

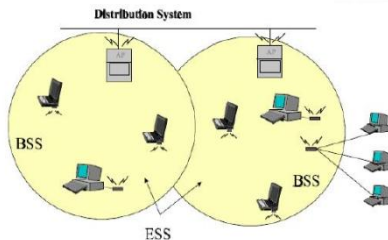
BAB II

DASAR TEORI

2.1 WIRELESS LOCAL AREA NETWORK (WLAN)

2.1.1. Pengenalan *Wireless LAN*

Wireless LAN (WLAN) merupakan suatu jaringan *Local Area Network* (LAN) yang tidak menggunakan kabel atau dapat dikatakan LAN nirkabel yang menggunakan media gelombang radio. WLAN menggunakan frekuensi radio untuk komunikasi antar perangkat komputer. Sistem jaringan WLAN atau Wi-Fi tidak memerlukan penghubung kabel antar komputer. Jaringan *wireless* hanya membutuhkan ruang, yang mana jarak jangkauan *network* dibatasi kekuatan pancaran sinyal radio dari masing-masing komputer. Jaringan ini memiliki kelebihan dan kekurangan yaitu *user* tidak dibatasi oleh ruang gerak tetapi hanya dibatasi pada jarak jangkauan dari satu titik pemancar Wi-Fi.



Gambar 2.1 Arsitektur Jaringan WLAN ^[4]

2.1.2. Kelebihan *Wireless LAN*

- a. Menyediakan pengaksesan informasi *real time*.
- b. Jaringan yang disediakan lebih luas dibandingkan dengan jaringan yang menggunakan kabel sebagai media transmisinya.
- c. Pemasangan dan instalasi yang mudah dan cepat.
- d. Memberikan keuntungan biaya dalam jangka panjang.
- e. Mudah dalam mengakses jaringan selama berada dalam area *access point*.

2.1.3. Kekurangan *Wireless LAN*

- a. Kecepatan transfer data akan semakin berkurang, jika terjadi penambahan komputer dalam suatu jaringan.
- b. Keamanan data lebih rendah.
- c. Mudah dipengaruhi oleh beberapa gangguan alam seperti hujan dan petir.
- d. Memiliki *bandwidth* yang rendah.

2.1.4. Standar *Wireless LAN*

Sistem 802.11 secara umum disebut dengan Wi-Fi. Langkah awal yang sangat diperhatikan dalam membangun perencanaan WLAN adalah mengenali

beberapa jenis standar IEEE (*Institute of Electrical and Electronics Engineers*) 802.11. Pada standar-standar tersebut tersedia kebutuhan yang sudah ditetapkan, seperti *radio band*, cakupan dan kebutuhan lainnya. Dengan mengetahui kebutuhan yang disediakan oleh standar tersebut, maka dapatlah ditentukan rancangan WLAN (IEEE 802.11).

Saat ini terdapat dua kategori standar IEEE 802.11, yang merupakan standar protokol dasar yang untuk sistem Wi-Fi, seperti 802.11a, 802.11b, dan 802.11g. Adapun inovasi terbaru dari standar WLAN untuk mengatasi kelemahan pada standar yang sudah ada. Contoh standar kedua adalah 802.11d, e, f, h, i, dan j.^[5]

Secara garis besar, spesifikasi standar WLAN yang umum digunakan, seperti pada Tabel 2.1.

Tabel 2.1 Standar IEEE 802.11^[5]

Standar	<i>Radio Band</i>	Modulasi	Cakupan	<i>Data Rate</i>
802.11b	2,4 GHz	DSSS	100 m	11 Mbps
802.11a	5 GHz	OFDM	50 m	54 Mbps
802.11g	2,4 GHz	OFDM	100 m	54 Mbps

2.2 VOICE OVER INTERNET PROTOCOL (VOIP)

VOIP merupakan suatu teknologi yang memiliki kemampuan untuk melakukan pertukaran suara secara interaktif melalui *Internet Protocol* (IP).^[1] Pertukaran suara dengan kata lain untuk mengirimkan suara dalam bentuk paket-paket melalui jaringan IP. VOIP dapat diimplementasikan pada setiap jaringan data yang menggunakan IP, seperti internet, intranet dan *Local Area Network* (LAN). Pada VOIP, sinyal suara di digitalisasi, dikompresi dan diubah ke IP paket dan kemudian ditransmisikan kembali melalui jaringan IP. *Protocol signaling* digunakan untuk mengatur dan mengatasi panggilan yang *down*, membawa informasi yang dibutuhkan untuk mencari pengguna dan kemampuan untuk bernegosiasi. VOIP biasa juga disebut dengan *Internet Telephony*. Proses konversi suara ke bentuk sinyal digital dilakukan dengan tiga cara yaitu :^[6]

1. *Voice Sampling*

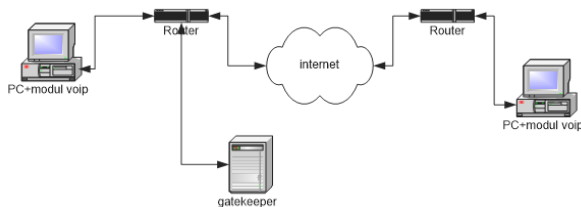
- *Codec* menyampling gelombang suara dalam interval waktu tertentu dan memberikan nilai pada setiap sampelnya.
- Interval waktu : 8000 siklus/detik (frekuensi sampling 8 kHz) atau 16000 siklus/detik (frekuensi sampling 16 kHz).

2. Kuantisasi

Mengubah nilai sampel menjadi bentuk diskrit agar bisa direpresentasikan dalam bit-bit.

