Link Code hingga bertemu dengan awal field Link Code pada pesan berikutnya	
6	Neighbor Interface Address	Field ini berisi informasi alamat interface dari node-nde tetangga	

### 2.2.2.2 TC (*Topology Control*) Messages

Setiap *node* yang telah terpilih sebagai MPR dalam jaringan mengirimkan TC Messages (pesan TC) untuk mendeklarasikan sebuah set/kumpulan jalur-jalur yang disebut *advertised link set* yang harus disertai minimal jalur ke seluruh *node* dari MPR *selector set*-nya atau *node-node* tetangga yang telah memilih *node* tersebut sebagai MPR-nya. Format dari paket TC Messages seperti digambarkan pada Gambar 2.10.



Gambar 2.10 Format Paket TC Messages [12].

Penjelasan mengenai *field-field* dari format paket TC message dapat dilihat tabel 2.4.

Tabel 2.4 Penjelasan *Field-field* Pada Format Paket TC

No	Nama Field	Deskripsi
1	Advertised Neighbour Sequence Number (ANSN)	Field ini berisi nomor urut yang berhubungan dengan advertised neighbour set. Setiap sebuah node mendeteksi adanya perubahan pada advertise neighbour set-nya, maka node tersebut akan meningkatkan nomor urut ini. Nomor urut ini berfungsi untuk menginformasikan baru tidaknya sebuah TC message
2	Advertised Neighbour Main Address	Field ini berisi alamat utama dari node tetangga. Semua alamat dari tetangga node utama akan dimasukkan ke dalam TC messages
3	Reserved	Field ini harus berisi "000000000000" agar dapat bekerja pada spesifikasi format paket ini

### 2.3 OpenWRT

OpenWRT merupakan program *firmware* berbasis sistem operasi Linux yang digunakan dalam suatu *embedded device* seperti *wireless router*. Dikembangkan pertama kali pada tahun 2004 oleh tim proyek OpenWRT yang terbentuk dalam rangka mengembangkan sebuah *third-party firmware* yang jumlahnya masih sangat terbatas pada saat itu [15].

Pada awalnya OpenWRT diciptakan hanya untuk mendukung *wireless router* Linksys WRT54G series, namun pada perkembangannya OpenWRT dapat pula digunakan untuk mendukung *wireless router* dari hasil manufaktur yang lain seperti ASUS, D-Link, DELL dan lain-lain. Perkembangan OpenWRT sangat

dipengaruhi akan kemudahannya dalam memodifikasi fitur-fitur tambahan diluar fitur-fitur yang telah disediakan oleh pihak manufaktur agar dapat digunakan sesuai dengan keperluan tertentu dari para pengguna. Hal ini dapat terjadi karena OpenWRT bersifat opensource karena dibuat berdasarkan GNU *General Public License/Linux*, sehingga setiap perubahan yang dibuat oleh pihak manufaktur harus didaftarkan dan dirilis melalui lisensi GPL. Berdasarkan sifat *opensource* ini pula maka para pengguna dapat dengan bebas memodifikasi ataupun menambahkan fitur-fitur lain pada *router* sesuai dengan kebutuhan.

Tidak seperti firmware bawaan dari *wireless router* yang merangkum semua fitur dalam satu firmware, OpenWRT hanya menyediakan konfigurasi minimal namun dengan kemampuan untuk mendukung paket-paket fitur tambahan. Bagi para pengguna ini berarti penghematan ruang, karena paket-paket fitur yang tidak diperlukan dapat dihilangkan. OpenWRT menggunakan sistem ipkg seperti yang terdapat pada *distro* Linux Debian untuk mengatur paket-paket fitur tersebut [16].

Sampai saat ini ada dua jenis OpenWRT yang telah dikembangkan [17] yaitu sebagai berikut.

1. WhiteRussian

Versi awal dari OpenWRT, lebih stabil karena telah dikembangkan lebih lama. Oleh karena itu telah banyak dokumentasi maupun tutorial yang tersedia untuk mendukung pemakaian OpenWRT versi ini. Walaupun demikian pengembangannya telah dihentikan pada permulaan tahun 2007. Seri terakhir dari whiterussian adalah WhiteRussian 0.9 yang dirilis pada tanggal 5 Februari 2007.

2. Kamikaze

Versi terbaru dari OpenWRT, walaupun sudah stabil namun masih dalam pengembangan. Dibuat berdasarkan desain berbeda dengan versi yang terdahulu sehingga dapat bekerja pada pilihan jenis *wireless router* yang lebih luas, selain itu mempunyai kernel yang lebih baru. Seri terakhir dari kamikaze adalah Kamikaze 7.09 yang dirilis pada September 2007.



## **BAB III**

### **PERANCANGAN DAN PEMBUATAN JARINGAN**

#### **3.1 Perancangan *Testbed*.**

Dalam membangun *testbed wireless mesh network* langkah pertama yang diambil adalah menentukan langkah-langkah pengerjaan pembangunan *testbed* jaringan tersebut. Adapun langkah-langkah yang dipakai dalam pembuatan *testbed* jaringan tersebut adalah :

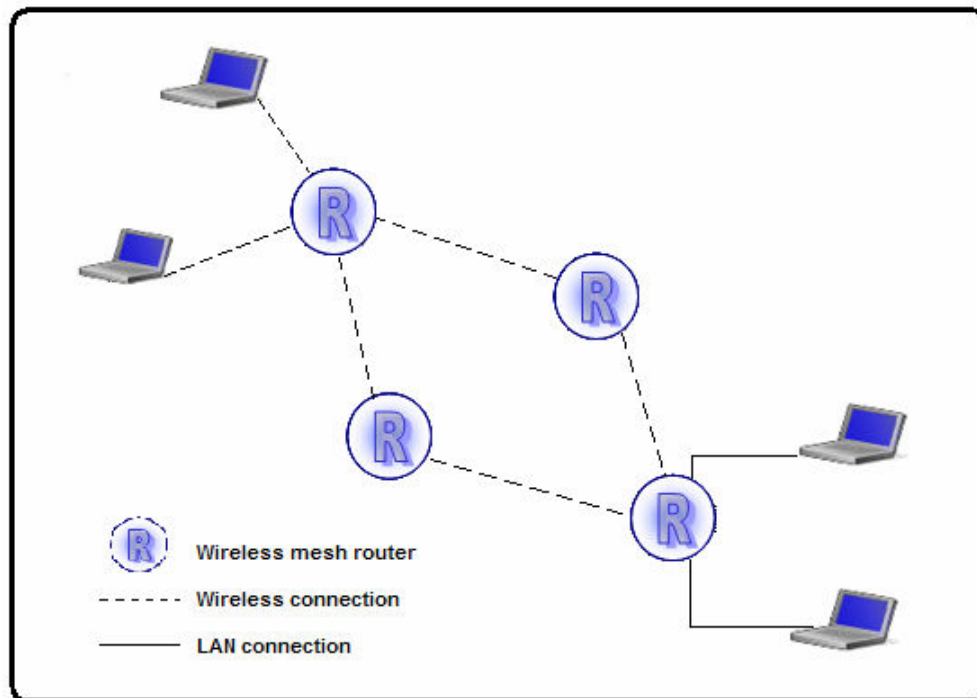
1. Menentukan topologi dari jaringan *testbed*.  
Pada langkah ini dilakukan perancangan bentuk topologi jaringan yang akan sesuai dengan keperluan *testbed*.
2. Menentukan spesifikasi dari perangkat-perangkat yang akan dipakai.  
Penentuan spesifikasi dari perangkat keras maupun perangkat lunak yang akan dipakai pada *testbed*.
3. Menentukan lokasi *testbed*.  
Penentuan lokasi dari *testbed* disesuaikan dengan bentuk topologi jaringan *testbed* yang diinginkan.
4. Mengkonfigurasi jaringan.  
Melakukan konfigurasi jaringan seperti pengalamatan, konfigurasi *olsr*, *packet forwarding* dan sebagainya.

#### **3.2 Pembangunan *Testbed*.**

Tahap berikutnya adalah merealisasikan langkah-langkah yang telah ditentukan dalam tahapan perancangan.

##### **3.2.1 Penentuan Topologi Jaringan.**

Pembangunan topologi jaringan *wireless mesh* yang dibangun adalah jaringan *wireless mesh* yang bertipe *infrastructured wireless mesh network* dimana terdapat jaringan infrastruktur yang terdiri dari beberapa *wireless router*. Adapun bentuk topologi *testbed* jaringan *wireless mesh* yang akan dibangun adalah seperti pada gambar 3.1.



Gambar 3.1 Bentuk Topologi *Testbed Wireless Mesh Network*.

Bentuk topologi seperti pada gambar 3.1 dinilai dapat memenuhi tujuan pembangunan *testbed* yaitu mengetes kinerja dari suatu jaringan *wireless mesh* dengan *routing protocol* OLSR. Pada gambar 3.1 terlihat bahwa jaringan infrastruktur terbentuk dari empat buah *wireless mesh router* yang dapat terhubung dengan berbagai koneksi *user* seperti *wireless* dan LAN (*Local Area Network*) berbasis Ethernet.

### 3.3 Spesifikasi Sistem.

#### 3.3.1 Spesifikasi *Hardware*.

Berdasarkan bentuk topologi jaringan *testbed* yang ingin dibentuk maka untuk membangun sebuah infrastruktur *mesh* memerlukan empat buah *wireless mesh router*. Selain itu juga memerlukan beberapa *user* dalam bentuk koneksi *wireless* dan LAN.

### 3.3.1.1 Spesifikasi *Hardware Untuk Wireless Mesh Router.*

Untuk mendapatkan sebuah *wireless mesh router* dapat menggunakan sebuah *wireless router* yang sudah biasa digunakan namun harus mempunyai sifat mudah untuk dimodifikasi agar dapat digunakan untuk keperluan jaringan *wireless mesh*. Berdasarkan kebutuhan tersebut *wireless router* yang digunakan untuk kepentingan *testbed* yang akan dibangun adalah Linksys *Wireless Router WRT54GL* seperti terlihat pada gambar 3.2.a dan 3.2.b.



Gambar 3.2.a. Tampak Depan *Wireless Router Linksys WRT54GL*



Gambar 3.2.b. Tampak Belakang *Wireless Router Linksys WRT54GL*

*Wireless router* ini dikembangkan untuk dapat beroperasi dengan menggunakan sistem operasi / *firmware* berbasis Linux yang bersifat *open source* sehingga mudah untuk dimodifikasi sesuai dengan kebutuhan *testbed*. Selain itu *wireless router* ini juga mempunyai empat buah port LAN yang digunakan untuk koneksi menggunakan kabel ethernet.

Adapun spesifikasi lengkap dari *wireless router* ini adalah :

- Tipe : *Wireless router* Linksys WRT54GL v 1.0
- *Wireless network standard* : IEEE 802.11g
- *Wireless data transfer rate* : 54 Mbps
- *Wireless indoor range* : 100 m
- *Wireless outdoor range* : 300 m
- *Security protocols* : WPA, 128 bit WEP, 64 bit WEP
- *Networking standard* : IEEE 802.3 Ethernet 10 base T
- *Routing* : TCP/IP
- *Management* : *Web based*
- *Wan port* : 1
- *10/100 Mbps port* : 4
- *Ethernet* : Cat6, Cat5e

### 3.3.1.2 Spesifikasi *Hardware* Untuk *User/client*.

Untuk menentukan spesifikasi *user* atau *client* berdasarkan bentuk topologi jaringan *testbed* yang diinginkan maka bentuk *user* yang ideal untuk digunakan dalam *testbed* ini adalah sebuah *laptop/notebook* karena dapat bersifat *mobile* maupun statis. Oleh karena itu *laptop/notebook* tersebut harus mempunyai spesifikasi minimal seperti :

- Port koneksi ethernet
- *Wireless* modem
- Processor Pentium III
- Memory 256 Mb

### 3.3.2 Spesifikasi *Software*.

Untuk spesifikasi *software* dibagi menjadi spesifikasi *software* yang digunakan pada *wireless mesh router* dan spesifikasi *software* untuk *user*.

#### 3.3.2.1 Spesifikasi *Software* Untuk *Wireless Mesh Router*

Spesifikasi *software* yang digunakan pada *wireless mesh router* adalah menyangkut penggunaan sistem operasi atau *firmware* yang ada pada *wireless router* tersebut. *Wireless router* Linksys WRT54GL sebenarnya sudah mempunyai *firmware* bawaan namun *firmware* tersebut tidak bersifat fleksibel dan tidak dapat untuk dimodifikasi untuk penggunaan sebagai *wireless mesh router*. Untuk itu digunakan sebuah *firmware* berbasis Linux yang disebut OpenWRT. OpenWRT bersifat *opensource* sehingga dapat mudah dimodifikasi serta memiliki banyak *package-package* tambahan yang dapat digunakan sesuai kebutuhan pembangunan *testbed*.

Spesifikasi OpenWRT yang digunakan pada *wireless mesh router* untuk *testbed* antara lain :

- Tipe : WhiteRussian 0.9
- Developed by : openwrt.org
- OS : Linux
- Genre : Router-OS
- License : GPL
- Package management : ipkg package system
- Package-package tambahan : - olsrd (untuk *routing protocol* OLSR)  
 - olsrd-mod-httpinfo (manajemen OLSR)  
 - wl (untuk mengatur power tx)  
 - snmpd (untuk pengaktifan SNMP)

#### 3.3.2.2 Spesifikasi *Software* Untuk *Client/user*

Tidak ada spesifikasi khusus dalam pemilihan *software* untuk digunakan pada *laptop user*. Setiap system operasi seperti Windows, Linux dan sebagainya dapat digunakan selama kompatibel untuk penggunaan jaringan baik itu *wireless* maupun *wired*.

### 3.3.2.3 Instalasi OpenWRT

Sebelum menginstall *firmware* OpenWRT, perlu ditentukan versi *image* OpenWRT yang cocok dengan tipe *wireless router* yang akan dipakai seperti terlihat pada tabel 3.1.

Tabel 3.1 Tabel Pasangan Antara Versi Image OpenWRT Dengan Jenis *Router*

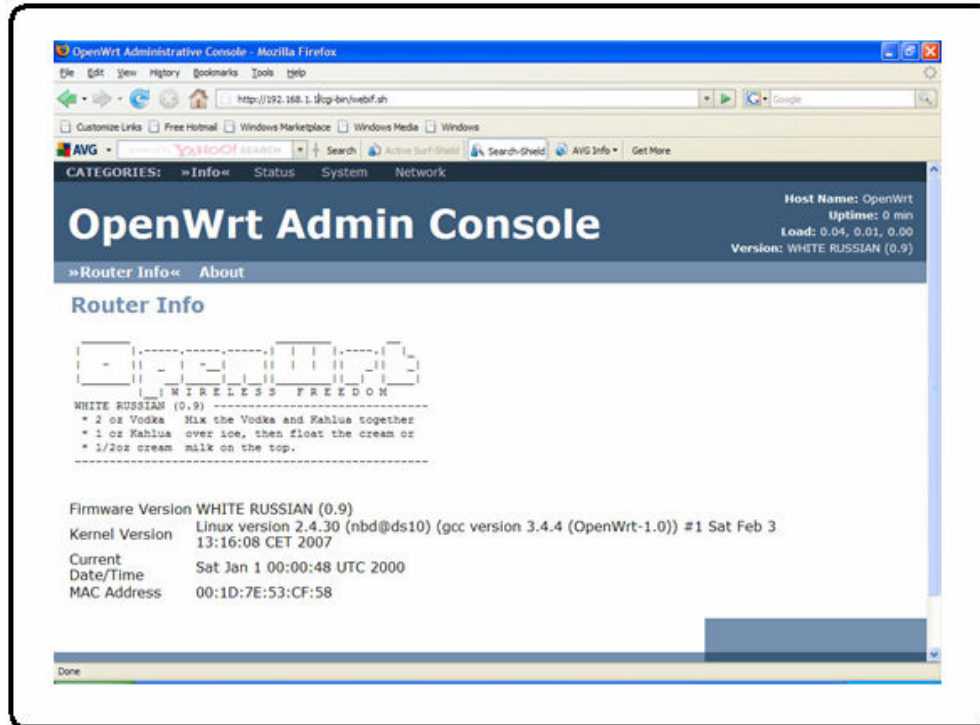
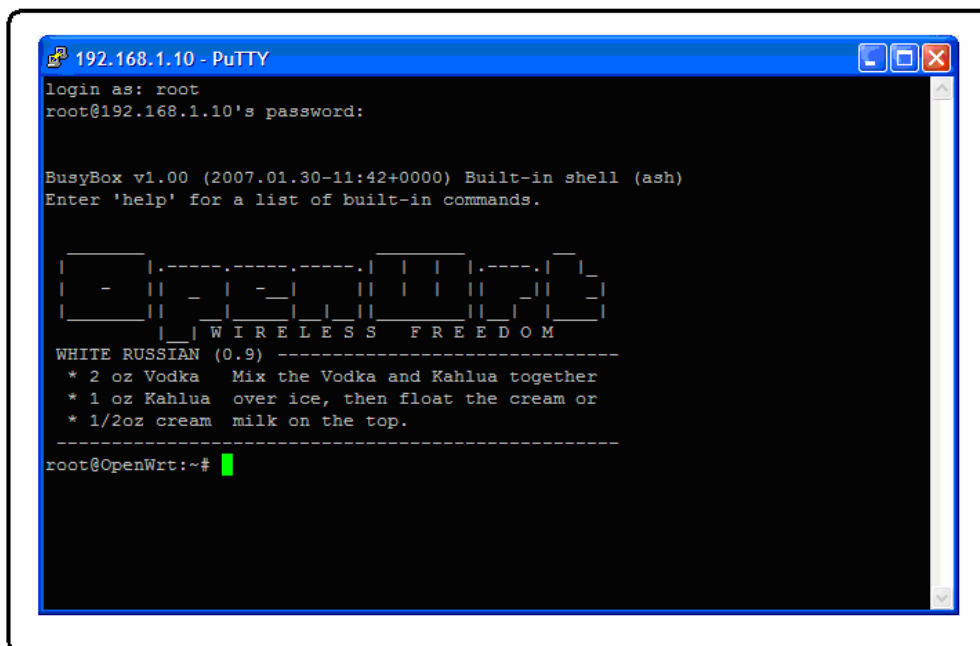
Device	OpenWrt Image
Linksys WRT54G (versions 1.0, 1.1, 2.0, 2.2, 3.0, and 4.0)	
Linksys WRT54GL (versions 1.0 and 1.1)	openwrt-wrt54g-<fstype>.bin
Linksys WRT54GS (versions 1.0, 1.1, 2.0, and 3.0)	openwrt-wrt54gs-<fstype>.bin
Linksys WRT54GS (version 4.0)	openwrt-wrt54gs_v4-<fstype>.bin
Linksys WRTSL54GS (all versions as of the time of this writing)	openwrt-wrtsl54gs-<fstype>.bin

Berdasarkan tabel 3.1 maka paket *image* OpenWRT yang digunakan adalah `openwrt-wrt54g-squashfs.bin`. Paket *image* ini dapat di-*download* pada situs [www.openwrt.org](http://www.openwrt.org). Untuk menginstall OpenWRT dapat menggunakan fitur *firmware upgrade* melalui *web interface* dari *firmware* Linksys bawaan pabrik melalui alamat *default-nya* yaitu 192.168.1.1.



Gambar 3.3 Jendela Firmware Upgrade.

Apabila *firmware* OpenWRT telah terinstall, maka OpenWRT dapat dimasuki melalui *web interface* maupun Telnet atau SSH untuk memasuki jendela *command line*. Gambar 3.4 dan 3.5 memperlihatkan halaman depan dari *web interface* serta jendela *command line* dari OpenWRT.

Gambar 3.4 Halaman Depan *Web Interface* OpenWRTGambar 3.5 Tampilan *Command Line Interface* OpenWRT



### 3.3.2.4 Instalasi Package Tambahan Pada OpenWRT

Untuk dapat menjalankan *routing protocol* OLSR, openWRT memerlukan paket tambahan bernama *olsrd* (*optimized link state routing daemon*). Paket *olsrd* dapat di-*download* pada situs [www.openwrt.org](http://www.openwrt.org) dan [www.olsr.org](http://www.olsr.org) atau pun diinstall langsung pada *router* yang terhubung dengan internet dengan menggunakan fasilitas *ipkg*. Selain itu diperlukan pula *package-package* seperti *olsrd-http\_info*, *wl* dan *snmpd*.

Adapun perintah-perintah yang dapat dijalankan untuk menginstall *package-package* tersebut adalah :

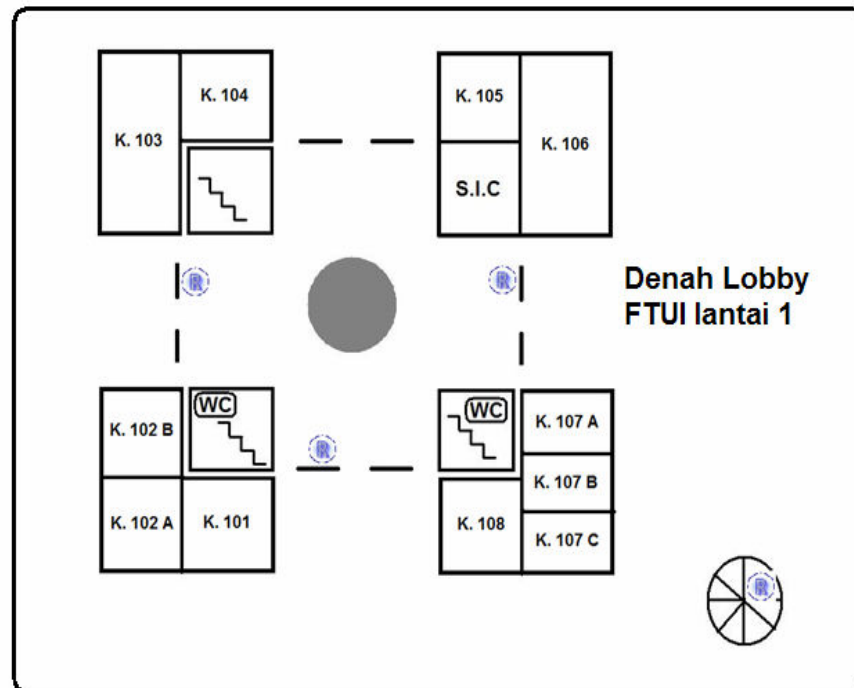
```
ipkg update
ipkg install olsrdm
ipkg install olsrd-mod-httpinfo
ipkg install wl
ipkg install snmpd
```

### 3.4 Penentuan Lokasi *Testbed*

Salah satu faktor yang menentukan dalam pembangunan *testbed* jaringan *wireless mesh* ini adalah penentuan lokasi untuk membangun *testbed* atau dalam hal ini adalah lokasi penempatan *router-router* sebagai infrastruktur jaringan *wireless mesh*. Lokasi penempatan *router* diambil dengan pertimbangan-pertimbangan sebagai berikut :

- Lokasi *router* harus dapat memenuhi bentuk rancangan topologi jaringan *testbed* seperti pada gambar 3.1.
- Lokasi mempunyai tempat sumber listrik untuk *router*.
- Lokasi terlindung dari gangguan-gangguan seperti : cuaca, hilir mudik manusia dan sebagainya.
- Mudah untuk dijangkau untuk keperluan pengetesan dan konfigurasi.

Berdasarkan pertimbangan-pertimbangan di atas maka lokasi paling cocok Untuk membangun *testbed* jaringan *wireless mesh* adalah daerah sekitar lobi Fakultas Teknik Universitas Indonesia. Penempatan *router* diatur sedemikian rupa agar dapat memenuhi tujuan rancangan seperti terlihat pada gambar 3.6 dan gambar 3.7.



Gambar 3.6 Denah Lokasi Penempatan *Wireless Router*



Gambar 3.7 Lokasi-lokasi Penempatan *Wireless Router*

### 3.5 Konfigurasi Jaringan

Setelah proses instalasi OpenWRT dan paket *olsrd* selesai, hal berikutnya yang perlu dilakukan adalah mengkonfigurasi jaringan serta *olsrd* agar dapat berjalan sesuai dengan ketentuan *routing protocol* OLSR.

Beberapa point penting yang perlu diperhatikan dalam mengkonfigurasi sebuah *nodemesh router* dalam suatu jaringan agar dapat bekerja dengan sistem OLSR adalah :

1. Pengalamatan (*address assignment*)

Setiap *node* dalam jaringan *wireless mesh* harus diberikan pengalamatan yang baik yaitu penggunaan alamat setiap *mesh router* dengan *netmask*-nya dengan menggunakan IPv4.

2. Konfigurasi *routing* (*routing configuration*)

Setiap *node* yang berhubungan dengan jaringan *mesh* harus dikonfigurasi agar dapat melakukan *set-up routing* terhadap *interface-interface* yang terhubung dengan jaringan *mesh*.

3. *Data packet forwarding*

OLSR sendiri tidak melakukan *package forwarding*, melainkan hanya melakukan penjagaan / *peng-update-an routing table* yang ada pada suatu *operating system* yang prosesnya terlihat *package forwarding*.

#### 3.5.1 Pengalamatan

Langkah pertama pengkonfigurasian adalah menentukan alamat-alamat pada masing-masing *router* baik alamat untuk *interface* LAN ataupun *wireless* LAN pada *router* tersebut. Pada *wireless router* Linksys WRT54GL secara *default* mempunyai alamat 192.168.1.1 pada *interface* br0. Dimana *interface* br0 bersifat *bridge* sehingga alamat untuk LAN dan *wireless* LAN hanya memerlukan satu alamat.

Untuk proyek *wireless mesh networking* ini dimana jaringan akan bersifat *ad-hoc* maka *interface* br0 yang bersifat *bridge* tersebut harus dihilangkan. Cara untuk menghilangkan *interface* br0 adalah dengan memberikan alamat yang berbeda pada masing-masing *interface* LAN dan *wireless* LAN.

