



UNIVERSITAS INDONESIA

**PEMBANGUNAN SIMULASI DAN ANALISA KINERJA
OPTIMALISASI VOIP - SIP DENGAN RESOURCE
RESERVATION PROTOCOL (RSVP)**

SKRIPSI

HENNANDA WULANDARI

06 06 07 396 6

**DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS INDONESIA
DEPOK**

JUNI 2010



UNIVERSITAS INDONESIA

**PEMBANGUNAN SIMULASI DAN ANALISA KINERJA
OPTIMALISASI VOIP - SIP DENGAN RESOURCE
RESERVATION PROTOCOL (RSVP)**

SKRIPSI

**Diajukan sebagai salah satu syarat untuk memperoleh gelar
Sarjana Teknik**

HENNANDA WULANDARI

06 06 07 396 6

**DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS INDONESIA
DEPOK**

JUNI 2010

HALAMAN PERNYATAAN ORISINALITAS

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

Nama : Hennanda Wulandari

NPM : 0606073966

Tanda Tangan :

Tanggal : Juni 2010

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :

Nama : Hennanda Wulandari

NPM : 0606073966

Program Studi : Teknik Elektro

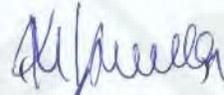
Judul Skripsi :

**PEMBANGUNAN SIMULASI DAN ANALISA KINERJA OPTIMALISASI
VOIP - SIP DENGAN RESOURCE RESERVATION PROTOCOL (RSVP)**

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

DEWAN PENGUJI

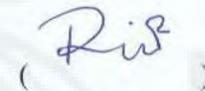
Pembimbing : Prof. Dr. Ing Kallamullah Ramli, M. Eng.



Penguji : Prof. Dr. Ir. Bagio Budiarjo, M. Sc.



Penguji : Prof. Dr. Ir. Riri Fitri Sari, MSc, MM.



Ditetapkan di : Depok

Tanggal : 28 Juni 2010

KATA PENGANTAR

Puji syukur saya panjatkan kepada Tuhan Yang Maha Esa, karena atas berkat dan rahmat-Nya, saya dapat menyelesaikan skripsi ini. Penulisan skripsi 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 skripsi ini, sangatlah sulit bagi saya untuk menyelesaikan skripsi ini. Oleh karena itu, saya ingin mengucapkan terima kasih kepada:

- (1) Prof. Dr. Ing Kallamullah Ramli, M.Eng, selaku dosen pembimbing yang telah menyediakan waktu, tenaga, dan pikiran untuk mengarahkan saya dalam penyusunan skripsi ini;
- (2) Papa, mama, dan semua keluarga saya yang telah memberikan bantuan dukungan baik material dan moral; dan
- (3) Sahabat-sahabat yang telah banyak membantu saya dalam menyelesaikan skripsi ini.

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

Depok, Juni 2010

Hennanda Wulandari

**HALAMAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK
KEPENTINGAN AKADEMIS**

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

Nama : Hennanda Wulandari

NPM : 0606073966

Program Studi : Teknik Elektro

Departemen : Teknik Elektro

Fakultas : Teknik

Jenis Karya : Skripsi

demikian pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif (Non-exclusive Royalty-Free Right)** atas karya ilmiah saya yang berjudul :

**PEMBANGUNAN SIMULASI DAN ANALISA KINERJA OPTIMALISASI
VOIP - SIP DENGAN RESOURCE RESERVATION PROTOCOL (RSVP)**

berserta perangkat yang ada. Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/format-kan, mengelola dalam bentuk pangkalan (*database*), merawat, dan memublikasikan tugas akhir saya selama dalam tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok

Tanggal : Juni 2010

Yang menyatakan,

Hennanda Wulandari

ABSTRAK

Nama : Hennanda Wulandari

Program Studi : Teknik Elektro

Judul :

PEMBANGUNAN SIMULASI DAN ANALISA KINERJA OPTIMALISASI VOIP - SIP DENGAN RESOURCE RESERVATION PROTOCOL (RSVP)

Skripsi ini membahas mengenai pembangunan simulasi dan perbandingan QoS objektif pada jaringan VoIP berbasis SIP dengan variasi penggunaan protokol RSVP, codec, serta besarnya bandwidth yang digunakan. Variasi codec yang digunakan yaitu G.711, GSM dan G.729 serta variasi bandwidth 64 Kbps dan 128 Kbps dengan utilitas jaringan sebesar 20 %. Parameter pengukuran yang diukur adalah delay dan jitter. Dari hasil simulasi diperoleh bahwa pada jaringan tanpa RSVP, codec G.711 memiliki nilai delay dan jitter yang paling besar. Sedangkan pada jaringan yang menggunakan RSVP, dapat dilihat bahwa nilai delay dan jitter menjadi lebih baik dibandingkan dengan tanpa menggunakan RSVP. Hal ini karena RSVP mereservasi sumber daya yang ada terlebih dahulu sehingga bisa mengurangi nilai delay dan jitter.

Kata Kunci : VoIP-SIP, RSVP, *QoS*

ABSTRACT

Name : Hennanda Wulandari
Study Program : Electrical Engineering
Tittle :

SIMULATION AND PERFORMANCE ANALYSIS OF VOIP – SIP WITH RESOURCE RESERVATION PROTOCOL (RSVP)

The focus of this study is to create simulation and to compare the objective QoS in SIP based VoIP with variations in RSVP protocol, codec, and also bandwidth. The codec variations are G.711, GSM and G.729 at 64 Kbps and 128 Kbps with 20 % network utility. The measured QoS parameters are delay and jitter. From the simulation, it is known that in the network without RSVP, G.711 has the biggest delay and jitter. Besides, in the network with RSVP, it is known that the value of delay and jitter are better than in the network without RSVP. It is happened because RSVP will reserve network resources so that delay and jitter can be reduced

Keywords : VoIP-SIP, RSVP, QoS

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PERNYATAAN ORISINALITAS	ii
HALAMAN PENGESAHAN	iii
KATA PENGANTAR	iv
HALAMAN PERSETUJUAN PUBLIKASI	v
ABSTRAK	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR GAMBAR	x
DAFTAR TABEL	xii
BAB 1 PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Tujuan	1
1.3. Pembatasan Masalah	2
1.4. Metodologi Penelitian	2
1.5. Sistematika Penulisan	3
BAB 2 TEORI DASAR	4
2.1. Voice Over Internet Protocol (VoIP)	4
2.2. Session Initiation Protocol (SIP)	7
2.3. Resource Reservation Protocol (RSVP)	11
2.4. Quality of Service (QoS)	14
2.5. Codec	19
BAB 3 PERANCANGAN DAN IMPLEMENTASI JARINGAN	23
3.1. Diagram Alir Perancangan dan Implementasi	23
3.2. Perancangan Topologi dan Skenario	24
3.3. Pembangunan Topologi Jaringan	26

3.3.1. Interface Hardware	27
3.3.2. Interface Software	27
3.3.3. Instalasi OPNET	27
3.3.4. Pembangunan Topologi	28
3.3.2. Konfigurasi Aplikasi dan Profile	28
3.3.6. Mendefinisikan Sumber dan Tujuan	30
3.3.7. Konfigurasi RSVP	31
BAB 4 ANALISA HASIL SIMULASI	33
4.1. Delay	33
4.2. Jitter	37
BAB 5 KESIMPULAN	41
DAFTAR ACUAN	42

DAFTAR GAMBAR

Gambar 2.1. Arsitektur jaringan VoIP	5
Gambar 2.2. Arsitektur SIP	7
Gambar 2.3. Proses koneksi dengan proxy server	9
Gambar 2.4. Proses koneksi dengan redirect server	10
Gambar 2.5. Proses registrasi	11
Gambar 2.6. Proses reservasi RSVP	12
Gambar 2.7. Sumber delay	16
Gambar 2.8. Akibat adanya jitter	18
Gambar 2.9. Akibat adanya packet loss	19
Gambar 3.1. Diagram alir perancangan dan simulasi	23
Gambar 3.2. Topologi Jaringan	25
Gambar 3.3. OPNET IT Guru Edition	28
Gambar 3.4. Pemodelan trafik	29
Gambar 3.5. RSVP parameters	32
Gambar 4.1. Grafik delay tanpa RSVP	33
Gambar 4.2. Grafik perbandingan delay tanpa RSVP	34
Gambar 4.3. Grafik perbandingan delay tanpa dan dengan RSVP pada bandwidth 64	35
Gambar 4.4. Grafik perbandingan delay tanpa dan dengan RSVP pada bandwidth 128	36
Gambar 4.5. Grafik perbandingan delay dengan RSVP	37
Gambar 4.6. Grafik jitter tanpa RSVP pada bandwidth 64	38
Gambar 4.7. Grafik jitter tanpa RSVP pada bandwidth 128	38
Gambar 4.8. Grafik perbandingan jitter tanpa dan dengan RSVP pada bandwidth 64	39

Gambar 4.8. Grafik perbandingan jitter tanpa dan dengan RSVP pada

bandwidth 128

40



DAFTAR TABEL

Tabel 2.1. Tabel reservasi RSVP	13
Tabel 2.2. Nilai kualitas MOS	15
Tabel 2.3. Pedoman batasan delay	17
Tabel 2.4. Beberapa jenis codec pada VoIP	20

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Seiring dengan majunya industri telekomunikasi, semakin banyak pula bermunculan teknologi yang menawarkan kemudahan dan efisiensi. Protokol merupakan salah satu teknologi yang diciptakan untuk memenuhi kebutuhan akan suatu jaringan yang handal dan optimal tanpa harus terus bergantung dengan kemampuan suatu perangkat. Demikian juga dengan protokol – protokol yang ada sekarang, jumlahnya semakin banyak dan beragam serta yang dapat disesuaikan dengan kebutuhan penggunanya. Oleh karena itu, agar dapat menciptakan suatu jaringan yang efisien, diperlukan suatu pemahaman tentang bagaimana cara kerja suatu protokol dalam sebuah jaringan, bagaimana protokol tersebut membantu jaringan agar tetap bekerja sesuai dengan harapan penggunanya dengan investasi perangkat keras dan lunak yang seminimal mungkin sehingga pengeluaran modal dan operasional dapat ditekan dan memperoleh keuntungan maksimal. Hal ini sangatlah krusial bagi seorang network engineer yang akan merancang dan mengelola suatu jaringan.

Pada tugas akhir ini akan dilakukan simulasi bagaimana protokol RSVP bekerja pada suatu jaringan IP serta peran protokol tersebut dalam meningkatkan kinerja aplikasi VoIP-SIP menggunakan beberapa variasi codec dan kondisi bandwidth jaringan.

1.2 Tujuan

- Mengetahui bagaimana suatu aplikasi audio bekerja dalam suatu jaringan dan nilai – nilai yang dapat menentukan kualitasnya.
- Menentukan efek dari penggunaan codec yang berbeda
- Memahami efek dari adanya perbedaan bandwidth jaringan.

- Mengetahui bagaimana protokol RSVP mempengaruhi kinerja VoIP-SIP pada suatu jaringan IP.

1.3 Pembatasan Masalah

Hanya menganalisa pengaruh penggunaan protokol RSVP terhadap aplikasi VoIP-SIP menggunakan 3 jenis codec, yaitu : G.711 , G.729 dan GSM dengan bandwidth 64 Kbps dan 128 Kbps pada software network simulator OPNET IT Guru Edition. Parameter QoS yang dianalisa adalah delay dan jitter. Hasil yang ditampilkan adalah berdasarkan pada hasil yang didapat penulis saat melakukan simulasi dengan kondisi sesuai dengan skenario yang telah ditetapkan.

1.4 Metodologi Penelitian

- Studi Literatur

Dilakukan dengan cara membaca berbagai tulisan dan dokumentasi tentang VoIP dan QoS serta mengenai software OPNET.

- Melakukan Simulasi

Membangun jaringan VoIP-SIP dengan menggunakan protocol RSVP pada simulator OPNET IT Guru Edition.

- Pengambilan Data

Mengukur kualitas VoIP-SIP dengan memperhatikan parameter QoS berupa delay dan jitter.

- Analisis.

Melakukan analisa terhadap hasil yang didapatkan.

1.5 Sistematika Penulisan

BAB I PENDAHULUAN

Menjelaskan latar belakang, tujuan, pembatasan masalah, metodologi penelitian dan sistematika penulisan.

BAB II DASAR TEORI

Menjelaskan teori tentang VoIP, arsitektur SIP, codec yang digunakan, serta parameter QoS pada VoIP.

BAB III SIMULASI

Membahas mengenai apa saja yang dibutuhkan untuk melakukan simulasi, bagaimana cara membangun jaringan simulasi, serta scenario yang akan dilakukan.