3. Coding

- Sampel-sampel diakumulasikan dalam periode waktu tertentu, dikodekan menjadi kelompok bit-bit yang dinamakan *frame*.
- Pada proses *coding* ini dilakukan kompresi informasi agar didapat jumlah bit yang minimal dalam setiap sampel suara.
- Proses *coding* ini menjadi bagian terpenting dari seluruh proses konversi, karena menentukan efisiensi *codec* dalam kaitannya dengan *bandwidth* yang diperlukan.



Gambar 2.2 Konfigurasi Jaringan VOIP [7]

2.2.1. Kelebihan VOIP

1. Biaya lebih murah untuk sambungan langsung jarak jauh (SLJJ).

2. Dengan adanya akses internet biaya percakapan menjadi sangat murah, terlebih lagi untuk lokasi yang berbeda dan berjauhan.
3. VOIP dapat menggabungkan ke banyak sambungan, seperti PC ke telepon biasa, IP Phone *handset*, dan sambungan lainnya.
4. VOIP menggunakan *bandwidth* yang lebih kecil daripada telepon biasa sehingga suara yang dilewatkan membutuhkan kecepatan akses yang tidak terlalu besar.
5. Penggunaan *gateway* berfungsi untuk menghubungkan antar pesawat telepon biasa untuk melakukan komunikasi.

2.2.2. Kekurangan VOIP

1. Berpotensi menyebabkan jaringan terhambat, karena semakin banyak penggunaan VOIP, maka ada potensi jaringan data akan penuh.
2. Kualitas suara tidak sejernih telepon biasa, tergantung dari koneksi akses internet yang digunakan.
3. Ada jeda dalam berkomunikasi, yang disebabkan oleh proses perubahan data menjadi suara dan jeda jaringan akibat akses internet yang digunakan.

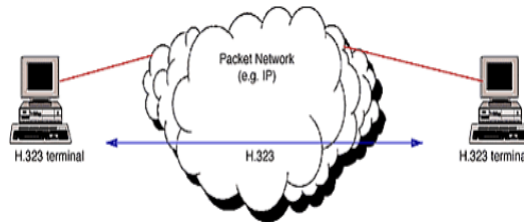
4. Tidak pernah ada jaminan kualitas yang dihasilkan akan baik.

2.2.3. Protocol Signaling VOIP

2.2.3.1. H.323

a) Pengenalan H.323

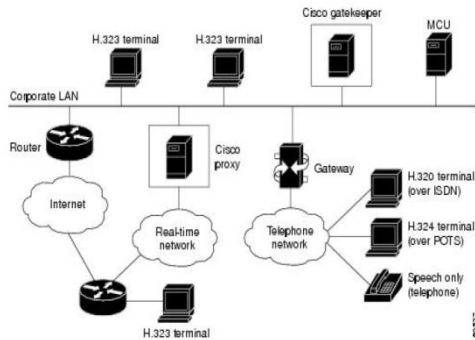
Standar H.323 merupakan sebuah protokol standar untuk transmisi pada audio, video, dan komunikasi data secara *real-time* melalui jaringan *packet-based* yang tidak menyediakan jaminan kualitas layanan atau QoS (*Quality of Service*).



Gambar 2.3 Terminal H.323 Pada Jaringan *Packet* [8]

b) Komponen H.323

Standar H.323 mempunyai komponen-komponen, untuk menyediakan layanan komunikasi multimedia *point to point* dan *point to multipoint*. Komponen-komponen tersebut seperti pada Gambar 2.4.

Gambar 2.4 Komponen H.323 ^[9]

c) Daerah H.323

Pada daerah H.323 yang dimaksud adalah pengumpulan semua terminal-terminal, gateway, dan MCU yang diatur oleh satu gatekeeper. Sebuah daerah H.323 hanya memiliki satu gatekeeper.

d) Spesifikasi Protokol H.323

a. *Audio Codec*

Protokol H.323 menyediakan layanan minimum *audio codec*. Semua terminal-terminal harus memiliki paling tidak mendukung satu *audio codec*. ITU-T merekomendasikan *audio codec* seperti G.711 (*bit rate* 64 kbps), G.722 (64,56, dan 48 kbps), G.723.1 (5,3 dan 6,3 kbps), G.728 (16 kbps) dan

G.729 (8 kbps). H.323 mendukung semua *audio codec* tersebut.

b. *Video Codec*

Untuk *video codec*, ITU-T merekomendasikan H.261 sebagai *video codec*. Setiap H.323 terminal harus menyediakan komunikasi video yang harus mendukung *video encoding* dan *decoding*.

2.2.3.2. *Session Initiation Protocol (SIP)*

a) Pengenalan SIP

SIP merupakan produk yang dikeluarkan oleh IETF (*Internet Engineering Task Force*). Berdasarkan RFC 2543 SIP didefinisikan sebagai arsitektur protokol pensinyalan yang di distribusikan untuk pembuatan aplikasi multimedia, termasuk VOIP [2]. Protokol SIP didukung oleh beberapa protokol untuk melaksanakan tugasnya diantaranya protokol RSVP sebagai protokol pemesanan pada jaringan. Protokol RTP (*Real time Transport Protocol*) dan RTCP (*Real time Transport Control Protocol*) sebagai protokol transmisi media. Protokol SDP (*Session Description Protocol*) sebagai protokol

untuk mendeskripsikan sesi media. SIP sebagai protokol pensinyalan yang mengontrol sesi multimedia seperti panggilan telepon internet maupun konferensi multimedia. Berikut 5 hal pengontrolan (baik dari segi pembuatan dan penghentian) komunikasi multimedia, yaitu ;

1. Letak pengguna (*user location*)

Merupakan mekanisme untuk menentukan *endpoint* (sistem terakhir) yang digunakan untuk komunikasi.