Memberi alamat untuk *interface* LAN :

```
nvrn set lan_ifname=vlan1      (memberi nama interface lan = vlan1)
nvrn set lan_ipaddress=192.168.2.10 (memberi alamat lan =192.168.2.10)
nvrn set lan_netmask=255.255.255.0 (memberi netmask lan =255.255.255.0)
nvrn set lan_proto=static      (menon-aktifkan DHCP)
```

Memberi alamat untuk *interface* wi-fi :

```
nvrn set wifi_ifname=eth1      (memberi nama interface wifi = eth1)
nvrn set wifi_ipaddress=192.168.2.1 (memberi alamat wifi =192.168.2.1)
nvrn set wifi_netmask=255.255.255.0 (memberi netmask wifi =255.255.255.0)
nvrn set wifi_proto=static     (menon-aktifkan DHCP)
```

Menjalankan perintah konfigurasi.

```
ifup lan      (menjalankan perintah konfigurasi untuk lan)
ifup wifi     (menjalankan perintah konfigurasi untuk wifi)
nvrn commit   (menuliskan perubahan pada nvrn)
reboot       (me-reboot router)
```

Kemudian untuk pengkonfigurasian parameter-parameter *wireless* lebih lanjut agar dapat bekerja maksimal dalam kondisi *ad-hoc* dilakukan dengan menjalankan perintah-perintah berikut :

```
nvrn set wl0_radio=1      (menghidupkan koneksi radio wireless)
nvrn set wl0_infra=0     (0 = adhoc mode , 1 = normal AP)
nvrn set wl0_SSID=OLSR   (memberi nama SSID = "OLSR")
nvrn set wl0_channel=11  (menentukan channel yang dipakai 0-11)
nvrn set wl0_closed=0    (0 = broadcast SSID, 1 = hide SSID)
nvrn commit             (menuliskan perubahan pada nvrn)
reboot                 (me-reboot router)
```

### 3.5.2 Konfigurasi *Routing*

Konfigurasi untuk kinerja OLSR dapat dilakukan dengan mengedit *file* `olsrd.conf` yang terdapat pada `/etc/olsrd.conf`. Konfigurasi dalam `olsrd.conf` terdiri dari *comments*, *single option* dan *option blocks*. *Comments* adalah setiap baris yang diikuti dengan tanda `#`. Data yang mengandung tanda tersebut akan

dihilangkan. *Single option* merupakan sebuah baris tunggal berisi *keyword* atau nilai tertentu. Sedangkan *option blocks* merupakan *option* konfigurasi yang terdiri dari sekumpulan *sub-option* yang ditandai dengan tanda {}.

Adapun konfigurasi dari `olsrd.conf` untuk proyek ini adalah seperti ditunjukkan sintaks berikut :

```

DebugLevel          0
IpVersion           4
AllowNoInt          yes
Pollrate            0.1
TcRedundancy        2
MprCoverage         7
LinkQualityfishEye  1
LinkQualityWinSize  100

Hna4
{
192.168.10.0 255.255.255.0
}

LoadPlugin "olsrd_httpinfo.so.0.1"
{
PIParam "port" "8080"
PIParam "Host" "192.168.10.0"
PIParam "Net" "192.168.10.0 255.255.255.0"
}

LinkQualityLevel 2
UseHysteresis no
Interface "eth1"

{
HelloInterval      5.0
HelloValidityTime  90.0
TcInterval         2.0
TcValidityTime     270.0
MidInterval        15.0
MidValidiyTime     90.0
HnaInterval        15.0
HnaValidityTime    90.0
}

```

Beberapa poin konfigurasi yang penting dalam pengkonfigurasian di atas adalah untuk *point* `DebugLevel` yang diset dengan nilai 0. `DebugLevel` merupakan fungsi kontrol yang mengatur jumlah *debug output* yang dapat ditampilkan,

dengan nilai default 0 berarti setiap proses yang dijalankan oleh olsrd akan dijalankan dari belakang layar.

Point IpVersion merupakan fungsi untuk menentukan jenis alamat IP yang digunakan pada jaringan, dalam hal ini jaringan menggunakan alamat IP versi 4. Point TcRedundancy merupakan fungsi untuk mengatur TC *redundancy* yang digunakan oleh *node* sumber dalam mengirimkan pesan TC (*topology control*), dengan memberinya nilai 0 maka *advertised link set* dari *node* tersebut adalah setiap *node* tetangga yang bersifat *symmetric*.

Poin MprCoverage dengan nilai 7 menunjukkan nilai berapa banyak MPR yang dapat dimiliki oleh sebuah *node* untuk setiap *2-hop neighbour*. Poin Interface menunjukkan *interface* yang digunakan dalam jaringan *wireless* yang dalam hal ini adalah “eth1”. Poin Hna4 menunjukkan alamat LAN yang dapat berhubungan dengan *wireless mesh router* dalam hal ini berisi alamat jaringan 192.168.10.0 dan *netmask* 255.255.255.0, sehingga setiap *client* LAN dengan alamat *subnet* yang sama dapat berhubungan dengan *node wireless mesh router*.

Untuk poin HelloInterval bernilai 5.0 artinya pesan HELLO akan dikirimkan setiap 5 detik dan poin HelloValidityTime bernilai 90.0 berarti waktu valid informasi dalam pesan HELLO tersebut hingga 90 detik. Untuk poin TcInterval bernilai 2.0 artinya pesan TC akan dikirimkan setiap 2 detik dan poin TcValidityTime bernilai 270.0 artinya waktu valid informasi dalam pesan TC tersebut hingga 270 detik.

### **3.5.3 Data Packet Forwarding**

Agar setiap paket data yang dikirimkan melalui jaringan *mesh* dapat diteruskan melalui *interface-interface node* khususnya dengan jenis *interface* yang berbeda seperti *interface* LAN dan *wireless* LAN yang telah dipisahkan pengalamatannya pada proses konfigurasi sebelumnya maka perlu dilakukan pengaturan *forwarding rules*.

Untuk mengkonfigurasi *forwarding rules* yang akan diterapkan untuk setiap *node mesh router* dapat dilakukan melalui *file* *firewall.user* yang terdapat pada direktori */etc/firewall.user*.

Adapun hasil konfigurasi untuk proyek ini pada firewall.user adalah seperti ditunjukkan pada sintaks dibawah ini :

```
#!/bin/sh
. /etc/functions.sh

WAN=$(nvram get wan_ifname)
LAN=$(nvram get lan_ifname)
WIFI=$(nvram get wifi_ifname)

iptables -F input_rule
iptables -F output_rule
iptables -F forwarding_rule
iptables -t nat -F prerouting_rule
iptables -t nat -F postrouting_rule

# For forwarding WAN (internet) to WIFI

iptables -A forwarding_rule -i $WIFI -o $WAN -j ACCEPT

#For forwarding LAN & WIFI in nodes

iptables -A forwarding_rule -i $LAN -o $WIFI -j ACCEPT

#For WIFI clients to connect to node

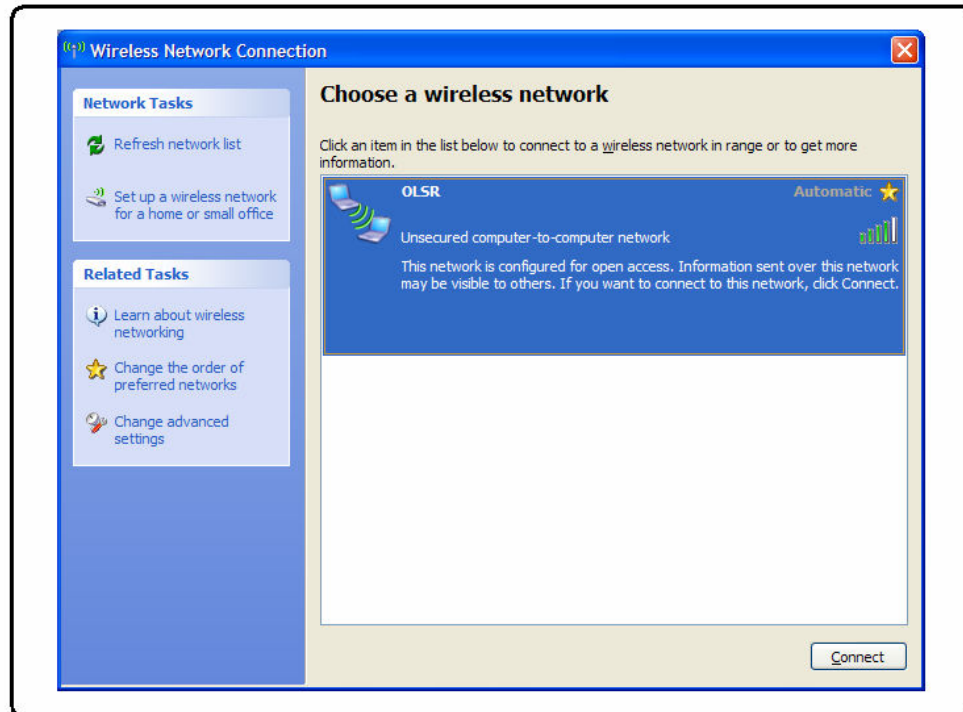
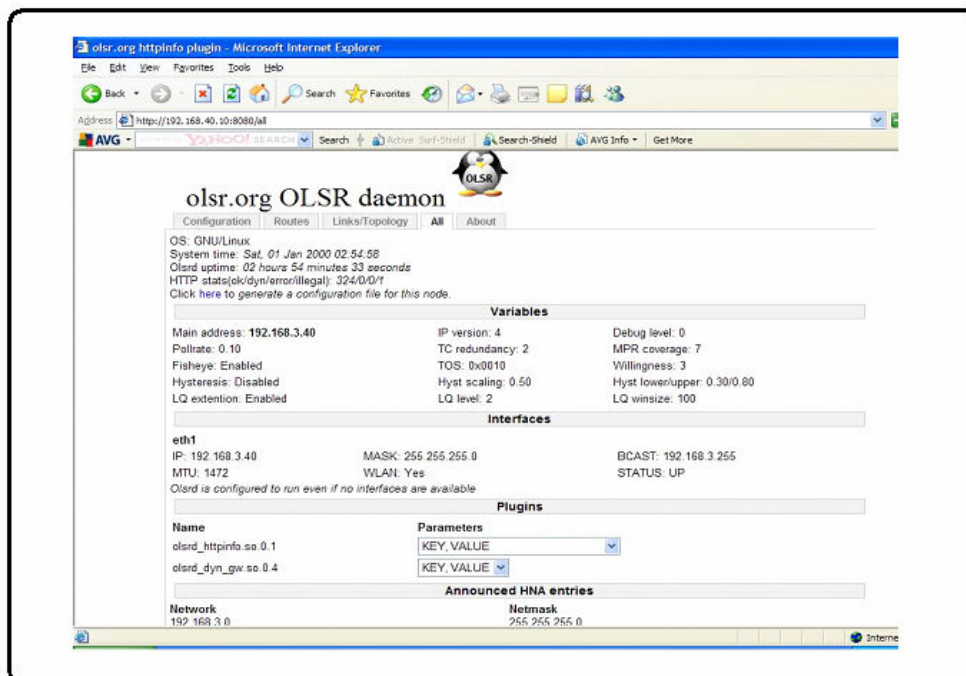
iptables -A forwarding_rule -i $WIFI -o $WIFI -j ACCEPT

#For connecting a Wired Lan client of node 1 to wired client of node 2

iptables -A forwarding_rule -i $LAN -o $LAN -j ACCEPT
```

Pada konfigurasi diatas telah diatur *forwarding rules* antar *interface-interface* yang berbeda-beda seperti dari LAN ke wifi, WAN (internet) ke *wifi*, hubungan interface antar *wifi client* maupun hubungan *interface* antar LAN *client*.

Setelah semua langkah pengkonfigurasi telah selesai dilakukan maka jaringan pun siap untuk digunakan, seperti terlihat pada jendela Wireless Network Connection pada gambar 3.8 serta jendela manajemen olsrd melalui *web interface* http-mod-info.

Gambar 3.8 Jendela *Wireless Network Connection*

Gambar 3.9 Web Interface olsrd http-mod-info

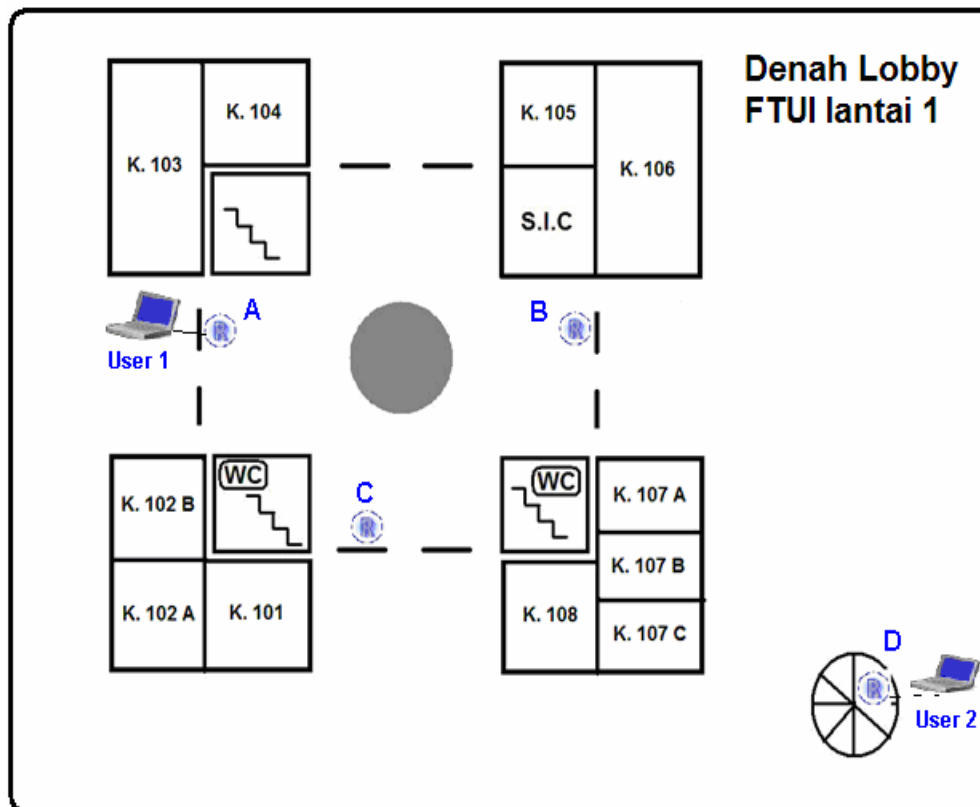


# BAB IV

## PENGUJIAN DAN ANALISIS

### 4.1 Hasil Rancang Bangun

Berdasarkan rancang bangun pada Bab 3, dibuatlah sebuah *testbed* yang dapat dipakai untuk menguji kinerja dari *wireless mesh network* dengan *routing protocol* OLSR. Gambar 4.1 memperlihatkan konfigurasi keseluruhan dari *testbed* beserta perangkatnya.



Gambar 4.1. Topologi *Testbed* Jaringan *Wireless Mesh*

Dengan keterangan sebagai berikut :

Router A : IP : 192.168.3.10  
Subnet Mask : 255.255.255.0

Router B : IP Address : 192.168.3.20  
Subnet Mask : 255.255.255.0

Router C : IP Address : 192.168.3.30  
Subnet Mask : 255.255.255.0

Router D : IP Address : 192.168.3.40  
Subnet Mask : 255.255.255.0

User 1 : IP Address : 192.168.30.10  
Subnet Mask : 255.255.255.0

User 2 : IP Address : 192.168.30.20  
Subnet Mask : 255.255.255.0

Monitoring IP Address : 192.168.30.30  
Subnet Mask : 255.255.255.0

## 4.2 Pengujian Jaringan

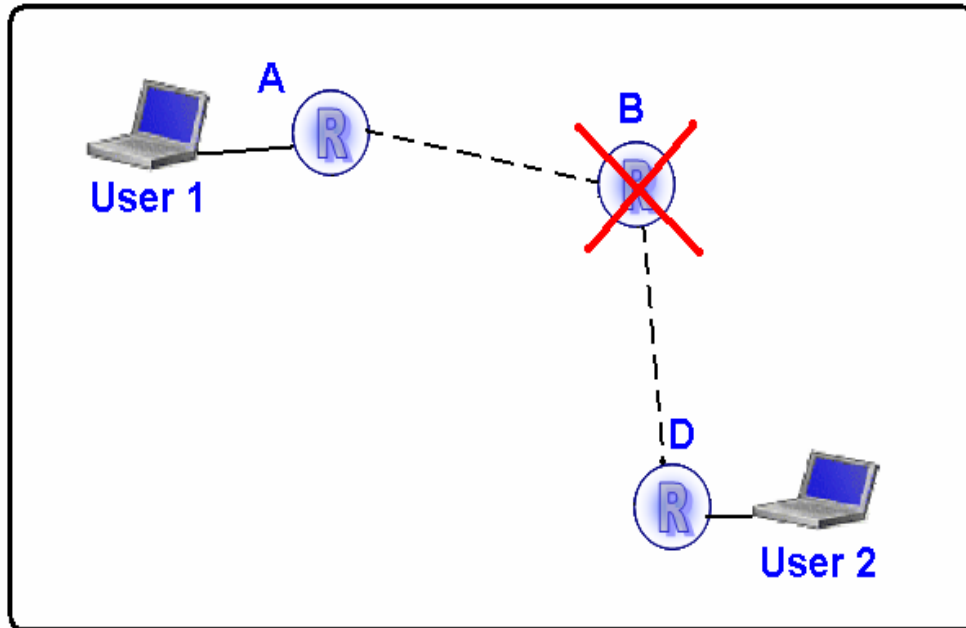
Pada testbed jaringan yang sudah dibangun dilakukan beberapa skenario pengujian seperti :

1. Skenario pengujian untuk menguji kinerja *self-configure* dan *self-healing* pada *wireless mesh network*.
2. Skenario pengujian untuk menguji pemakain *bandwidth*, *troughtput* serta *latency* dan *jitter* dari *wireless mesh network* dengan berbagai kondisi seperti :
  - a. Kondisi lingkungan dengan tingkat interferensi rendah
  - b. Kondisi lingkungan dengan tingkat interferensi tinggi
3. Skenario pengujian untuk menguji performa *end-to-end throughput* dan *jitter*.

### 4.2.1 Pengujian Self-configure

Skenario untuk pengujian kemampuan *self-configure* dari wireless mesh network digunakan untuk mendapatkan data waktu yang diperlukan suatu *mesh router* untuk melakukan *self-configure* pada dirinya sendiri dan bergabung dengan *wireless mesh network* yang sudah ada.

Adapun skenario pengujian *self-configure* pada wireless mesh network digambarkan pada gambar 4.2.



Gambar 4.2 Skenario Pengujian *Self-Configure*

Langkah-langkah yang dilakukan pada skenario pengujian *self-configure* adalah :

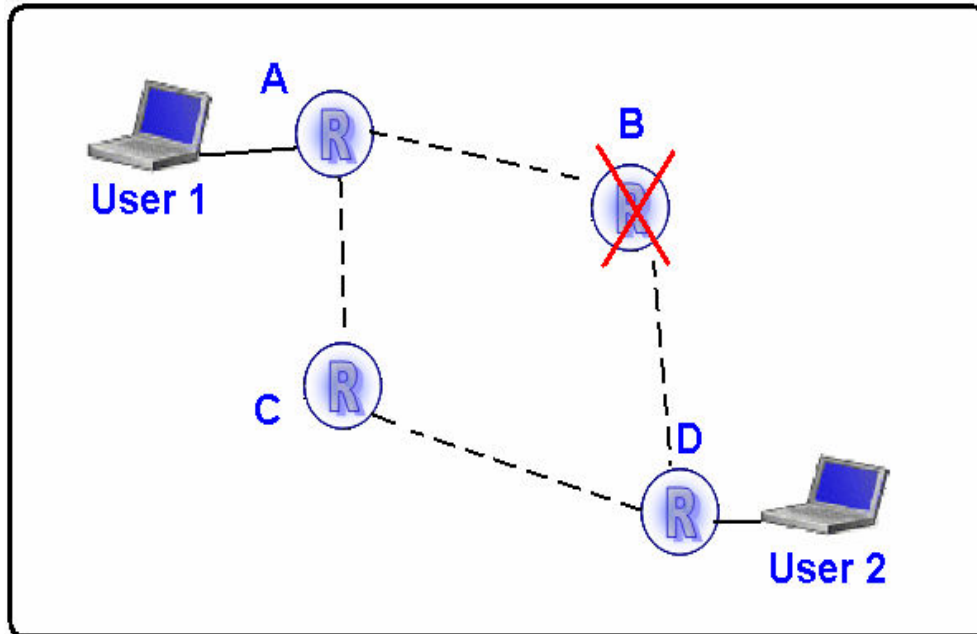
1. User 1 pada *router A* akan berusaha berhubungan dengan user 2 pada *router D* dengan melakukan ping terus menerus.
2. Karena *router D* berada diluar jangkauan *router A* maka perintah ping pun hanya akan menghasilkan Request Time Out, yang berarti tujuan tidak dapat dicapai.
3. Kemudian agar *router A* dapat berhubungan dengan *router D*, *router B* dinyalakan.
4. Dengan menyalakan *router B* maka *router* dapat berhubungan dengan *router D* melalui *router B* atau dengan kata lain dengan menggunakan sistem *multihop*.
5. Kemampuan *router B* untuk mengkonfigurasi inilah yang dikatakan sebagai kemampuan *self-configure*.

6. Data yang dicatat adalah waktu yang dibutuhkan *router B* untuk melakukan *self-configure* dengan mencatat waktu yang diperlukan agar *user 1* dapat berhubungan dengan *user 2* atau mencatat waktu mulainya respon *reply* pada *user 1* semenjak dinyalainya *router B*.

Langkah-langkah tersebut dilakukan berulang-ulang dengan merubah variabel besarnya interval dari *HELLO message* dan *TC message* dari paket *OLSR* yang dapat diubah pada *file etc/olsrd.conf*. Hasil dari pengujian *self-configure* dari *wireless mesh network* terlihat pada lampiran 1.a. Selanjutnya dengan menggunakan tambahan program aplikasi *audio-video conference*, langkah-langkah pengujian *self-configure* dilakukan ulang sehingga menghasilkan data seperti terlihat pada lampiran 1.b.

#### **4.2.2 Pengujian Self-Healing**

Skenario untuk pengujian kemampuan *self-healing* dalam *wireless mesh network* digunakan untuk mendapatkan data waktu yang diperlukan jaringan untuk mencari rute baru ataupun memperbaiki hubungan apabila terjadi kerusakan pada suatu jalur ataupun pada suatu *router*. Adapun gambaran tentang skenario untuk pengujian *self-healing* pada *wireless mesh network* ini dapat terlihat pada gambar 4.3.



Gambar 4.3 Skenario Pengujian *Self-Healing*

Langkah-langkah yang dilakukan pada skenario pengujian *self-healing* adalah :

1. *User 1* pada *router A* akan berusaha berhubungan dengan *user 2* pada *router D* dengan jalur melalui *router B* dengan melakukan ping terus menerus.
2. Kemudian *router B* dimatikan untuk mensimulasikan bahwa telah terjadi kerusakan pada *router B* sehingga jalur melalui *router B* tidak dapat dilalui sehingga ping akan mengeluarkan pesan *request time out*.
3. Protokol OLSR dengan otomatis akan mencari jalur baru melalui *router C* yang masih aktif, kemampuan inilah yang disebut dengan *self-healing*.
4. Data yang dicatat adalah waktu yang dibutuhkan *user 1* pada *router A* untuk dapat berhubungan kembali dengan *user 2* pada *router D* melalui jalur yang baru yaitu dengan munculnya kembali pesan *reply* pada perintah ping di *user 1*.

Langkah-langkah tersebut dilakukan berulang-ulang dengan merubah besarnya variabel besarnya interval dari *HELLO message* dan *TC message* dari paket OLSR yang dapat diubah pada *file etc/olsrd.conf*. Hasil dari pengujian *self-*

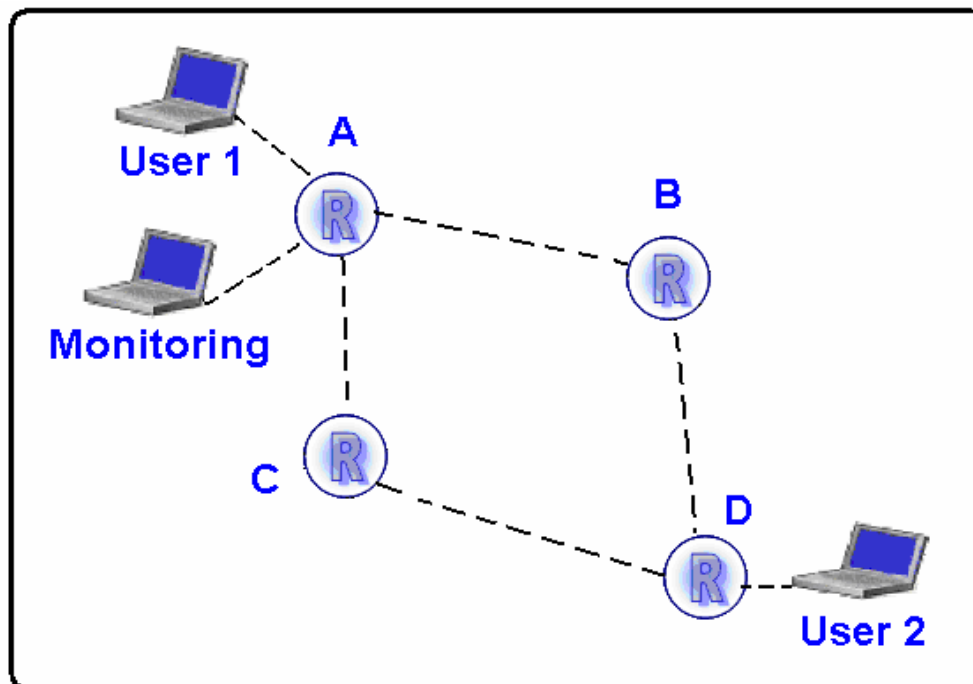
*healing* dari wireless mesh network dengan jaringan tanpa beban aplikasi terlihat pada lampiran 2.a.