BAB IV PENGAMBILAN DATA DAN ANALISIS

Membahas mengenai hasil simulasi berdasarkan scenario yang telah dibuat dan analisis parameter QoSnya.

BAB V KESIMPULAN

Memaparkan beberapa kesimpulan yang diperoleh dari hasil simulasi dan analisisnya.

BAB 2

TEORI DASAR

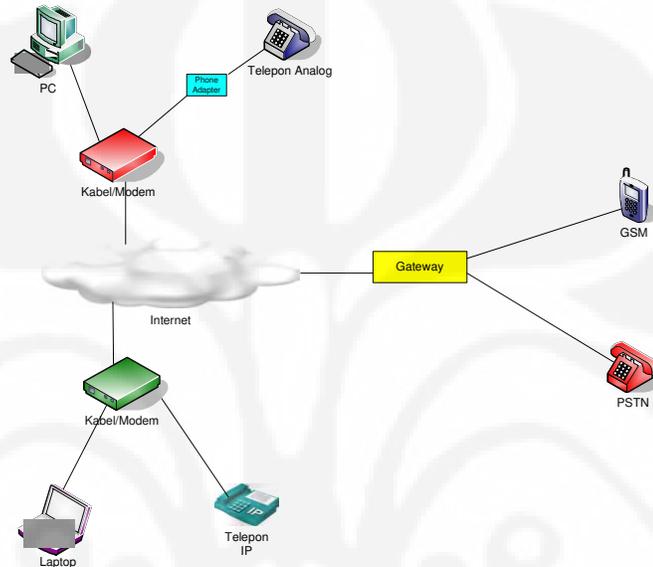
2.1 Voice Over Internet Protocol (VOIP)

Voice over internet protocol adalah suatu teknologi yang memiliki kemampuan untuk melakukan pertukaran suara secara interaktif melalui *internet protocol* [1]. Data suara diubah ke bentuk digital menggunakan kompresi audio, lalu dimasukkan ke dalam paket dan dikirim menggunakan jaringan berbasis IP. Seluruh paket dirutekan tanpa jaminan bahwa paket tersebut akan melewati jalur yang sama. Selama transmisi, paket – paket tersebut kemungkinan mengalami *delay*, *lost*, maupun *error*. Setelah paket – paket ditransmisikan dan tiba di tujuan, paket – paket tersebut disatukan kembali dan di dekompresi untuk membentuk ke bentuk aslinya. Banyak istilah lain yang digunakan untuk menyebut VoIP ini seperti *IP telephony*, *internet telephony*, *voice over broadband (VoBB)*, *broadband telephony* dan *broadbandphone* [2]. Istilah *internet telephony* mengacu kepada layanan komunikasi berupa suara, faksimili dan aplikasi pesan suara yang dialirkan melalui internet, bukan melalui jalur PSTN biasa.

Suatu sistem VoIP memerlukan protokol untuk mengontrol suatu sesi pada saat penyambungan dan pemutusan sambungan serta membutuhkan codec suara yang dapat meng-encode dan men-decode sinyal suara. Pada setiap implementasinya,, penggunaan codec untuk aplikasi VoIP sangat beragam tergantung kepada kebutuhan pengguna. Codec dengan bit rate yang kecil biasanya digunakan pada jaringan yang memiliki ketersediaan bandwidth yang sedikit, sedangkan codec dengan bit rate yang besar digunakan untuk kebutuhan akan suara dengan kualitas baik.

Semua kemampuan yang ada pada telepon konvensional juga terdapat pada VoIP bahkan dengan beberapa fitur – fitur tambahan lain seperti *voicemail*, *instant messaging* dan *video conference*. Bentuk nomor telepon pada VoIP dapat berupa beberapa digit nomor seperti pada PSTN, alamat – alamat, atau suatu alias.

Pada PSTN, digit nomor telepon digunakan langsung untuk menemukan lokasi telepon yang dituju. Sedangkan pada VoIP, nomor tersebut diubah terlebih dahulu ke alamat IP sebanyak 32 bit atau 128 bit yang baru kemudian akan disambungkan dengan tujuan. Gambar 2.1 di bawah ini menjelaskan tentang gambaran umum dari teknologi VoIP.



Gambar 2.1. Arsitektur jaringan VoIP

User atau *end point* pada VoIP dapat berupa *hardphone* yang bentuknya seperti telepon biasa, contohnya adalah *IP phone* dan telepon analog, ataupun berupa *softphone* yang diinstal pada PC (*personal computer*). Oleh karena itu, terdapat empat mode komunikasi dalam VoIP yaitu *phone to phone*, *phone to PC*, *PC to Phone* dan *PC to PC*. Agar dapat terhubung ke jaringan VoIP, komputer dan IP phone hanya memerlukan akses ke jaringan internet, sedangkan telepon analog memerlukan alat tambahan berupa phone adapter. Sementara itu, agar VoIP bisa terhubung dengan user di jaringan lain seperti PSTN atau GSM, maka VoIP perlu disambungkan ke *gateway* terlebih dahulu.

Beberapa kelebihan penggunaan VoIP dibandingkan dengan telepon konvensional :

- Penggunaan *bandwidth* yang lebih efisien dari telepon biasa. Telepon konvensional menggunakan sambungan 64 kbps untuk setiap sambungan telepon, sedangkan VoIP yang didukung dengan teknik pemampatan suara, dapat membagi *bandwidth* tersebut untuk beberapa sambungan.
- Biaya yang relatif lebih rendah untuk sambungan langsung jarak jauh maupun sambungan langsung internasional. Hal ini merupakan kelebihan utama VoIP dibandingkan dengan telepon konvensional. Pada VoIP hanya diperlukan sambungan internet untuk melakukan komunikasi suara.
- Menggunakan infrastruktur jaringan data yang ada untuk komunikasi suara. Jaringan yang sudah ada dapat dibangun jaringan VoIP sehingga dapat mengurangi biaya operasional telepon bulanan dan peralatan.
- Memungkinkan digabung dengan jaringan telepon lokal yang sudah ada. Dengan adanya *gateway*, jaringan VoIP dapat disambungkan dengan jaringan PSTN yang telah ada sebelumnya.

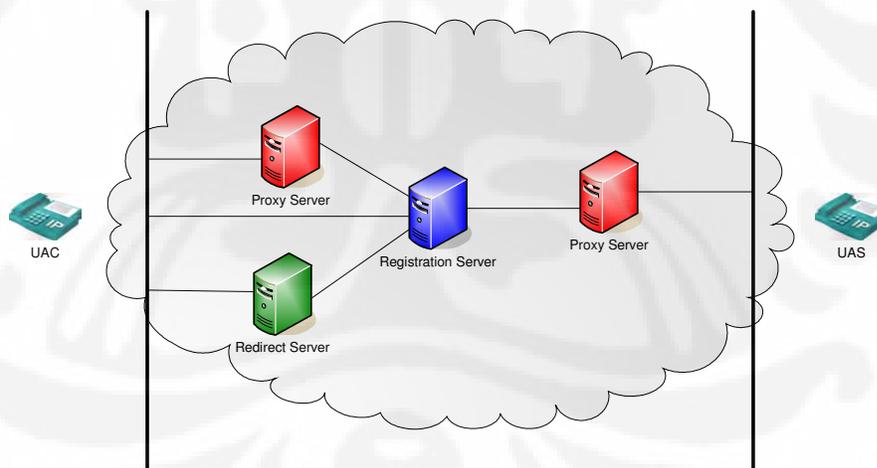
Sedangkan kekurangan penggunaan VoIP, diantaranya :

- Kualitas suara tidak sejernih PSTN sebagai efek dari kompresi suara. Dengan adanya kompresi suara untuk meminimalkan penggunaan *bandwidth* maka kualitas suara VoIP tidak akan sejernih telepon konvensional. Namun hal ini bisa diminimalisasi dengan menggunakan koneksi *broadband*.
- Adanya jeda dalam proses komunikasi. Adanya *delay* pada saat kompresi dan dekompresi, *delay* jaringan, *delay* peralatan dan *jitter* mengakibatkan adanya jeda dalam berkomunikasi menggunakan VoIP. Hal ini juga dapat dihindari dengan menggunakan koneksi *broadband*.
- Belum ada jaminan kualitas jika VoIP melewati internet.
- Peralatan relatif mahal. Peralatan VoIP yang menghubungkan VoIP dengan PABX relatif berharga mahal. Diharapkan dengan makin populernya VoIP maka harga peralatannya juga ikut turun.
- Pengaturan yang tidak baik dapat menyebabkan terganggu saat jumlah pemakai VoIP meningkat. Pengaturan *bandwidth* perlu agar jaringan tidak menjadi jenuh akibat pemakaian VoIP.

- Sistem penomoran yang perlu di maintain secara teratur.

2.2 Session Initiation Protocol (SIP)

Session Initiation Protocol adalah salah satu standar IETF untuk protocol pensinyalan dan banyak digunakan sebagai pengontrol sesi komunikasi multimedia seperti panggilan *voice* dan *video* melalui *internet protocol* [3]. SIP merupakan protokol layer aplikasi yang berfungsi untuk manajemen pengaturan dan pemutusan panggilan. SIP dapat membuat, memodifikasi, dan mengakhiri sesi multimedia seperti konferensi multimedia, panggilan telepon internet dan aplikasi – aplikasi serupa, baik unicast maupun multicast. Karena SIP termasuk salah satu signaling protocol VoIP, maka SIP bukanlah protokol yang membawa paket data, suara, maupun video. Protokol ini merupakan elemen permanen dalam arsitektur *IP Multimedia Subsystem* (IMS) untuk *streaming multimedia* berbasis IP dalam sistem selular. Pada Gambar 2.2 diperlihatkan bagaimana arsitektur SIP.



Gambar 2.2. Arsitektur SIP

Protokol SIP didukung oleh beberapa protokol antara lain RSVP (*Resource Reservation Protocol*) untuk melakukan pemesanan jaringan, RTP (*Real time Transport Protocol*) dan RTCP (*Real time Transport Control Protocol*)

untuk melakukan transmisi media dan mengetahui kualitas layanan, serta SDP (*Session Description Protocol*) untuk mendeskripsikan sesi media. Arsitektur SIP terdiri dari 2 komponen dasar, yaitu *user agent* dan *server SIP*.

- *User Agent*

Merupakan terminal akhir yang bertindak sesuai keinginan dari user. Dapat berupa *software (softphone)* maupun *hardware (hardphone)*. Terbagi menjadi 2 yaitu :

1. *User Agent Client (UAC)* : berfungsi untuk melakukan inisiasi *request* ke *server SIP*.
2. *User Agent Server (UAS)* : berfungsi untuk menerima dan memberikan respon terhadap *request SIP*.

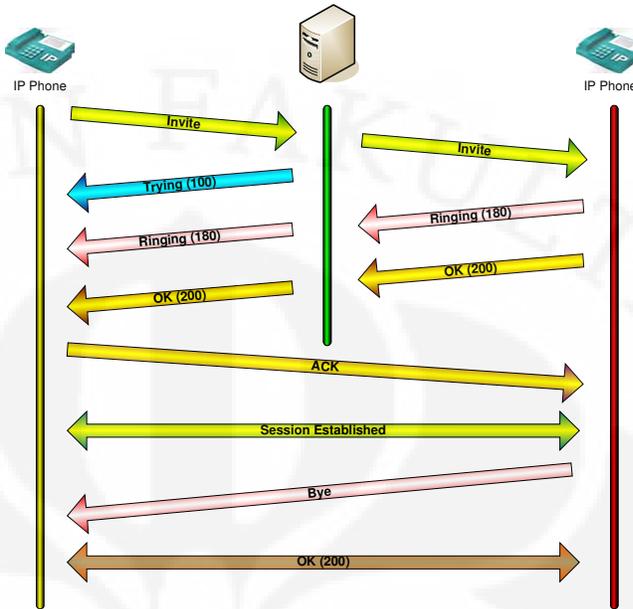
Baik UAC maupun UAS dapat menutup sesi komunikasi.

- *Server SIP*

Komponen penengah antar *user agent*. Ada berbagai jenis *server SIP* untuk membantu layanan dan pengaturan panggilan SIP.

1. *Proxy Server* : menerima *request SIP* dan meneruskan ke UAS atau *server* lain yang dituju yang memiliki informasi tentang *user* yang dipanggil. Gambar 2.3 menjelaskan bagaimana proses koneksi pada *proxy server*.

Ketika sebuah *request message* dikirim dari client (UAC) ke server (UAS) dan ditulis dalam *cleartext* karena SIP merupakan *text-based protocol*. Mula – mula *client* akan mengirimkan pesan INVITE kepada *proxy server* untuk membangun sebuah sesi SIP, yaitu dengan mengundang *user agent* lain untuk bergabung dalam sesi komunikasi. Pesan tersebut lalu diteruskan oleh *server* ke *user agent* tujuan, lalu mengirimkan *response message* Trying (100) kepada client yang mengindikasikan bahwa request yang tadi dikirimkan sudah diterima dan sedang diproses.