2. Kemampuan pengguna (*user capabilities*)

Merupakan mekanisme kemampuan pengguna untuk menentukan media dan parameter media yang digunakan.

3. Keberadaan pengguna (*user availability*)

Merupakan mekanisme keberadaan pengguna untuk menentukan jika user terpanggil sedang akan ikut serta di dalam sesi komunikasi.

4. Pengaturan panggilan (*call setup*)

Merupakan mekanisme pengaturan panggilan untuk mengatur pembuatan panggilan,

baik dari sisi pemanggil (*caller*) dan sisi yang dipanggil (*callee*).

5. Penanganan panggilan (*call handling*)

Merupakan mekanisme penanganan panggilan untuk mengatur panggilan sisipan dan penanganan panggilan pihak ketiga seperti *call transfer* dan *call waiting*. Pada hal ini pemberhentian tiap panggilan juga dapat diatur.

b) Fungsi SIP

Fungsi pada SIP terdiri dari 3 fungsi, diantaranya :

1. Permulaan panggilan (*call initiation*)

Memiliki fungsi untuk membangun sebuah sesi komunikasi, melakukan negosiasi media transfer protokol, dan mengundang *user* lain untuk bergabung di dalam sesi komunikasi yang ada.

2. Perubahan panggilan (*call modification*)

Memiliki fungsi di mana jika diperlukan, SIP dapat melakukan perubahan atau modifikasi pada sesi komunikasi,

3. Penghentian panggilan (*call termination*)

Memiliki fungsi untuk memberhentikan atau menutup sesi komunikasi yang ada.

c) Komponen SIP

Adapun komponen dari protokol SIP, yaitu :

a. *User agent*

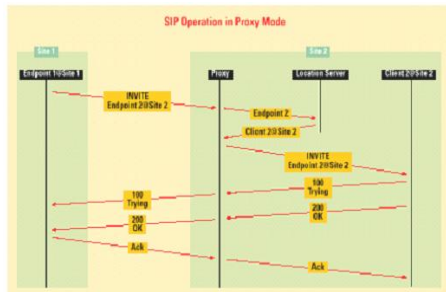
Merupakan komponen SIP yang memulai, menerima dan menutup sesi komunikasi. Pada user agent terdiri dari 2 komponen yaitu *user agent client* (UAC) dan *user agent server* (UAS). UAC bertindak sebagai *client* yang membangkitkan atau mengenali *requests*, dan UAS bertindak sebagai *server* yang menerima panggilan dan merespon *request* yang dilakukan oleh *client*.

b. *Network server*

Pada *network server*, terdapat 3 tipe *server*, yaitu :

1. *Proxy server*

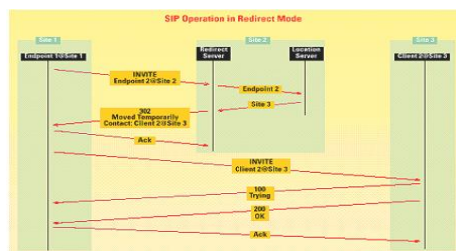
Merupakan server yang berfungsi untuk menerima dan mengirim *request*, melakukan *routing*, memastikan *request* dapat diterima oleh *client* dan melakukan *forward request* pada *user agent* lainnya. Untuk proses panggilan (*call flow*) pada SIP, seperti pada Gambar 2.5.



Gambar 2.5 Proses *Call Flow* Pada SIP *Proxy Server* ^[10]

2. *Redirect server*

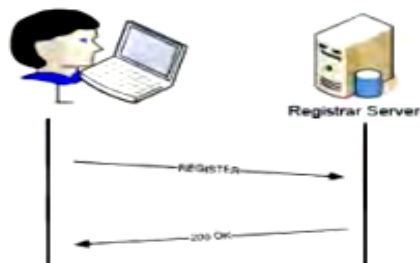
Merupakan server yang berfungsi untuk menerima *request*, memberikan atau memetakan sebuah daftar alamat baru ke alamat SIP pengguna yang meminta. Pada bagian ini terdapat *location server* yang memberikan informasi ke *proxy* atau *redirect server* mengenai lokasi dari *client*, seperti pada Gambar 2.6.



Gambar 2.6 Proses *Call Flow* Pada SIP *Redirect Server* ^[10]

3. *Restart server*

Merupakan server yang berfungsi untuk menerima dan memproses pesan pendaftaran dari *client*. *Registrar server* tidak melakukan *forward request*. Untuk *call flow* pada *restart server* seperti pada Gambar 2.7.



Gambar 2.7 Proses *Call Flow* Pada *Restart Server* ^[11]

c. *SIP message*

SIP banyak mendefinisikan sebuah pesan. Setiap pesan digunakan untuk melakukan komunikasi antara *client* dan *server*. Berikut pesan-pesan pada SIP, yaitu :

1. *INVITE* : pesan yang digunakan untuk mengundang *user* untuk melakukan sebuah panggilan
2. *BYE* : pesan yang digunakan untuk mengakhiri koneksi antara dua *end-points*

3. *ACK* : pesan yang digunakan untuk mempercayai pertukaran dari undangan pesan.
4. *OPTIONS* : pesan yang digunakan untuk memperoleh informasi mengenai kemampuan pada sebuah panggilan.
5. *REGISTER* : pesan yang digunakan untuk memberikan informasi mengenai lokasi dari *user* menuju *server* pendaftaran SIP.
6. *CANCEL* : pesan yang digunakan untuk mengakhiri pencarian *user*.

