



UNIVERSITAS INDONESIA



**IMPLEMENTASI LAYANAN VIDEO CALL
PADA JARINGAN IMS**

SKRIPSI

**R. KHARISMA ADHIPUTRA SUARDY
0606074230**

**FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK ELEKTRO
DEPOK
JUNI 2010**



UNIVERSITAS INDONESIA

**IMPLEMENTASI LAYANAN VIDEO CALL
PADA JARINGAN IMS**

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik

**R. KHARISMA ADHIPUTRA SUARDY
0606074230**

**FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK ELEKTRO
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 : R.KHARISMA ADHIPUTRA SUARDY
NPM : 0606074230
Tanda Tangan :
Tanggal : 14 Juni 2010

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :
Nama : R. Kharisma Adhiputra Suardy
NPM : 0606074230
Program Studi : Teknik Elektro
Judul Skripsi : Implementasi Layanan Video Call Pada Jaringan
IMS

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

DEWAN PENGUJI

Pembimbing : Ir. Djamhari Sirat, M.Sc, Ph.D

Penguji : Ir. Muhammad Asvial, M.Sc, Ph.D

Penguji : Prima Dewi Purnamasari ST., MT., MSc.

Ditetapkan di : Depok

Tanggal : 28 Juni 2010

UCAPAN TERIMA KASIH

Puji syukur saya panjatkan kehadirat Allah SWT, karena atas segala rahmat dan hidayat-Nya saya dapat menyelesaikan skripsi ini. Saya menyadari bahwa skripsi ini tidak akan terselesaikan tanpa bantuan dari berbagai pihak. Oleh karena itu, saya mengucapkan terima kasih kepada :

1. Bapak Ir. Djamhari Sirat M.Sc., Ph.D selaku pembimbing skripsi ini, yang telah meluangkan waktunya, serta masukan-masukan selama bimbingan;
2. Orang tua saya yang selalu memberi dukungan baik secara langsung maupun tidak langsung
3. Rekan satu bimbingan: Deolens, Hartono, Swapo, Pelangi atas dukungan dan kerjasamanya dalam menyelesaikan skripsi ini.
4. Pak Nooriza, Pal Yulianus, Pak Panca, Bambang yang telah memberi masukan melalui hasil diskusi-diskusi yang telah dilakukan
5. Asisten lab STL atas support yang diberikannya
5. Teman-teman dan seluruh Sivitas Akademik Departemen Teknik Elektro yang tidak dapat saya sebutkan satu persatu.

Akhir kata, semoga Allah SWT berkenan membalas kebaikan semua pihak yang telah membantu. Semoga skripsi ini bermanfaat bagi perkembangan ilmu pengetahuan.

Depok, Juni 2010

Kharisma Adhiputra S.

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

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

Nama : R.Kharisma Adhiputra Suardy
NPM : 0606074230
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 :

IMPLEMENTASI LAYANAN VIDEO CALL PADA JARINGAN IMS

Berserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non Eksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia / formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta sebagai pemegang Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok
Pada tanggal : 14 Juni 2010
Yang menyatakan

R. Kharisma Adhiputra Suardy

ABSTRAK

Nama : R.Kharisma Adhiputra Suardy
Program Studi : Teknik Elektro
Judul : Implementasi Layanan Video Call Pada Jaringan IMS

Skripsi ini membahas rancang bangun jaringan IMS yang berbasis SIP menggunakan Open IMS core serta pengimplementasian layanan *video call* pada jaringan tersebut. Kualitas layanan video call diukur dengan memvariasikan besarnya *bandwidth* jaringan, parameter QoS objektif yang diukur adalah *delay* dan *packet loss*. Dari hasil pengukuran diperoleh bahwa layanan *video call* ini dapat bekerja dengan baik pada bandwidth 384 kbps keatas, dimana besarnya *delay* dan *packet loss* memenuhi standard yang dikeluarkan oleh ITU-T.

Kata kunci:
IMS, QoS, SIP, *Video call*

ABSTRACT

Name : R.Kharisma Adhiputra Suardy
Study Program: Electrical Engineering
Title : Implementation of Video Call service on IMS network

The focus of this study is build and design of IMS network that based on SIP using Open IMS Core and the implementation of video call service on it. The quality of video call service was measured by varying network's bandwidth, parameters of the objective QoS that were measured are delay and packet loss. From the measurement, video call service works well on bandwidth 384 kbps and above, where delay and packet loss that occurred satisfy ITU-T standard.