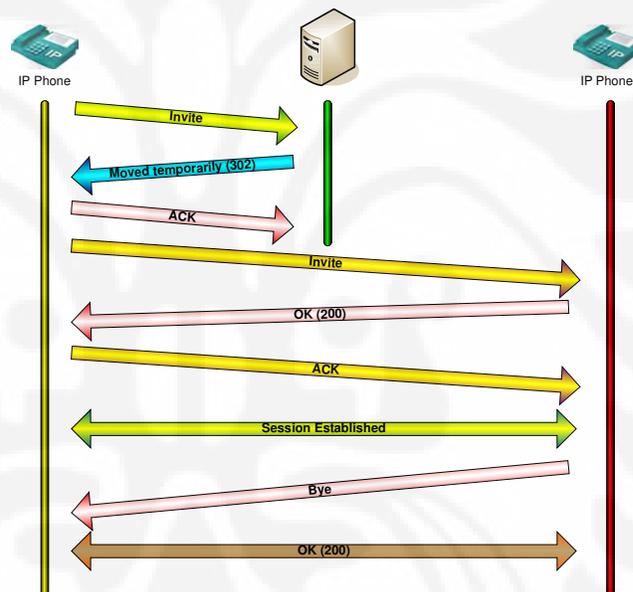


Gambar 2.3. Proses koneksi dengan *proxy server*

Bila request telah sampai, penerima akan mengirimkan response message berupa Ringing (180) yang artinya request sedang diproses kepada server dan akan diteruskan server ke client. Apabila request tersebut diterima, maka penerima akan mengirimkan OK (200) kepada client melalui server. Pesan ACK digunakan untuk konfirmasi bahwa user agent telah menerima pesan terakhir dari serangkaian pesan INVITE sehingga dapat dilakukan suatu sesi komunikasi antara UAC dan UAS. Ketika salah satu user ingin mengakhiri komunikasi, maka ia akan mengirimkan pesan BYE kepada server dan diteruskan ke user yang lain. Lalu user yang menerima pesan tersebut akan mengirimkan respons OK (200) sebagai tanda diakhirinya sebuah sesi komunikasi.

Request dapat dilayani sendiri atau disampaikan (forward) kepada server lain. Sebelum disampaikan kepada tujuan lain, terlebih dahulu proxy server ini akan menerjemahkan dan/atau menulis ulang request message. Server ini juga menyimpan state sesi komunikasi antara UAS dan UAC.

2. *Redirect server* : menerima *request* SIP, menentukan UAS / *server* yang dituju selanjutnya dan mengembalikan alamat UAS / *server* yang dituju selanjutnya kepada *client* daripada meneruskan *request* ke *server* yang dituju tersebut. Tidak seperti *proxy server*, *redirect server* tidak dapat menerima dan menutup sesi komunikasi dan tidak seperti UAS, *redirect server* juga tidak dapat menerima dan menutup sesi komunikasi. Server ini juga tidak menyimpan *state* sesi komunikasi antara UAC dan UAS setelah pengembalian alamat kepada UAC. Gambar 2.4 menjelaskan bagaimana proses koneksi pada *redirect server*.

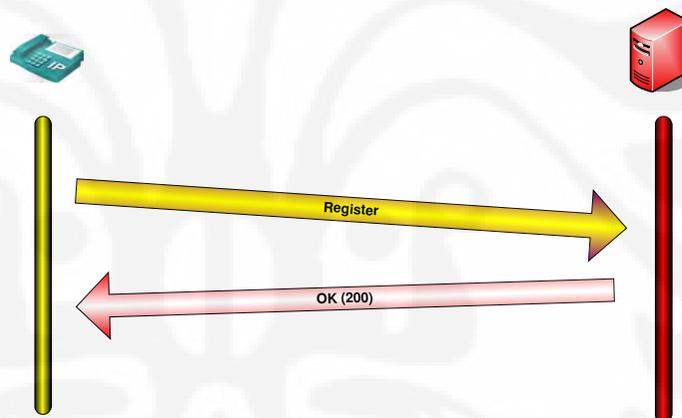


Gambar 2.4. Proses koneksi dengan redirect server

Awalnya client akan mengirim request INVITE kepada redirect server. Namun pesan itu tidak akan diteruskan, melainkan hanya akan dipetakan dan ditentukan user agent atau server tujuan, lalu alamatnya akan dikirimkan kembali kepada client. Client yang menerima hasil pemetaan akan mengirimkan response message ACK ke redirect server, lalu mengirimkan kembali request INVITE kepada user agent / server yang dituju. Ketika request tersebut telah diterima,

maka akan dikirimkan response message OK (200) kepada client. Client merespon kembali dengan mengirim pesan ACK sebagai pemberitahuan kepada penerima sehingga suatu sesi komunikasi dapat terjadi. Proses pengakhiran sebuah sesi pada redirect server ini sama seperti yang terjadi pada proxy yaitu salah satu user akan mengirimkan pesan BYE dan akan direspon oleh penerima dengan mengirimkan OK (200) sebagai tanda berakhirnya suatu sesi.

3. *Registration server* : menerima request registrasi dari user SIP dan melakukan update terhadap lokasi user yang berupa IP dan port agar dapat dihubungi oleh komponen SIP yang lain. Gambar 2.5 menunjukkan proses registrasi pada registration server.

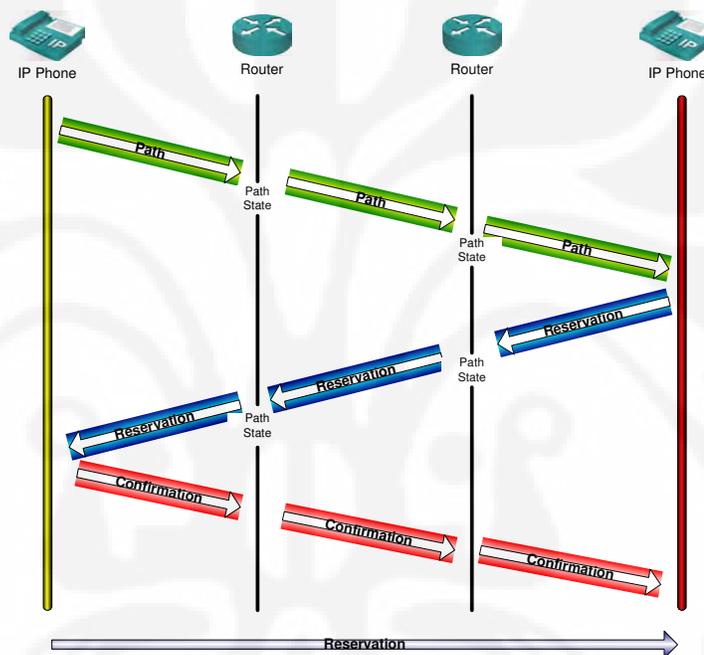


Gambar 2.5. Proses registrasi

2.3 Resource reSerVation Protocol (RSVP)

RSVP adalah *signalling protocol* pada *layer network* yang dapat melakukan reservasi pada sumber daya jaringan untuk suatu aplikasi. Protokol ini digunakan ketika trafik yang dihasilkan sensitif terhadap delay (contohnya pada aplikasi VoIP) karena RSVP akan mereservasi sumber daya yang ada sehingga menyebabkan delay dan jitter yang dihasilkan lebih rendah. Protokol ini juga

merupakan *hop-by-hop QoS signaling protocol* yang artinya RSVP akan mendistribusikan permintaan QoS ke tiap node yang akan dilewati data sehingga node – node tersebut akan melakukan reservasi [4]. RSVP digunakan oleh host untuk meminta QoS pada network untuk data aplikasi tertentu dan oleh router untuk mengirimkan permintaan QoS ke seluruh node yang dilewati data serta untuk menjaga kondisi agar permintaan QoS dapat terpenuhi. Gambar 2.6 menjelaskan proses reservasi pada RSVP.



Gambar 2.6. Proses reservasi RSVP

Saat pengirim mengirimkan data, modul RSVP yang ada pada sisi pengirim akan mengirimkan RSVP path messages yang akan menjelaskan trafik yang dihasilkan oleh pengirim dan membuat path state pada tiap node yang dilewati data. Path messages ini akan dirouting bersama data menuju ke alamat penerima. Path messages ini akan membuat path state pada tiap router yang dilewati. Dengan mekanisme seperti ini maka seluruh peralatan yang dilewati akan mengenali node tetangganya yang akan dilewati oleh data.

Ketika modul RSVP pada penerima menangkap RSVP path messages tersebut, ia akan menyampaikan pesan itu kepada aplikasi penerima dan aplikasi penerima inilah yang akan memutuskan apakah sumber daya yang ada harus dipeservasi atau tidak. Ketika permintaan disetujui, maka aplikasi penerima akan meminta kepada modul RSVP untuk membantunya dalam mereservasi sumber daya. Lalu protocol RSVP akan membawa permintaan reservasi berupa Resv messages ke seluruh node kembali ke sumber data. Proses reservasi tersebut dilakukan secara hop-by-hop sehingga setiap node dapat memeriksa apakah sumber daya yang ada cukup untuk memenuhi permintaan reservasi. Bila proses reservasi berhasil, tiap node akan membuat Resv state dan meneruskan permintaan reservasi tersebut ke node selanjutnya.

Pihak penerima dapat meminta *notification* tentang status permintaan reservasinya sehingga ketika pengirim menerima Resv message, ia akan langsung mengirimkan Resv Confirmation message kepada penerima. Jika penerima mengirim data maka ia akan mengirimkan Path messages kepada pengirim dan mengulang mekanisme ini dari awal dengan penerima sebagai pengirim dan pengirim sebagai penerima.

Saat terjadi multicast atau terdapat banyak pengirim dengan data flow yang sama, maka penerima akan memperlakukan trafik - trafik yang datang dengan beberapa jenis perlakuan, yaitu Fixed Filter (FF), Shared Explicit (SE) dan Wildcard Filter (WF). Perlakuan yang akan dijalankan tergantung kepada konfigurasi dari reservation option seperti yang digambarkan pada Tabel 2.1.

Tabel 2.1. Tabel reservasi RSVP

	<i>Reservation Option</i>		
		Distinct	Shared
<i>Sender Selection</i>	Explicit	Fixed Filter (FF)	Shared Explicit (SE)
	Wildcard	-	Wildcard Filter (WF)

Reservasi untuk pengirim yang berbeda pada sesi yang sama dapat berupa distinct atau shared. Distinct akan membuat reservasi yang berbeda bagi tiap pengirim, sedangkan *shared* akan membuat suatu reservasi yang dipakai bersama. Selain itu juga dalam pemilihan pengirim terdapat *explicit* dan *wildcard*. *Explicit* hanya akan memilih pengirim yang telah ditentukan sebelumnya sedangkan *wildcard* akan memilih seluruh pengirim dalam suatu sesi.

Reservasi shared, yaitu WF dan SE, cocok untuk aplikasi multicast yang pengirimnya tidak akan melakukan pengiriman data secara terus – menerus. Contoh aplikasi yang cocok untuk shared adalah aplikasi packetized audio karena sedikit orang yang akan berbicara pada waktu bersamaan. Sedangkan FF lebih cocok untuk aplikasi video karena ia akan membuat reservasi yang berbeda untuk setiap pengirim.

2.4 Quality of Service (QoS)

QoS singkatan dari Quality of Service yang merupakan jaminan kualitas atas layanan. QoS sangatlah penting agar suatu aplikasi dapat berjalan dengan baik. Real time applications, seperti VoIP memiliki karakteristik dan kebutuhan yang berbeda dibandingkan dengan aplikasi data. Aplikasi ini memiliki sedikit toleransi terhadap delay. Packet loss dan jitter juga sangat menentukan kualitas dari layanan VoIP ini. QoS bertujuan untuk menyediakan kualitas layanan yang berbeda – beda untuk beragam kebutuhan akan layanan di dalam jaringan IP, contohnya untuk menyediakan bandwidth khusus, menurunkan hilangnya paket – paket, menurunkan waktu tunda dan variasi waktu tunda dalam proses transmisi.

Qos dapat dilihat secara subjektif dan objektif. Secara subjektif, tingkat kualitas dari suatu layanan diukur berdasarkan subjektifitas masing – masing pengguna. Setiap pengguna dapat memberikan nilai yang berbeda untuk suatu aplikasi yang sama. Metode pengukuran Qos secara subjektif ini umumnya dilakukan dengan menggunakan nilai MOS, dimana dilakukan uji coba langsung suatu layanan VoIP oleh beberapa pengguna, yang kemudian akan memberikan penilaian terhadap kualitas VoIP tersebut. Range nilai yang diberikan adalah

antara 1 sampai 5, 1 untuk kualitas yang terburuk sedangkan 5 untuk kualitas yang terbaik seperti yang ada pada Tabel 2.2.