2.2.4. *Coder – Encoder (CODEC)*

Codec dalam aplikasi VOIP bertujuan untuk mengkonversikan sinyal suara input menjadi bentuk digital, mentransmisikan sinyal tersebut ke penerima dan merekonstruksi sinyal suara asli agar didengar oleh penerima.^[6]

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. Setiap jenis *codec* memiliki *bit rate* dan kualitas yang berbeda-beda. Berikut spesifikasi pada masing-masing *codec*, seperti pada Tabel 2.2.

Tabel 2.2 Spesifikasi *Codec* ^[12]

<i>Codec</i>	<i>Bit rate (Kbps)</i>	<i>Sample size (Bytes)</i>	<i>Sample rate (ms)</i>	<i>MOS Quality</i>	<i>Voice Payload Size (Bytes)</i>	<i>Voice Payload Size (ms)</i>	<i>Packets Per Second (PPS)</i>	<i>Bandwidth Ethernet (Kbps)</i>
G.711	64	80	10	4.3	160	20	50	87.2
G.729	8	10	10	3.7	20	20	50	31.2
G.723.1	6.3	24	30	3.9	24	30	33.3	21.9
G.723.1	5.3	20	30	3.8	20	30	33.3	20.8
G.726	32	20	5	3.85	80	20	50	55.2
G.726	24	15	5	-	60	20	50	47.2
G.728	16	10	5	3.61	60	30	33.3	31.5
G.722	64	80	10	4.13	160	20	50	87.2
iLBC	15.2	38	20	4.14	38	20	50	38.4
iLBC	13.33	50	30	-	50	30	33.3	28.8

Untuk penjelasan beberapa *codec* sebagai berikut :

a. G.711

Codec G.711 memiliki *bit rate* 64 Kbps dan tidak membutuhkan lisensi. Dengan tidak adanya lisensi, maka *codec* ini dapat digunakan secara bebas. *Codec* ini menggunakan teknik kompresi PCM (*Pulse Code Modulation*). Dalam perhitungan untuk menghasilkan 64 Kbps yaitu kecepatan sampling 8000 sample/detik dan terdiri dari 8 bit untuk setiap sampel, sehingga menghasilkan 64000 bit per detiknya (64 Kbps). *Codec* ini tidak

membutuhkan daya CPU yang terlalu besar sehingga cocok untuk pengimplementasian yang sederhana.

b. G.726

Codec G.726 menggunakan teknik kompresi ADPCM (Adaptive Differential Pulse-Code Modulation). Codec ini memiliki bit rate diantaranya 16 Kbps, 24 Kbps 32 Kbps dan 40 Kbps. Codec yang paling sering digunakan adalah codec dengan bit rate 32 Kbps. G.726 memberikan kualitas yang identik mendekati codec G.711. Sama seperti dengan codec G.711, codec G.726 ini juga tidak membutuhkan lisensi. Codec G.726 pernah mengalami kegagalan pada tahun 1990 karena tidak mampu membawa sinyal-sinyal modem dan fax, tetapi karena performansi dari bandwidth dan CPU, membuat codec ini kembali lagi. G.726 secara khusus menarik karena tidak memerlukan banyak proses dari sebuah sistem.

c. G.723.1

Codec G.723.1 memiliki bit rate 6,3 Kbps (menggunakan 24 byte frame), yang mana codec dengan bit rate 6,3 Kbps ini menggunakan algoritma MPC-MLQ. Untuk codec G.723.1 dengan bit rate

5,3 Kbps (menggunakan 20 *byte frame*), yang mana menggunakan algoritma ACELP. *Codec* ini membutuhkan lisensi untuk hak paten yang mencakup algoritma tersebut. Hak paten terakhir akan berakhir pada tahun 2014. *Codec* ini dengan sepenuhnya berbeda dari *codec* G.723.

d. G.729A

Codec G.729A memiliki *bit rate* 8 Kbps. *Codec* G.729A ini mempertimbangkan bagaimana menggunakan *bandwidth* yang kecil, namun dapat mengirimkan kualitas suara yang mengesankan/bagus. *Codec* ini menggunakan teknik kompresi CS-ACELP (*Conjugate-Structure Algebraic-Code-Excited Linear Prediction*). *Codec* ini membutuhkan lisensi, sehingga tidak dapat digunakan secara bebas. Karena hak paten, *codec* ini tidak dapat digunakan tanpa membayar biaya lisensi, bagaimanapun, *codec* ini sangat terkenal dan dengan sangat baik banyak mendukung sistem dan telepon yang berbeda. Untuk mencapai kualitas suara yang mengesankan, maka *codec* ini memerlukan upaya/usaha dari CPU. CELP merupakan metode yang paling terkenal dalam hal pengkompresian *speech*.

e. iLBC

Codec iLBC menyediakan kesatuan yang menarik dari penggunaan *bandwidth* yang rendah/kecil dan kualitas suara yang mengesankan. *Codec* ini memiliki *bit rate* 13,3 Kbps (menggunakan 30 ms *sample frame*) dan 15,2 Kbps (menggunakan 20 ms *sample frame*).