Key words:
IMS, QoS, SIP, *Video call*

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PERNYATAAN ORISINALITAS	ii
HALAMAN PENGESAHAN	iii
UCAPAN TERIMAKASIH	iv
LEMBAR PERSETUJUAN PUBLIKASI KARYA ILMIAH	v
ABSTRAK	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR GAMBAR	x
DAFTAR TABEL	xi
BAB 1 PENDAHULUAN	1
1.1 LATAR BELAKANG	1
1.2 TUJUAN	2
1.3 PEMBatasan MASALAH	2
1.4 METODOLOGI PENELITIAN	2
1.5 SISTEMATIKA PENULISAN	3
BAB 2 LANDASAN TEORI	5
2.1 SESSION INITIATION PROTOCOL (SIP)	5
2.1.1 Komponen – Komponen SIP	5
2.1.2 SIP Request/Method dan Response	6
2.1.3 Session Establishment	8
2.1.4 Komponen – Komponen SIP Request message	10
2.1.5 SIP Registration	12
2.2 SESSION DESCRIPTION PROTOCOL (SDP)	12
2.3 REAL-TIME TRANSPORT PROTOCOL (RTP)	14
2.4 REAL-TIME TRANSPORT CONTROL PROTOCOL (RTCP)	14
2.5 IP MULTIMEDIA SUBSYSTEM (IMS)	15
2.5.1 Arsitektur IMS	15
2.5.2 Policy and Charging Control	19

2.6 VIDEO CALL.....	19
2.7 PARAMETER QUALITY OF SERVICE (QOS)	20
2.7.1 Delay.....	20
2.7.2 Jitter	21
2.7.3 Packet Loss	21
BAB 3 INSTALASI DAN KONFIGURASI JARINGAN IMS	23
3.1 KONFIGURASI OPENIMSCORE	23
3.1.1 Download dan Compile Source Code	23
3.1.2 Konfigurasi Database	25
3.1.3 Konfigurasi DNS.....	25
3.2 KONFIGURASI CLIENT	28
3.3 KONFIGURASI ROUTER	30
BAB 4 PENGUKURAN DAN ANALISIS	32
4.1 SKENARIO PENGUJIAN	32
4.2 SIP MESSAGE FLOW	33
4.2.1 Registrasi	33
4.2.2 Session Establishment	33
4.3 RESPONSITIVITAS CORE IMS.....	35
4.3.1 Registration Time.....	35
4.3.2 Ringing Time	36
4.4 PERFORMANSI LAYANAN VIDEO CALL	37
4.4.1 Jaringan Dengan Bandwidth 96 kbps.....	37
4.4.2 Jaringan Dengan Bandwidth 128 kbps.....	38
4.4.3 Jaringan Dengan Bandwidth 256 kbps.....	39
4.4.4 Jaringan Dengan Bandwidth 384 kbps.....	40
4.4.5 Jaringan Dengan Bandwidth 512 kbps.....	40
BAB 5 KESIMPULAN.....	42
DAFTAR ACUAN	43
LAMPIRAN	44

DAFTAR GAMBAR

Gambar 2.1 SIP message flow peer to peer	8
Gambar 2.2 SIP message flow dengan proxy.....	9
Gambar 2.3 Protokol SDP didalam body message SIP	13
Gambar 2.4 Arsitektur IMS.....	15
Gambar 2.5 IMS Core.....	16
Gambar 2.6 Breakout Gateway Control Function (BGFC)	18
Gambar 3.1 Konfigurasi proxy subversion	24
Gambar 3.2 Konfigurasi hosts.....	26
Gambar 3.3 Konfigurasi resolve.conf.....	26
Gambar 3.4 Konfigurasi BIND	27
Gambar 3.5 FHoSS webinterface	28
Gambar 3.6 Konfigurasi client	29
Gambar 3.7 User interface client.....	30
Gambar 4.1 Topologi jaringan	32
Gambar 4.2 SIP message flow pada proses registrasi	33
Gambar 4.3 SIP message flow pada proses call establishment.....	34
Gambar 4.4 Contact pada pesan 183 session progress	35
Gambar 4.5 Grafik paket vs delay pada bandwidth 96 kbps.....	37
Gambar 4.6 Grafik paket vs delay pada bandwidth 128 kbps.....	38
Gambar 4.7 Grafik paket vs delay pada bandwidth 256 kbps.....	39
Gambar 4.8 Grafik paket vs delay pada bandwidth 384 kbps.....	40
Gambar 4.9 Grafik paket vs delay pada bandwidth 512 kbps.....	41