Tabel 2.2. Nilai kualitas MOS

Nilai	Kualitas Suara
1	Jelek
2	Kurang
3	Cukup
4	Baik
5	Baik Sekali

Sedangkan secara objektif, QoS pada suatu layanan diukur berdasarkan parameter – parameter tertentu yang dapat mempengaruhi tingkat QoS pada VoIP. Parameter – parameter tersebut yaitu :

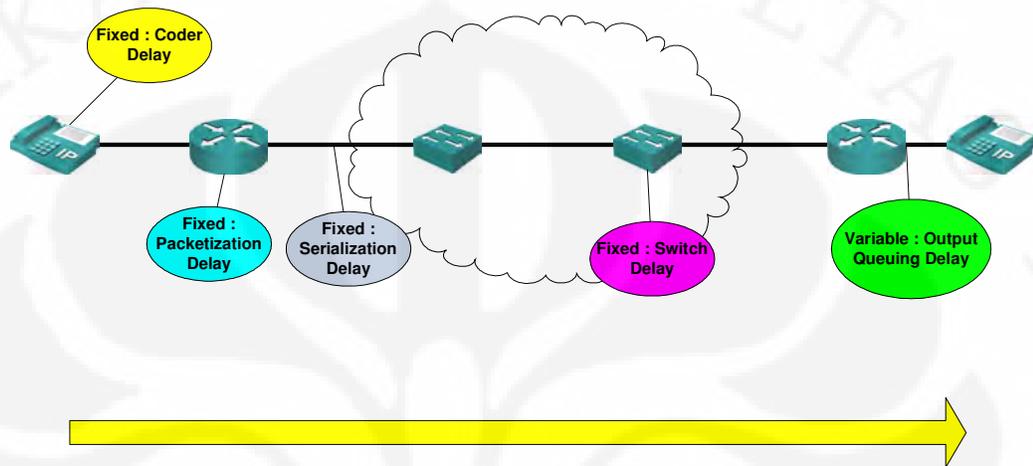
- *Delay*

Waktu total yang dibutuhkan paket mulai dari dikirim oleh *sender* sampai diterima oleh *receiver*. *Delay* merupakan faktor penting dalam menentukan kualitas VoIP dan merupakan musuh terbesar dalam suatu jaringan VoIP. Semakin besar *delay* yang terjadi maka akan semakin rendah kualitas VoIP yang dihasilkan. Ada dua jenis *delay* yaitu *fixed delay*, *delay* yang dihasilkan oleh komponen, dan *variable delay*, *delay* akibat adanya antrian dalam saluran yang mengakibatkan munculnya *variable delay* atau lebih dikenal dengan sebutan *jitter*. Beberapa sumber *delay* antara lain seperti digambarkan pada Gambar 2.7.[5]

1. *Coder delay*

Disebut juga *processing delay*, yaitu waktu yang dibutuhkan oleh codec untuk melakukan kompresi dari sampel suara. Karena penggunaan codec yang berbeda – beda maka nilai dari *delay* ini

bervariasi tergantung kepada jenis codec yang digunakan dan kecepatan prosesor.



Gambar 2.7. Sumber delay

2. Packetization delay

Adalah waktu yang dibutuhkan untuk mengisi suatu paket dengan suara yang sudah dikompresi. RFC 1890 menetapkan bahwa *packetization delay* sebaiknya 20 ms.

3. Queuing delay

Merupakan delay yang terjadi karena adanya antrian dalam jaringan. Queuing delay bergantung kepada kecepatan *trunk* dan terjadinya antrian atau tidak.

4. Serialization delay

Yaitu *delay* pada saat peletakan *data frame* ke dalam jaringan. *Delay* ini juga terjadi pada *router* atau *switch* dan merupakan *fixed delay*.

5. Network delay

Yaitu waktu yang diperlukan paket untuk melewati media transmisi dari pengirim ke penerima dan tergantung jarak yang harus ditempuh oleh paket. Delay ini merupakan jenis delay yang paling sulit untuk dihitung. Terdiri dari delay komponen (delay akibat proses pada komponen intermediate), delay propagasi (delay pada saluran yang dilewati), dan queuing delay. Untuk menghitung nilai delay propagasi, digunakanlah perkiraan 10 ms per mil atau 6 ms per km (berdasarkan G.114).

Besarnya *delay* yang direkomendasikan oleh ITU untuk aplikasi dalam Recommendation G.144 adalah seperti pada Tabel 2.3.

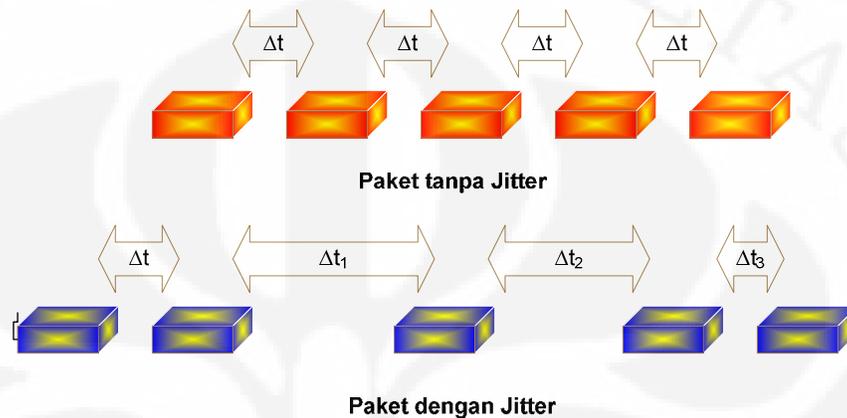
Tabel 2.3. Pedoman batasan delay

Delay (ms)	Penjelasan
0 - 150	Dapat diterima.
150 - 400	Dapat diterima. Namun administrator jaringan harus waspadaterhadap segala sesuatu yang dapat mempengaruhi kualitas jaringan.
Di atas 400	Secara umum, tidak dapat diterima. Namun untuk kasus – kasus khusus, nilai batas ini dapat berubah.

- *Jitter*

Merupakan variasi waktu tunda atau variasi *delay* yang terjadi akibat adanya selisih waktu atau *interval* kedatangan paket di penerima. Jitter diilustrasikan seperti pada Gambar 2.8. Awalnya, paket dikirim secara kontinu dari pengirim menuju ke penerima, namun diperjalanan paket tersebut mengalami gangguan akibat dari jaringan yang terlalu padat ataupun adanya kesalahan dalam konfigurasi, yang mengakibatkan

munculnya delay antar paket yang dikirimkan. Untuk mengatasi *jitter* maka paket data yang datang di sisi penerima terlebih dahulu dikumpulkan dengan menggunakan mekanisme *jitter buffer* selama beberapa saat sampai paket tersusun dengan urutan yang benar di sisi penerima.



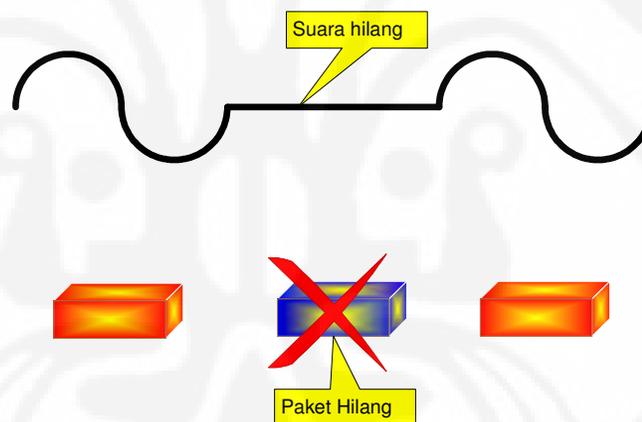
Gambar 2.8. Akibat adanya jitter

- *Packet Loss*

Adalah hilangnya paket data yang sedang dikirimkan. Pada jaringan internet, setiap paket yang dikirimkan belum tentu sampai di tujuan. Hal ini disebabkan karena sifat jaringan internet yang *best effort*, dimana jaringan hanya berusaha agar paket sampai tujuan. Sebenarnya adanya paket yang hilang ini tidak akan menjadi masalah besar apabila penerima dapat meminta pengiriman ulang paket yang hilang tersebut. Namun sayangnya hal ini tidak berlaku pada VoIP karena suara yang dikirim harus dimainkan secara real time dan tidak memungkinkan dilakukan pengiriman kembali. *Packet loss* dapat disebabkan oleh *router* yang sibuk atau jalur komunikasi yang terlampau padat. Agar diperoleh kualitas suara yang dapat diterima, maka jumlah paket yang hilang harus kurang dari 15%. Gambar 2.9 mengilustrasikan bagaimana efek yang ditimbulkan akibat adanya packet loss untuk aplikasi VoIP.

- *Bandwidth*

Bandwidth yang besar merupakan hal utama yang harus disediakan untuk memperoleh QoS yang baik. Dengan alokasi *bandwidth* ini, setiap aliran paket data yang berisi suara mendapatkan jatah *bandwidth* yang tetap dan tidak perlu berkompetisi dengan paket data lain. *Bandwidth* sendiri adalah kecepatan maksimum yang dapat digunakan untuk melakukan transmisi data antar computer pada jaringan IP atau internet. Dalam perancangan VoIP, *bandwidth* merupakan suatu yang harus diperhitungkan agar dapat memenuhi kebutuhan pelanggan yang dapat digunakan menjadi parameter untuk menghitung jumlah peralatan yang dibutuhkan dalam suatu jaringan. Perhitungan ini juga sangat diperlukan dalam efisiensi jaringan dan biaya serta sebagai acuan pemenuhan kebutuhan untuk pengembangan di masa mendatang.



Gambar 2.9. Akibat adanya *packet loss*

2.5 Codec

Codec adalah suatu hardware atau software yang dapat melakukan *sampling* terhadap sinyal suara analog, kemudian mengkonversi ke dalam bit – bit digital dan / atau mengembalikannya lagi ke bentuk analog. Codec sendiri merupakan kepanjangan dari *compressor-decompressor* atau lebih dikenal dengan

coder – encoder [6]. Beberapa jenis codec melakukan kompresi agar dapat menghemat *bandwidth*.

Codec yang digunakan untuk aplikasi VoIP merupakan speech encoding, yaitu kompresi data audio digital yang mengandung speech (pembicaraan). Speech encoding sendiri termasuk jenis audio codec pada kategori lossy data compression, artinya ketika data yang terkompresi itu di-dekompresi akan sedikit berbeda dengan data asli tapi masih dianggap cukup merepresentasikan.

Untuk VoIP, banyak jenis speech encoding yang dapat dipakai antara lain yang distandarisasi oleh ITU-T seperti G.711, G.723.1, G.726, G.728, dan G.729. Sedangkan yang non ITU seperti GSM, Speex dan iLBC. Berbagai jenis codec ini memiliki bit rate dan kualitas yang berbeda satu sama lain. Besar kecilnya bit rate ini akan mempengaruhi kerja prosesor. Semakin kecil nilai bit rate maka kerja prosesor akan semakin berat, dan sebaliknya. Namun nilai bit rate yang kecil dibutuhkan untuk menghemat konsumsi bandwidth.

Pada Tabel 2.3 di bawah ini, terdapat beberapa jenis codec yang digunakan secara umum, bersama dengan *bit rate* dan *packetization delay*.

Tabel 2.4. Beberapa jenis codec pada VoIP

Nama Codec	Bit Rate
G.711	64.0 kbps
G.711a	64.0 kbps
GSM	13 kbps
G.729	8.0 kbps
G.723.1 MPMLQ	6.3 kbps
G.723.1 ACELP	5.3 kbps

- G.711

G.711 merupakan codec yang dikeluarkan oleh ITU untuk digunakan pada telepon digital seperti ISDN dan sambungan E1 atau T1. Codec ini menggunakan teknik kompresi PCM (Pulse Code Modulation) dalam pengiriman suara dan memiliki bitrate yang tinggi yaitu sebesar 64 Kbps, dengan kecepatan sampling 8000 sample / detik dan terdiri dari 8 bit untuk setiap sampel. G.711 memiliki payload sebesar 160 byte dan waktu untuk satu frame adalah 20 ms. Sedangkan ukuran header totalnya adalah 40 byte yang berasal 20 byte header IP, 12 byte header UDP dan 8 byte header RTP.

Codec ini dapat digunakan secara bebas pada aplikasi VoIP karena tidak ada biaya lisensi dan paling baik digunakan pada jaringan local yang memiliki bandwidth cukup besar. Keuntungan lain dari penggunaan codec ini adalah implementasi yang sederhana sehingga tidak memerlukan daya CPU yang terlalu besar.