2.3 RESOURCE RESERVATION PROTOCOL (RSVP)

2.3.1. Pengenalan RSVP

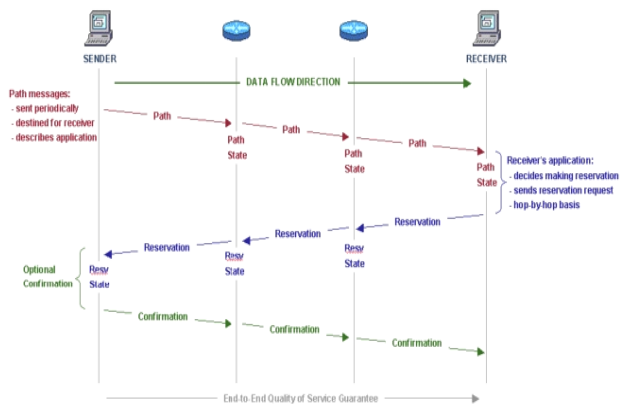
RSVP merupakan sebuah protokol yang menyediakan pemesanan bandwidth atau *resource* pada jaringan. RSVP mengatur aliran trafik dengan QoS yang spesifik sepanjang *path* yang dipesan oleh *receiver*. Kemampuan yang dimiliki oleh protokol RSVP antara lain;

- a. RSVP mendukung semua *signaling protocol*, termasuk SIP
- b. RSVP mendukung reservasi *bandwidth* untuk aplikasi audio dan video
- c. RSVP memiliki kemampuan *retry*
- d. RSVP mengintegrasikan *Differentiated Services* (DiffServ) QoS dengan profil *Differentiated Services Code Point* (DSCP).

2.3.2. Fungsi Kerja RSVP

Adapun fungsi kerja dari protokol RSVP yaitu menyediakan *message*. RSVP *Message* akan dipertukarkan oleh *sender* dan *receiver* dalam proses *resource reservation*. Ada beberapa macam *message* RSVP, tetapi *message* yang dianggap paling utama yaitu *Resv Message* dan *Path Message*. *Sender* akan mengirim *Path Message* ke *receiver*, dan *receiver* akan mengirim *Resv Message* ke *sender* dengan membalikkan jalur yang didefinisikan pada *Path Message* sebelumnya.

RSVP merupakan protokol yang bersifat *connection-oriented* dimana terdapat tiga fase dalam mengadakan reservasi, yaitu fase pembukaan hubungan, fase hubungan, fase pemutusan hubungan. Proses *call flow* RSVP seperti pada Gambar 2.8.

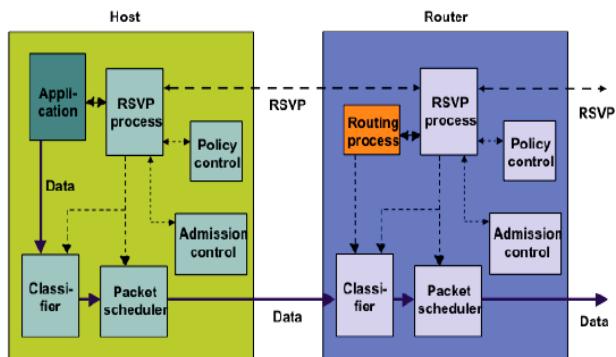


Gambar 2.8 Proses *Call Flow* RSVP [13]

Pada RSVP *data flow* dapat berupa *unicast* dan *multicast*. RSVP *message* ditransmisikan dari satu *node* ke *node* yang lain sepanjang jalur data. ^[14]

2.3.3. Arsitektur RSVP

Protokol RSVP digunakan oleh *host* untuk meminta QoS dari jaringan yang digunakan pada aplikasi seperti *video conference*, VOIP dan komunikasi multimedia lainnya. Protokol RSVP digunakan oleh *router* untuk mengantar dan mengirim permintaan QoS ke semua *node* dan ke semua *router* lain sepanjang jalur aliran data.



Gambar 2.9 Arsitektur RSVP ^[15]

Proses RSVP dilakukan dengan menjalankan dua modul yaitu *policy control* dan *admission control*. Pada saat kedua modul ini berjalan dengan baik, maka RSVP

akan membentuk *packet scheduler* dan *classifier*. Proses terhadap permintaan ini adalah pemesanan sumber daya (*resource reservation*) pada jalur yang akan digunakan oleh aplikasi. Protokol RSVP memungkinkan *router* memesan *bandwidth* yang cukup pada *interface* untuk meningkatkan performansi dan kualitas dari jaringan.

2.3.4. RSVP Message

Protokol RSVP menyediakan empat tipe pesan, yaitu :

a. *Path message*

Pada tipe pesan ini, pesan *path* dikirim oleh masing-masing pengirim di sepanjang rute. Pada *path message* terdapat pesan-pesan yang dibawa seperti, parameter RSVP untuk panggilan, termasuk nilai rata-rata dalam satuan *bit per second* (bps), ukuran paket, *burst depth*, dan *peak rate* (puncak maksimum *path message*).

b. *Reservation-request message*

Pada tipe pesan ini, dikirimkan oleh masing-masing *host* penerima ke pengirim. Protokol RSVP akan membawa permintaan reservasi berupa *Resv message* ke seluruh *node* kembali ke sumber data, hal ini dilakukan oleh *router*. *Reservation message* membawa pesan (meneruskan) berupa pesan yang diterima dari *path message* melalui *router*.

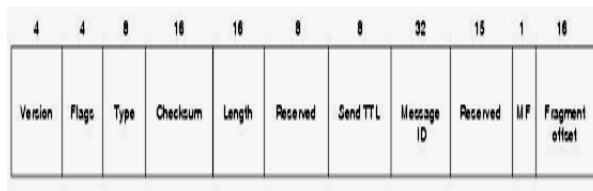
c. *Error and confirmation message*

Pada tipe pesan ini, terdapat tiga kesalahan dan konfirmasi bentuk pesan yang ada, yaitu : pesan *path - error*, pesan *reservation-request error*, dan pesan *reservation-request acknowledgment*.

d. *Teardown message*

Pada tipe pesan ini akan menghapus *path* dan *state reservation* tanpa menunggu waktu pada *teardown* selesai. Pada tipe pesan ini, RSVP juga mendukung dua tipe pesan pada *teardown message*, yaitu *path-teardown* dan *reservation-request teardown*. Pesan ini muncul pada saat panggilan berakhir.