DAFTAR TABEL

Tabel 2.1 ITU-T G1010 performance target for audio and video application	20
Tabel 4.1 Registration time	36
Tabel 4.2 RINGING time.....	36

BAB 1

PENDAHULUAN

1.1 LATAR BELAKANG

Perkembangan teknologi telekomunikasi menuju ke konvergensi jaringan dan layanan telekomunikasi menyebabkan teknologi telekomunikasi bermigrasi ke jaringan IP based network, hal tersebut tidak terlepas dari perkembangan dunia internet yang pada beberapa tahun terakhir ini telah menunjukkan hasil yang sangat luar biasa, banyak inovasi-inovasi baru yang bermunculan lewat internet. Adanya IP dan internet ini telah memunculkan berbagai bentuk bisnis model baru, seperti IP *telephony*, *search engine*, *online shop*, *online game*, dll Tidak hanya itu, dampak dari internet juga ialah cakupannya yang global, siapapun dibelahan dunia manapun asalkan mempunyai koneksi dan IP dapat terhubung dengan internet. Hal-hal tersebutlah yang membuat operator telekomunikasi ingin bermigrasi ke jaringan IP.

Tujuan utama pemanfaatan IP dalam bidang telekomunikasi ialah bagaimana menciptakan suatu keadaan dimana pelanggan suatu operator telekomunikasi dimanapun dia berada, asalkan dapat mengakses internet maka pelanggan tersebut dapat menikmati layanan-layanan operator telekomunikasi tersebut serta layanan-layanan multimedia yang ditawarkan oleh pihak ketiga atau bahkan dari operator telekomunikasi lain. Selain itu melihat trend perkembangan trafik telekomunikasi yang pada saat ini trafik untuk data sudah melampaui trafik suara, maka tidak heran migrasi ke jaringan yang berbasis IP menjadi pilihan bagi operator telekomunikasi. Namun untuk dapat menggunakan *all IP based network* operator telekomunikasi harus menghadapi tantangan seperti masalah keamanan karena celah keamanan pada internet pada umumnya masih banyak serta masalah QoS, yaitu menjamin atau menyediakan layanan yang lebih dari sekedar *best effort*. Untuk itu 3GPP mengeluarkan suatu framework baru yang dikenal dengan IP Multimedia Subsystem (IMS). IMS merupakan *session-control subsystem* atau sering juga dikenal sebagai *session delivery platform* (SDP) yang berbasiskan SIP, RTP dan protokol-protokol lainnya yang bertujuan menghadirkan atau

mengantarkan layanan multimedia yang berbasis IP. Dengan adanya IMS konvergensi antara dunia internet dan dunia telco dapat terjadi.

Layanan dasar yang harus dimiliki Operator telekomunikasi ialah layanan *voice*, *video*, dan *data*, oleh karena itu kemampuan IMS dalam menyelenggarakan layanan tersebut haruslah memiliki kualitas dan keamanan yang lebih baik ataupun menyamai *circuit switch network*. Layanan *video call* merupakan salah satu layanan multimedia yang memiliki *value added service* karena dengan *video call* user dapat saling melihat wajah lawan bicara sehingga menghadirkan pengalaman yang berbeda dengan layanan *voice* biasa.

1.2 TUJUAN

1. Membangun jaringan IMS
2. Mempelajari cara kerja jaringan IMS dengan cara melihat alur SIP message dalam mengatur sesi multimedia
3. Implementasi dan unjuk kerja dari kualitas layanan *video call*

1.3 PEMBATAAN MASALAH

Dalam skripsi ini masalah yang akan dibahas yaitu pembangunan jaringan IMS dengan menggunakan Open IMS Core, SIP message flow dalam membangun maupun memutuskan suatu session, serta analisis *quality of service* layanan video call dengan menggunakan jaringan IMS dengan melihat parameter-parameter seperti *delay* dan *packet loss*. Sedangkan untuk masalah seperti keamanan maupun *charging* tidak dibahas dalam skripsi ini.

1.4 METODOLOGI PENELITIAN

Metode penelitian yang digunakan dalam menyusun skripsi ini ialah sebagai berikut :

1. Studi Literatur
Melakukan studi pustaka dari berbagai tulisan baik buku, *ebook*, website, dokumentasi, *manual* maupun forum diskusi mengenai IMS.