- GSM

GSM adalah jenis codec yang memiliki kecepatan penuh atau disebut full rate (FR) yang dikeluarkan oleh ETSI. Standar ini mendefinisikan sebuah referensi konfigurasi untuk rantai transmisi suara terutama pada telekomunikasi digital selular. Input bagi encoder suara ini adalah sinyal PCM 13 bit uniform. Codec ini merupakan speech encoding yang didasarkan pada algoritma RPE – LTP (Regular Pulse Excitation-Long Term Prediction). GSM memiliki bit rate sebesar 13 kbps dengan 160 sampel rate 8 KHz. Encoder memproses blok suara sebesar 20 ms yang tiap blok berisi 260 bit sehingga dihasilkan kecepatan 13 Kbps ($260 \text{ bits}/20 \text{ ms} = 13.000 \text{ bits/s} = 13 \text{ kbits/s}$), walaupun dalam kondisi sebenarnya GSM akan mentransmisikan bit rate yang lebih tinggi akibat adanya *error recovery* dan *packet information* [7]. Dan memiliki ukuran payload 33 byte dengan ukuran header 40 byte.

- G.729

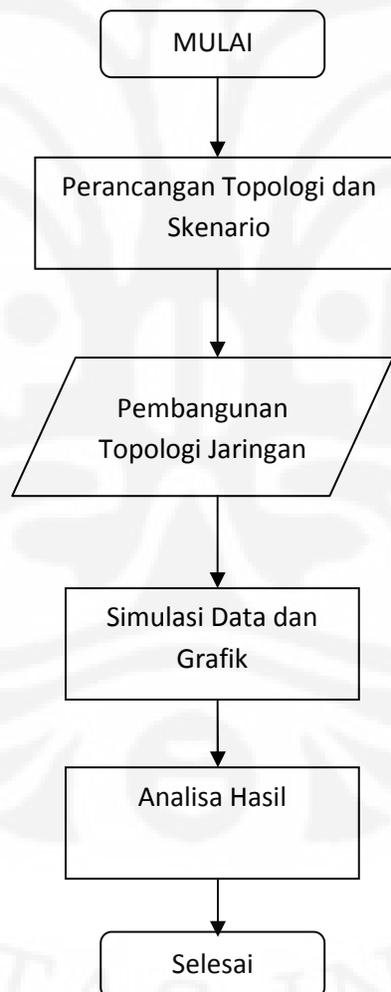
G.729 adalah algoritma pemrosesan data audio untuk suara yang dapat mengkompresi suara digital ke dalam paket berdurasi 10 ms. Codec ini memiliki bit rate sebesar 8 kbit/s dengan menggunakan teknik kompresi CS-ACELP (Conjugate Structure Algebraic Code Excited Linear Prediction). Kelebihan dari codec ini adalah kebutuhan akan bandwidth yang kecil sehingga umumnya digunakan pada aplikasi VoIP, seperti Skype, dimana ketersediaan bandwidth terbatas. Kekurangannya adalah tidak dapat digunakan untuk mentransfer high quality audio secara reliable.

BAB 3

PERANCANGAN DAN IMPLEMENTASI SIMULASI JARINGAN

3.1 Diagram Alir Perancangan dan Implementasi

Dalam merancang dan mengimplementasikan jaringan ke dalam simulator, dibutuhkan langkah – langkah untuk mempermudah prosesnya. Gambar 3.1 berikut merupakan diagram alir perancangan dan implementasi skripsi ini.



Gambar 3.1. Diagram alir perancangan dan simulasi

Perencanaan implementasi ini dibagi menjadi empat tahap utama seperti pada Gambar 3.1 di atas, yaitu tahap perancangan topologi, pembangunan topologi jaringan, simulasi data dan grafik serta yang terakhir adalah tahap analisa hasil. Berikut adalah penjabaran lebih lanjut mengenai tahap – tahap implementasi :

1. Perancangan topologi

Langkah awal yang dalam merancang topologi jaringan adalah dengan melakukan studi literatur. Topologi jaringan ini berupa jumlah node, letak node, besarnya bandwidth, codec yang digunakan, serta protokol yang akan dianalisa. Dengan adanya perancangan topologi ini akan membantu proses pembangunan simulasi jaringan nantinya.

2. Pembangunan topologi jaringan

Tahap ini merupakan tahapan inti dari implementasi ini, yaitu pembangunan topologi jaringan yang telah dirancang sebelumnya ke dalam simulator OPNET IT Guru Edition. Pada simulasi ini akan dibangun jaringan VoIP yang menggunakan protokol pensinyalan SIP, dimana user akan mencoba melakukan komunikasi VoIP dengan dan tanpa adanya protokol RSVP.

3. Simulasi data dan grafik

Pada tahap ini, simulasi akan dijalankan pada simulator OPNET dan hasilnya yang berupa grafik kemudian akan ditampilkan.

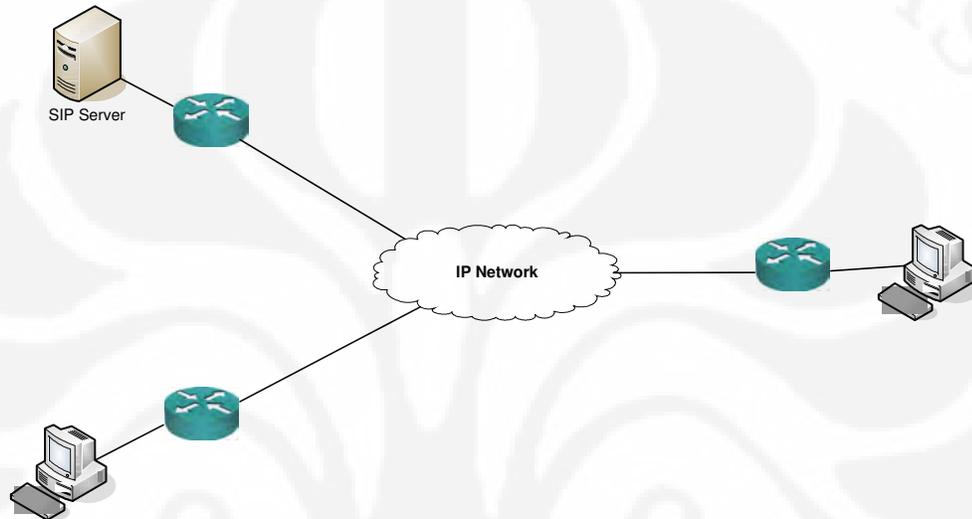
4. Analisa hasil

Dalam tahap ini dilakukan analisa berdasarkan hasil grafik yang didapatkan dari kegiatan simulasi sebelumnya. Analisa ini meliputi nilai – nilai dari parameter QoS yaitu throughput, delay, jitter dan paket yang hilang.

3.2 Perancangan Topologi dan Skenario

Simulasi ini dilakukan dengan membuat 2 jenis skenario dengan topologi yang sama, namun berbeda kondisi. Pada skenario pertama, seorang user melakukan komunikasi VoIP kepada user lain dengan variasi bandwidth dan codec yang berbeda

tanpa menggunakan protokol RSVP pada jaringan, sedangkan pada skenario kedua, user yang sama melakukan komunikasi VoIP dengan protokol RSVP telah dijalankan dalam jaringan serta variasi bandwidth dan codec yang sama. Gambar 3.2 berikut merupakan topologi yang akan digunakan pada simulasi ini menggunakan simulator OPNET (Optimized Network Engineering Tools) IT Guru Edition.



Gambar 3.2. Topologi jaringan

Client dan receiver dibuat pada jaringan yang berbeda. Hal ini bertujuan untuk membuat sistem yang dibangun serealistis mungkin dengan kondisi jaringan internet. Utilitas jaringan dibuat 20% sehingga tersisa bandwidth 80%. Pengambilan data QoS VoIP diambil berdasarkan kondisi jaringan secara global.

Sedangkan skenario pengujian yang akan dilakukan untuk melakukan pengujian terhadap kinerja protokol RSVP terhadap QoS dan performa jaringan VoIP-SIP adalah sebagai berikut.

- Skenario pertama :

Pada skenario pertama akan dijalankan aplikasi VoIP dengan menggunakan protokol signaling SIP dari client di Bandung ke receiver di

Cirebon dengan bandwidth jaringan 64 Kbps dan utilitas 20 %. Akan digunakan 3 jenis codec yaitu codec G.711 dengan bitrate 64 Kbps, codec G.729 yang memiliki bitrate 8 Kbps dan codec GSM yang memiliki bitrate 13 Kbps. Ketika simulasi dijalankan maka OPNET akan mengambil data sesuai dengan statistik yang diinginkan. Setelah selesai dilakukan simulasi, data – data tersebut ditampilkan dalam bentuk grafik. Dari grafik tersebut, dapat dianalisa bagaimana QoS dan performa dari jaringan tersebut.

- Skenario kedua :

Skenario kedua pada dasarnya hamper sama dengan scenario pertama, akan dijalankan aplikasi VoIP dengan menggunakan protokol signaling SIP dari client di Bandung ke receiver di Cirebon dengan bandwidth jaringan 64 Kbps dan utilitas 20 % serta penggunaan 3 jenis codec yaitu codec G.711, codec G.729 dan codec GSM. Namun yang membedakan dari scenario pertama adalah penggunaan protokol RSVP pada jaringan yang bertujuan untuk meningkatkan QoS jaringan. Hal ini dapat diketahui setelah menganalisa delay dan jitter. Setelah didapatkan data tersebut, akan dianalisa bagaimana performa dari codec tersebut serta pada bandwidth berapa yang paling optimal.

3.3 Pembangunan Topologi Jaringan

Setelah ditentukan bentuk dan scenario jaringan yang akan dibuat, langkah kedua adalah membangun rancangan yang telah di buat ke dalam simulator yang akan digunakan.

3.3.1 Interface Hardware

Pada tugas akhir ini, akan dibangun sebuah simulasi jaringan dengan menggunakan 1 buah PC (laptop) dengan spesifikasi sebagai berikut :

- Lenovo ThinkPad R61i
- Intel Core 2 Duo processor 2.00 GHz
- 1 GB memory DDR2
- HDD 80 GB

3.3.2 Interface Software

Spesifikasi software yang digunakan untuk melakukan simulasi ini adalah :

- Sistem operasi : Microsoft Windows 7 Professional
- Aplikasi simulator : OPNET IT Guru Edition release 9.1

3.3.3 Instalasi OPNET

Simulator OPNET dapat berjalan di atas sistem operasi windows 7 professional. OPNET yang digunakan dalam simulasi ini adalah OPNET IT Guru Edition 9.1. Proses instalasi diawali dengan mengunduh *resource file* ITG_Academic_Edition_v1998.zip dari website resmi OPNET (www.opnet.com). Waktu yang dibutuhkan kurang lebih 15 menit, tergantung spesifikasi komputer yang digunakan. Proses instalasi sebaiknya dilakukan saat computer dapat terhubung dengan internet karena pada akhir proses, software akan meminta kode registrasi license yang akan diperoleh pada website resmi opnet. Setelah mendapatkan kode dan memasukkannya ke dalam software, selesailah proses instalasi simulator. Gambar 3.3 memperlihatkan tampilan awal dari software ini.



Gambar 3.3. OPNET IT Guru Edition

3.3.4 Pembangunan Topologi

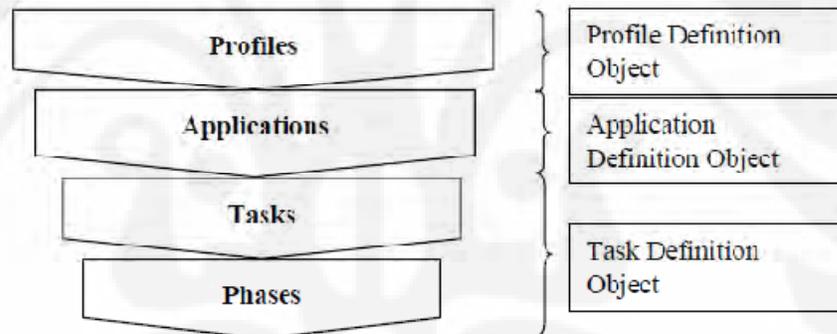
Pembangunan topologi jaringan pada OPNET adalah dengan memilih objek – objek yang ingin digunakan lalu menghubungkan satu sama lain dan mengaturnya sesuai dengan topologi yang ingin dicapai. Pada simulasi ini akan digunakan 2 buah workstation sebagai sumber dan penerima, 3 buah ethernet4_slip8_gtwy yang berperan sebagai router, sebuah ethernet_server sebagai server SIP (proxy server), 3 buah subnet dan sebuah IP_32_cloud. Sambungan dari tiap workstation menggunakan kabel 10Base_T, server menggunakan 1000Base_X, sedangkan sambungan dari tiap subnet menggunakan sambungan point_to_point.