Berdasarkan pesan-pesan di atas, sehingga RSVP *messages* yang berisi pesan-pesan tersebut dapat dinyatakan dalam enkapsulasi *header fragment* RSVP *message*, sebagai berikut :



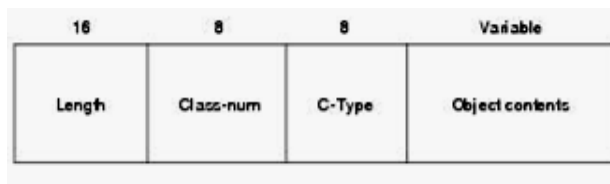
Gambar 2.10 RSVP *Header* ^[16]

Adapun penjelasan dari RSVP *header* sebagai berikut :

1. *Version* – terdiri dari 4 bit yaitu 00, 01, 02, dan 03 bit. Mengindikasikan protokol versi angka (versi 1).
2. *Flags* – terdiri dari 4 bit yaitu 04, 05, 06, dan 07, dengan *flags* yang belum didefinisikan.
3. *Type* – terdiri dari 8 bit, yaitu 08, 09, 10, 11, 12, 13, 14, 15, dengan beberapa tipe pesan didalamnya, yaitu : *path*, *reservation-request*, *path-error*, *reservation-request error*, *path-teardown*, *reservation-teardown* dan *reservation-request acknowledgment*.
4. *Checksum* – terdiri dari 16 bit yaitu 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30 dan 31. *Frame* ini merepresentasikan standar protokol TCP/UDP untuk memeriksa jumlah RSVP *message*.
5. *Length* – terdiri dari 16 bit yang merepresentasikan paket RSVP dalam bentuk *bytes*.
6. *Reserved* – paket reservasi berupa *frame* sebesar 8 bit dikirimkan.
7. *Send TTL* – terdiri dari 8 bit yang mengindikasikan IP TTL (*time-to-live*) yang menunjukkan sukses tidaknya pesan reservasi berhasil atau tidak.

8. *Message ID* – terdiri dari 32 bit yang menyediakan sebuah label yang dibagikan oleh semua *fragment* dari satu pesan ke pesan selanjutnya atau sebelumnya, yang dikenal dengan RSVP *hop*.
9. *More Fragments (MF) Flag* – terdiri dari 1 bit kata dengan 7 bit yang lebih tinggi yang dispesifikasikan sebagai reservasi. MF diatur untuk semua *fragment* tetapi terletak pada *fragment* terakhir dari sebuah pesan.
10. *Fragment Offset* – terdiri dari 24 bit yang merepresentasikan *byte offset* dari *fragment* dalam sebuah pesan.

Pada bagian ini akan membahas mengenai enkapsulasi *fragment* pada RSVP, seperti pada Gambar 2.11



Gambar 2.11 Enkapsulasi RSVP ^[16]

Setiap enkapsulasi RSVP *message* konsisten terdiri dari 32 bit *header*. Pada *frame length* memiliki 16 bit yang berisi total panjang objek dalam *bytes*. Total panjang objek ini dapat bernilai penjumlahan 4 dan

paling tidak lebih kecil dari 4. Pada *class-num* terdiri dari 8 bit yang mengidentifikasi kelas objek. Setiap kelas objek memiliki sebuah nama. Sehingga, pada saat RSVP mendefinisikan QoS (termasuk permintaan pesan reservasi), hal ini dilakukan pada *frame length*. Begitu juga halnya dengan pesan *reservation-confirmation*, yang menyatakan permintaan reservasi telah dikonfirmasi.

Kelas objek dari *reservation-confirmation* yaitu 15. Pada *frame C-type*, hal ini mengindikasikan panjang maksimum dari sebuah objek yaitu 65528 bytes. Pada *frame object contents* merupakan *frame* yang menspesifikasikan ketiga *frame* sebelumnya (*length*, *class-num* dan *c-type*).

2.4 PARAMETER KUALITAS LAYANAN VOIP

Untuk menjamin sebuah kualitas yang akan diterima oleh *user*, sangat penting diperhatikan level parameter-parameter untuk menjamin sebuah kualitas layanan, seperti pada Tabel 2.3.

Tabel 2.3 Level Parameter VOIP^[17,3]

<i>Parameters</i>	Level yang dapat diterima
<i>Delay</i>	≤ 150 ms
<i>Jitter</i>	≤ 30 ms
<i>Packet Loss</i>	$\leq 1\%$

Merujuk pada Tabel 2.3, adapun penjelasan untuk masing-masing parameter sebagai berikut:

- 1) *Delay*, merupakan suatu faktor penting yang menentukan kualitas VOIP. *Delay* merupakan waktu total yang dibutuhkan paket mulai dari dikirim sampai diterima. *Delay* yang direkomendasikan oleh ITU-T yaitu kurang dari 150 ms.^[3]

Tabel 2.4 Pedoman Batasan *Delay* ITU-T G.114 ^[18,19]

<i>Delay</i> (ms)	Keterangan	Kategori
0 – 150	Dapat diterima.	Baik
150 – 400	Dapat diterima. Namun administrator jaringan harus waspada terhadap segala sesuatu yang dapat mempengaruhi kualitas jaringan.	Cukup
Di atas 400	Secara umum, tidak dapat diterima. Namun untuk kasus-kasus khusus, nilai batas ini dapat berubah.	Buruk

Adapun rumus untuk perhitungan *packet delay* pada persamaan (1) : ^[20]