Selanjutnya dengan menggunakan tambahan program aplikasi *audio-video conference*, langkah-langkah pengujian *self-healing* dilakukan ulang sehingga menghasilkan data seperti terlihat pada lampiran 2.b.

#### 4.2.3 Pengujian Penggunaan *Bandwidth*

Skenario pengujian penggunaan *bandwidth* ini berfungsi untuk mendapatkan data tentang kinerja pemakaian *bandwidth* pada masing-masing *mesh router* dalam sistem *multihop* pada *wireless mesh network*. Pengujian dilakukan dengan menggunakan beberapa alat bantu seperti aplikasi *audio-video conference*, aplikasi *monitoring* jaringan seperti PRTG dan JPERF. Selain itu pengujian dilakukan dalam kondisi yang berbeda-beda untuk melihat performa jaringan dalam berbagai kondisi tersebut.

Adapun gambaran tentang skenario untuk pengujian *self-healing* pada *wireless mesh network* ini dapat terlihat pada gambar 4.4.



Gambar 4.4 Skenario Pengujian Penggunaan *Bandwidth*.

Pengujian dilakukan dengan durasi 2 jam dimana waktu tersebut dibagi-bagi lagi menjadi tiga bagian dimana sejam pertama beban yang diberikan pada jaringan tidak terlalu besar seperti ping ataupun pengiriman data ukuran kecil, bagian kedua selama setengah jam berikutnya jaringan diberi beban *voip/audio conference*, setengah jam terakhir jaringan diberi beban *video conference*.

Langkah-langkah yang dilakukan pada skenario pengujian penggunaan *bandwidth* adalah :

1. *User 1* pada *router A* akan berusaha berhubungan dengan *user 2* pada *router D* dengan melakukan ping terus menerus.
2. *User monitoring* menjalankan aplikasi PRTG yang mengawasi dan mencatat kinerja pemakaian *bandwidth* serta *latency* dari masing-masing *router*.
3. Memasuki jam kedua jaringan diberi beban tambahan seperti aplikasi voip atau *audio conference*.
4. Tiga puluh menit kemudian jaringan diberi beban *audio-video conference*.

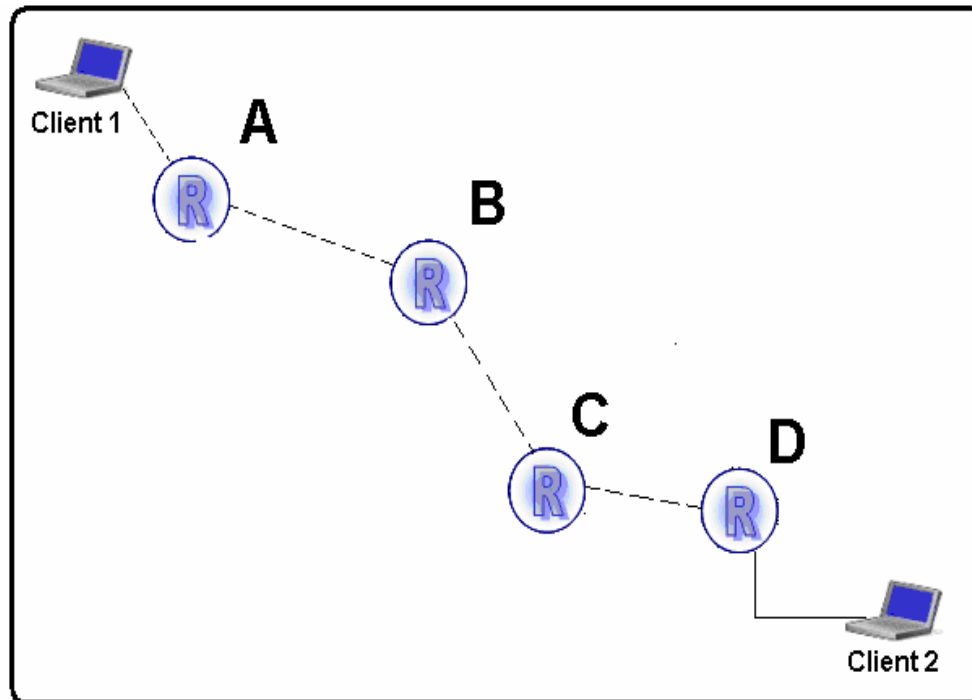
Langkah-langkah diatas dilakukan dalam dua kondisi yang berbeda, kondisi pertama adalah kondisi dimana tingkat interferensi jaringan *wireless* lain di sekitar *testbed* rendah, sedangkan yang kedua adalah keadaan dimana tingkat interferensi jaringan sedang tinggi. Hasil lengkap pengujian selengkapnya dari kondisi pertama berdasarkan pembacaan PRTG dapat dilihat pada lampiran 3.a.dan lampiran 3.b. Sedangkan pembacaan latency oleh PRTG pada setiap *router* dapat dilihat pada lampiran 4.a dan lampiran 4.b

Untuk hasil pengujian untuk kondisi kedua dimana tingkat interferensi pada jaringan tinggi dapat dilihat pada lampiran 5.a dan lampiran 5.b untuk pengamatan pemakaian *bandwidth* serta lampiran 6.a dan lampiran 6.b untuk pengamatan *latency*.

#### **4.2.4 Pengujian Sistem *Multihop***

Pengujian ini dilakukan untuk mengamati performansi sistem *multihop* dari *wireless mesh network*. Dalam skenario pengujian ini parameter yang akan

diamati adalah *throughput* serta *jitter* dari sistem *multihop*. Skenario pengujian dilakukan dengan variasi jumlah *hop* mulai dari 1 *hop* hingga 3 *hop* sesuai dengan jumlah *router* yang tersedia. Pada skenario pengujian hitungan *hop* hanya diperuntukkan hubungan antar *router* sehingga kehadiran *client* tidak diperhitungkan. Adapun topologi jaringan yang digunakan untuk skenario pengujian ini adalah seperti terlihat pada gambar 4.5.



Gambar 4.5 Skenario Pengujian Performansi Sistem *Multihop*.

Langkah-langkah yang dilakukan untuk menjalankan skenario pengujian ini adalah :

1. Menempatkan *router* sesuai dengan topologi *skenario* mulai 1 *hop* (A-B), 2 *hop* (A-B-C) hingga 3 *hop* (A-B-C-D).
2. Menjalankan program pengujian Iperf untuk mengamati *throughput* serta *jitter* dari sistem *multihop*.
3. Langkah 1 dan 2 dilakukan kembali namun kali ini dengan memberikan beban pada jaringan dengan menjalankan aplikasi *audio-video conference*. Hasil pengujian berupa grafik performansi untuk *throughput* dari sistem *multihop*



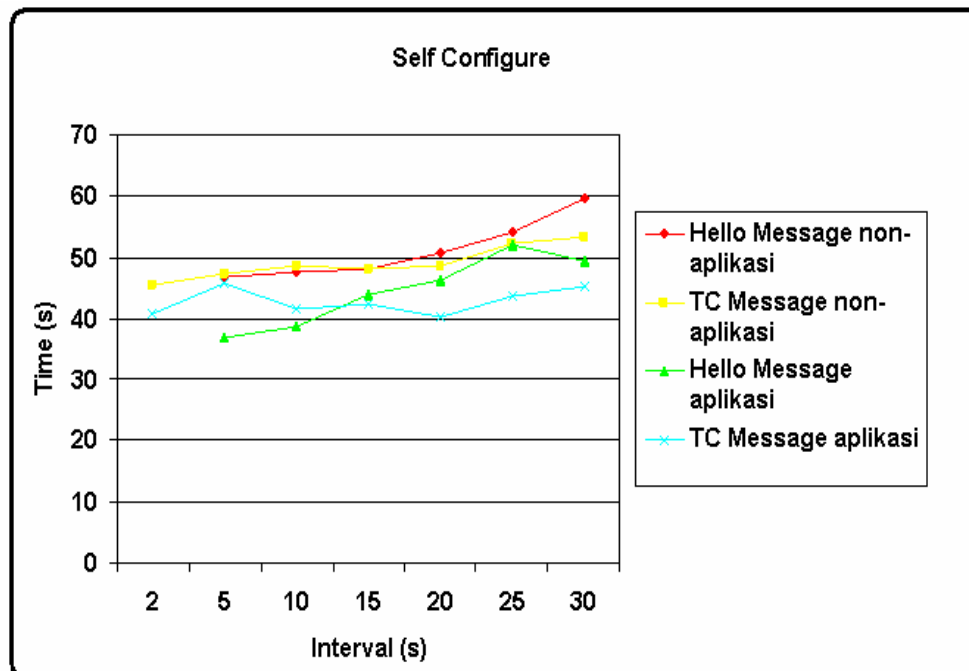
terlihat pada lampiran 7. Sedangkan hasil pengujian *jitter* dengan menggunakan Iperf dapat dilihat pada lampiran 8.

### 4.3 ANALISIS SISTEM

Dari skenario pengujian yang dilakukan pada bagian 4.2, kita dapat menganalisis pengujian yang dilakukan di atas. Analisa yang dihasilkan adalah sebagai berikut.

#### 4.3.1 Analisis Performansi *Self Configure*

Berdasarkan data kemampuan *self configure* tanpa aplikasi yang terlihat pada lampiran 1.a serta data *self-configure* dengan aplikasi yang terlihat pada lampiran 1.b didapatkan grafik perbandingan seperti pada gambar 4.6.



Gambar 4.6 Grafik Perbandingan Performansi *Self Configure*

Pada grafik perbandingan antara pengaruh parameter interval Hello message dan TC message terhadap performansi *self configure* terlihat bahwa parameter interval untuk Hello message lebih berpengaruh terhadap performansi *self-configure* jaringan mesh dibandingkan dengan parameter interval TC

*message*. Grafik *TC message* walaupun mengalami perubahan waktu performansi namun tidak terlalu signifikan seperti yang terjadi pada perubahan parameter interval *Hello message* yang menghasilkan peningkatan waktu performansi seiring dengan peningkatan nilai interval.

Informasi dari *Hello message* mempengaruhi beberapa tabel informasi pada *Local Link set* dari protokol OLSR seperti *Link set*, *Neighbor set*, *2-hop Neighbor set*, *MPR set* maupun *MPR Selector set* sedangkan *TC message* hanya mempengaruhi tabel informasi seperti *Topology set*. Berdasarkan hal tersebut parameter interval *Hello message* jelas sangat mempengaruhi performansi *self-configure* dari jaringan *mesh* karena menangani lebih banyak tabel informasi khususnya tabel informasi yang berkaitan dengan pengenalan *node* tetangga. Waktu yang dibutuhkan untuk pengenalan *node* baru akan mempengaruhi kecepatan memproses informasi *routing* dari *node* tersebut, sehingga dapat dikenali oleh *node* lain pada jaringan *mesh*.

Sedangkan *TC message* yang hanya berfungsi memberikan informasi kepada *Topology set* baru menjalankan tugasnya setelah mendapatkan informasi dari *Local Link set*. Oleh karena itu pada nilai interval *TC message* yang tidak berbeda jauh dari nilai interval *Hello Message* waktu performansi *self configure* yang didapat tidak jauh berbeda dengan waktu performansi *Hello message*.

Pada pengetesan jaringan *mesh* yang diberi beban aplikasi *audio-video conference* performansi *self-configure* baik pada parameter interval *Hello message* maupun *TC message* mengalami peningkatan meskipun tidak terlalu signifikan. Peningkatan performansi tersebut karena aplikasi *audio-video conference* yang digunakan dalam pengujian menggunakan protocol yang memakai fitur *acknowledgement* untuk menjaga stabilitas koneksi seperti *Transmission Control Protocol* (TCP) dimana pada sinyal-sinyal *acknowledgement* tersebut mengandung *header* berisi informasi-informasi mengenai alamat dari *node-node* lain yang dapat dipakai protocol OLSR sebagai informasi tambahan selain informasi alamat *node* dari *Hello message*. Informasi tambahan tersebutlah yang mempengaruhi performansi *self-configure* dalam mengenali *node* baru pada jaringan *wireless mesh*.

Adapun hasil perhitungan persentase untuk perubahan performansi *self-configure* untuk jaringan dalam kondisi tanpa beban aplikasi serta jaringan dengan beban aplikasi adalah seperti pada terlihat pada tabel 4.1.a dan tabel 4.1.b

Tabel 4.1.a Tabel Persentase Perbedaan Nilai *Self Configure* Untuk Interval *Hello Message*

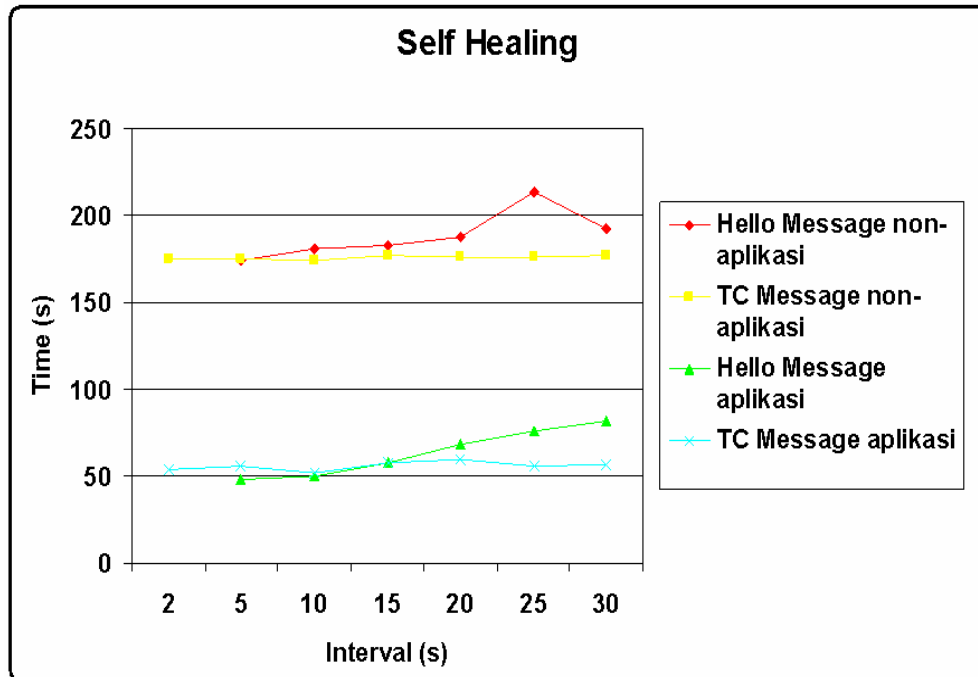
Hello Interval (s)	TC Interval (s)	Waktu tanpa aplikasi (s)	Waktu dengan aplikasi (s)	Perubahan (%)
5	2	46,68	36,78	21,2
10	2	47,64	38,74	18,68
15	2	48,08	43,88	8,73
20	2	50,60	46,32	8,45
25	2	54,14	51,88	4,17
30	2	59,68	49,42	17,19
Rata-rata				13,07 %

Tabel 4.1.b Tabel Persentase Perbedaan Nilai *Self Configure* Untuk Interval TC *Message*

Hello Interval (s)	TC Interval (s)	Waktu tanpa aplikasi (s)	Waktu dengan aplikasi (s)	Perubahan (%)
5	2	45,46	40,64	10,6
5	5	47,24	45,6	3,4
5	10	48,52	41,66	14,13
5	15	47,94	42,26	11,84
5	20	48,48	40,28	20,35
5	25	52,32	43,78	16,9
5	30	53,32	45,06	15,4
Rata-rata				13,23 %

#### 4.3.2 Analisis Performansi *Self Healing*

Untuk performansi *self-healing* berdasarkan data yang terdapat pada lampiran 2.a mengenai performansi *self-healing* jaringan tanpa aplikasi serta data yang terdapat lampiran 2.b mengenai performansi *self-healing* pada jaringan yang diberi beban aplikasi *audio-video conference* didapatkan grafik perbandingan seperti pada gambar 4.7.



Gambar 4.7 Grafik Perbandingan Performansi *Self Healing*

Pada gambar grafik perbandingan antara pengaruh parameter interval Hello message dan TC message terhadap performansi *self configure* terlihat bahwa parameter interval untuk Hello message juga lebih berpengaruh terhadap performansi *self-healing* jaringan *mesh* dibandingkan dengan parameter interval TC message walaupun perbedaan yang terjadi tidak terlalu signifikan.. Pada proses *self-healing*, hal yang paling berpengaruh adalah proses pengenalan adanya kerusakan pada jalur yang dilewati karena putusnya hubungan menuju *node* tersebut.

Putusnya suatu jalur atau hubungan pada suatu *node* dapat dideteksi dari perubahan isi pada *Neighbor set* dan *2-hop Neighbor set*, dimana perubahan tersebut terjadi akibat adanya tidak adanya update informasi tentang suatu *node* yang didapat dari Hello message. Hello message yang berhenti diterima dari suatu *node* hingga suatu batas waktu tertentu akan menandakan hubungan terhadap *node* tersebut telah hilang atau putus. Perubahan pada parameter interval TC message tidak terlalu berpengaruh pada performansi *self healing*. Berbeda dengan kondisi pada skenario *self-configure* dimana terjadi pembentukan *routing table* baru, pada kondisi skenario *self healing* setiap *node* sudah mempunyai

*routing table* masing-masing yang selalu terjaga dengan adanya informasi yang dibroadcast oleh setiap *node* sehingga pemilihan jalur baru tidak akan memakan banyak waktu kecuali harus menggunakan jalur yang benar-benar baru. Selain itu TC *message* dapat dikirimkan lebih cepat dari intervalnya sendiri apabila terjadi perubahan pada MPR *Selector set* yang dapat dianggap sebagai putusnya hubungan, dalam hal ini MPR *Selector set* sendiri dipengaruhi oleh Hello *message*.

Perbedaan yang signifikan justru terjadi pada perbandingan antara performansi *self-healing* antara jaringan tanpa beban dengan jaringan yang diberi beban aplikasi *audio-video conference*. Perbedaan ini terjadi akibat pengaruh performansi protokol OLSR yang memiliki fitur tambahan pada tabel informasi *Link set*-nya. Fitur tambahan tersebut adalah dimana OLSR dapat menangkap informasi yang terdapat pada *link layer* yang berisikan informasi konektivitas dengan *node-node* tetangga seperti informasi hilangnya konektivitas dari ketiadaan pesan *acknowledgement* pada *link layer* seperti yang terdapat pada protocol TCP.

Dengan adanya pesan-pesan *acknowledgement* pada protocol TCP, OLSR akan mendapatkan informasi tersebut sebagai informasi tambahan dari informasi yang didapatkan dari Hello *message* dari protocol OLSR itu sendiri sehingga dapat mempercepat performansi dari *self-healing* karena informasi hilangnya suatu hubungan dapat diketahui dengan lebih cepat.

Adapun hasil perhitungan persentase untuk perubahan performansi *self-healing* untuk jaringan dalam kondisi tanpa beban aplikasi serta jaringan dengan beban aplikasi adalah seperti pada terlihat pada tabel 4.2.a dan tabel 4.2.b

Tabel 4.2.a Persentase Perbedaan Nilai *Self Healing* Untuk Interval *HelloMessage*

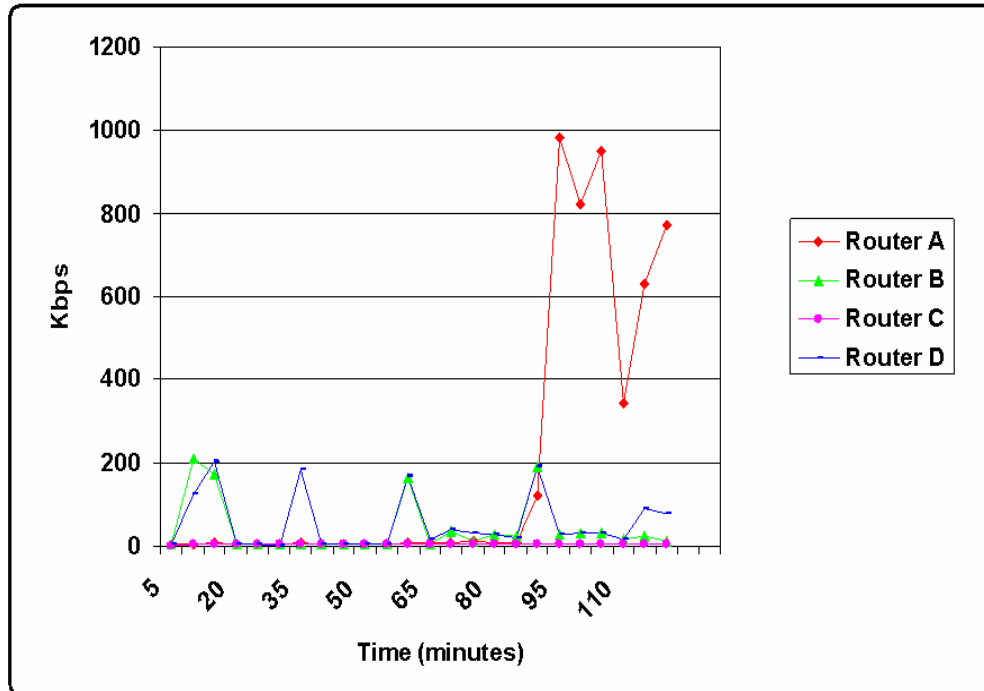
Hello Interval (s)	TC Interval (s)	Waktu tanpa aplikasi (s)	Waktu dengan aplikasi (s)	Perubahan (%)
5	2	173,6	48,2	72,23
10	2	180,4	50,04	72,26
15	2	182,4	57,98	68,21
20	2	187,8	68,62	63,46
25	2	213,2	76,02	64,34
30	2	192,4	81,82	57,53
<b>Rata-rata</b>				<b>66,33 %</b>

Tabel 4.2.b Persentase Perbedaan Nilai *Self Healing* Untuk Interval *TC Message*

Hello Interval (s)	TC Interval (s)	Waktu tanpa aplikasi (s)	Waktu dengan aplikasi (s)	Perubahan (%)
5	2	175	53,98	69,15
5	5	174,6	55,94	67,96
5	10	174,4	52,22	70,05
5	15	176,8	57,42	67,52
5	20	175,8	59,72	66,02
5	25	176,2	55,72	68,76
5	30	177,4	57,18	67,76
<b>Rata-rata</b>				<b>68,17 %</b>

#### 4.3.3 Analisis Performansi Pemakaian *Bandwidth*

Pada pengujian ini dilakukan dengan beberapa kondisi jaringan seperti adanya interferensi dari jaringan *wireless* yang ada disekitar *testbed* hingga pemberian beban aplikasi pada jaringan seperti aplikasi *chatting* serta *audio-video conference*. Hal tersebut dilakukan untuk menguji performa *wireless mesh network* dalam berbagai kondisi tersebut. Berdasarkan pembacaan PRTG dari pada lampiran 3.a dan lampiran 3.b setiap *router* didapatkan grafik perbandingan seperti pada gambar 4.8.



Gambar 4.8 Grafik Perbandingan Pemakaian *Bandwidth* Untuk Kondisi Tingkat Interferensi Rendah

Pengamatan pertama adalah kemampuan jaringan *wireless mesh* dalam menjalankan berbagai aplikasi pada saat kepadatan tingkat interferensi rendah. Pada gambar 4.8 tentang grafik perbandingan pemakaian *bandwidth* terlihat bahwa jalur yang terpakai adalah jalur yang melalui *router B* karena aktifitas *router C* tidak terlalu besar. Pada setiap *router* yang dilewati seperti *router A*, *B* dan *D* terlihat bahwa terjadi peningkatan pemakaian *bandwidth* pada setiap saat dijalankannya aplikasi secara bertahap. Untuk satu jam pertama dimana tidak banyak aplikasi yang digunakan yaitu hanya penggunaan aplikasi pengiriman teks atau *chatting*, pemakaian *bandwidth* yang terlihat belum terlalu besar, kemudian setelah jaringan mulai diberi beban audio conference pada menit ke 60 aktifitas pemakaian *bandwidth* pada *router* yang dilalui mulai menanjak hingga mencapai puncaknya mulai sekitar menit ke 90 setelah jaringan diberi beban *audio-video conference*.