3.3.5 Konfigurasi Aplikasi dan Profile

Apabila topologi jaringan telah dibangun, maka langkah selanjutnya adalah memodelkan traffic jaringannya. Ada dua teknik untuk merepresentasikan traffic

jaringan, yang pertama adalah dengan mengimport traffic sebagai “conversation pair traffic” dan yang kedua adalah dengan memodelkan trafik aplikasi dengan cara menyeting beberapa attribute aplikasi. Pada simulasi ini akan dilakukan konfigurasi aplikasi dan profile dengan metode yang kedua [8].

Untuk menggambarkan trafik yang ada pada suatu workstation / LAN, atau trafik seorang pengguna / sekelompok pengguna, perlu didefinisikan bagaimana kebiasaannya. Kebiasaan pengguna atau profile dapat digambarkan oleh aplikasi yang biasa digunakan oleh user tersebut dan berapa lama serta seberapa sering aplikasi tersebut digunakan. Sebuah aplikasi dapat didefinisikan berdasarkan apa saja yang dilakukan oleh aplikasi tersebut atau disebut task pada OPNET. Gambar 3.4 berikut merupakan penggambaran dari pemodelan pada OPNET.



Gambar 3.4. Pemodelan trafik

Sebuah profile terdiri dari sebuah aplikasi atau lebih. Aplikasi tersebut dapat direpresentasikan sebagai sebuah sumber trafik, protokol atau suatu set dari tasks. Tasks dapat terdiri dari banyak fasa, dimana tiap fasa mendeskripsikan pola dari pertukaran data antara sumber dan tujuan. Pada simulasi ini yang akan digambarkan hanya profile dan aplikasi karena task dan fasa hanya digunakan pada custom application.

Sebelum mengkonfigurasi profile, akan didefinisi dulu aplikasi voice. Untuk mendefinisikan aplikasi, harus ditambahkan objek baru yaitu application config. Dalam application config akan dibuat aplikasi voice yang memiliki nilai silence length exponential (0.65) seconds yang mendefinisikan waktu diam saat terjadi antara penelpon dan penerima saat melakukan komunikasi, talk spurt length (0.352) seconds yang mendefinisikan waktu bicara antara penelpon dan penerima saat melakukan komunikasi, Type of service interactive voice (6) yang mendefinisikan prioritas dari paket yang dikirim oleh aplikasi dan menggunakan signaling SIP.

Setelah selesai pendefinisian aplikasi, selanjutnya adalah pengkonfigurasi profile. Profile mendefinisikan bagaimana pola aktivitas dari user. Untuk mengkonfigurasi profile, hal yang pertama dilakukan adalah menambahkan objek profile config. Dalam simulasi ini akan dibuat profile yang memiliki nilai start time offset uniform (5,10) seconds yang mendefinisikan waktu mulai dari sebuah aplikasi, duration end of profile yang menunjukkan durasi dari aplikasi tersebut, operation mode serial (ordered) yang mendefinisikan aktivitas yang dilakukan user, start time uniform (100,110) seconds yang mendefinisikan waktu mulai dari profile dan duration end of simulation yang menunjukkan durasi profile.

Ketika aplikasi dan profile telah didefinisikan, berarti profile tersebut telah siap dipasang pada suatu workstation, server atau LAN. Biasanya profile dipasang pada workstation atau LAN karena keduanya merupakan sumber trafik, namun profile dapat dipasang juga pada server apabila server tersebut menjadi sumber dari aplikasi. Pada simulasi ini, profile dipasang pada workstation sumber karena merupakan sumber trafik pada jaringan.

3.3.6 Mendefinisikan Sumber dan Tujuan

Pendefinisian sumber dan tujuan dari suatu aplikasi penting untuk dilakukan agar suatu trafik tiba pada objek yang mendukung aplikasi yang dibawanya. Apabila

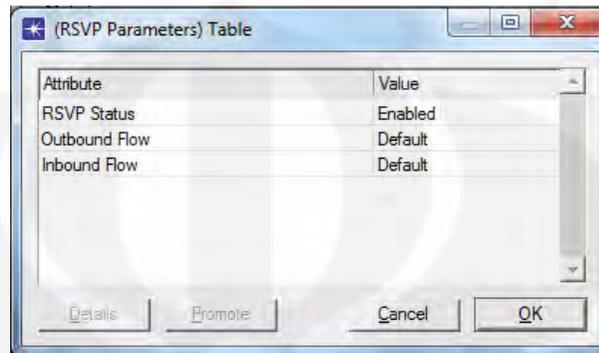
tidak dilakukan pendefinisian, maka OPNET secara default akan memilih end station tujuan secara random. Sebelum melakukan pendefinisian sumber dan tujuan, setiap objek yang ada harus diberikan sebuah nama yang unik agar tidak terjadi kekeliruan. Pada simulasi ini, workstation Cirebon merupakan tujuan dari trafik yang dikirimkan workstation Bandung. Oleh karena itu, pada pengirim Application : Destination Preferences di set menjadi nama unik tujuan yaitu Receiver_no_RSVP untuk scenario pertama dan Receiver_RSVP untuk scenario kedua. Sedangkan workstation tujuan berfungsi sebagai penyedia layanan sehingga pada tujuan Application : Supported Service di set menjadi Voice RSVP Not Used untuk skenario pertama dan Voice RSVP Used untuk skenario kedua.

3.3.7 Konfigurasi RSVP

Parameter RSVP pada OPNET didefinisikan dengan dua objek yaitu QoS Attribute Config dan Application Config. Agar dapat menjalankan simulasi RSVP, kedua objek tersebut harus diikutsertakan dalam skenario. Ada dua set atribut yang harus didefinisikan pada QoS attribute config yaitu RSVP flow specification, yang mendefinisikan kebutuhan trafik, dan RSVP profiles, yang digunakan untuk membuat keputusan reservasi untuk suatu aplikasi yang dikirimkan oleh host. Sedangkan parameter RSVP untuk path messages didefinisikan pada Application config.

Proses konfigurasi RSVP harus dilakukan secara berurutan. Yang pertama dikonfigurasi adalah dua atribut pada QoS attribute config. RSVP flow specification mendefinisikan kebutuhan trafik yang akan direservasi, berupa bandwidth dan buffer size, oleh RSVP. Dapat juga digunakan untuk menspesifikan trafik specification yang dikirim pada path message, serta dapat digunakan untuk menspesifikan flow reservation yang digunakan pada resv message. Sedangkan RSVP profiles mendefinisikan seluruh profile resevasi yang ada dalam jaringan. Tiap profile mengandung informasi tentang resv parameters, resv style, resv retry parameter, yang digunakan oleh ketika keputusan reservasi dibuat. Sedangkan pada application config,

didefinisikan RSVP parameters yang mencakup RSVP status, outbond bandwidth dan buffer size, inbound bandwidth dan buffer size yang digunakan oleh node diperlihatkan pada Gambar 3.5 berikut.



Gambar 3.5. *RSVP Parameters*

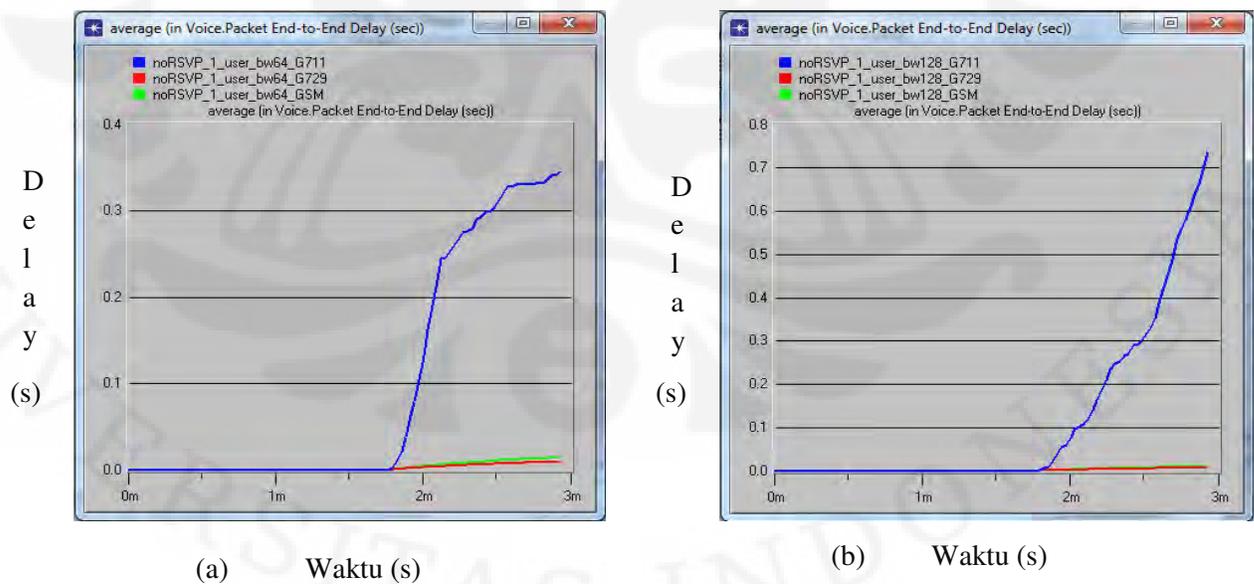
BAB 4

ANALISA HASIL SIMULASI

Pada bab ini, akan disajikan data hasil simulasi berupa grafik yang menggambarkan bagaimana kondisi jaringan. Data yang ditampilkan meliputi delay dan jitter sebagai parameter QoS jaringan. Lalu akan dilakukan analisa terhadap data tersebut untuk melihat codec manakah yang memiliki performansi terbaik saat dijalankan di atas aplikasi VoIP-SIP dengan atau tanpa menggunakan RSVP.

4.1 Delay

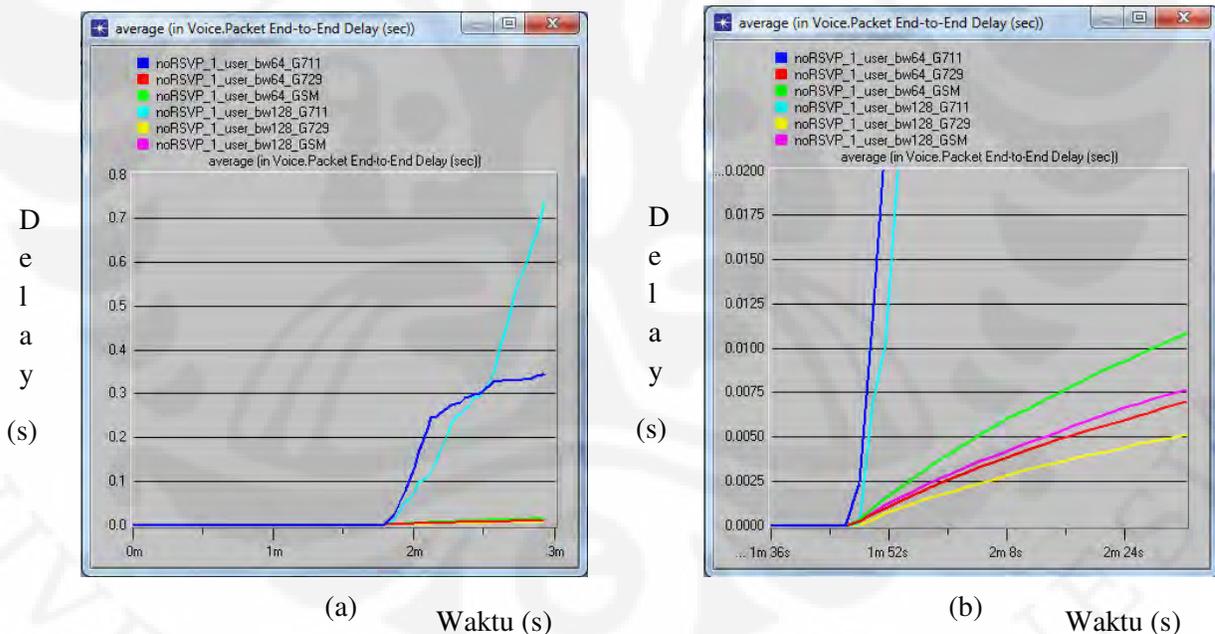
Delay masing – masing skenario dapat dilihat pada parameter *voice.packet end to end delay*. Hasil yang didapatkan dari kegiatan simulasi tanpa protokol RSVP untuk parameter *delay* diperlihatkan pada Gambar 4.1 berikut ini.