2. Studi lapangan

Melakukan studi dengan cara diskusi maupun tanya jawab dengan pihak-pihak yang dapat menjadi sumber acuan yang berhubungan dengan topik skripsi ini.

3. Pembangunan Jaringan

Membangun jaringan IMS dengan menggunakan FOKUS Open IMS Core

4. Pengukuran

Pengukuran kualitas layanan *video call* pada jaringan IMS dengan melihat beberapa parameter QoS seperti *delay*, dan *packet loss*

1.5 SISTEMATIKA PENULISAN

Skripsi ini terdiri dari beberapa bagian yaitu :

BAB 1 PENDAHULUAN

Berisi mengenai gambaran umum mengenai pembahasan skripsi ini. Bagian ini mencakup latar belakang pengambilan tema, perumusan masalah, tujuan penelitian, batasan masalah, metodologi penelitian, dan sistematika penulisan penelitian.

BAB 2 LANDASAN TEORI

Berisi tentang teori dasar mengenai arsitektur IMS, protokol SIP, SDP, RTP, RTCP *video call*, parameter QoS

BAB 3 INSTALASI DAN KONFIGURASI JARINGAN IMS

Berisi mengenai pembahasan tentang Instalasi dan konfigurasi IMS core, client serta yang komponen lainnya yang dibutuhkan.

BAB 4 UJI COBA DAN ANALISIS

Berisi tentang pembahasan hasil pegujian jaringan IMS yang telah dibangun serta performansi layanan *video call*

BAB 5 KESIMPULAN

Berisi tentang kesimpulan mengenai jaringan IMS yang dibangun serta QoS layanan *video call*



BAB 2 LANDASAN TEORI

2.1 SESSION INITIATION PROTOCOL (SIP)

Session Initiation Protocol (SIP) merupakan protokol yang berfungsi untuk melakukan sambungan, memelihara, dan memutuskan multimedia *session*, melalui internet. SIP merupakan protokol yang dikembangkan oleh Internet Engineering Task Force (IETF), SIP tidak mendefinisikan tipe dari *session* yang akan dilakukan, melainkan hanya menginisiasinya saja. SIP merupakan text encoded based protokol seperti yang digunakan dalam protokol *Hyper Text Transport Protocol* (HTTP) pada *webbrowsing* dan *Simple Mail Transport Protocol* (SMTP) pada email. SIP dapat memenuhi empat kebutuhan dasar dalam komunikasi yaitu :

1. User Location Services
2. *Session* Establishment
3. *Session* Participant Management
4. Limited Feature Establishment

2.1.1 Komponen-komponen SIP

User Agent

User agent merupakan *end-user device*, seperti PC, *SIP phone*, dll. *User Agent* dapat dibedakan menjadi dua jenis yaitu *User Agent Client* dan *User Agent Server*. *User Agent Client* merupakan *User Agent* yang memulai *message (request)* *User Agent Server* merupakan *User Agent* yang menerima *request* tersebut dan membalasnya (*respon*).

Registrar Server

Registrar Sever merupakan database yang menyimpan semua lokasi dari *User Agent* yang berada pada domainnya.

Proxy Server

Proxy Server menerima request dari User Agent, kemudian melakukan query ke Registrar Server untuk mendapatkan informasi alamat dari tujuan, kemudian meneruskan request tersebut ke User Agent tujuan.

Redirect Server

Redirect Server meneruskan request ke eksternal domain. Redirect Server dan Registrar Server dapat berada di hardware yang sama dengan Proxy Server.

2.1.2 SIP Request /Method dan Response

SIP menggunakan request/ method dan respon dalam mendirikan call *session*. Request dan respons merupakan komunikasi antar user agent dengan user agent, user agent dengan proxy ataupun proxy dengan user agent untuk membuat suatu *session*.

Request/Method

Ada enam request dasar yaitu :

INVITE

INVITE berfungsi untuk melakukan *session* setup. Untuk dapat melakukan suatu *session*, user agent harus memberikan INVITE terlebih dahulu terhadap user agent tujuan melalui Proxy Server.

ACK

ACK merupakan jawaban final terhadap INVITE.

BYE

BYE berfungsi untuk menghentikan *session* yang sedang terjadi antar user agent.

CANCEL

CANCEL berfungsi untuk membatalkan perintah INVITE yang belum mendapatkan response dari user agent tujuan.