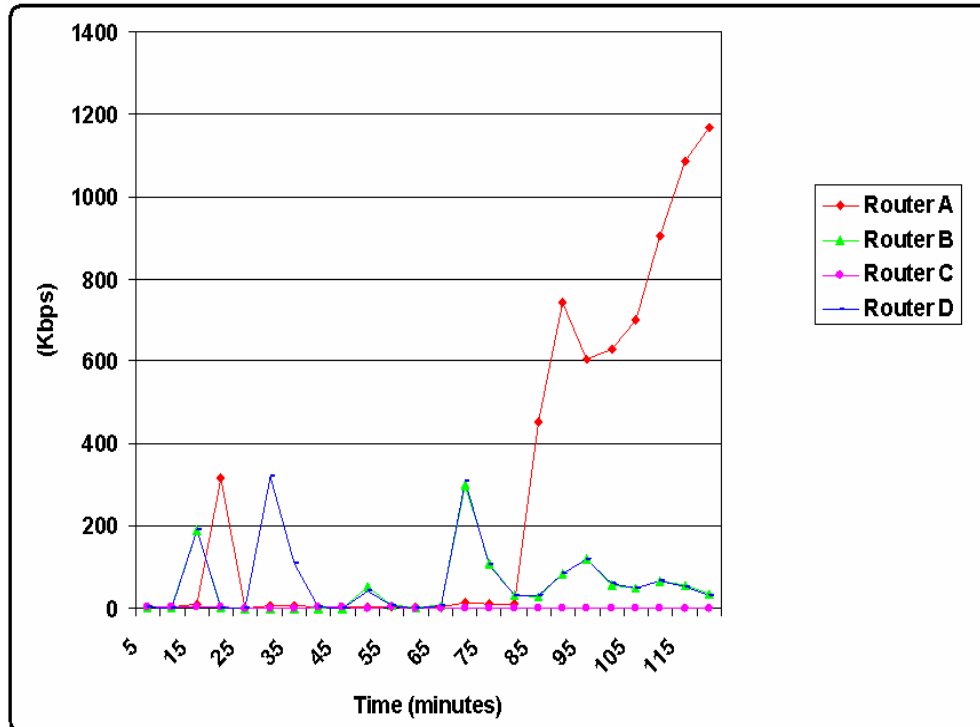
Pada grafik, *router A* sebagai *gateway* dari *user* pengirim memiliki aktifitas pemakaian *bandwidth* paling besar hingga mencapai puncaknya dengan

982.18 Kbps pada saat jaringan diberi beban *audio-video conference*, diikuti oleh aktifitas pada *router B* dengan nilai maksimum *outgoing* 208,507 Kbps dan *router D* dengan nilai 202,768 Kbps dan *router C* yang tidak dilalui jalur dengan nilai maksimum *outgoing* 3.304 Kbps.

Terjadi fenomena overshoot pada *router A* dikarenakan *router A* digunakan sebagai gateway oleh dua user yaitu user pengirim aplikasi serta user yang digunakan sebagai server monitoring. Dari banyaknya paket yang dikirimkan oleh kedua user tersebut, terutama dari user pengirim aplikasi maka akan mengakibatkan aktifitas pemakaian bandwidth yang tinggi pada *router A*. Selain itu karena program aplikasi *audio-video conference* yang digunakan menggunakan protocol TCP maka semakin banyaknya paket yang melewati *router A* maka makin besar kemungkinan terjadi collision yang dapat menyebabkan user pengirim aplikasi mengirimkan kembali paket yang hilang karena tidak adanya pesan acknowledgement yang diterima dari user penerima. Hal tersebut akan semakin memperbesar aktifitas pemakaian bandwidth pada user A

Pengamatan kedua adalah data pengujian jaringan dalam kondisi kepadatan *user* tinggi atau tingkat interferensi yang tinggi. Grafik perbandingan dari aktifitas pemakaian *bandwidth* untuk setiap *router* pada kondisi interferensi tinggi berdasarkan data pada lampiran 5.a dan lampiran 5.b dapat dilihat pada gambar 4.9.

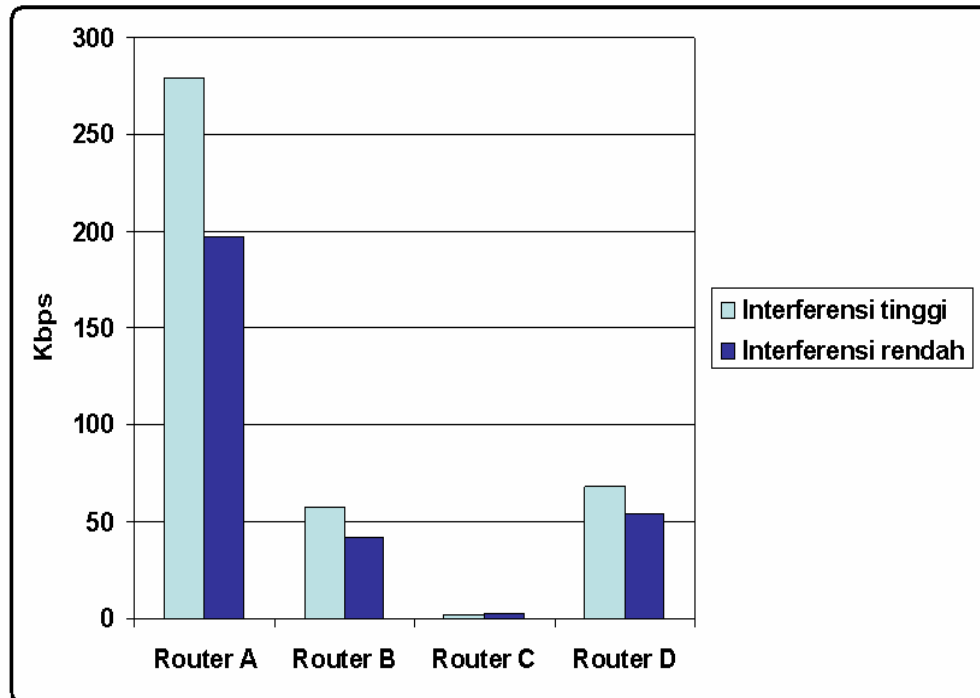




Gambar 4.9 Grafik Perbandingan Pemakaian *Bandwidth* Untuk Kondisi Tingkat Interferensi Tinggi.

Dengan pengaruh dari tingkat interferensi yang tinggi dari *user-user* pengguna laptop serta *router-router* lain yang ada disekitar lokasi *testbed*, jaringan *wireless mesh* tetap mampu melayani kebutuhan pemakaian *bandwidth* untuk setiap aplikasi yang diujikan meskipun mengalami peningkatan pemakaian *bandwidth* di setiap *router*. Pada grafik, *router A* sebagai *gateway* dari *user* pengirim memiliki aktifitas pemakaian *bandwidth* paling besar hingga mencapai puncaknya dengan 1169.139 Kbps , diikuti oleh aktifitas pada *router B* dengan nilai maksimum *outgoing* 208,507 Kbps dan *router D* dengan nilai 202,768 Kbps dan *router C* yang tidak dilalui jalur dengan nilai maksimum *outgoing* 3.304 Kbps diikuti dengan *router B* pada 231,511 Kbps, *router C* dengan nilai 2,351 Kbps serta *router D* dengan nilai 321,05 Kbps.

Grafik perbandingan antara rata-rata pemakaian *bandwidth* pada setiap *router* pada berdasarkan tingkat interferensi dapat dilihat pada gambar 4.10.



Gambar 4.10 Grafik Perbandingan Pemakaian *Bandwidth* Berdasarkan Tingkat Interferensi

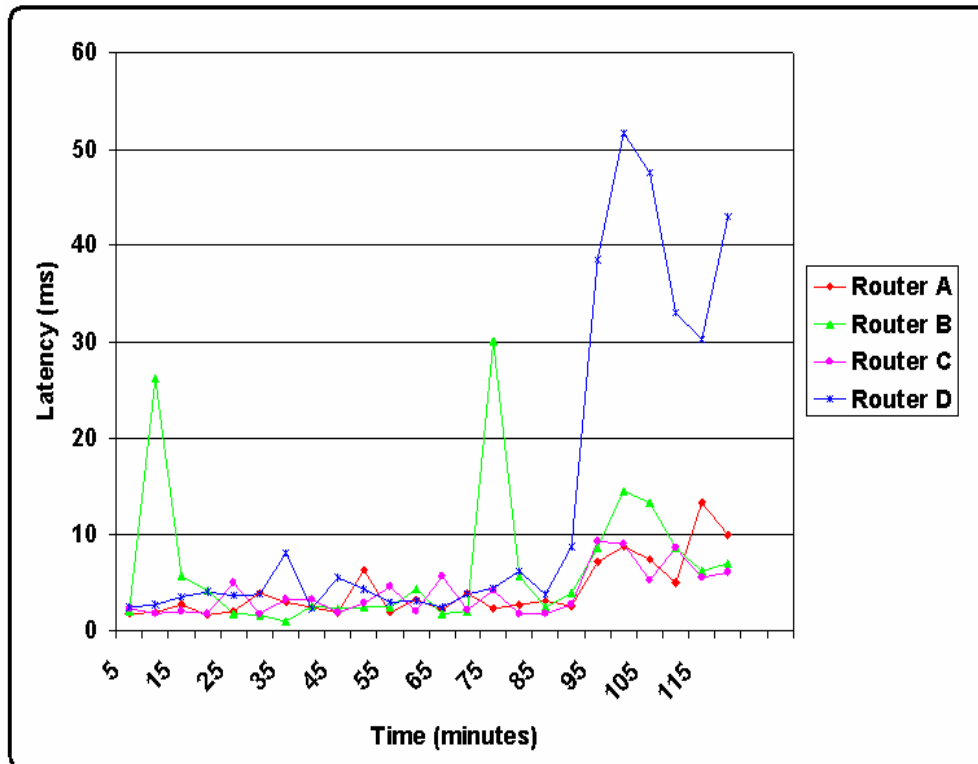
Dengan pengaruh dari tingkat interferensi yang tinggi dari *user-user* pengguna *laptop* serta *router-router* lain yang ada disekitar lokasi *testbed*, jaringan *wireless mesh* tetap mampu melayani kebutuhan pemakaian *bandwidth* untuk setiap aplikasi yang diujikan meskipun mengalami peningkatan pemakaian *bandwidth* di setiap *router*. Secara persentase peningkatan pemakaian *bandwidth* yang terjadi pada setiap *router* adalah:

- Router A =  $(278,804 - 196,685) / 196,685 \times 100 \% = 41,75 \%$
- Router B =  $(57,418 - 41,673) / 41,673 \times 100 \% = 37,78 \%$
- Router C =  $(1,472 - 2,681) / 1,472 \times 100 \% = - 81 \%$
- Router D =  $(67,819 - 54,108) / 67,819 \times 100 \% = 20,21 \%$

#### 4.3.4 Analisis Performansi *Latency*

Pengujian *latency* untuk setiap node *router* juga dilakukan dengan skenario yang sama dimana setiap *router* akan diamati aktifitas *latency* terhadap kondisi-kondisi yang berbeda-beda. Grafik perbandingan *latency* yang terjadi

pada setiap *router* saat kondisi jaringan pada tingkat interferensi rendah berdasarkan data dari lampiran 4.a dan lampiran 4.b dapat terlihat pada gambar 4.11.



Gambar 4.11 Grafik Perbandingan *Latency* Untuk Kondisi Tingkat Interferensi Rendah.

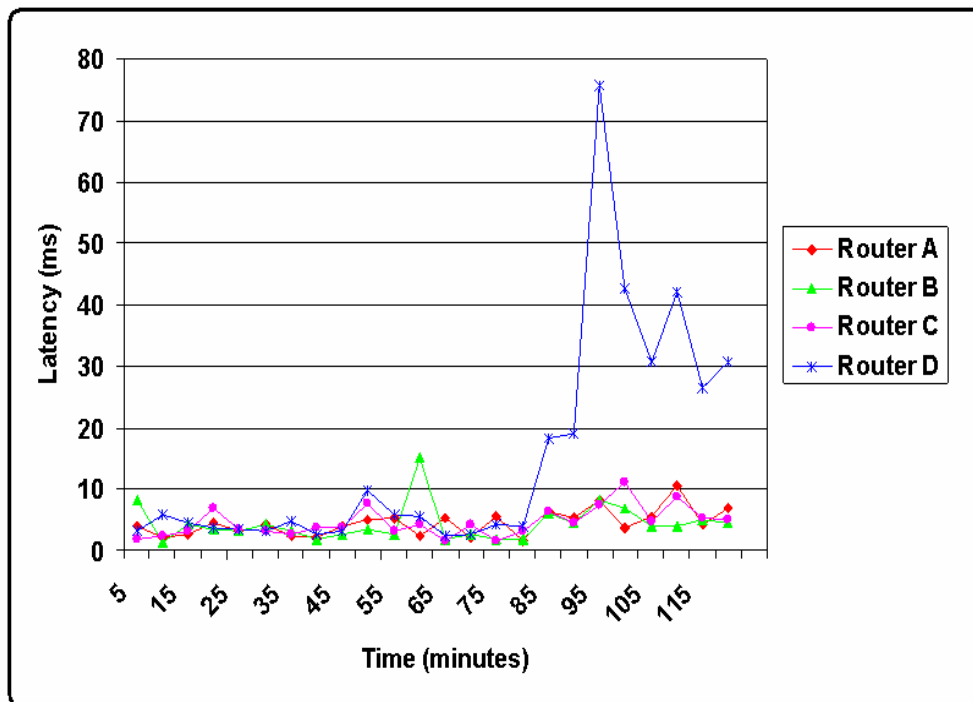
Pengamatan pertama yaitu pada kondisi dimana tingkat interferensi rendah, pada grafik terlihat bahwa terjadi peningkatan *latency* terhadap beban aplikasi yang diberikan pada jaringan. Pada 60 menit pertama dimana jaringan hanya diberi beban aplikasi *chatting* aktifitas *latency* pada setiap *router* cenderung kecil. Namun saat jaringan diberikan beban yang lebih besar seperti *audio-video conference* pada 60 menit kedua maka aktifitas *latency* terlihat meningkat.

Aktifitas *latency* paling besar terdapat pada *router* D dimana mencapai puncaknya dengan nilai 51,65 ms yaitu pada saat jaringan diberi beban *audio-video conference*. Diikuti oleh *router* B yang mempunyai nilai *latency* maksimum

30,08 ms, *router A* dengan nilai maksimum 13,317 ms serta *router C* dengan nilai maksimum 9,233 ms, yang kesemuanya terjadi pada saat jaringan diberi beban *audio-video conference*.

Fenomena overshoot nilai latency pada *router D* terjadi karena letak *router D* yang terletak paling jauh dari jarak *router-router* lainnya yang masih berada dalam satu ruangan dengan user pengirim maupun monitoring.

Pengamatan kedua adalah pengamatan aktifitas *latency* pada setiap *node router* pada saat kondisi jaringan mengalami tingkat interferensi yang tinggi. Grafik perbandingan dari aktifitas *latency* untuk setiap *router* berdasarkan data yang ada pada lampiran 6.a dan lampiran 6.b dapat terlihat pada gambar 4.12.



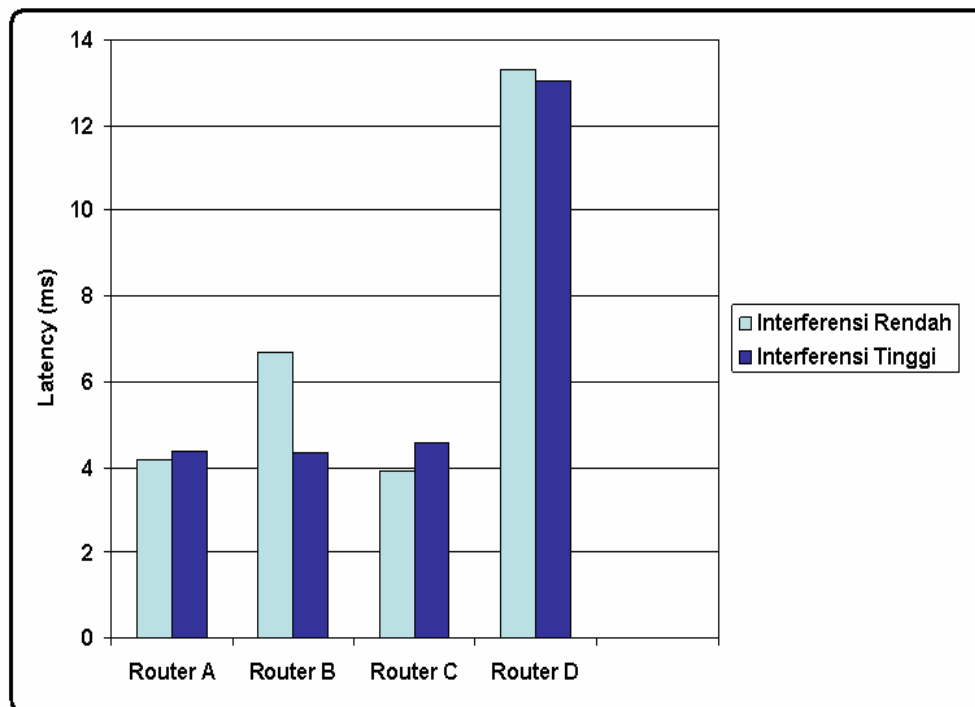
Gambar 4.12 Grafik Perbandingan *Latency* Untuk Kondisi Tingkat Interferensi Tinggi

Kondisi yang sama terjadi pada saat pengamatan pertama yaitu aktifitas *latency* untuk setiap *router* bertambah pada saat dijalankannya aplikasi yang berbeda. Grafik menunjukkan aktifitas *latency* untuk setiap *router* cenderung meningkat pada menit 60 keatas yaitu pada saat jaringan diberi beban *audio-video*

*conference*. Namun akibat adanya interferensi pada beberapa waktu tertentu nilai dari *latency* memiliki nilai yang lebih besar dibandingkan pada saat kondisi jaringan dengan tingkat interferensi yang rendah.

Aktivitas *latency* paling besar terdapat pada router D yang mencapai nilai puncaknya dengan nilai 75,883 ms yaitu pada saat jaringan diberi beban *audio-video conference*. Diikuti oleh *router B* yang mempunyai nilai *latency* maksimum 15,2 ms, *router A* dengan nilai maksimum 10,533 ms serta *router C* dengan nilai maksimum 11,05 ms, yang kesemuanya juga terjadi pada saat jaringan diberi beban *audio-video conference*.

Namun apabila dilihat secara rata-rata perbandingan nilai *latency* pada saat jaringan pada kondisi interferensi rendah maupun tinggi mempunyai nilai yang bervariasi untuk setiap *router* seperti terlihat pada gambar 4.13.



Gambar 4.13 Grafik Perbandingan *Latency* Berdasarkan Tingkat Interferensi

Seperti terlihat pada gambar 4.13 dimana grafik menunjukkan walaupun terjadi perbedaan nilai maksimum pada setiap *router* untuk kondisi tingkat interferensi yang berbeda, namun secara rata-rata keseluruhan pengaruh

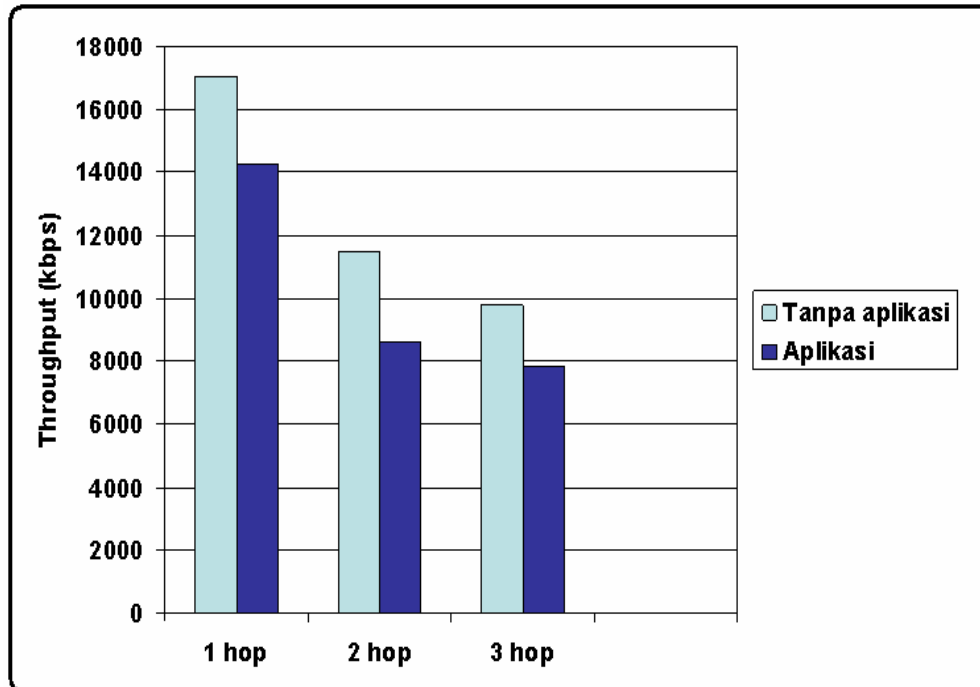
interferensi tidak menunjukkan perubahan yang signifikan bahkan pada *router* B dan *router* D nilai latency cenderung menurun. Apabila dihitung secara persentase maka perbedaan nilai rata-rata latency pada setiap router adalah sebagai berikut :

- Router A =  $(4,383 - 4,198) / 4,383 \times 100 \% = 4,22 \%$
- Router B =  $(4,363 - 6,682) / 4,363 \times 100 \% = (-) 53,15 \%$
- Router C =  $(4,565 - 3,913) / 4,565 \times 100 \% = 14,28 \%$
- Router D =  $(13,049 - 13,307) / 13,049 \times 100 \% = (-) 1,97 \%$

Walaupun begitu dari grafik ini pula terlihat bahwa pengaruh jarak atau jumlah *hop* yang dilalui pada suatu jalur memiliki pengaruh yang cukup signifikan untuk setiap *router* dimana nilai *latency* bertambah dengan bertambahnya *jumlah hop*. Dalam hal ini jalur yang dilewati oleh data adalah jalur yang melewati *router* A, B dan D, pada grafik tiap *router* mengalami kenaikan nilai *latency* seiring dengan jumlah *hop* yang dilewati.

#### **4.3.5 Analisis Performansi *End-to-End Throughput***

Pada pengujian ini dilakukan skenario untuk mengamati performansi sistem *multihop* yang terbentuk pada jaringan *wireless mesh*, dimana parameter yang diambil adalah kemampuan *end-to-end throughput*. Dari nilai rata-rata *throughput* yang didapat dari data pada lampiran 7 didapatkan grafik perbandingan *throughput* untuk setiap jumlah hop yang dapat terlihat dari gambar 4.14.



Gambar 4.14 Grafik Perbandingan *End-to-End Throughput* Sistem *Multihop*.

Berdasarkan grafik pada gambar 4.14, terlihat bahwa performansi *end-to-end throughput* akan terus menurun seiring dengan bertambahnya jumlah *hop* dari jalur yang terbentuk pada jaringan *wireless mesh*. Pada keadaan jaringan tanpa beban aplikasi, perhitungan penurunan performansi *end-to-end throughput* dalam persentase adalah sebagai berikut :

$$- 1 \text{ hop} \rightarrow 2 \text{ hop} = (17.052 - 11.520) / 17.502 \times 100 \% = 31,6 \%$$

$$- 2 \text{ hop} \rightarrow 3 \text{ hop} = (11.520 - 9.787) / 11.520 \times 100 \% = 15,04 \%$$

Sedangkan untuk keadaan jaringan dengan pemberian beban audio-video conference persentase dari penurunan performansi throughput adalah sebagai berikut :

$$- 1 \text{ hop} \rightarrow 2 \text{ hop} = (14.263 - 8.638) / 14.263 \times 100 \% = 39,43 \%$$

$$- 2 \text{ hop} \rightarrow 3 \text{ hop} = (8.638 - 7.830) / 8628 \times 100 \% = 9,35 \%$$

Dari hasil perhitungan tersebut terlihat kecenderungan turunnya nilai *throughput* berdasarkan dari bertambahnya jumlah *hop*. Hasil ini sebanding nilai pemakaian *bandwidth* atau *throughput* per *node* yang terdapat pada analisa 4.3.3

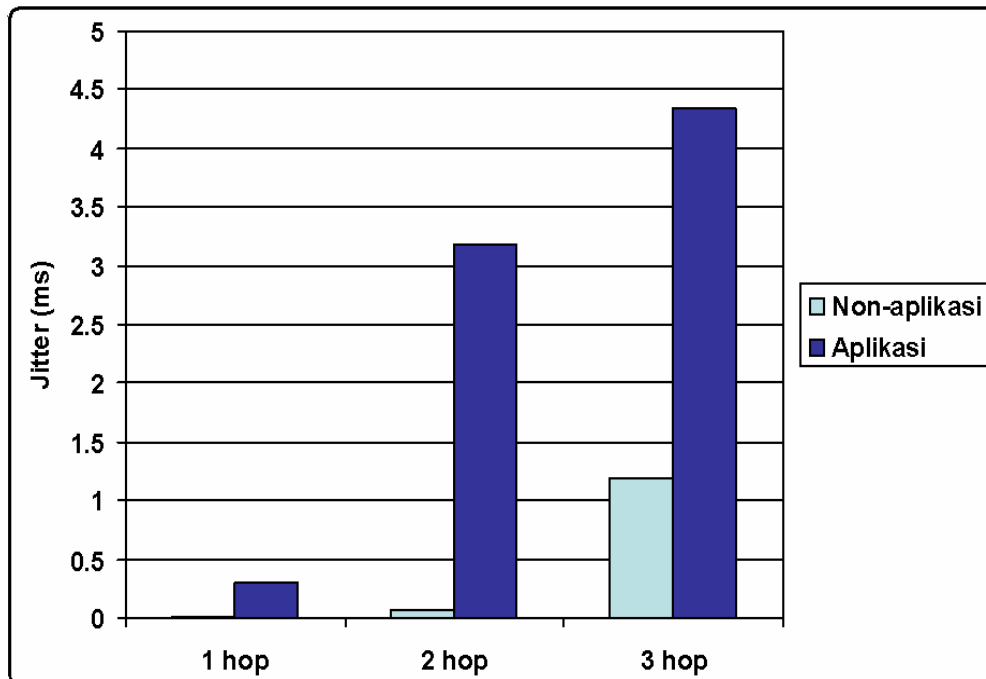
dimana pemakaian *bandwidth* setiap *node* cenderung berubah untuk setiap *node* yang dilalui jalur pengiriman data.