Gambar 4.1. (a) Grafik delay tanpa RSVP pada bandwidth 64, (b) Grafik delay tanpa RSVP pada bandwidth 128

Dari grafik di atas, dapat diketahui bahwa pada jaringan dengan bandwidth 64 Kbps maupun pada bandwidth 128 Kbps, codec G.279 memiliki tingkat performa yang paling baik karena memiliki delay yang paling kecil, disusul oleh GSM, yang memiliki sedikit perbedaan waktu delay dengan G.729, dan yang terakhir adalah G.711, dengan perbedaan yang cukup signifikan bila dibandingkan dengan kedua codec yang lain. Hal ini disebabkan karena G.711 memiliki payload yang paling besar yaitu 214 bytes, disusul oleh GSM dan G.729. Semakin besar payload, maka delay paketasasi, routing, transmisi dan switching akan semakin besar.

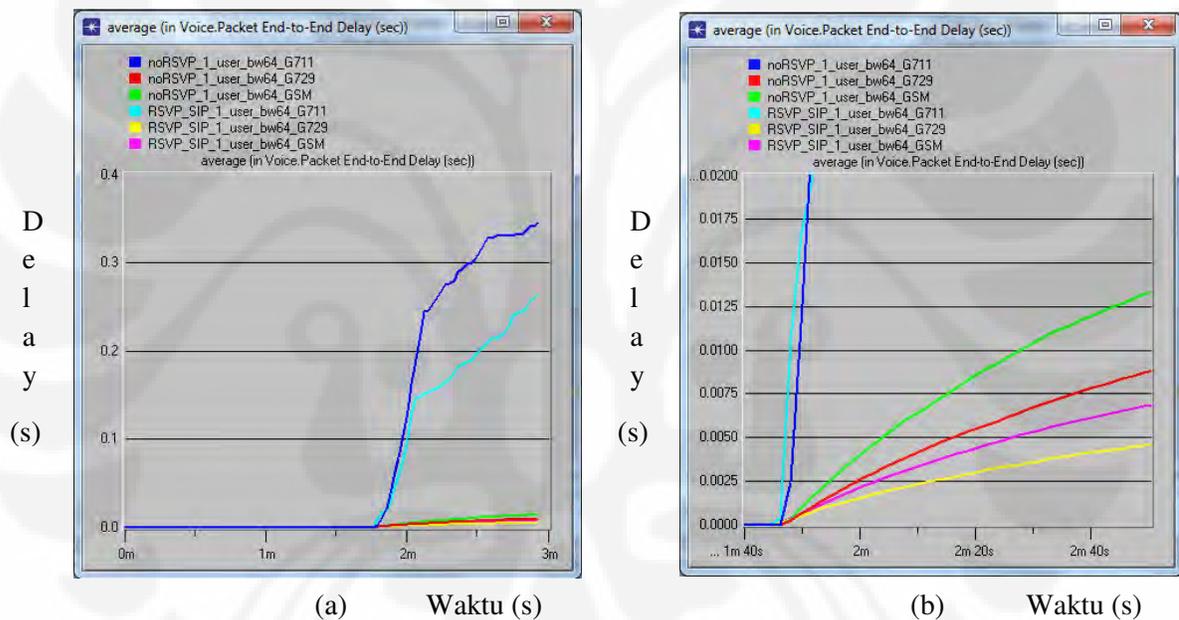
Jika kedua grafik tersebut digabungkan, maka akan terlihat seperti pada Gambar 4.2 bahwa kombinasi G.729 dengan bandwidth 128 Kbps memiliki delay terkecil diantara kombinasi yang lainnya, hal ini disebabkan karena G.729 memiliki payload yang paling kecil dan didukung oleh bandwidth yang cukup besar.



Gambar 4.2. (a) Grafik perbandingan delay tanpa RSVP, (b) Detail grafik perbandingan delay tanpa RSVP

Disusul oleh G.729 pada 64 Kbps, GSM pada 128 Kbps, GSM pada 64 Kbps, G.711 pada 128 Kbps dan G.711 pada 64 Kbps. G.729 pada 64 Kbps memiliki nilai delay yang lebih kecil beberapa milisekon bila dibandingkan dengan GSM pada 128 Kbs dan G.711 pada 128Kbps memiliki delay yang lebih besar dibandingkan dengan GSM dan G.729 pada 64 Kbps. Hal ini menunjukkan bahwa pemilihan codec juga mempengaruhi besarnya delay.

Jika dibandingkan dengan jaringan yang menggunakan protocol RSVP, berikut gambar perbandingannya pada Gambar 4.3 dan Gambar 4.4.

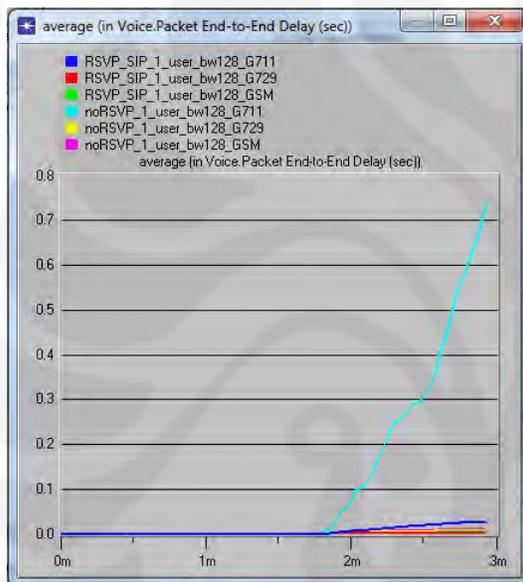


Gambar 4.3. (a) Grafik perbandingan delay tanpa dan dengan RSVP pada bandwidth 64, (b) Detail grafik perbandingan delay tanpa dan dengan RSVP pada bandwidth 64

Dari gambar 4.3 dan 4.4, secara umum terlihat bahwa jaringan yang menggunakan protokol RSVP memiliki nilai delay yang lebih kecil dibandingkan dengan jaringan yang tidak menggunakan RSVP. Hal ini disebabkan karena protocol RSVP akan mereservasi sumber daya yang ada terlebih dahulu sehingga memastikan aplikasi VoIP diutamakan dalam jaringan. Pada bandwidth 64 Kbps, G.729 dengan RSVP memiliki performa delay terkecil dibandingkan yang lain, disusul oleh GSM

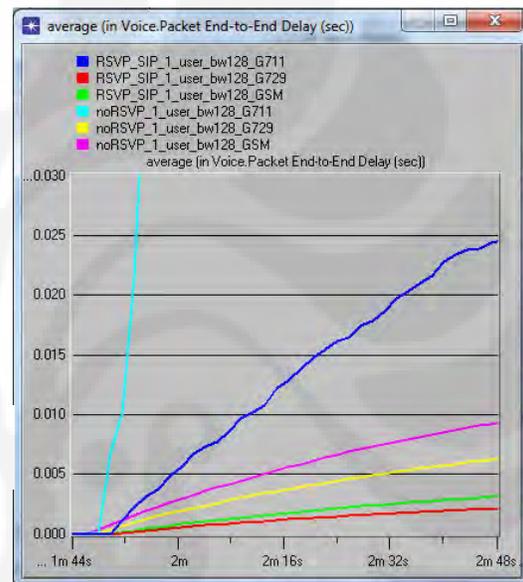
dengan RSVP, G.729 tanpa RSVP, GSM tanpa RSVP, G.711 dengan RSVP dan G.711 tanpa RSVP. Codec GSM yang awalnya memiliki delay lebih besar dari G.729, dengan dikombinasi dengan RSVP, nilai delaynya menjadi lebih kecil dibanding G.729 tanpa RSVP. Demikian pula pada bandwidth 128 Kbps, memiliki urutan yang sama dengan bandwidth 64 Kbps. Codec G.711 memiliki pengurangan delay yang signifikan pada bandwidth 128 Kbps ini.

D
e
l
a
y
(s)



(a) Waktu (s)

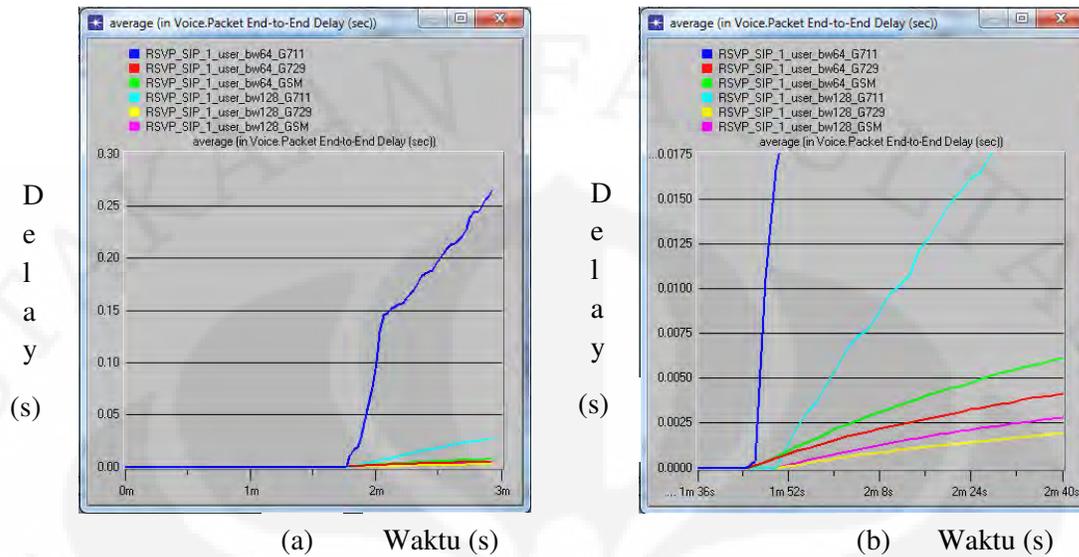
D
e
l
a
y
(s)



(b) Waktu (s)

Gambar 4.4. (a) Grafik perbandingan delay tanpa dan dengan RSVP pada bandwidth 128, (b) Detail grafik perbandingan delay tanpa dan dengan RSVP pada bandwidth

Apabila seluruh grafik jaringan yang menggunakan protokol RSVP digabungkan, maka dapat dilihat pada Gambar 4.5 bahwa codec G.729 pada bandwidth 128 Kbps dan 64 Kbps serta GSM pada bandwidth 128 Kbps dan 64 Kbps perbedaan nilai delay diantara keempatnya cukup kecil sehingga dapat disimpulkan bahwa penggunaan protocol RSVP ini merupakan salah satu cara untuk mengurangi nilai delay pada aplikasi VoIP, yang sensitive terhadap delay, selain dengan melebarkan bandwidth dan pemilihan codec.

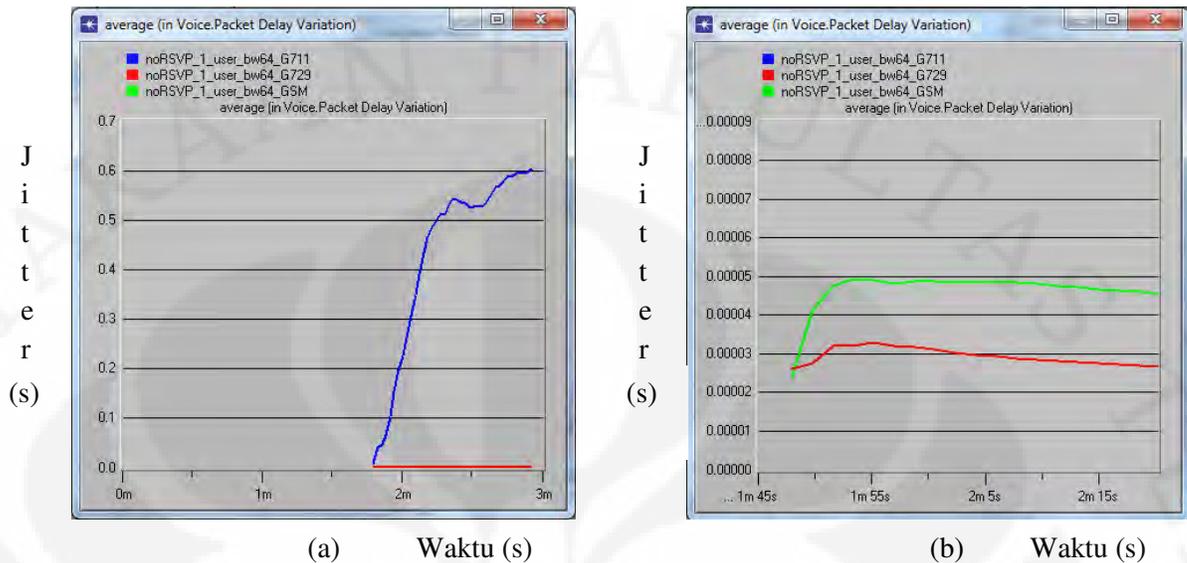


Gambar 4.5. (a) Grafik perbandingan delay dengan RSVP, (b) Detail grafik perbandingan delay dengan RSVP