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

- 2) *Jitter*, merupakan variasi waktu tunda yang terjadi akibat adanya selisih waktu atau *interval* kedatangan

paket di sisi penerima. Parameter *jitter* sangat mempengaruhi kualitas suara. Semakin besar *jitter* maka suara yang dihasilkan juga akan semakin tidak jelas. *Jitter* yang direkomendasikan oleh ITU-T G.1010 yaitu kurang dari 30ms. ^[21]

Tabel 2.5 Pedoman Batasan *Jitter* ^[21, 22]

<i>Jitter</i> (ms)	Keterangan	Kategori
0 – 20	Dapat diterima (<i>good</i>)	Baik
20 -50	Dapat diterima (<i>acceptable</i>)	Cukup
Di atas 50	Tidak dapat diterima (<i>poor</i>).	Buruk

Adapun rumus untuk perhitungan *jitter* pada persamaan (2) : ^[23]

$$J_1 = \text{abs}(t_2-t_1), J_2 = \text{abs}(t_3-t_2) \dots\dots\dots (2)$$

- 3) *Packet Loss*, merupakan hilangnya paket data pada saat proses pengiriman dari tujuan ke penerima. *Packet Loss* yang direkomendasikan oleh ITU-T yaitu kurang dari 1%. ^[18]

Tabel 2.6 Pedoman Batasan *Packet Loss* ^[18, 22]

<i>Packet loss</i> (%)	Keterangan	Kategori
0 – 0,5	Dapat diterima (<i>good</i>)	Baik
0,5 – 1,5	Dapat diterima (<i>acceptable</i>)	Cukup
> 1,5	Secara umum, tidak dapat diterima (<i>poor</i>).	Buruk

Adapun rumus untuk perhitungan *packet loss* pada persamaan (3) : ^[23]

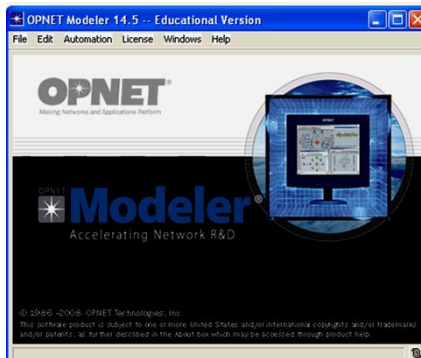
$$Packet\ loss = \frac{(Packet\ transmitted - Packet\ received)}{Packet\ transmitted} \times 100\% \dots\dots\dots (3)$$

- 4) *Throughput*, merupakan jumlah total kedatangan paket yang sukses yang diamati pada tujuan selama interval waktu tertentu dibagi oleh lamanya pengamatan. Adapun rumus untuk perhitungan *throughput* pada persamaan (4) : ^[24]

$$Throughput = \frac{Paket\ data\ yang\ diterima}{Lama\ pengamatan} \dots\dots\dots (4)$$

2.5 OPNET MODELER 14.5

OPNET (*Optimized Network Engineering Tools*) Modeler 14.5 merupakan sebuah simulator untuk merancang sebuah jaringan yang mana jaringan tersebut terlihat *real*. OPNET juga menyediakan sebagian besar komponen-komponen dari model jaringan yang beragam, topologi, protokol-protokol dan lainnya. Komponen-komponen tersebut diantaranya, *routing protocol*, topologi, aplikasi, jaringan, dan *link (interface)*. Hasil simulasi pada simulator OPNET ini berupa grafik dan tabel yang menampilkan hasil dari setiap parameter yang telah dikonfigurasi. Pada OPNET juga mendukung fasilitas *export file* dari hasil grafik ke dalam bentuk Excel (.xml), sehingga hasil simulasi tersebut berupa tabel.



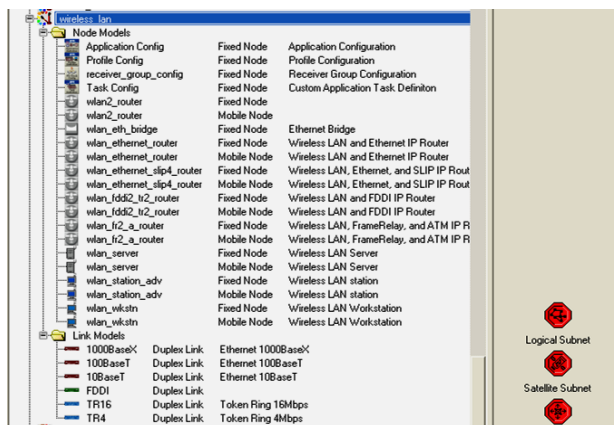
Gambar 2.12 Tampilan Simulator OPNET Modeler 14.5

Komponen-komponen untuk jaringan WLAN yang terdapat dalam OPNET Modeler 14.5, yaitu :

- a) *Wireless LAN router*, merupakan *access point* yang dilengkapi dengan kemampuan *routing*.
- b) *Wireless LAN workstation*, merupakan sebuah *workstation* yang mendukung *data rate* WLAN pada 1 Mbps, 2 Mbps, 5.5 Mbps, dan 11 Mbps.
- c) *Link 100BaseT*, merupakan koneksi *ethernet* yang beroperasi pada kecepatan 100 Mbps dengan menggunakan teknologi *baseband* sebagai metode pensinyalan untuk komunikasi jaringan. *Interface* ini dapat disebut juga sebagai *Fast Ethernet*.
- d) *Router*, bekerja untuk menyampaikan paket dari sumber ke tujuan sesuai alamat yang akan dituju, proses penyampaian paket ini disebut *routing*.

- e) *Switch*, merupakan penghubung beberapa perangkat untuk membentuk jaringan kecil atau LAN.