Selain itu dengan bertambahnya beban aplikasi pada jaringan juga terlihat berpengaruh terhadap performansi *end-to-end throughput* pada setiap perbedaan jumlah *hop*. Dengan bertambahnya beban aplikasi *audio-video conference* performansi *throughput* terlihat berkurang untuk setiap perbedaan jumlah *hop*. Apabila dihitung secara persentase maka perbedaan nilai rata-rata *throughput* pada setiap *router* adalah sebagai berikut :

- *Throughput 1 hop* =  $(17.052 - 14.263) / 17.502 \times 100 \% = 18,5 \%$
- *Throughput 2 hop* =  $(11.502 - 8638) / 11.502 \times 100 \% = 21,07 \%$
- *Throughput 3 hop* =  $(9787 - 7830) / 9787 \times 100 \% = 19,99 \%$

#### 4.3.6 Analisis Performansi Jitter

Pada pengujian ini dilakukan skenario untuk mengamati performansi *end-to-end jitter* untuk sistem *multihop* yang terbentuk pada sebuah jaringan *wireless mesh*. Dari nilai rata-rata *jitter* yang didapat dari data pada lampiran 8 didapatkan grafik perbandingan *jitter* untuk setiap jumlah *hop* yang dapat terlihat dari gambar 4.15.



Gambar 4.15 Grafik Perbandingan *End-to-End Jitter* Sistem *Multihop*



Berdasarkan grafik pada gambar 4.15, terlihat bahwa nilai *jitter* terus bertambah seiring dengan bertambahnya jumlah *hop* yang dilalui. Perhitungan persentase untuk kenaikan nilai *jitter* untuk setiap bertambahnya jumlah *hop* pada jaringan tanpa beban adalah sebagai berikut :

$$- 1 \text{ hop} \rightarrow 2 \text{ hop} = (0,068 - 0,014) / 0,014 \times 100 \% = 385,71 \%$$

$$- 2 \text{ hop} \rightarrow 3 \text{ hop} = (1,192 - 0,068) / 0,068 \times 100 \% = 1652,94 \%$$

Sedangkan untuk keadaan jaringan dengan pemberian beban *audio-video conference* persentase dari penurunan performansi *throughput* adalah sebagai berikut :

$$- 1 \text{ hop} \rightarrow 2 \text{ hop} = (3,19 - 0,302) / 0,302 \times 100 \% = 956,29 \%$$

$$- 2 \text{ hop} \rightarrow 3 \text{ hop} = (4,35 - 3,19) / 3,19 \times 100 \% = 36,36 \%$$

Dari hasil perhitungan tersebut terlihat kecenderungan naiknya nilai *jitter* berdasarkan dari bertambahnya jumlah *hop*. Hasil ini sebanding dengan nilai *latency per node* yang terdapat pada analisa 4.3.4 dimana besarnya nilai *latency* setiap *node* cenderung bertambah untuk setiap *node* yang dilalui jalur pengiriman data.

Bertambahnya beban dari jaringan setelah diberi beban *audio-video conference* juga terlihat berpengaruh pada nilai *end-to-end jitter* pada jaringan *wireless mesh*. Dengan bertambahnya beban aplikasi *audio-video conference* performansi *end-to-end jitter* terlihat bertambah untuk setiap bertambahnya jumlah *hop*. Apabila dihitung secara persentase maka perbedaan nilai rata-rata *jitter* pada setiap *router* adalah sebagai berikut :

$$- \text{Jitter } 1 \text{ hop} = (0,032 - 0,014) / 0,014 \times 100 \% = 128,57 \%$$

$$- \text{Jitter } 2 \text{ hop} = (3,19 - 0,068) / 0,068 \times 100 \% = 4591,17 \%$$

$$- \text{Jitter } 3 \text{ hop} = (4,34 - 1,192) / 1,192 \times 100 \% = 264,09 \%$$

## **BAB V**

### **KESIMPULAN DAN SARAN**

Berdasarkan hasil pengujian dan analisis sistem pada Bab 4, dapat diambil beberapa kesimpulan sebagai berikut:

1. Nilai interval Hello message lebih berpengaruh terhadap performansi self configure pada jaringan wireless mesh daripada nilai interval TC message. Pemberian beban aplikasi pada jaringan juga mempengaruhi performansi self configure dimana terjadi kenaikan 13,07 % untuk perubahan interval Hello message dan kenaikan 13,23 % untuk perubahan interval TC message.
2. Nilai interval Hello message juga lebih berpengaruh terhadap performansi self healing. Pemberian beban aplikasi pada jaringan juga mempengaruhi performansi self healing dimana terjadi kenaikan 66,33 % untuk perubahan interval Hello message dan kenaikan 68,17 % untuk perubahan interval TC message.
3. Untuk pemakaian bandwidth serta latency untuk setiap node mesh router performannya dipengaruhi oleh kedudukannya dalam jalur multihop yang dilalui oleh pengiriman data. Selain itu gangguan interferensi juga memberikan pengaruh untuk performa setiap wireless mesh router .
4. Nilai end-to-end throughput untuk sistem multihop serta besarnya data atau beban aplikasi pada jaringan wireless mesh sangat dipengaruhi dari jumlah hop yang dilalui jalur data serta besarnya data atau beban aplikasi, dimana nilai rata-rata throughput paling besar didapatkan pada jalur 1 hop yaitu 17.052 Kbps. Seiring dengan bertambahnya jumlah hop, performansi end-to-end throughput juga semakin berkurang.
5. Nilai end-to-end jitter juga dipengaruhi jumlah hop yang dilalui serta besarnya data yang dikirimkan melalui jalur data. Dimana seiring dengan bertambahnya jumlah hop maka nilai jitter yang didapatkan

juga semakin besar. Nilai jitter paling kecil didapatkan pada keadaan 1 hop dengan nilai 0,014 ms.

6. Untuk mendapatkan data performansi yang lebih baik pengujian bisa dilakukan dengan menggunakan lebih banyak node sehingga kinerja protocol OLSR yang menggunakan sistem flooding akan lebih terlihat.

## Daftar Referensi

- [1] Zhang Y, Luo J, Hu H, *Wireless Mesh Networking : Architecture, Protocols and Standard* (Boca Raton : Aurbach Publications, 2007)
- [2] B.H Walke, S. Mangold, L. Berlemann, *IEEE 802 Wireless Systems : Protocols, Multi-Hop Mesh/Relaying, Performance and Spectrum Coexistence* (Chicester : John Wiley & Sons Ltd, 2006)
- [3] Gilbert Held, *Wireless Mesh Networks* (Boca Raton : Aurbach Publications, 2005)
- [4] E. Hosain, K.K. Leung, *Wireless Mesh Network : Architecture and Protocols* (New York : Springer, 2008)
- [5] Ian F. Akyldiz, Xudong Wang, Weilin Wang, “Wireless mesh networks: a survey.” Diakses 21 Juni 2008.  
[http://users.ece.gatech.edu/~wxudong/Xudong\\_Wang\\_WirelessMesh\\_COMNET\\_0305.pdf](http://users.ece.gatech.edu/~wxudong/Xudong_Wang_WirelessMesh_COMNET_0305.pdf)
- [6] Thomas B. Krag, “Introduction to Wireless Mesh” Presentasi, diakses 23 Juni 2008.  
<http://www.wireless.dk>
- [7] Khrisna N, Ramachandran et al., “On the Design and Implementation of Infrastructure Mesh Network”
- [8] [http://en.wikipedia.org/wiki/Wireless\\_mesh\\_network](http://en.wikipedia.org/wiki/Wireless_mesh_network). Diakses 20 Juni 2008
- [9] S. Ahson, M. Ilyas, *Wimax Application* (Boca Raton : CRC Press, 2008)

- [10] Andrew S. Tanenbaum, *Computer Network* (New Jersey : Prentice Hall, 2003)
- [11] <http://en.wikipedia.org/wiki/OLSR> . Diakses 23 Juni 2008
- [12] T. Clausen, P. Jaquet, “Optimized Link State Protocol (OLSR)” RFC 3626
- [13] Andreas Tonnesen “Implementing and Extending the Optimized Link State Protocol” Jurnal, diakses 20 Juni 2008.  
<http://www.olsr.org>
- [14] P. Jaquet “Optimized Link State Routing Protocol for Ad-hoc Networks”
- [15] <http://en.wikipedia.org/wiki/OpenWRT>. Diakses 20 Juni 2008
- [16] Paul A, Larry P, Linksys WRT54G Ultimate Hacking (Burlington : Syngress Publishing, 2007)
- [17] <http://www.openwrt.org/docs>. Diakses 20 Juni 2008
- [18] Zhenyu Y, Chandrakanth C, Haiyun L, “Bandwidth Measurement in Wireless Mesh Network”

**DAFTAR PUSTAKA**

- Zhang Y, Luo J, Hu H, *Wireless Mesh Networking : Architecture, Protocols and Standard* (Boca Raton : Aurbach Publications, 2007)
- B.H Walke, S. Mangold, L. Berlemann, *IEEE 802 Wireless Systems : Protocols, Multi Hop Mesh/Relaying, Performance and Spectrum Coexistence* (Chicester : John Wiley & Sons Ltd, 2006)
- Gilbert Held, *Wireless Mesh Networks* (Boca Raton : Aurbach Publications, 2005)
- E. Hosain, K.K. Leung, *Wireless Mesh Network : Architecture and Protocols* (New York : Springer, 2008)
- Ian F. Akyldiz, Xudong Wang, Weilin Wang, "Wireless mesh networks: a survey."  
Diakses 21 Juni 2008.  
[http://users.ece.gatech.edu/~wxudong/Xudong\\_Wang\\_WirelessMesh\\_CO\\_MNET\\_0305.pdf](http://users.ece.gatech.edu/~wxudong/Xudong_Wang_WirelessMesh_CO_MNET_0305.pdf)
- Paul A, Larry P, Linksys WRT54G Ultimate Hacking (Burlington : Syngress Publishing, 2007)

### Lampiran 1.a : Tabel Performansi Self Configure Tanpa Aplikasi

Tabel hasil pengujian *self-configure* berdasarkan parameter *Hello Message*

Hello Interval (s)	TC Interval (s)	Time (s)
5	2	50,5
5	2	45,7
5	2	41,2
5	2	46,1
5	2	49,9
<b>Rata-rata =</b>		<b>46,68</b>
10	2	48,6
10	2	46,2
10	2	45,0
10	2	47,6
10	2	50,8
<b>Rata-rata =</b>		<b>47,64</b>
15	2	52,2
15	2	45,7
15	2	50,2
15	2	48,1
15	2	44,2
<b>Rata-rata =</b>		<b>48,08</b>
20	2	60,2
20	2	43,2
20	2	42,6
20	2	52,9
20	2	56,1
<b>Rata-rata =</b>		<b>50,60</b>
25	2	58,3
25	2	48,7
25	2	43,9
25	2	53,4
25	2	60,4
<b>Rata-rata =</b>		<b>54,14</b>
30	2	50,2
30	2	76,2
30	2	50,5
30	2	70,1
30	2	51,3
<b>Rata-rata =</b>		<b>59,68</b>

(Lanjutan)

Tabel hasil pengujian *self-configure* berdasarkan parameter TC *Message*

Hello Interval (s)	TC Interval (s)	Time (s)
5	2	49,7
5	2	50,2
5	2	42,3
5	2	40,6
5	2	44,5
<b>Rata-rata =</b>		<b>45,46</b>
5	5	51,5
5	5	42,3
5	5	47,6
5	5	46,2
5	5	48,6
<b>Rata-rata =</b>		<b>47,24</b>
5	10	47,8
5	10	48,5
5	10	51,9
5	10	40,3
5	10	54,1
<b>Rata-rata =</b>		<b>48,52</b>
5	15	43,7
5	15	42,5
5	15	54,3
5	15	43,3
5	15	55,9
<b>Rata-rata =</b>		<b>47,94</b>
5	20	45,2
5	20	50,4
5	20	45,5
5	20	47,1
5	20	54,2
<b>Rata-rata =</b>		<b>48,48</b>
5	25	58,3
5	25	45,1
5	25	61,0
5	25	42,5
5	25	54,7
<b>Rata-rata =</b>		<b>52,32</b>
5	30	48,8
5	30	50,5
5	30	48,2
5	30	51,4
5	30	48,1
<b>Rata-rata =</b>		<b>53,32</b>



### Lampiran 1.b : Tabel Performansi Self Configure Dengan Aplikasi

Tabel hasil pengujian *self-configure* berdasarkan parameter *Hello Message* dengan aplikasi

Hello Interval (s)	TC Interval (s)	Time (s)
5	2	11,8
5	2	17,5
5	2	22,3
5	2	20,4
5	2	19,4
<b>Rata-rata =</b>		<b>18,28</b>
10	2	16,9
10	2	21,8
10	2	24,5
10	2	19,3
10	2	22,7
<b>Rata-rata =</b>		<b>21,04</b>
15	2	22,4
15	2	23,5
15	2	28,7
15	2	30,7
15	2	25,3
<b>Rata-rata =</b>		<b>26,12</b>
20	2	24,3
20	2	29,7
20	2	30,8
20	2	28,3
20	2	28,7
<b>Rata-rata =</b>		<b>28,36</b>
25	2	24,7
25	2	29,2
25	2	27,9
25	2	32,5
25	2	28,8
<b>Rata-rata =</b>		<b>28,62</b>
30	2	32,4
30	2	29,4
30	2	35,0
30	2	37,4
30	2	27,5

(Lanjutan)

Tabel hasil pengujian *self-configure* berdasarkan parameter TC *Message* dengan aplikasi

Hello Interval (s)	TC Interval (s)	Time (s)
5	2	15,3
5	2	19,6
5	2	17
5	2	21,4
5	2	22,3
<b>Rata-rata =</b>		<b>19,12</b>
5	5	16,7
5	5	22,3
5	5	20,8
5	5	24,1
5	5	18,3
<b>Rata-rata =</b>		<b>20,44</b>
5	10	22,6
5	10	19,9
5	10	17,4
5	10	24,9
5	10	21,0
<b>Rata-rata =</b>		<b>21,16</b>
5	15	24,3
5	15	20,6
5	15	25,6
5	15	24,8
5	15	19,7
<b>Rata-rata =</b>		<b>23,01</b>
5	20	25,1
5	20	20,2
5	20	20,9
5	20	23,4
5	20	19,7
<b>Rata-rata =</b>		<b>21,86</b>
5	25	25,6
5	25	22,2
5	25	20,2
5	25	23,0
5	25	21,8
<b>Rata-rata =</b>		<b>22,56</b>
5	30	27,4
5	30	23,2
5	30	25,4
5	30	25,7
5	30	23,1
<b>Rata-rata =</b>		<b>24,96</b>

### Lampiran 2.a : Tabel Performansi Self Healing Tanpa Aplikasi

Tabel hasil pengujian *self-healing* dengan parameter Hello Message tanpa aplikasi

Hello Interval (s)	TC Interval (s)	Time (s)
5	2	235
5	2	173
5	2	135
5	2	75
5	2	250
<b>Rata-rata =</b>		<b>173,6</b>
10	2	276
10	2	176
10	2	118
10	2	137
10	2	195
<b>Rata-rata =</b>		<b>180,4</b>
15	2	254
15	2	194
15	2	132
15	2	147
15	2	185
<b>Rata-rata =</b>		<b>182,4</b>
20	2	295
20	2	145
20	2	165
20	2	123
20	2	211
<b>Rata-rata =</b>		<b>187,8</b>
25	2	175
25	2	211
25	2	281
25	2	210
25	2	189
<b>Rata-rata =</b>		<b>213,2</b>
30	2	308
30	2	83
30	2	103
30	2	259
30	2	209
<b>Rata-rata =</b>		<b>192,4</b>

(Lanjutan)

Tabel hasil pengujian *self-healing* dengan parameter TC Message  
tanpa aplikasi

Hello Interval (s)	TC Interval (s)	Time (s)
5	2	212
5	2	85
5	2	194
5	2	145
5	2	239
<b>Rata-rata =</b>		<b>175</b>
5	5	125
5	5	193
5	5	278
5	5	105
5	5	172
<b>Rata-rata =</b>		<b>174,6</b>
5	10	112
5	10	67
5	10	219
5	10	221
5	10	253
<b>Rata-rata =</b>		<b>174,4</b>
5	15	215
5	15	138
5	15	257
5	15	111
5	15	163
<b>Rata-rata =</b>		<b>176,8</b>
5	20	241
5	20	142
5	20	118
5	20	211
5	20	167
<b>Rata-rata =</b>		<b>175,8</b>
5	25	157
5	25	203
5	25	109
5	25	189
5	25	223
<b>Rata-rata =</b>		<b>176,2</b>
5	30	101
5	30	62
5	30	230
5	30	237
5	30	257
<b>Rata-rata =</b>		<b>177,4</b>

### Lampiran 2.b : Tabel Performansi Self Healing Dengan Aplikasi

Tabel hasil pengujian *self-healing* dengan parameter Hello Message dengan aplikasi

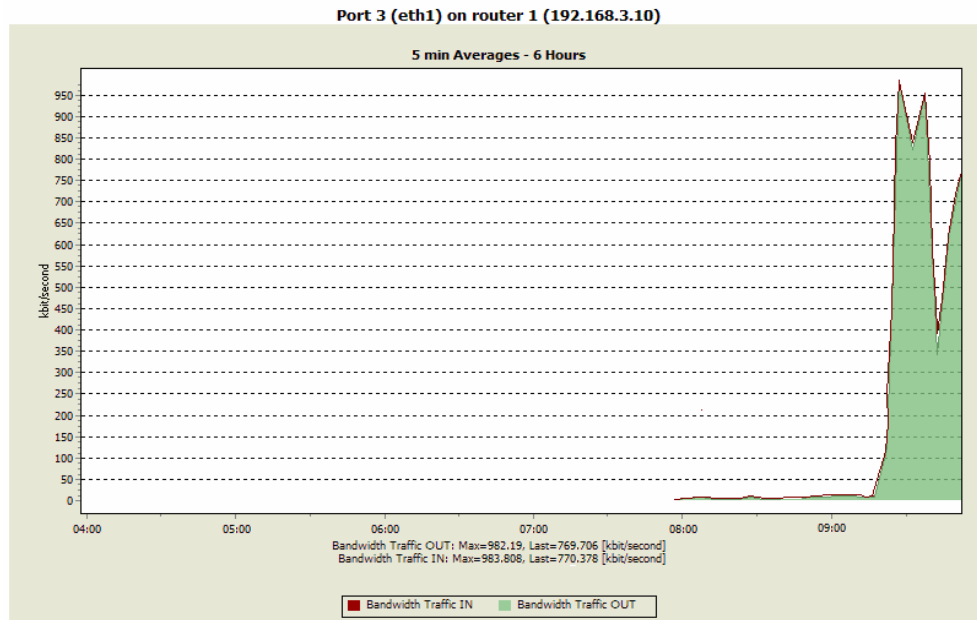
Hello Interval (s)	TC Interval (s)	Time (s)
5	2	43,6
5	2	46,2
5	2	61,2
5	2	55
5	2	35
<b>Rata-rata =</b>		<b>48,2</b>
10	2	54,2
10	2	81,0
10	2	35,3
10	2	46,6
10	2	33,1
<b>Rata-rata =</b>		<b>50,04</b>
15	2	34,5
15	2	54,4
15	2	75,3
15	2	56,7
15	2	69,0
<b>Rata-rata =</b>		<b>57,98</b>
20	2	75,6
20	2	85,4
20	2	50,1
20	2	65,7
20	2	66,3
<b>Rata-rata =</b>		<b>68,62</b>
25	2	66,3
25	2	98,0
25	2	45,6
25	2	93,2
25	2	77,0
<b>Rata-rata =</b>		<b>76,02</b>
30	2	94,0
30	2	111,3
30	2	64,5
30	2	67,3
30	2	72,0
<b>Rata-rata =</b>		<b>81,82</b>

(Lanjutan)

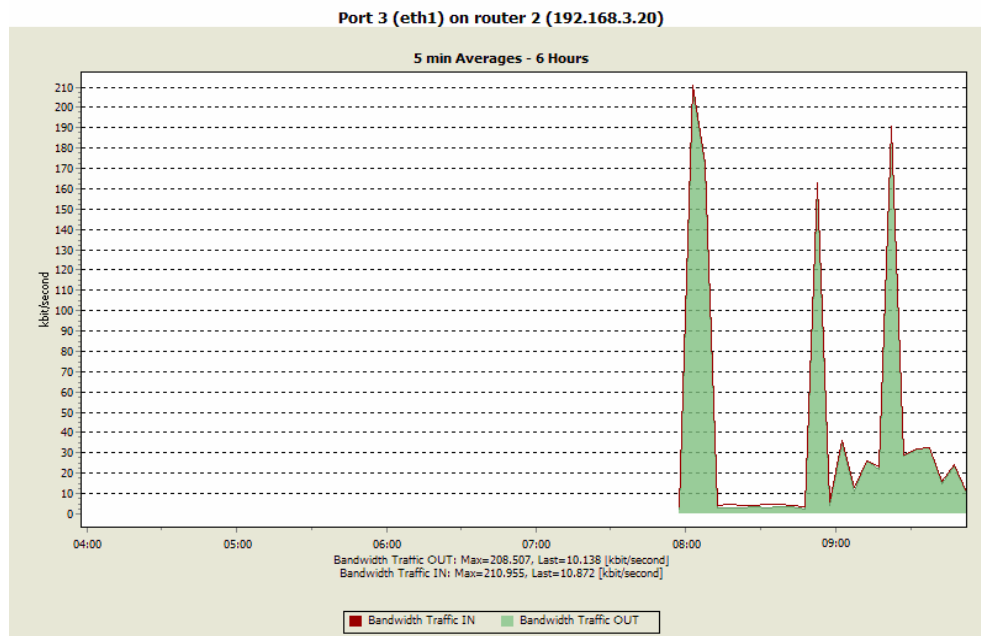
Tabel hasil pengujian *self-healing* dengan parameter *TC Message*  
dengan aplikasi

Hello Interval (s)	TC Interval (s)	Time (s)
5	2	49,0
5	2	52,5
5	2	73,7
5	2	50,1
5	2	44,6
<b>Rata-rata =</b>		<b>53,98</b>
5	5	65,4
5	5	77,0
5	5	34,9
5	5	47,1
5	5	55,3
<b>Rata-rata =</b>		<b>55,94</b>
5	10	73
5	10	40,2
5	10	45,2
5	10	65,7
5	10	37,0
<b>Rata-rata =</b>		<b>52,22</b>
5	15	85,4
5	15	60,4
5	15	73,0
5	15	30,3
5	15	38,0
<b>Rata-rata =</b>		<b>57,42</b>
5	20	50,6
5	20	45,3
5	20	85,0
5	20	75,6
5	20	42,1
<b>Rata-rata =</b>		<b>59,72</b>
5	25	40,6
5	25	45,3
5	25	75,0
5	25	75,6
5	25	42,1
<b>Rata-rata =</b>		<b>55,72</b>
5	30	68,2
5	30	45,4
5	30	89,4
5	30	47,0
5	30	35,9
<b>Rata-rata =</b>		<b>57,18</b>

### Lampiran 3.a : Pembacaan PRTG Untuk Kondisi Interferensi Rendah

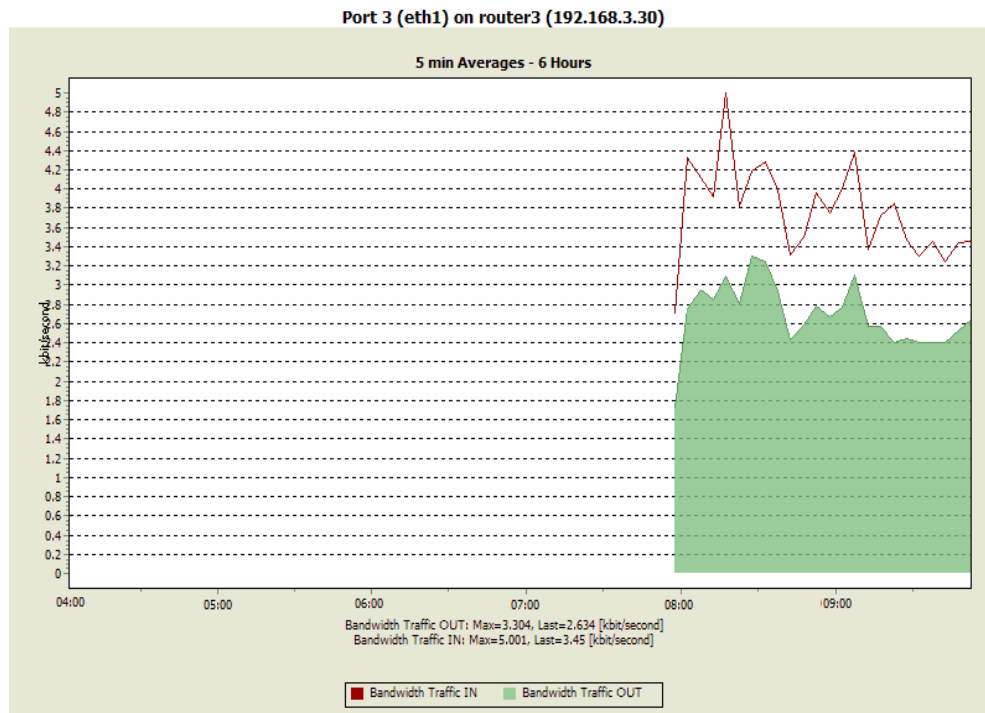


Pembacaan PRTG untuk *Router A* dengan tingkat interferensi rendah

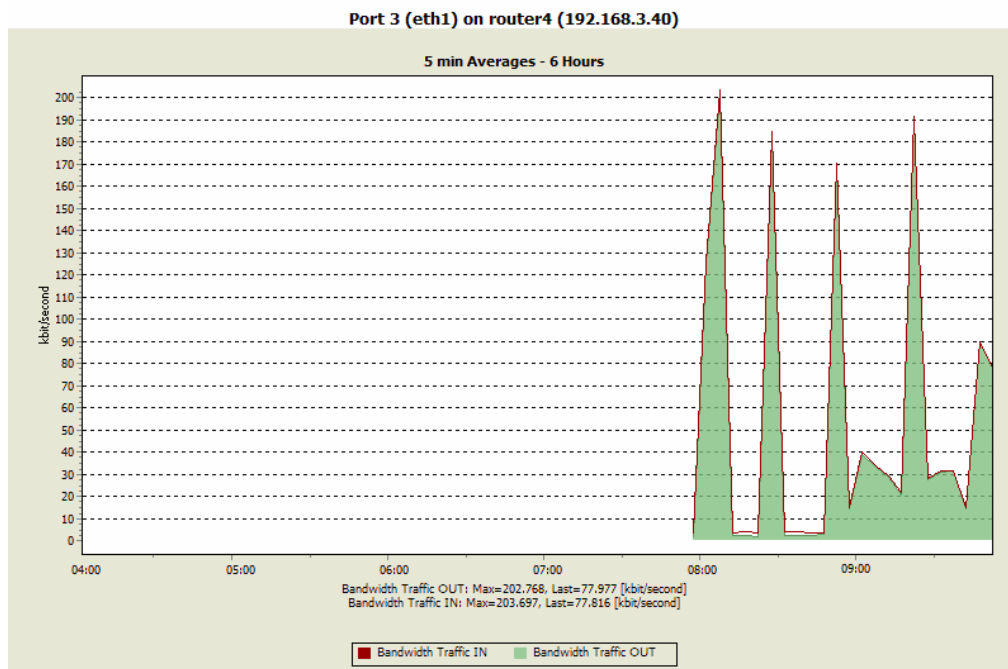


Pembacaan PRTG untuk *Router B* dengan tingkat interferensi rendah

(Lanjutan)