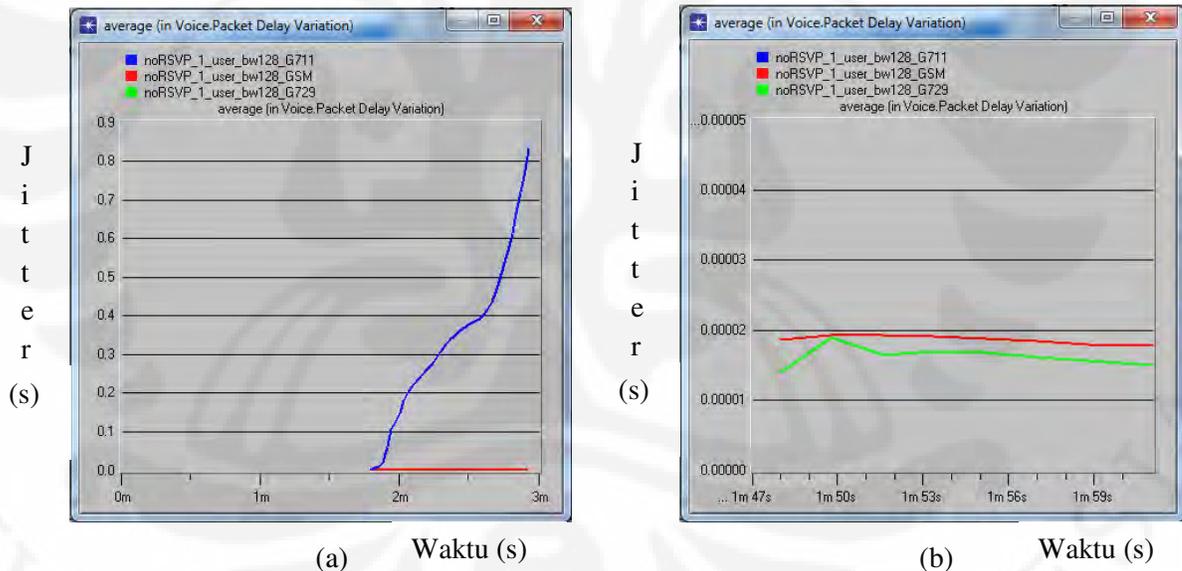
4.2 Jitter

Jitter adalah variasi delay antar paket yang terjadi karena waktu kedatangan paket yang berbeda - beda. Atau dengan kata lain, jitter adalah perbedaan waktu kedatangan antara 1 paket dengan paket setelahnya. Parameter jitter perlu untuk dianalisis untuk mengetahui delay kedatangan antar satu paket dengan paket lainnya. Semakin besar jitter maka semakin besar juga perbedaan antara suara asli dengan suara yang terdengar. Hal tersebut dapat disebabkan oleh besarnya collision antara paket. Berikut pada Gambar 4.6 dan 4.7 adalah hasil simulasi jitter pada jaringan tanpa menggunakan RSVP.

ITU-T merekomendasikan jitter yang baik adalah kurang dari 30 ms. Pada hasil simulasi terlihat bahwa untuk codec G.711, nilai jitter melebihi nilai 30 ms baik pada bandwidth 64 Kbps maupun pada bandwidth 128 Kbps. Sedangkan untuk codec GSM dan G.729, keduanya memiliki nilai jitter kurang dari 30 ms pada bandwidth 64 Kbps dan 128 Kbps. Parameter jitter sangat mempengaruhi kualitas suara. Semakin besar jitter maka suara yang dihasilkan akan semakin tidak jelas (terputus – putus).



Gambar 4.6. (a) Grafik jitter tanpa RSVP pada bandwidth 64, (b) Detail grafik jitter tanpa RSVP pada bandwidth 64

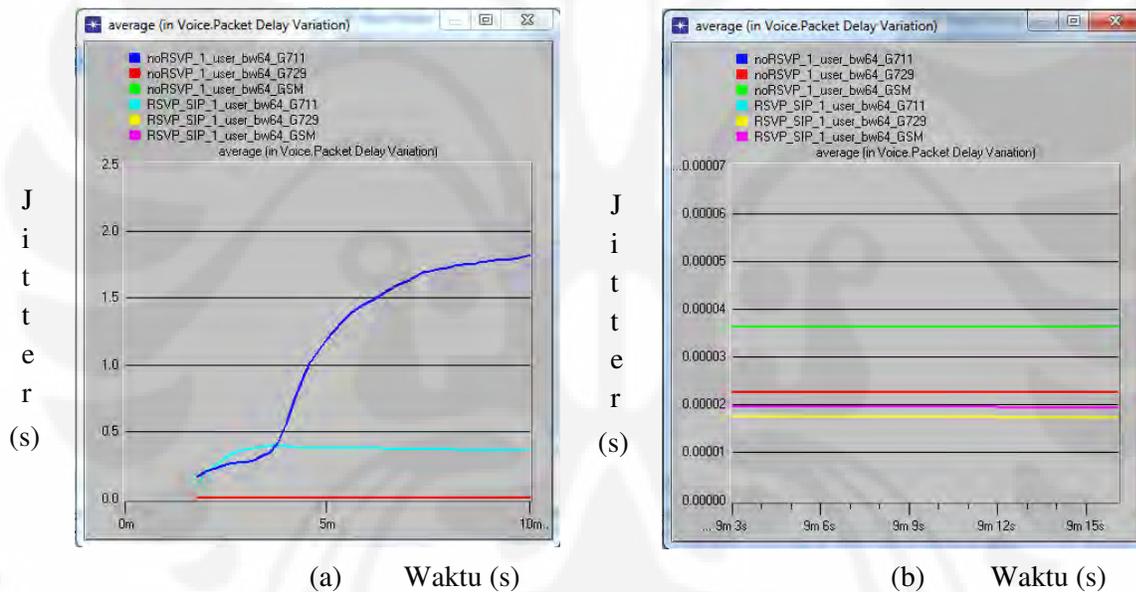


Gambar 4.7. (a) Grafik jitter tanpa RSVP pada bandwidth 128, (b) Detail grafik jitter tanpa RSVP pada bandwidth 128

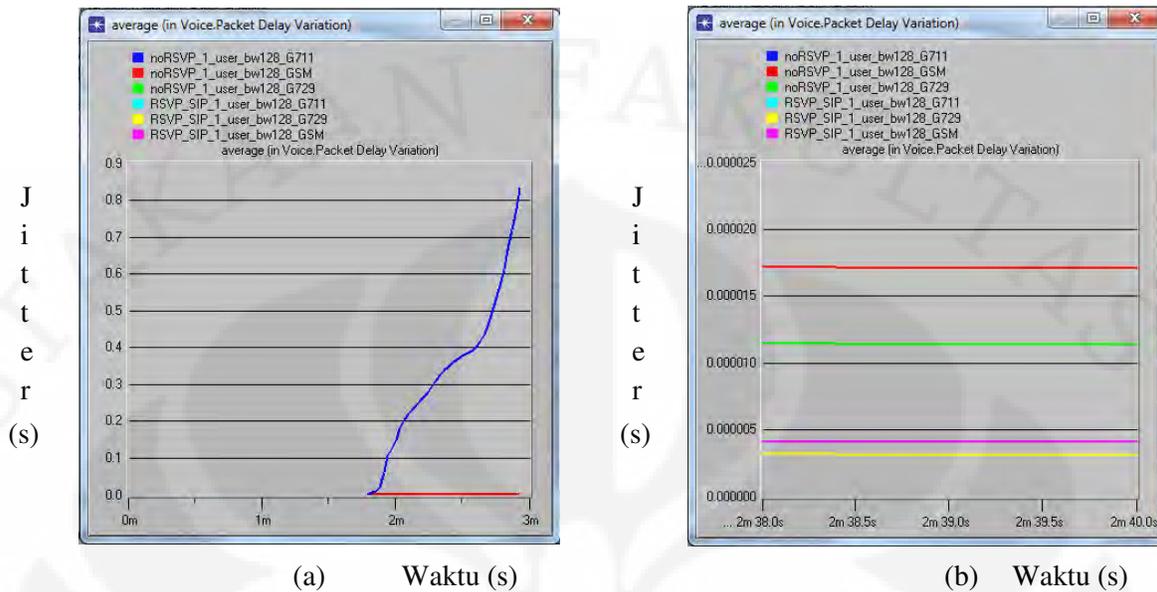
Nilai jitter berpengaruh ketika paket RTP yang datang akan diproses menjadi suara. Ketika nilai jitter lebih kecil dari waktu pemrosesan paket data, maka sebelum paket selesai diproses, paket selanjutnya telah datang untuk menunggu diproses

sehingga suara yang dihasilkan berkualitas baik. Namun ketika nilai jitter lebih besar dari delay processing maka suara akan terdengar terputus – putus. Hal tersebut dapat diantisipasi dengan menggunakan buffer jitter, sehingga paket yang datang akan dibuffer terlebih dahulu sebelum diproses. Tetapi ketika jitter dari paket jauh lebih besar dari buffer jitter maka kualitas suara akan menjadi jelek.

Sedangkan bila dibandingkan dengan jaringan yang menggunakan RSVP, terlihat pada Gambar 4.8 dan Gambar 4.9, nilai jitter yang terjadi pada codec G.711 menjadi jauh lebih kecil hingga nilainya mendekati nilai 30 ms. Sedangkan pada codec G.729 dan GSM, adanya RSVP turut berperan dalam menurunkan nilai jitter.



Gambar 4.8. (a) Grafik perbandingan jitter tanpa dan dengan RSVP pada bandwidth 64, (b) Detail grafik perbandingan jitter tanpa dan dengan RSVP pada bandwidth 64



Gambar 4.9. (a) Grafik perbandingan jitter tanpa dan dengan RSVP pada bandwidth 128, (b) Detail grafik perbandingan jitter tanpa dan dengan RSVP pada bandwidth 128

BAB 5

KESIMPULAN

- Aplikasi VoIP yang dilakukan pada jaringan harus memenuhi persyaratan kualitas (QoS) seperti delay dan jitter.
- Pemilihan codec, besarnya bit rate, serta penggunaan protokol RSVP turut berperan dalam mempengaruhi parameter QoS.
- Pada pengujian parameter delay tanpa protokol RSVP dengan bandwidth 64 Kbps, codec G.711 memiliki nilai delay yang paling besar diantara yang lain. Begitu pula dengan bandwidth 128, codec G.711 juga memiliki nilai delay yang paling kecil. Hal ini disebabkan karena G.711 memiliki payload yang paling besar yaitu 214 bytes.
- Pada pengujian parameter delay dengan menggunakan protocol RSVP, nilai delay yang terjadi pada codec menjadi lebih kecil bila dibandingkan dengan yang tidak menggunakan protocol RSVP. Hal ini terjadi karena protocol RSVP akan mereservasi terlebih dahulu sumber daya yang ada sehingga nilai delay yang terjadi dapat dikurangi.
- Pada pengujian parameter jitter tanpa menggunakan RSVP, nilai jitter yang terjadi pada G.711 melebihi batas nilai yang direkomendasikan oleh ITU-T, yaitu sebesar 30 ms, baik pada bandwidth 64 Kbps maupun pada bandwidth 128 Kbps.
- Pada pengujian parameter jitter dengan menggunakan RSVP, nilai jitter yang terjadi pada codec G.711 menjadi lebih kecil, mendekati standar yang direkomendasikan. Hal ini terjadi karena protocol RSVP akan mereservasi sumber daya yang ada terlebih dahulu sehingga nilai jitter yang terjadi dapat dikurangi.
- Performansi VoIP paling baik jika menggunakan codec G729. codec ini memiliki nilai delay dan jitter paling rendah dan kualitas suara yang cukup bagus.

DAFTAR ACUAN

- [1] Hardy, William C. (2003). *VoIP Service Quality*. New York : McGraw-Hill.
- [2] Voice over internet protocol.
http://en.wikipedia.org/wiki/Voice_over_Internet_Protocol, diakses tanggal 7 Juni 2010.
- [3] Session initiation protocol.
http://en.wikipedia.org/wiki/Session_Initiation_Protocol, diakses tanggal 6 Juni 2010.
- [4] OPNET IT Guru [Computer software]. *RSVP Model Description*. Model Description Reference Manual.
- [5] *Cisco Voice Over IP Student Guide (Version 4.2)* . (2004). Cisco System,Inc.
- [6] Codec. <http://en.wikipedia.org/wiki/Codec>, diakses tanggal 6 Juni 2010.
- [7] Huerta, Juan M. , Richard M. Stern. *Speech Recognition from GSM Codec Parameters*. Pittsburgh.
http://www.cs.cmu.edu/~robust/Papers/icslp98_juan.pdf.
- [8] OPNET IT Guru [Computer software]. *Configuring Applications and Profiles*.