REGISTER

REGISTER berfungsi untuk melakukan registrasi URI user, method ini mendaftarkan URI user ke proxy server yang berada di domainnya.

OPTIONS

OPTIONS berfungsi untuk memberi informasi mengenai capabilities dari SIP *devices* pemanggil dan penerima.

Response

Request *message* akan ditanggapi dengan response *message*, response terdiri atas enam kelas yaitu

1XX merupakan informational response

2XX merupakan success response

3XX merupakan redirect response

4XX merupakan request failure

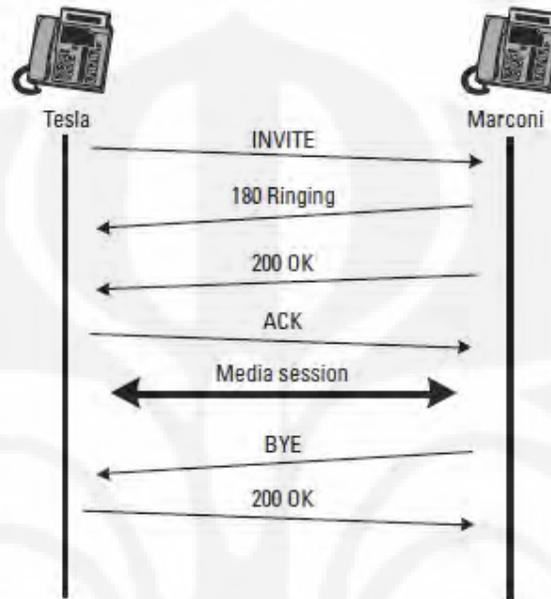
5XX merupakan server failure

6XX merupakan global failure

2.1.3 *Session establishment*

Berikut merupakan contoh atau alur suatu *session establishment* sederhana

Tanpa Proxy (Peer to Peer)



Gambar 2.1 SIP Message Flow peer to peer

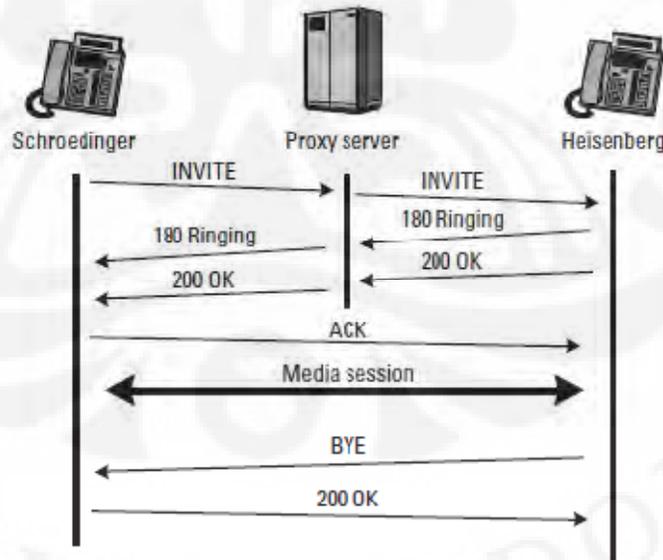
sumber : A.B.Johnston - SIP - Understanding The *Session* Initiation Protocol. 2004

Gambar diatas menunjukkan *SIP message flow* yang terjadi untuk melakukan suatu *session* antar dua user tanpa melalui proxy server atau dikenal dengan *peer to peer connection*, kedua user tersebut mengetahui ip address satu sama lainnya. Prinsip kerjanya menggunakan prinsip *client server*, user yang ditandai oleh anak panah pada flow diagram diatas bertindak sebagai *user agent server* dan sebaliknya sebagai *user agent client*. User Tesla pertama-tama mengirim *message INVITE* pada user Marconi, kemudian User Agent Marconi mengirim *ringing message* kepada Tesla sebagai respon terhadap *message INVITE* tersebut yang menandakan SIP *device* Marconi sedang di bel. Saat Marconi menjawab/mengangkat SIP *device* miliknya maka *Ok message* akan dikirim pada Tesla. *Message ACK* merupakan konfirmasi dari Tesla bahwa respon dari Marconi telah diterima. Setelah itu *media session* dapat dilaksanakan, *media session* tersebut biasanya menggunakan Real Time Protocol (RTP).

Message BYE menandakan Marconi mengakhiri media *session* yang sedang berlangsung, *message* Ok dikirim Tesla sebagai respon, *session* keduanya pun berakhir [1].