Pembacaan PRTG untuk *Router C* dengan tingkat interferensi rendah



Pembacaan PRTG untuk *Router D* dengan tingkat interferensi rendah



### Lampiran 3.b : Tabel Pembacaan PRTG Untuk Interferensi Rendah.

#### Pembacaan PRTG Untuk Router A Dengan Interferensi Rendah

	Port 3 (eth1) on router	Port 3 (eth1) on router	Port 3 (eth1) on router 1	Port 3 (eth1) on router
	Bandwidth Traffic IN	Bandwidth Traffic IN	Bandwidth Traffic OUT	Bandwidth Traffic OUT
	kbyte	kbit/second	kbyte	kbit/second
7/20/2008 9:50 PM - 9:55 PM	21,230.78	770.378	21,102.09	769.706
7/20/2008 9:45 PM - 9:50 PM	23,967.11	631.098	23,052.52	630.391
7/20/2008 9:40 PM - 9:45 PM	12,671.09	343.73	12,525.29	342.515
7/20/2008 9:35 PM - 9:40 PM	35,120.88	951.556	34,750.05	950.302
7/20/2008 9:30 PM - 9:35 PM	29,987.13	821.878	28,981.56	821
7/20/2008 9:25 PM - 9:30 PM	36,155.62	983.808	35,917.33	982.19
7/20/2008 9:20 PM - 9:25 PM	4,432.99	122.09	4,392.56	120.118
7/20/2008 9:15 PM - 9:20 PM	289.76	9.483	288.635	8.148
7/20/2008 9:10 PM - 9:15 PM	320.89	10.005	319.89	9.063
7/20/2008 9:05 PM - 9:10 PM	350.01	12.301	349.918	10.287
7/20/2008 9:00 PM - 9:05 PM	354.67	9.899	353.507	9.667
7/20/2008 8:55 PM - 9:00 PM	255.223	7.231	254.677	6.964
7/20/2008 8:50 PM - 8:55 PM	294.99	8.578	293.519	8.424
7/20/2008 8:45 PM - 8:50 PM	175.38	6.41	174.027	5.031
7/20/2008 8:40 PM - 8:45 PM	152.664	4.408	152.626	4.481
7/20/2008 8:35 PM - 8:40 PM	141.333	5.233	140.71	3.852
7/20/2008 8:30 PM - 8:35 PM	144.32	5.224	143.135	4.055
7/20/2008 8:25 PM - 8:30 PM	279.04	8.23	278.894	7.627
7/20/2008 8:20 PM - 8:25 PM	116.773	3.88	116.277	3.18
7/20/2008 8:15 PM - 8:20 PM	133.4	4.322	132.552	3.825
7/20/2008 8:10 PM - 8:15 PM	138.55	4.877	138.427	3.785
7/20/2008 8:05 PM - 8:10 PM	273.00	7.984	272.511	7.452
7/20/2008 8:00 PM - 8:05 PM	214.56	6.322	213.349	5.834
7/20/2008 7:55 PM - 8:00 PM	33.012	2.675	32.859	2.542
Total	166,112.89		164,376.91	
Average	6,622.41	197.884	6,572.14	196.685

#### Pembacaan PRTG Untuk Router B Dengan Interferensi Rendah

	Port 3 (eth1) on router	Port 3 (eth1) on router	Port 3 (eth1) on router 2	Port 3 (eth1) on router 2
	Bandwidth Traffic IN	Bandwidth Traffic IN	Bandwidth Traffic OUT	Bandwidth Traffic OUT
	kbyte	kbit/second	kbyte	kbit/second
7/20/2008 9:50 PM - 9:55 PM	310.414	10.872	289.44	10.138
7/20/2008 9:45 PM - 9:50 PM	894.848	24.47	874.569	23.916
7/20/2008 9:40 PM - 9:45 PM	555.021	15.726	531.935	15.071
7/20/2008 9:35 PM - 9:40 PM	1,185.82	32.428	1,183.17	32.356
7/20/2008 9:30 PM - 9:35 PM	1,180.91	32.293	1,177.04	32.187
7/20/2008 9:25 PM - 9:30 PM	1,065.00	29.123	1,045.11	28.579
7/20/2008 9:20 PM - 9:25 PM	6,724.48	190.52	6,719.51	190.379
7/20/2008 9:15 PM - 9:20 PM	835.451	22.846	806.758	22.061
7/20/2008 9:10 PM - 9:15 PM	962.155	26.311	944.173	25.819
7/20/2008 9:05 PM - 9:10 PM	263.19	13.069	223.086	11.008
7/20/2008 9:00 PM - 9:05 PM	1,193.12	36.073	1,156.86	34.977
7/20/2008 8:55 PM - 9:00 PM	210.556	5.944	155.908	4.401
7/20/2008 8:50 PM - 8:55 PM	5,963.10	163.066	5,918.88	161.857
7/20/2008 8:45 PM - 8:50 PM	135.074	3.694	97.433	2.664
7/20/2008 8:40 PM - 8:45 PM	160.962	4.402	124.109	3.394
7/20/2008 8:35 PM - 8:40 PM	175.142	4.79	124.661	3.409
7/20/2008 8:30 PM - 8:35 PM	151.256	4.459	97.848	2.874
7/20/2008 8:25 PM - 8:30 PM	1.191	4.3	0.968	3.493
7/20/2008 8:20 PM - 8:25 PM	157.14	4.297	114.897	3.142
7/20/2008 8:15 PM - 8:20 PM	181.326	4.959	116.952	3.198
7/20/2008 8:10 PM - 8:15 PM	144.047	3.939	102.144	2.793
7/20/2008 8:05 PM - 8:10 PM	6,299.99	172.279	6,263.54	171.282
7/20/2008 8:00 PM - 8:05 PM	4,467.59	210.955	4,442.23	208.507
7/20/2008 7:55 PM - 8:00 PM	47.351	3.591	34.814	2.64
Total	33,265.13		32,546.04	
Average	1,451.09	42.684	1,418.15	41.673

(Lanjutan)

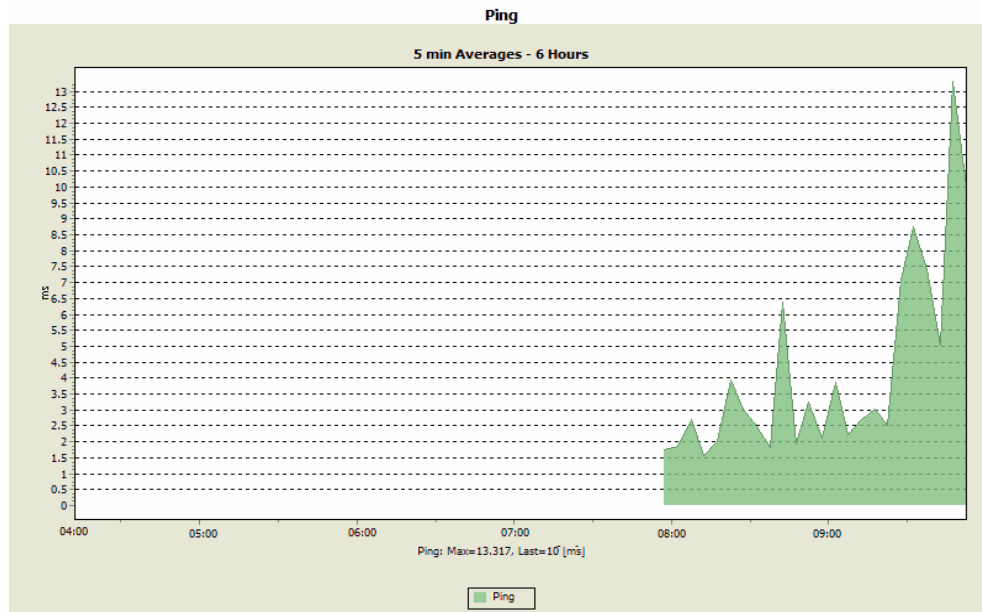
## Pembacaan PRTG Untuk Router C Dengan Interferensi Rendah

	Port 3 (eth1) on routerC	Port 3 (eth1) on routerC	Port 3 (eth1) on router3	Port 3 (eth1) on router3
	Bandwidth Traffic IN	Bandwidth Traffic IN	Bandwidth Traffic OUT	Bandwidth Traffic OUT
	kbyte	kbit/second	kbyte	kbit/second
7/20/2008 9:50 PM - 9:55 PM	102.618	3.45	78.351	2.634
7/20/2008 9:45 PM - 9:50 PM	121.314	3.438	89.477	2.535
7/20/2008 9:40 PM - 9:45 PM	118.683	3.245	87.582	2.395
7/20/2008 9:35 PM - 9:40 PM	126.566	3.461	87.829	2.402
7/20/2008 9:30 PM - 9:35 PM	120.488	3.295	87.989	2.406
7/20/2008 9:25 PM - 9:30 PM	126.916	3.471	89.306	2.442
7/20/2008 9:20 PM - 9:25 PM	140.499	3.842	87.976	2.406
7/20/2008 9:15 PM - 9:20 PM	136.174	3.724	93.985	2.57
7/20/2008 9:10 PM - 9:15 PM	123.293	3.372	93.753	2.564
7/20/2008 9:05 PM - 9:10 PM	160.161	4.38	113.57	3.106
7/20/2008 9:00 PM - 9:05 PM	141.061	3.998	97.532	2.764
7/20/2008 8:55 PM - 9:00 PM	137.294	3.755	97.53	2.667
7/20/2008 8:50 PM - 8:55 PM	144.731	3.958	101.762	2.783
7/20/2008 8:45 PM - 8:50 PM	128.081	3.502	94.438	2.582
7/20/2008 8:40 PM - 8:45 PM	121.248	3.316	88.921	2.432
7/20/2008 8:35 PM - 8:40 PM	145.791	3.987	107.354	2.936
7/20/2008 8:30 PM - 8:35 PM	156.669	4.284	118.763	3.248
7/20/2008 8:25 PM - 8:30 PM	153.132	4.188	120.819	3.304
7/20/2008 8:20 PM - 8:25 PM	135.074	3.827	98.925	2.803
7/20/2008 8:15 PM - 8:20 PM	182.868	5.001	113.181	3.095
7/20/2008 8:10 PM - 8:15 PM	143.289	3.918	104.197	2.849
7/20/2008 8:05 PM - 8:10 PM	150.388	4.112	107.82	2.948
7/20/2008 8:00 PM - 8:05 PM	91.677	4.328	58.808	2.76
7/20/2008 7:55 PM - 8:00 PM	33.521	2.707	21.314	1.721
Total	3,141.54		2,241.18	
Average	139.031	3.773	99.661	2.681

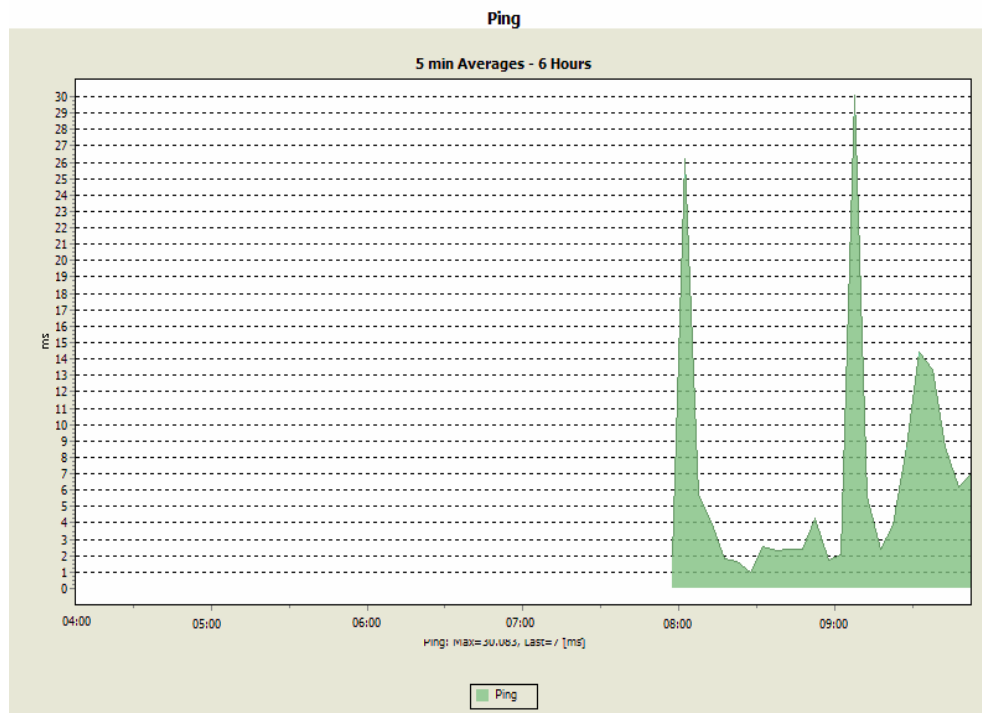
## Pembacaan PRTG Untuk Router D Dengan Interferensi Rendah

	Port 3 (eth1) on routerD	Port 3 (eth1) on routerD	Port 3 (eth1) on router4	Port 3 (eth1) on router4
	Bandwidth Traffic IN	Bandwidth Traffic IN	Bandwidth Traffic OUT	Bandwidth Traffic OUT
	kbyte	kbit/second	kbyte	kbit/second
7/20/2008 9:50 PM - 9:55 PM	2,293.63	77.816	2,298.39	77.977
7/20/2008 9:45 PM - 9:50 PM	3,277.35	89.622	3,281.57	89.737
7/20/2008 9:40 PM - 9:45 PM	550.009	15.04	530.401	14.504
7/20/2008 9:35 PM - 9:40 PM	1,122.69	31.704	1,121.08	31.659
7/20/2008 9:30 PM - 9:35 PM	1,141.22	31.208	1,139.68	31.165
7/20/2008 9:25 PM - 9:30 PM	1,021.80	27.942	1,003.66	27.446
7/20/2008 9:20 PM - 9:25 PM	6,999.43	191.405	7,000.47	191.434
7/20/2008 9:15 PM - 9:20 PM	789.473	21.589	763.213	20.871
7/20/2008 9:10 PM - 9:15 PM	1,064.07	29.099	1,053.25	28.803
7/20/2008 9:05 PM - 9:10 PM	1,231.69	33.682	1,205.90	32.976
7/20/2008 9:00 PM - 9:05 PM	1,462.41	39.991	1,429.89	39.102
7/20/2008 8:55 PM - 9:00 PM	546.767	14.952	516.537	14.125
7/20/2008 8:50 PM - 8:55 PM	6,235.45	170.514	6,184.92	169.132
7/20/2008 8:45 PM - 8:50 PM	128.296	3.508	94.66	2.589
7/20/2008 8:40 PM - 8:45 PM	116.929	3.335	87.1	2.484
7/20/2008 8:35 PM - 8:40 PM	139.375	3.923	84.461	2.377
7/20/2008 8:30 PM - 8:35 PM	147.659	4.038	80.172	2.192
7/20/2008 8:25 PM - 8:30 PM	6,613.90	184.937	6,587.98	184.212
7/20/2008 8:20 PM - 8:25 PM	125.654	3.48	67.52	1.87
7/20/2008 8:15 PM - 8:20 PM	152.373	4.167	93.007	2.543
7/20/2008 8:10 PM - 8:15 PM	122.597	3.353	77.77	2.127
7/20/2008 8:05 PM - 8:10 PM	7,448.66	203.697	7,414.71	202.788
7/20/2008 8:00 PM - 8:05 PM	4,603.34	125.882	4,553.58	124.521
7/20/2008 7:55 PM - 8:00 PM	40.035	3.164	24.863	1.965
Total	47,374.80		46,694.78	
Average	2,093.59	54.919	2,064.50	54.108

### Lampiran 4.a : Pembacaan Latency Untuk Kondisi Interferensi Rendah

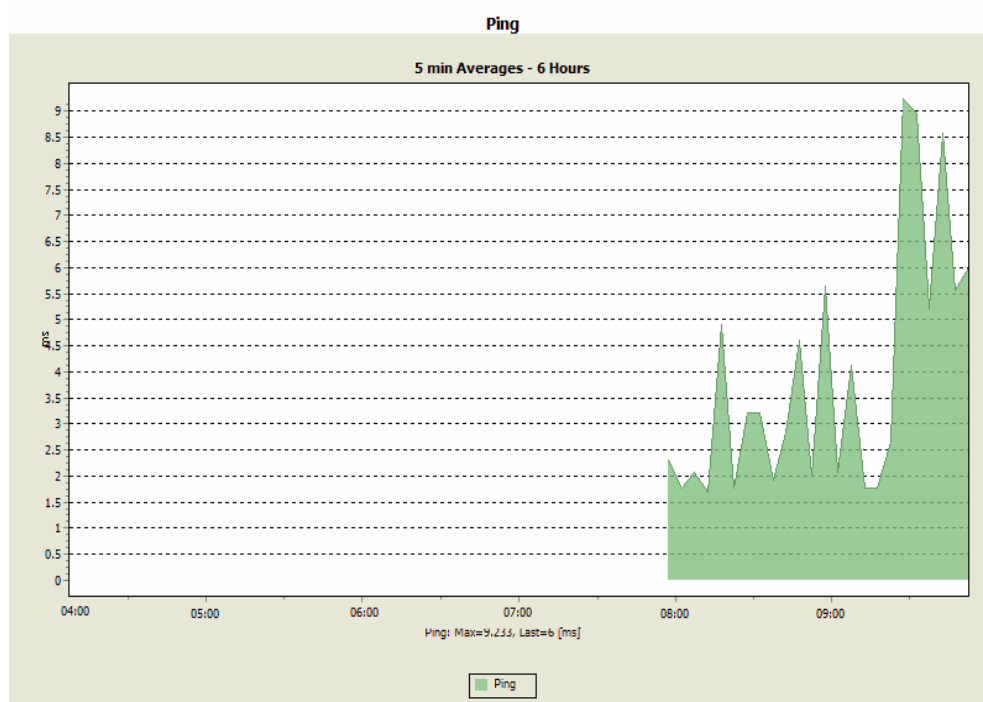


Pembacaan PRTG untuk latency pada Router A dengan tingkat interferensi rendah

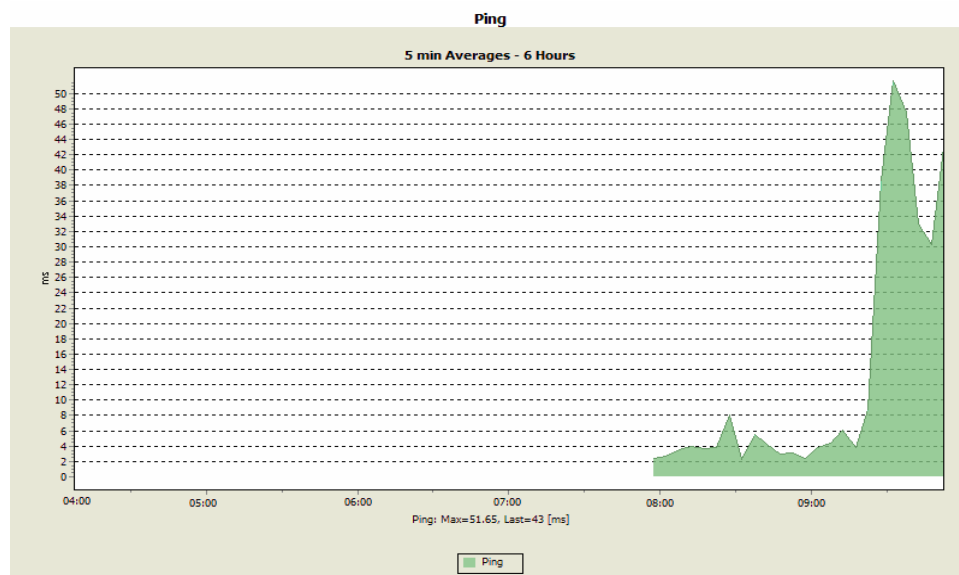


Pembacaan PRTG untuk *latency* pada Router B dengan tingkat interferensi rendah

(Lanjutan)



Pembacaan PRTG untuk *latency* pada *Router C* dengan tingkat interferensi rendah



Pembacaan PRTG untuk *latency* pada *Router D* dengan tingkat interferensi rendah

### Lampiran 4.b Tabel Pembacaan Latency Untuk Kondisi Interferensi Rendah

#### Pembacaan PRTG Untuk Latency Router A Dengan Interferensi Rendah

	Ping Latency ms
7/20/2008 9:50 PM - 9:55 PM	10
7/20/2008 9:45 PM - 9:50 PM	13.317
7/20/2008 9:40 PM - 9:45 PM	5.017
7/20/2008 9:35 PM - 9:40 PM	7.4
7/20/2008 9:30 PM - 9:35 PM	8.75
7/20/2008 9:25 PM - 9:30 PM	7.083
7/20/2008 9:20 PM - 9:25 PM	2.525
7/20/2008 9:15 PM - 9:20 PM	3.033
7/20/2008 9:10 PM - 9:15 PM	2.717
7/20/2008 9:05 PM - 9:10 PM	2.217
7/20/2008 9:00 PM - 9:05 PM	3.883
7/20/2008 8:55 PM - 9:00 PM	2.117
7/20/2008 8:50 PM - 8:55 PM	3.25
7/20/2008 8:45 PM - 8:50 PM	1.917
7/20/2008 8:40 PM - 8:45 PM	6.368
7/20/2008 8:35 PM - 8:40 PM	1.833
7/20/2008 8:30 PM - 8:35 PM	2.467
7/20/2008 8:25 PM - 8:30 PM	3.017
7/20/2008 8:20 PM - 8:25 PM	3.95
7/20/2008 8:15 PM - 8:20 PM	2
7/20/2008 8:10 PM - 8:15 PM	1.559
7/20/2008 8:05 PM - 8:10 PM	2.717
7/20/2008 8:00 PM - 8:05 PM	1.867
7/20/2008 7:55 PM - 8:00 PM	1.75
Total	N/A
Average	4.198

#### Pembacaan PRTG Untuk Latency Router B Dengan Interferensi Rendah

	Ping Latency ms
7/20/2008 9:50 PM - 9:55 PM	7
7/20/2008 9:45 PM - 9:50 PM	6.15
7/20/2008 9:40 PM - 9:45 PM	8.65
7/20/2008 9:35 PM - 9:40 PM	13.317
7/20/2008 9:30 PM - 9:35 PM	14.433
7/20/2008 9:25 PM - 9:30 PM	8.65
7/20/2008 9:20 PM - 9:25 PM	3.847
7/20/2008 9:15 PM - 9:20 PM	2.417
7/20/2008 9:10 PM - 9:15 PM	5.6
7/20/2008 9:05 PM - 9:10 PM	30.083
7/20/2008 9:00 PM - 9:05 PM	2.073
7/20/2008 8:55 PM - 9:00 PM	1.767
7/20/2008 8:50 PM - 8:55 PM	4.254
7/20/2008 8:45 PM - 8:50 PM	2.407
7/20/2008 8:40 PM - 8:45 PM	2.414
7/20/2008 8:35 PM - 8:40 PM	2.3
7/20/2008 8:30 PM - 8:35 PM	2.576
7/20/2008 8:25 PM - 8:30 PM	1
7/20/2008 8:20 PM - 8:25 PM	1.65
7/20/2008 8:15 PM - 8:20 PM	1.8
7/20/2008 8:10 PM - 8:15 PM	4.133
7/20/2008 8:05 PM - 8:10 PM	5.7
7/20/2008 8:00 PM - 8:05 PM	26.2
7/20/2008 7:55 PM - 8:00 PM	1.952
Total	N/A
Average	6.682

(Lanjutan)