Berikut pada Gambar 2.13 memberikan gambaran mengenai komponen-komponen WLAN yang terdapat pada OPNET.



Gambar 2.13 Komponen WLAN Pada OPNET Modeler 14.5

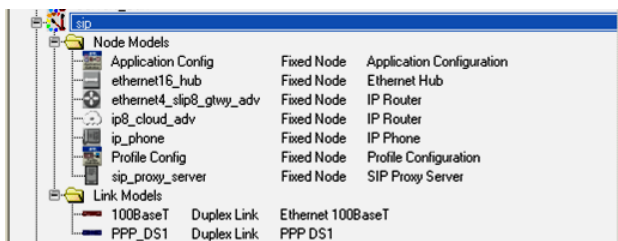
Komponen-komponen untuk SIP yang terdapat dalam OPNET Modeler 14.5 , yaitu :

1. *IP Cloud*, *IP cloud* direpresentasikan sebagai internet. Pada komponen ini terdapat atribut *packet latency* yang menspesifikasikan *delay (second)*.
2. *SIP Proxy Server*, merupakan sebuah *node server*, yang mana mendukung layanan SIP UAS (*User Agent*

Server). Komponen ini mendukung koneksi *ethernet* dengan kecepatan 10 Mbps, 100 Mbps, dan 1000 Mbps.

3. *Link* PPP DS1, komponen ini merupakan *link* yang memiliki *data rate* sebesar 1,544 Mbps.

Berikut pada Gambar 2.14 memberikan gambaran mengenai komponen-komponen SIP yang terdapat pada OPNET Modeler 14.5.

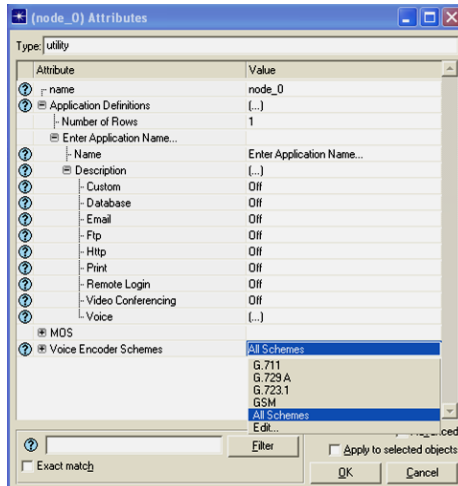


Gambar 2.14 Komponen SIP Pada OPNET Modeler 14.5

Pada komponen aplikasi konfigurasi, yang terdapat pada SIP dan WLAN mendukung jenis-jenis aplikasi seperti FTP, video *conferencing*, *voice*, dan *codec* untuk *voice*. Gambar 2.15 memberikan gambaran mengenai aplikasi apa saja yang disediakan.

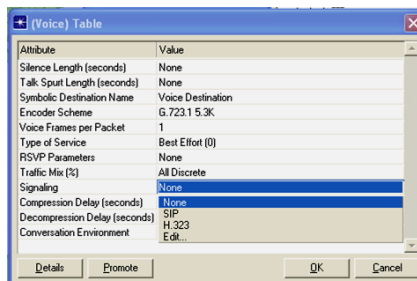
Pemilihan *codec* untuk *voice*, juga terdapat di dalam aplikasi konfigurasi. Penggunaan *codec* yang diinginkan dapat di tentukan secara manual, dengan memilih jenis-jenis *codec* yang telah terdapat di dalam OPNET Modeler 14.5.

Penyediaan protokol *signaling*, dapat di konfigurasi melalui komponen aplikasi konfigurasi, seperti pada Gambar 2.15.



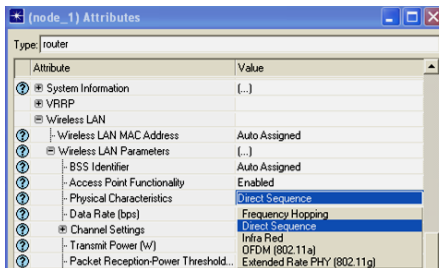
Gambar 2.15 Tampilan Aplikasi Konfigurasi Pada OPNET Modeler 14.5

Pada OPNET Modeler 14.5, protokol *signaling* yang disediakan yaitu SIP dan H.323. Penentuan protokol SIP ini terdapat pada komponen aplikasi konfigurasi.

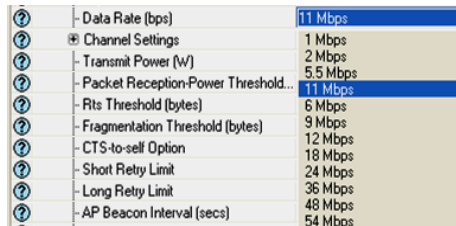


Gambar 2.16 Penyediaan *Protocol Signaling*

Pada pengaturan tipe standar yang akan digunakan pada *access point*, OPNET Modeler 14.5 menyediakan lima karakteristik fisik, yaitu *frequency hopping*, *direct sequence* (802.11b), *infra red*, OFDM (802.11a), dan 802.11g. *Data rate* yang tersedia mulai dari 1 Mbps – 54 Mbps.



Gambar 2.17 Penyediaan Standar 802.11



Gambar 2.18 Penyediaan *Data Rate*

Penyediaan teknologi fisik *wireless* pada OPNET Modeler menyediakan lima karakteristik fisik yaitu *frequency hopping*, *direct sequence* (802.11b), *infra red*, OFDM (802.11a), dan *extended rate* (802.11g). Seperti pada Gambar 2.17 dan Gambar 2.18 di atas penyediaan standar fisik *wireless* dan *data rate* yang disediakan oleh OPNET Modeler.