Melalui Proxy

Pada contoh sebelumnya kedua user mengetahui IP address satu sama lainnya, namun dalam kenyataan, mengetahui IP address user tujuan tidaklah mudah, karena IP address mungkin saja dapat berubah-ubah. Oleh karena itu diperlukan SIP URI dan Proxy server. Dengan adanya URI, user dapat mendaftarkan / meregistrasikan SIP URI dan IP address *device* yang sedang digunakannya ke Proxy server yang ada di domainnya. Proxy server tersebut akan menyimpan IP address *device* yang telah didaftarkan user tersebut dalam databasenya, sehingga ketika user lain ingin menghubungi user tersebut, maka ia cukup memanggil SIP URI user yang dituju, selanjutnya Proxy server yang akan mencari IP address SIP URI tersebut. Berikut contoh pembentukan *session* dengan melalui Proxy server.



Gambar 2.2 Sip *message* flow dengan proxy

sumber : A.B.Johnston - SIP - Understanding The *Session* Initiation Protocol. 2004

SIP URI Heisenberg adalah sip:werner.heisenberg@munich.de. Untuk menghubungi Heisenberg user agent Schroedinger melakukan query DNS domain minich.de yang merupakan IP address proxy domain tersebut. setelah itu mengirim INVITE *message* ke Proxy tersebut, proxy kemudian mencari IP address SIP URI tujuan dalam databasenyanya, kemudian memforward INVITE *message* tersebut. Menerima INVITE *message* tersebut user agent Heisenberg merespon dengan mengirim response *message* ringing, menandakan user agent *device* Heisenberg sedang bordering. Setelah Heisenberg mengangkat user agent *device* nya, maka response *message* ok akan dikirim ke Schrodinger melalui proxy. Kemudian komunikasi selanjutnya antara keduanya dapat dilakukan tanpa melalui proxy karena pada *message* ringing dan ok sebelumnya IP address Heisenberg telah dicantumkan dalam header contact [1].