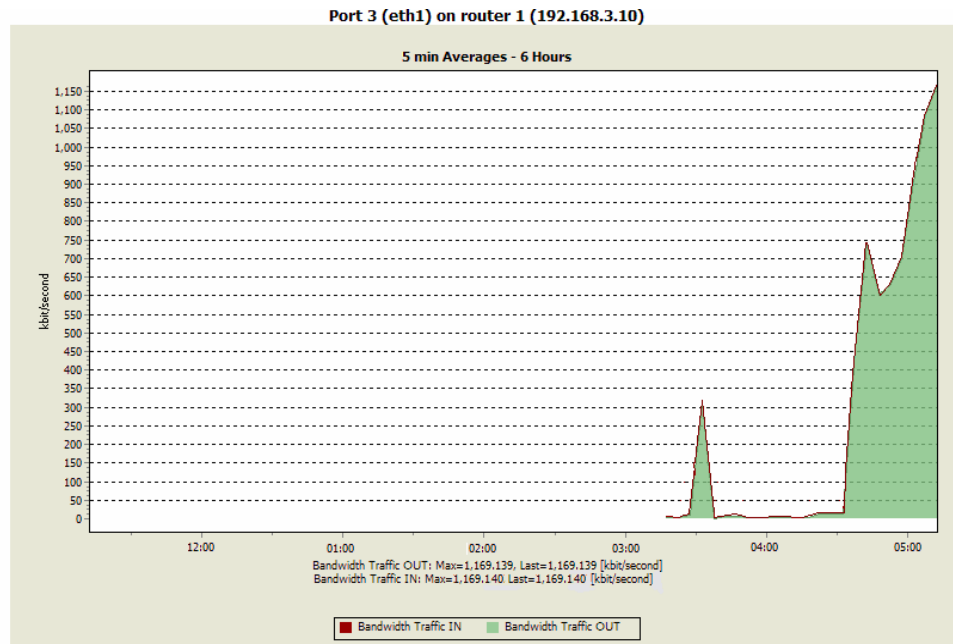
## Pembacaan PRTG Untuk Latency Router C Dengan Interferensi Rendah

	Ping
	Latency
	ms
7/20/2008 9:50 PM - 9:55 PM	6
7/20/2008 9:45 PM - 9:50 PM	5.567
7/20/2008 9:40 PM - 9:45 PM	8.567
7/20/2008 9:35 PM - 9:40 PM	5.217
7/20/2008 9:30 PM - 9:35 PM	8.967
7/20/2008 9:25 PM - 9:30 PM	9.233
7/20/2008 9:20 PM - 9:25 PM	2.65
7/20/2008 9:15 PM - 9:20 PM	1.783
7/20/2008 9:10 PM - 9:15 PM	1.783
7/20/2008 9:05 PM - 9:10 PM	4.117
7/20/2008 9:00 PM - 9:05 PM	2.083
7/20/2008 8:55 PM - 9:00 PM	5.633
7/20/2008 8:50 PM - 8:55 PM	2
7/20/2008 8:45 PM - 8:50 PM	4.6
7/20/2008 8:40 PM - 8:45 PM	2.833
7/20/2008 8:35 PM - 8:40 PM	1.917
7/20/2008 8:30 PM - 8:35 PM	3.217
7/20/2008 8:25 PM - 8:30 PM	3.22
7/20/2008 8:20 PM - 8:25 PM	1.783
7/20/2008 8:15 PM - 8:20 PM	4.917
7/20/2008 8:10 PM - 8:15 PM	1.683
7/20/2008 8:05 PM - 8:10 PM	2.068
7/20/2008 8:00 PM - 8:05 PM	1.783
7/20/2008 7:55 PM - 8:00 PM	2.3
Total	N/A
Average	3.913

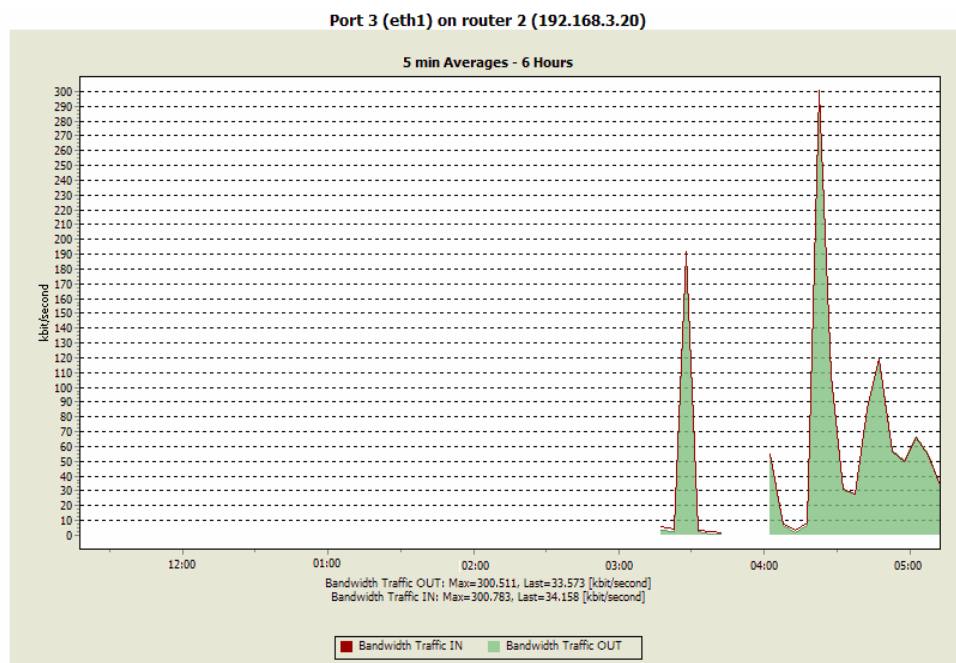
## Pembacaan PRTG Untuk Latency Router D Dengan Interferensi Rendah

	Ping
	Latency
	ms
7/20/2008 9:50 PM - 9:55 PM	43
7/20/2008 9:45 PM - 9:50 PM	30.254
7/20/2008 9:40 PM - 9:45 PM	33.017
7/20/2008 9:35 PM - 9:40 PM	47.517
7/20/2008 9:30 PM - 9:35 PM	51.65
7/20/2008 9:25 PM - 9:30 PM	38.583
7/20/2008 9:20 PM - 9:25 PM	8.75
7/20/2008 9:15 PM - 9:20 PM	3.763
7/20/2008 9:10 PM - 9:15 PM	6.117
7/20/2008 9:05 PM - 9:10 PM	4.39
7/20/2008 9:00 PM - 9:05 PM	3.81
7/20/2008 8:55 PM - 9:00 PM	2.414
7/20/2008 8:50 PM - 8:55 PM	3.085
7/20/2008 8:45 PM - 8:50 PM	2.915
7/20/2008 8:40 PM - 8:45 PM	4.271
7/20/2008 8:35 PM - 8:40 PM	5.525
7/20/2008 8:30 PM - 8:35 PM	2.25
7/20/2008 8:25 PM - 8:30 PM	8.033
7/20/2008 8:20 PM - 8:25 PM	3.75
7/20/2008 8:15 PM - 8:20 PM	3.683
7/20/2008 8:10 PM - 8:15 PM	4
7/20/2008 8:05 PM - 8:10 PM	3.483
7/20/2008 8:00 PM - 8:05 PM	2.661
7/20/2008 7:55 PM - 8:00 PM	2.45
Total	N/A
Average	13.307

### Lampiran 5.a : Pembacaan PRTG Untuk Interferensi Tinggi.

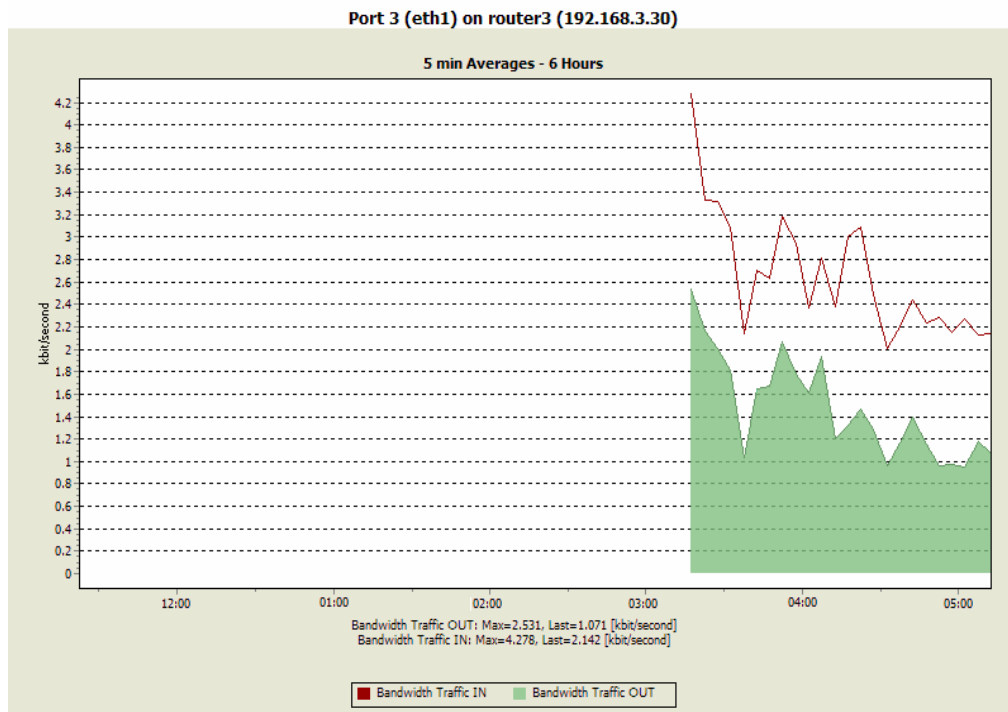


Pembacaan PRTG untuk *Router A* dengan tingkat interferensi tinggi

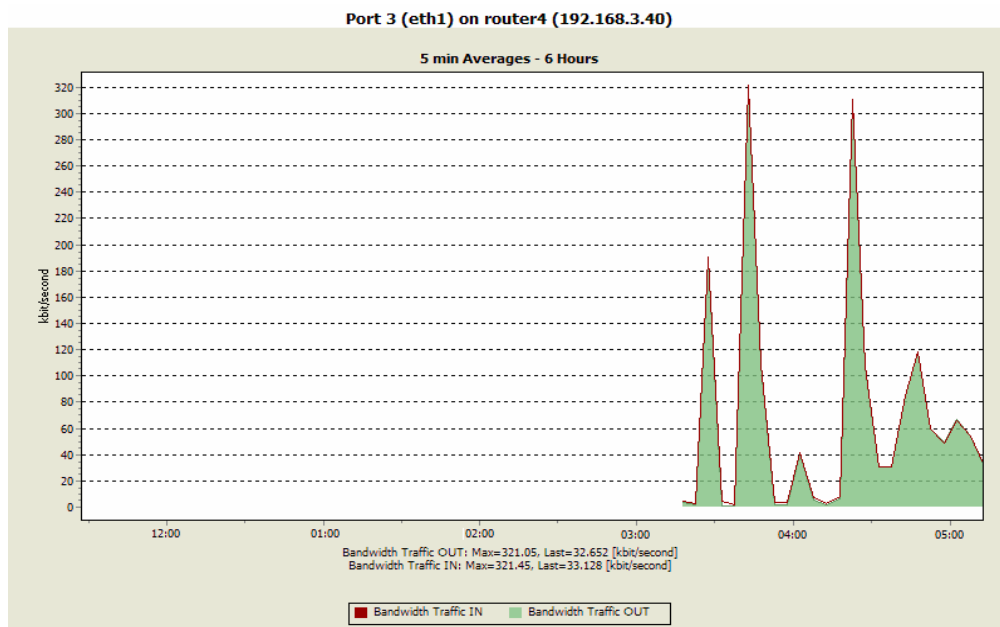


Pembacaan PRTG untuk *Router B* dengan tingkat interferensi tinggi

(Lanjutan)



Pembacaan PRTG untuk *Router C* dengan tingkat interferensi tinggi



Pembacaan PRTG untuk *Router D* dengan tingkat interferensi tinggi



### Lampiran 5.b : Tabel Pembacaan PRTG Untuk Interferensi Tinggi.

#### Pembacaan PRTG Untuk Router A Dengan Interferensi Tinggi

	Port 3 (eth1) on router 1 (192.168.3.10)	Port 3 (eth1) on router 1	Port 3 (eth1) on router 1	Port 3 (eth1) on router 1
	Bandwidth Traffic IN	Bandwidth Traffic IN	Bandwidth Traffic OUT	Bandwidth Traffic OUT
	kbyte	kbit/second	kbyte	kbit/second
7/18/2008 5:10 PM - 5:15 PM	12,591.67	1,169.14	12,590.51	1,169.14
7/18/2008 5:05 PM - 5:10 PM	39,897.67	1,087.79	39,697.43	1,085.56
7/18/2008 5:00 PM - 5:05 PM	33,165.98	906.75	33,107.48	905.383
7/18/2008 4:55 PM - 5:00 PM	23,797.80	699.59	23,774.70	699.351
7/18/2008 4:50 PM - 4:55 PM	23,049.78	631.33	23,038.79	630.015
7/18/2008 4:45 PM - 4:50 PM	22,223.96	603.97	22,084.16	603.91
7/18/2008 4:40 PM - 4:45 PM	27,450.76	742.34	27,103.77	741.176
7/18/2008 4:35 PM - 4:40 PM	16,015.56	453.76	15,946.42	452.754
7/18/2008 4:30 PM - 4:35 PM	443.09	12.03	432.082	11.816
7/18/2008 4:25 PM - 4:30 PM	444.66	12.78	437.131	11.954
7/18/2008 4:20 PM - 4:25 PM	460.08	13.403	459.482	12.565
7/18/2008 4:15 PM - 4:20 PM	110.9	3.713	108.33	3.065
7/18/2008 4:10 PM - 4:15 PM	73.111	2.243	72.675	1.987
7/18/2008 4:05 PM - 4:10 PM	118.089	3.714	117.954	3.226
7/18/2008 4:00 PM - 4:05 PM	171.887	4.797	171.378	4.686
7/18/2008 3:55 PM - 4:00 PM	144.031	4.556	143.394	3.921
7/18/2008 3:50 PM - 3:55 PM	82.33	2.908	81.495	2.309
7/18/2008 3:45 PM - 3:50 PM	297.89	8.675	296.813	8.117
7/18/2008 3:40 PM - 3:45 PM	290.77	8.978	289.928	7.928
7/18/2008 3:35 PM - 3:40 PM	44.56	1.439	43.368	1.186
7/18/2008 3:30 PM - 3:35 PM	11,556.98	315.994	11,514.85	314.883
7/18/2008 3:25 PM - 3:30 PM	327.88	9.588	325.536	9.569
7/18/2008 3:20 PM - 3:25 PM	88.09	2.884	87.613	2.396
7/18/2008 3:15 PM - 3:20 PM	119.65	4.9	118.982	4.4
Total	215,677.17		212,044.24	
Average	9,207.64	279.017	9,184.83	278.804

#### Pembacaan PRTG Untuk Router B Dengan Interferensi Tinggi

	Port 3 (eth1) on router	Port 3 (eth1) on router	Port 3 (eth1) on router	Port 3 (eth1) on router
	Bandwidth Traffic IN	Bandwidth Traffic IN	Bandwidth Traffic OUT	Bandwidth Traffic OUT
	kbyte	kbit/second	kbyte	kbit/second
7/18/2008 5:10 PM - 5:15 PM	354.672	34.158	348.596	33.573
7/18/2008 5:05 PM - 5:10 PM	1,991.94	54.471	2,000.66	54.71
7/18/2008 5:00 PM - 5:05 PM	2,416.39	66.078	2,447.13	66.919
7/18/2008 4:55 PM - 5:00 PM	1,766.63	50.153	1,778.55	50.492
7/18/2008 4:50 PM - 4:55 PM	2,072.64	56.678	2,092.73	57.228
7/18/2008 4:45 PM - 4:50 PM	4,348.87	118.924	4,366.17	119.397
7/18/2008 4:40 PM - 4:45 PM	3,134.86	85.728	3,115.65	85.203
7/18/2008 4:35 PM - 4:40 PM	1,029.47	28.152	1,019.85	27.889
7/18/2008 4:30 PM - 4:35 PM	1,136.84	31.088	1,124.84	30.76
7/18/2008 4:25 PM - 4:30 PM	3,944.92	107.877	3,935.04	107.607
7/18/2008 4:20 PM - 4:25 PM	10,999.21	300.783	10,989.28	300.511
7/18/2008 4:15 PM - 4:20 PM	299.163	8.181	240.446	6.575
7/18/2008 4:10 PM - 4:15 PM	111.627	3.053	70.471	1.927
7/18/2008 4:05 PM - 4:10 PM	274.385	7.774	248.506	7.041
7/18/2008 4:00 PM - 4:05 PM	1,527.92	54.782	1,513.88	54.031
7/18/2008 3:55 PM - 4:00 PM				
7/18/2008 3:50 PM - 3:55 PM				
7/18/2008 3:45 PM - 3:50 PM				
7/18/2008 3:40 PM - 3:45 PM	2.908	1.94	1.554	1.036
7/18/2008 3:35 PM - 3:40 PM	82.316	2.251	43.245	1.183
7/18/2008 3:30 PM - 3:35 PM	138.07	3.776	98.166	2.684
7/18/2008 3:25 PM - 3:30 PM	7,006.16	191.59	6,978.78	190.841
7/18/2008 3:20 PM - 3:25 PM	159.474	4.361	97.481	2.666
7/18/2008 3:15 PM - 3:20 PM	85.934	5.737	53.062	3.513
Total	42,884.39		42,564.07	
Average	2,391.97	57.978	2,375.61	57.418

(Lanjutan)

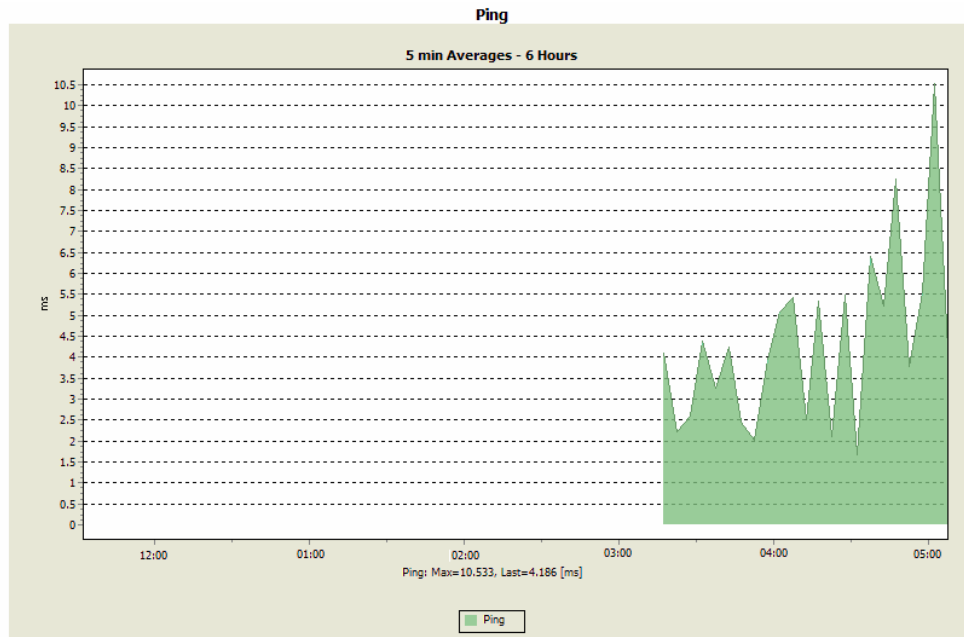
## Pembacaan PRTG Untuk Router C Dengan Interferensi Tinggi

	Port 3 (eth1) on router	Port 3 (eth1) on router	Port 3 (eth1) on router	Port 3 (eth1) on router
	Bandwidth Traffic IN	Bandwidth Traffic IN	Bandwidth Traffic OUT	Bandwidth Traffic OUT
	kbyte	kbit/second	kbyte	kbit/second
7/18/2008 5:10 PM - 5:15 PM	24.453	2.142	12.231	1.071
7/18/2008 5:05 PM - 5:10 PM	77.918	2.131	42.99	1.176
7/18/2008 5:00 PM - 5:05 PM	83.129	2.273	34.746	0.95
7/18/2008 4:55 PM - 5:00 PM	76.116	2.154	34.436	0.974
7/18/2008 4:50 PM - 4:55 PM	83.336	2.279	35.293	0.965
7/18/2008 4:45 PM - 4:50 PM	81.942	2.241	42.111	1.152
7/18/2008 4:40 PM - 4:45 PM	89.37	2.444	50.853	1.391
7/18/2008 4:35 PM - 4:40 PM	80.248	2.194	42.676	1.167
7/18/2008 4:30 PM - 4:35 PM	73.476	2.009	35.031	0.958
7/18/2008 4:25 PM - 4:30 PM	91.144	2.492	46.886	1.282
7/18/2008 4:20 PM - 4:25 PM	113.104	3.093	53.584	1.465
7/18/2008 4:15 PM - 4:20 PM	106.101	3.002	46.551	1.317
7/18/2008 4:10 PM - 4:15 PM	87.08	2.381	43.853	1.199
7/18/2008 4:05 PM - 4:10 PM	102.891	2.814	70.904	1.939
7/18/2008 4:00 PM - 4:05 PM	86.508	2.366	58.884	1.61
7/18/2008 3:55 PM - 4:00 PM	103.788	2.941	62.812	1.78
7/18/2008 3:50 PM - 3:55 PM	116.453	3.185	75.618	2.068
7/18/2008 3:45 PM - 3:50 PM	96.168	2.63	61.09	1.671
7/18/2008 3:40 PM - 3:45 PM	98.973	2.706	60.098	1.643
7/18/2008 3:35 PM - 3:40 PM	78.049	2.134	37.939	1.037
7/18/2008 3:30 PM - 3:35 PM	112.848	3.086	66.183	1.81
7/18/2008 3:25 PM - 3:30 PM	121.377	3.319	73.357	2.006
7/18/2008 3:20 PM - 3:25 PM	121.778	3.33	79.301	2.169
7/18/2008 3:15 PM - 3:20 PM	68.22	4.278	41.01	2.531
Total	2,174.47		1,208.44	
Average	95.221	2.651	52.815	1.472

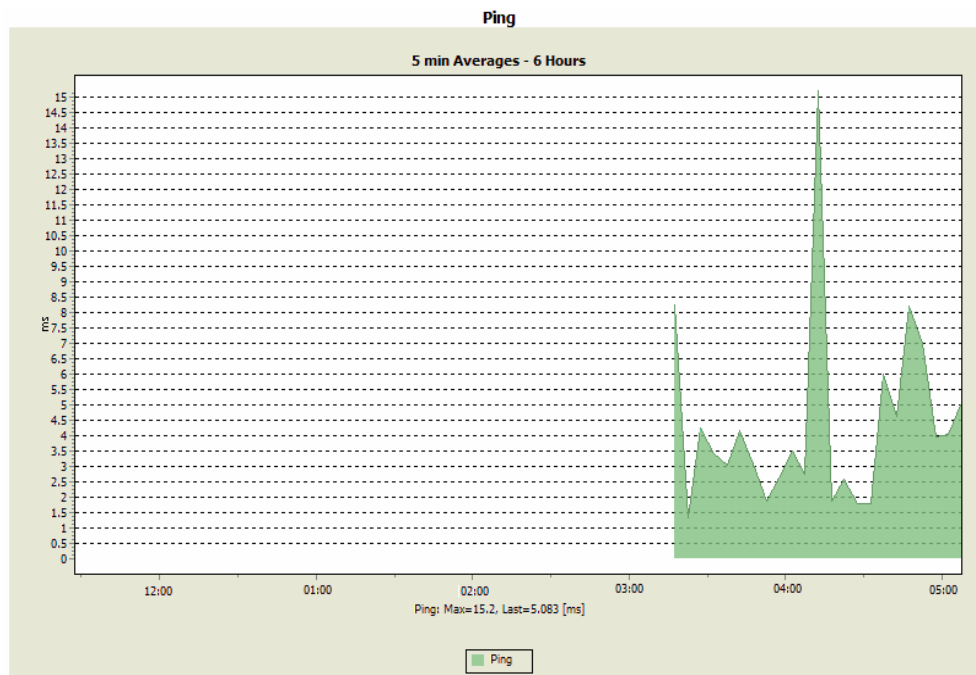
## Pembacaan PRTG Untuk Router D Dengan Interferensi Tinggi

	Port 3 (eth1) on router	Port 3 (eth1) on router	Port 3 (eth1) on router	Port 3 (eth1) on router
	Bandwidth Traffic IN	Bandwidth Traffic IN	Bandwidth Traffic OUT	Bandwidth Traffic OUT
	kbyte	kbit/second	kbyte	kbit/second
7/18/2008 5:10 PM - 5:15 PM	361.329	33.128	356.133	32.652
7/18/2008 5:05 PM - 5:10 PM	1,976.54	54.05	1,986.33	54.318
7/18/2008 5:00 PM - 5:05 PM	2,336.60	66.11	2,364.97	66.912
7/18/2008 4:55 PM - 5:00 PM	1,790.27	48.956	1,800.75	49.243
7/18/2008 4:50 PM - 4:55 PM	2,169.58	59.329	2,190.04	59.889
7/18/2008 4:45 PM - 4:50 PM	4,339.12	118.657	4,341.08	118.711
7/18/2008 4:40 PM - 4:45 PM	3,114.87	85.179	3,102.17	84.831
7/18/2008 4:35 PM - 4:40 PM	1,138.30	31.128	1,118.20	30.578
7/18/2008 4:30 PM - 4:35 PM	1,084.48	30.682	1,070.93	30.299
7/18/2008 4:25 PM - 4:30 PM	3,941.26	107.777	3,906.64	106.83
7/18/2008 4:20 PM - 4:25 PM	10,966.62	310.312	10,949.73	309.835
7/18/2008 4:15 PM - 4:20 PM	280.373	8.212	222.196	6.484
7/18/2008 4:10 PM - 4:15 PM	15.129	2.367	9.549	1.494
7/18/2008 4:05 PM - 4:10 PM	248.526	6.796	198.7	5.434
7/18/2008 4:00 PM - 4:05 PM	1,515.81	41.452	1,492.09	40.804
7/18/2008 3:55 PM - 4:00 PM	118.281	3.235	63.655	1.741
7/18/2008 3:50 PM - 3:55 PM	130.139	3.559	73.368	2.006
7/18/2008 3:45 PM - 3:50 PM	3,852.17	109.129	3,813.58	108.036
7/18/2008 3:40 PM - 3:45 PM	11,754.96	321.45	11,740.34	321.05
7/18/2008 3:35 PM - 3:40 PM	81.567	2.231	42.982	1.175
7/18/2008 3:30 PM - 3:35 PM	154.523	4.226	47.202	1.291
7/18/2008 3:25 PM - 3:30 PM	6,978.52	190.834	6,918.14	189.182
7/18/2008 3:20 PM - 3:25 PM	102.21	2.896	58.651	1.662
7/18/2008 3:15 PM - 3:20 PM	94.145	4.124	72.824	3.19
Total	58,545.32		57,940.26	
Average	2,800.99	68.576	2,773.28	67.819