2.1.4 Komponen-komponen SIP request *message*

Berikut merupakan contoh dari INVITE request *message* dari suatu user ke user lain, karena SIP merupakan text encoded based protokol maka SIP *message* akan terlihat seperti berikut :

```
INVITE sip:marconi@radio.org SIP/2.0
Via: SIP/2.0/UDP lab.high-
voltage.org:5060;branch=z9hG4bKfw19b
Max-Forwards: 70
To: G. Marconi <sip:Marconi@radio.org>
From: Nikola Tesla <sip:n.tesla@high-voltage.org>;tag=76341
Call-ID: 123456789@lab.high-voltage.org
CSeq: 1 INVITE
Subject: About That Power Outage...
Contact: <sip:n.tesla@lab.high-voltage.org>
Content-Type: application/sdp
Content-Length: 158
```

```
v=0
o=Tesla 2890844526 2890844526 IN IP4 lab.high-voltage.org
s=Phone Call
c=IN IP4 100.101.102.103
t=0 0
m=audio 49170 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

Header

Header field pertama pada contoh *message* diatas ialah

INVITE sip:marconi@radio.org SIP/2.0

Header tersebut berisi method yang digunakan yaitu INVITE, URI tujuan atau penerima serta versi SIP yang digunakan yaitu SIP 2.0. URI atau Unified Resource Indicator merupakan nama yang merepresentasikan alamat atau lokasi didalam internet.

Via

Setiap SIP *devices* yang memforward ataupun memulai SIP *message* akan mencantumkan address nya pada header Via, biasanya ditulis nama address yang dapat di resolve kedalam ip menggunakan DNS query. Dari contoh diatas header Via nya berisi

Via: SIP/2.0/UDP lab.high-voltage.org:5060;branch=z9hG4bKfw19b

SIP/2.0 berarti SIP yang digunakan merupakan versi 2.0, sedangkan transport protocol yang digunakan adalah UDP, 5060 adalah port yang digunakan dan branch adalah transaction identifier.

Max-Forwards

Header tersebut menunjukkan jumlah maksimum yang diizinkan untuk melakukan forward *message*, nilai max-forward akan di decrement kan oleh setiap SIP proxy server yang menerima ataupun memforward sip *message* tersebut.

To dan From

Header tersebut menunjukkan tujuan dan asal dari SIP *message* request tersebut.

Call-ID

Header ini berfungsi sebagai penanda yang digunakan untuk memeberikan sebuah identitas pada suatu SIP *session*.

SIP request *message* minimal harus memiliki header Via, To, From, Call-ID,dan CSeq

2.1.5 SIP Registration

Selain *call setup*, fungsi yang paling penting pada SIP yaitu registrasi. Registrasi merupakan proses mendaftarkan ip address *device* yang digunakan SIP URI user tersebut pada registrar server agar dapat dihubungi oleh user lain, dengan kata lain registrasi merupakan prosedur login ke sistem agar dapat online. Dalam proses registrasi user memberikan IP address *device* yang ia gunakan ke dalam SIP URI nya, IP address tersebut disebut dengan Contact URI. Sehingga sesama user untuk dapat saling menghubungi hanya butuh SIP URI dari user yang bersangkutan saja, tidak perlu mengetahui IP address nya. *Device* yang digunakan dapatlah berupa *IP Phone*, PC, dll. *Device* yang didaftarkan dapat saja lebih dari satu yang diatur berdasarkan prioritas seting, sebagai contoh user mendaftarkan *IP phone* dan PC nya, jika ada *incoming call* yang berupa voice maka akan disambungkan ke *IP phonenya*, sedangkan jika *incoming call* tersebut berupa *videoconference* maka akan dihubungkan dengan PC nya.

2.2 SESSION DESCRIPTION PROTOCOL (SDP)

Dalam melakukan *session establishment* SIP tidak mendeskripsikan sesi multimedia yang akan dibangun, fungsi tersebut dijalankan oleh protokol tersendiri yang disebut dengan *session description protocol* (SDP). SDP adalah protokol layer aplikasi yang berfungsi untuk memberikan deskripsi tentang suatu sesi multimedia[6], deskripsi tersebut antara lain mengenai :