### Lampiran 6.a Pembacaan Latency Untuk Kondisi Interferensi Tinggi

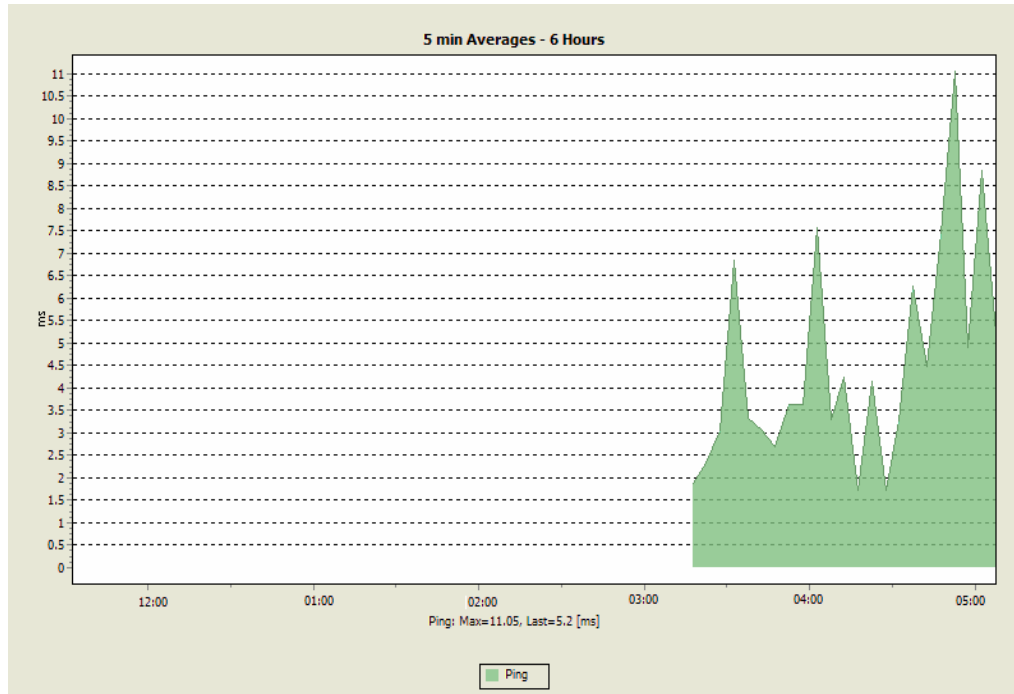


Pembacaan PRTG untuk *latency* pada Router A dengan tingkat interferensi tinggi

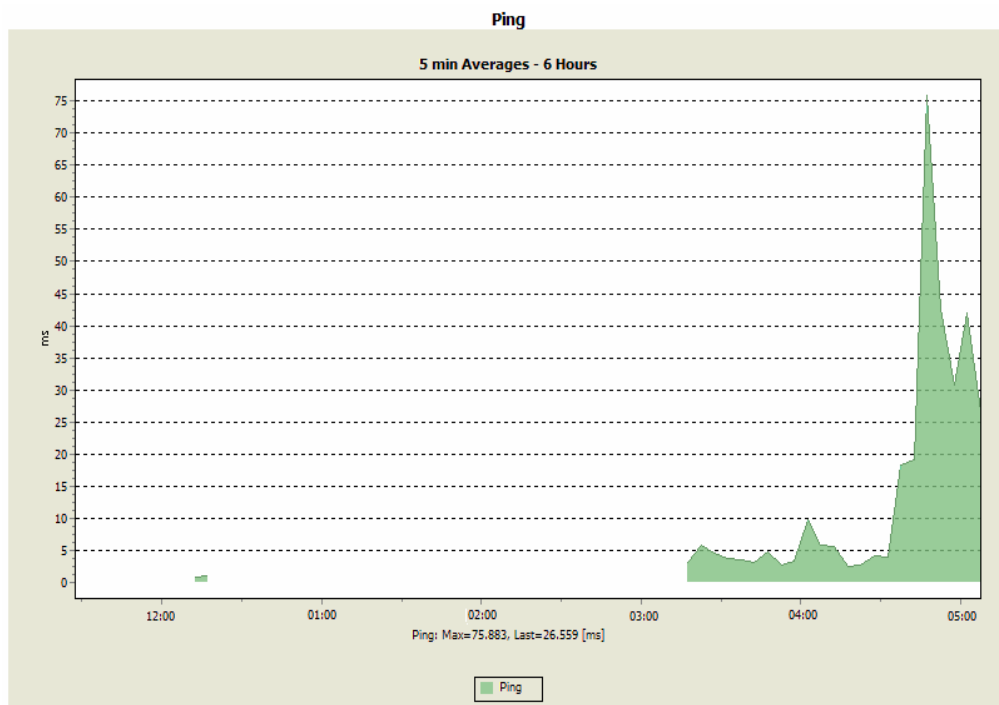


Pembacaan PRTG untuk *latency* pada Router B dengan tingkat interferensi tinggi

(Lanjutan)



Pembacaan PRTG untuk *latency* pada *Router C* dengan tingkat interferensi tinggi



Pembacaan PRTG untuk *latency* pada *Router D* dengan tingkat interferensi tinggi

### Lampiran 6.a : Tabel Pembacaan Latency Untuk Kondisi Interferensi Tinggi

#### Pembacaan PRTG Untuk Latency Router A Dengan Interferensi Tinggi

	Ping
	Latency
	ms
7/18/2008 5:10 PM - 5:15 PM	5.567
7/18/2008 5:05 PM - 5:10 PM	4.186
7/18/2008 5:00 PM - 5:05 PM	10.533
7/18/2008 4:55 PM - 5:00 PM	5.55
7/18/2008 4:50 PM - 4:55 PM	3.783
7/18/2008 4:45 PM - 4:50 PM	8.267
7/18/2008 4:40 PM - 4:45 PM	5.217
7/18/2008 4:35 PM - 4:40 PM	6.4
7/18/2008 4:30 PM - 4:35 PM	1.683
7/18/2008 4:25 PM - 4:30 PM	5.483
7/18/2008 4:20 PM - 4:25 PM	2.119
7/18/2008 4:15 PM - 4:20 PM	5.35
7/18/2008 4:10 PM - 4:15 PM	2.5
7/18/2008 4:05 PM - 4:10 PM	5.417
7/18/2008 4:00 PM - 4:05 PM	5.067
7/18/2008 3:55 PM - 4:00 PM	3.967
7/18/2008 3:50 PM - 3:55 PM	2.033
7/18/2008 3:45 PM - 3:50 PM	2.45
7/18/2008 3:40 PM - 3:45 PM	4.25
7/18/2008 3:35 PM - 3:40 PM	3.25
7/18/2008 3:30 PM - 3:35 PM	4.383
7/18/2008 3:25 PM - 3:30 PM	2.6
7/18/2008 3:20 PM - 3:25 PM	2.22
7/18/2008 3:15 PM - 3:20 PM	4.098
Total	N/A
Average	4.383

#### Pembacaan PRTG Untuk Latency Router B Dengan Interferensi Tinggi

	Ping
	Latency
	ms
7/18/2008 5:10 PM - 5:15 PM	5.083
7/18/2008 5:05 PM - 5:10 PM	4.051
7/18/2008 5:00 PM - 5:05 PM	3.95
7/18/2008 4:55 PM - 5:00 PM	6.983
7/18/2008 4:50 PM - 4:55 PM	8.183
7/18/2008 4:45 PM - 4:50 PM	4.633
7/18/2008 4:40 PM - 4:45 PM	5.983
7/18/2008 4:35 PM - 4:40 PM	1.797
7/18/2008 4:30 PM - 4:35 PM	1.783
7/18/2008 4:25 PM - 4:30 PM	2.567
7/18/2008 4:20 PM - 4:25 PM	1.867
7/18/2008 4:15 PM - 4:20 PM	15.2
7/18/2008 4:10 PM - 4:15 PM	2.75
7/18/2008 4:05 PM - 4:10 PM	3.5
7/18/2008 4:00 PM - 4:05 PM	2.61
7/18/2008 3:55 PM - 4:00 PM	1.864
7/18/2008 3:50 PM - 3:55 PM	3.083
7/18/2008 3:45 PM - 3:50 PM	4.147
7/18/2008 3:40 PM - 3:45 PM	3.05
7/18/2008 3:35 PM - 3:40 PM	3.433
7/18/2008 3:30 PM - 3:35 PM	4.25
7/18/2008 3:25 PM - 3:30 PM	1.35
7/18/2008 3:20 PM - 3:25 PM	8.226
7/18/2008 3:15 PM - 3:20 PM	N/A
Average	4.363

(Lanjutan)

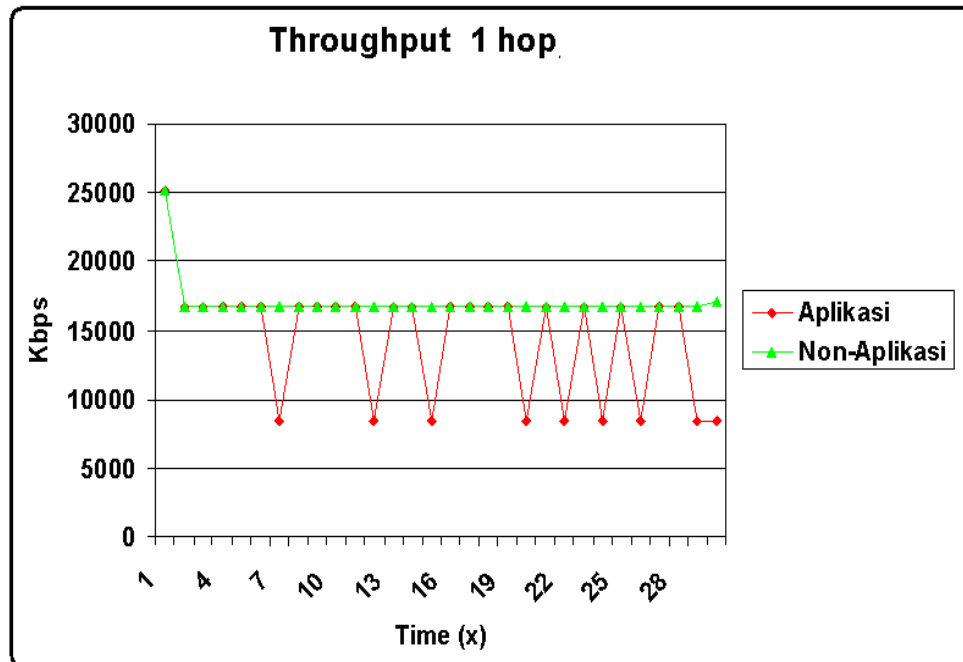
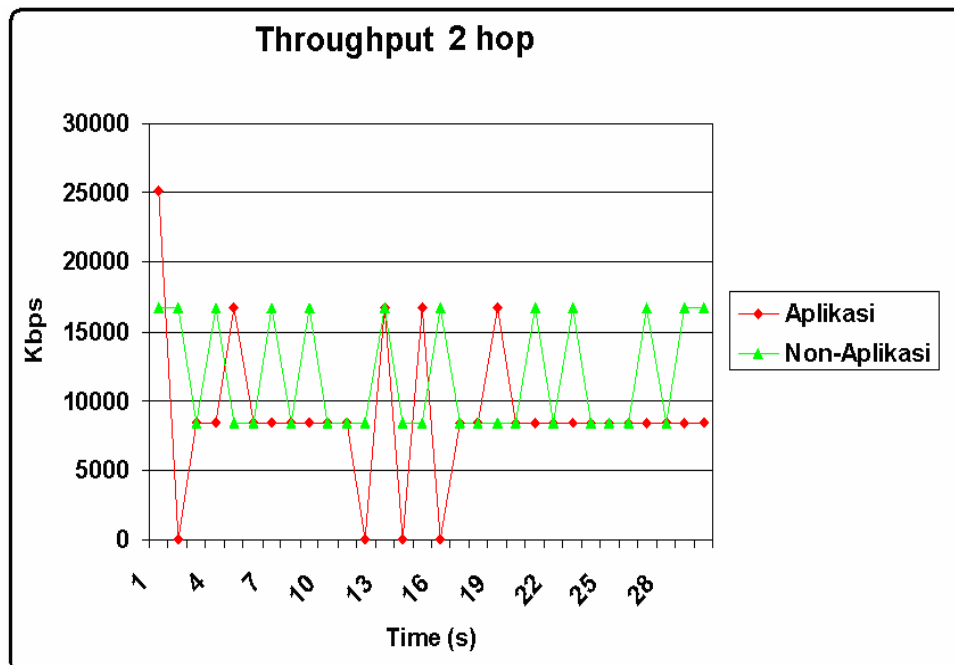
## Pembacaan PRTG Untuk Latency Router C Dengan Interferensi Tinggi

	Ping
	Latency
	ms
7/18/2008 5:10 PM - 5:15 PM	5
7/18/2008 5:05 PM - 5:10 PM	5.2
7/18/2008 5:00 PM - 5:05 PM	8.833
7/18/2008 4:55 PM - 5:00 PM	4.9
7/18/2008 4:50 PM - 4:55 PM	11.05
7/18/2008 4:45 PM - 4:50 PM	7.441
7/18/2008 4:40 PM - 4:45 PM	4.467
7/18/2008 4:35 PM - 4:40 PM	6.254
7/18/2008 4:30 PM - 4:35 PM	3.283
7/18/2008 4:25 PM - 4:30 PM	1.717
7/18/2008 4:20 PM - 4:25 PM	4.133
7/18/2008 4:15 PM - 4:20 PM	1.717
7/18/2008 4:10 PM - 4:15 PM	4.233
7/18/2008 4:05 PM - 4:10 PM	3.3
7/18/2008 4:00 PM - 4:05 PM	7.583
7/18/2008 3:55 PM - 4:00 PM	3.633
7/18/2008 3:50 PM - 3:55 PM	3.627
7/18/2008 3:45 PM - 3:50 PM	2.695
7/18/2008 3:40 PM - 3:45 PM	3.05
7/18/2008 3:35 PM - 3:40 PM	3.333
7/18/2008 3:30 PM - 3:35 PM	6.85
7/18/2008 3:25 PM - 3:30 PM	3.067
7/18/2008 3:20 PM - 3:25 PM	2.317
7/18/2008 3:15 PM - 3:20 PM	1.879
Total	N/A
Average	4.565

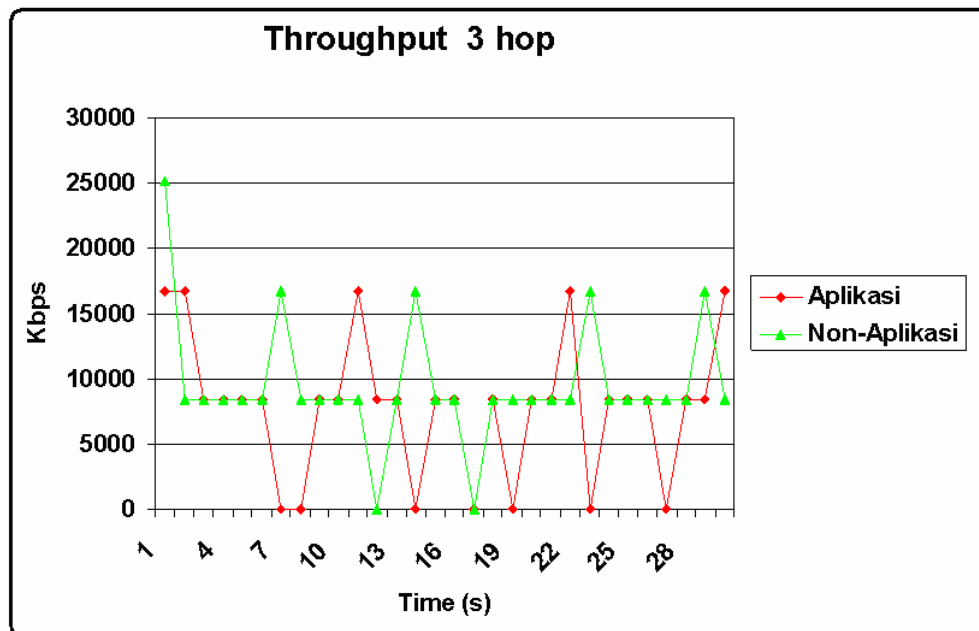
## Pembacaan PRTG Untuk Latency Router D Dengan Interferensi Tinggi

	Ping
	Latency
	ms
7/18/2008 5:10 PM - 5:15 PM	26.559
7/18/2008 5:05 PM - 5:10 PM	42.017
7/18/2008 5:00 PM - 5:05 PM	30.683
7/18/2008 4:55 PM - 5:00 PM	42.6
7/18/2008 4:50 PM - 4:55 PM	75.883
7/18/2008 4:45 PM - 4:50 PM	19.186
7/18/2008 4:40 PM - 4:45 PM	18.233
7/18/2008 4:35 PM - 4:40 PM	3.883
7/18/2008 4:30 PM - 4:35 PM	4.15
7/18/2008 4:25 PM - 4:30 PM	2.75
7/18/2008 4:20 PM - 4:25 PM	2.417
7/18/2008 4:15 PM - 4:20 PM	5.6
7/18/2008 4:10 PM - 4:15 PM	5.836
7/18/2008 4:05 PM - 4:10 PM	9.78
7/18/2008 4:00 PM - 4:05 PM	3.305
7/18/2008 3:55 PM - 4:00 PM	2.729
7/18/2008 3:50 PM - 3:55 PM	4.717
7/18/2008 3:45 PM - 3:50 PM	3.183
7/18/2008 3:40 PM - 3:45 PM	3.508
7/18/2008 3:35 PM - 3:40 PM	3.831
7/18/2008 3:30 PM - 3:35 PM	4.559
7/18/2008 3:25 PM - 3:30 PM	5.797
7/18/2008 3:20 PM - 3:25 PM	3.091
7/18/2008 3:15 PM - 3:20 PM	N/A
Average	13.049

## Lampiran 7 : Performansi End-to-End Throughput

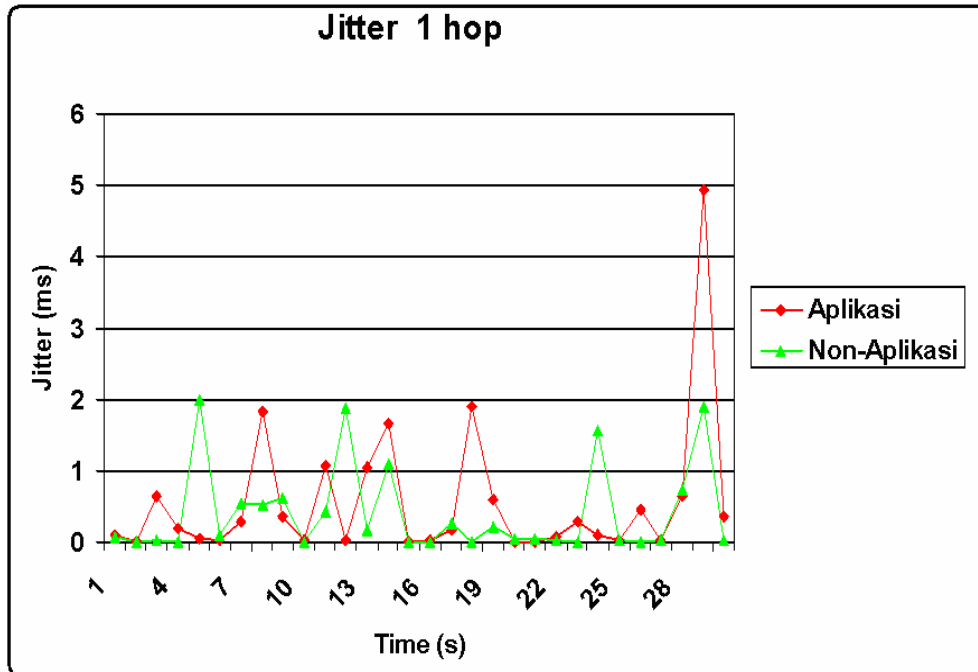
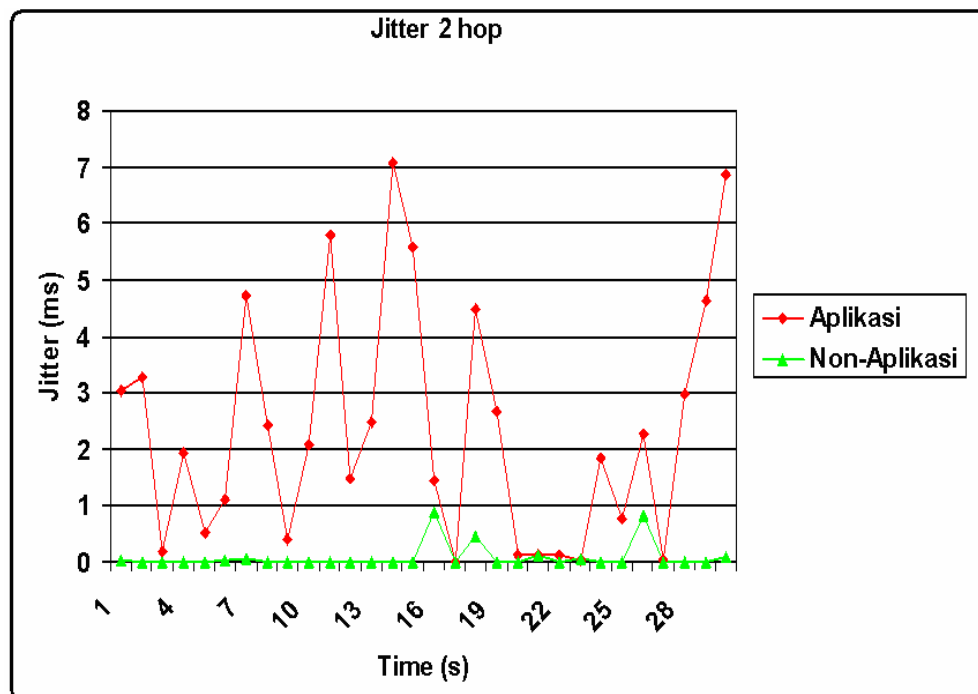
Performansi *throughput* 1 hopPerformansi *throughput* 2 hop

(Lanjutan)

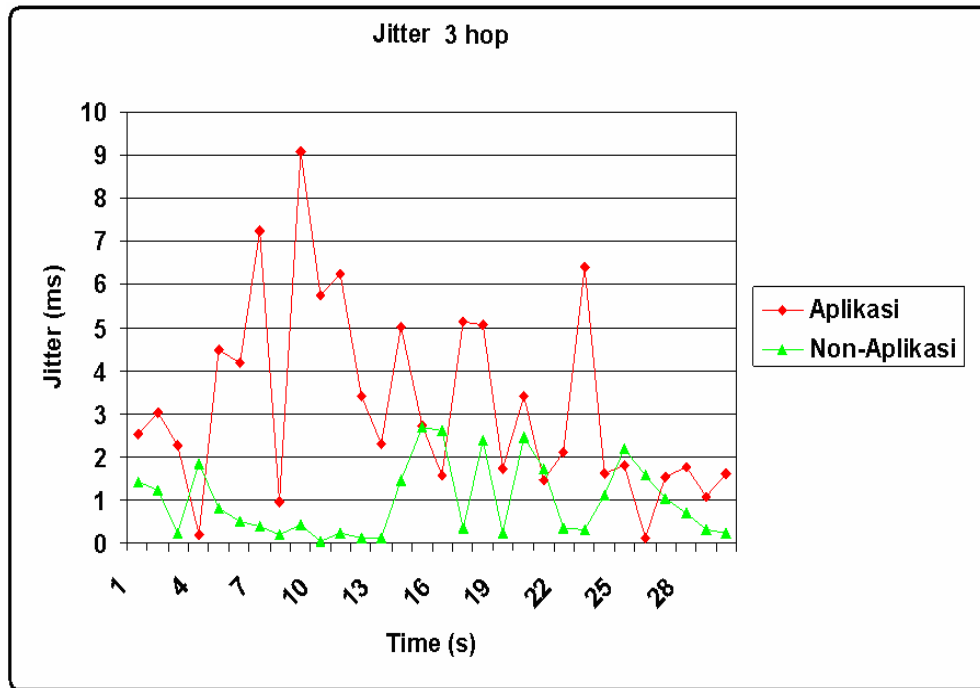
Performansi *throughput* 3 hop



## Lampiran 8 : Performansi End-to-End Jitter

Hasil pengamatan *jitter 1 hop*Hasil pengamatan *jitter 2 hop*

(Lanjutan)

Hasil pengamatan *jitter 3 hop*