- Nama dan tujuan *session*
- Waktu aktif *session* tersebut
- Tipe media yang akan digunakan
- *Address, port*, dll

Secara umum, protokol SDP digunakan pada saat melakukan *session announcement* serta *session invitation*. Informasi yang diberikan oleh sebuah pesan SDP antara lain adalah nama *session* dan tujuan penggunaan *session*, waktu aktif dari sebuah *session*, jenis media yang digunakan, format media, dan

informasi untuk menerima media tersebut. Berikut adalah isi dari pesan SDP, *field-field* yang bersifat *optional* diberi tanda *[7]

Session Description

v= (protocol version)
 o= (owner/creator and session identifier)
 s= (session name)
 i=* (session information)
 u* (URI of description)
 e* (email address)
 p* (phone number)
 c* (connection information)
 b* (bandwidth information)

One or more time description

z* (time zone adjustments)
 k* (encryption key)
 a* (zero or more session attributes lines)

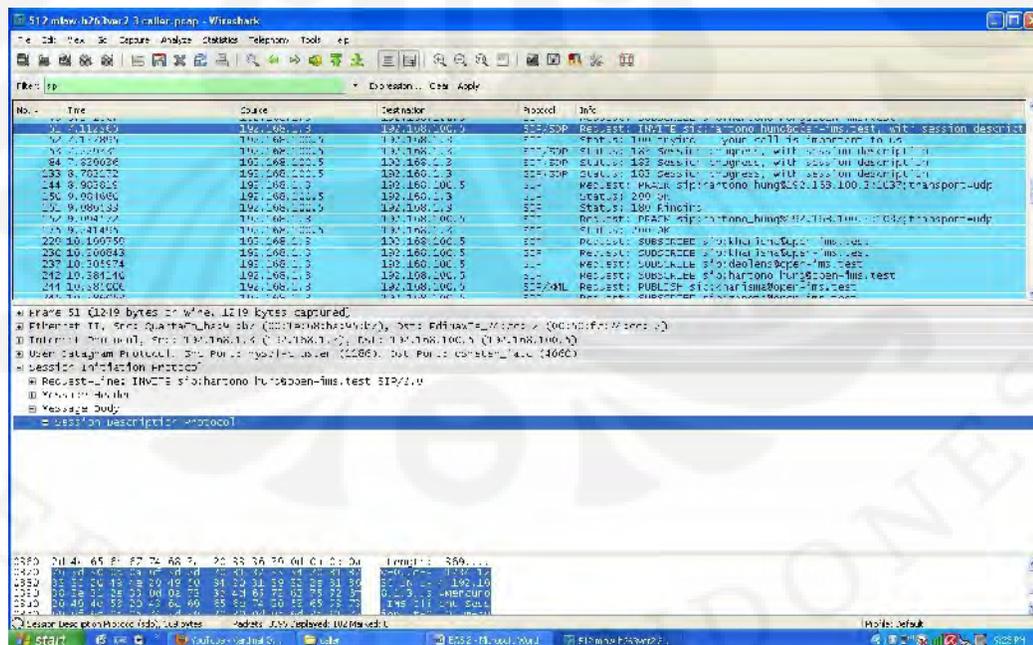
Time description

t* (time the session is active)
 r* (zero or more repeat times)

Media description

m= (media name and transport address)
 i* (media title)

Dalam penggunaannya SIP mencantumkan SDP pada bagian *message body*



Gambar 2.3 Protokol SDP didalam *body message SIP*

2.3 REAL-TIME TRANSPORT PROTOCOL (RTP)

RTP merupakan protokol yang didefinisikan oleh IETF untuk membawa *time-synchronized unicast* maupun *multicast* data melalui IP network. RTP protokol menyediakan *payload type identification* dan *sequencing*, namun feature yang paling pentingnya adalah *payload timestamping*. Packet switched network dapat mendelay data *stream* untuk beberapa waktu tertentu yang bervariasi. RTP didisain untuk membawa *timing information* bersama data, *marking traffic* dengan *timestamp* memungkinkan bagi stream data yang diterima yang timing nya terganggu akibat *delay variation* dapat kembali teratur timingnya seperti pada saat dikirimkan[6].

Selain *delay* yang tidak dapat diprediksi, paket pada IP network dapat saja hilang pada saat di jalur transmisi yang diakibatkan congesti ataupun *transmission error*. Untuk menghadapi situasi tersebut, *sequence* number diikuti sertakan dalam RTP. Nomor tersebut membolehkan penerima untuk mengkalkulasi seberapa banyak paket yang hilang dan menentukan paket yang mana yang hilang dalam aliran data. Tujuan penting lainnya dari RTP ialah *payload identification*. Ada banyak cara untuk mengirim multimedia stream melalui packet network. RTP menyediakan transport framework bersama. Tipe payload yang berbeda dapat di perlakukan secara berbeda di penerima. RTP memberitahu penerima tipe data yang sedang dikirim dalam payload dan bagaimana data tersebut di *encoded*.

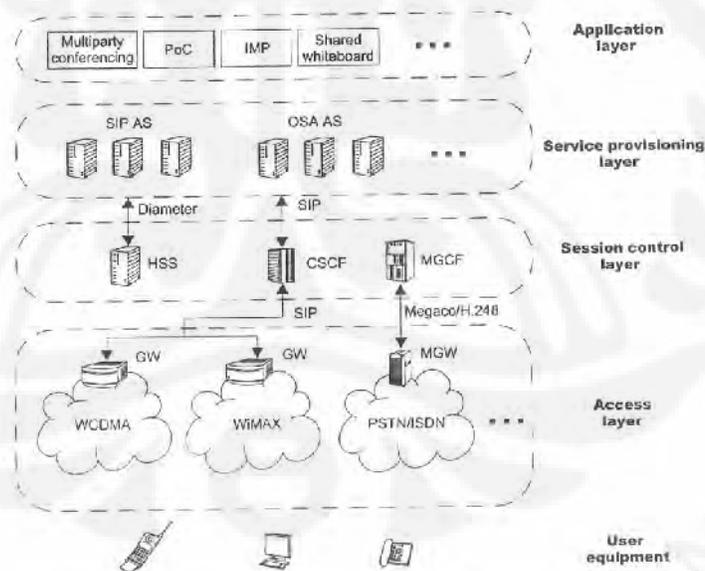
2.4 REAL-TIME TRANSPORT CONTROL PROTOCOL (RTCP)

RTCP merupakan protokol komplemen dari RTP yang mempunyai fungsi utama untuk memberikan *feedback* tentang kualitas transmisi. Pada implementasinya RTCP dimultiplex bersama dengan RTP dimana RTP menggunakan nomor port genap dan RTCP menggunakan nomor port ganjil pada saat dikirimkan di dalam UDP. Protokol ini memungkinkan partisipan pada suatu sesi RTP saling mengirimkan laporan kualitas. Fungsi laporan tersebut adalah mengetahui kualitas dari koneksi yang dibuat termasuk informasi seperti jumlah paket yang dikirim dan diterima, jumlah paket yang hilang dan jitter dari paket.

2.5 IP MULTIMEDIA SUBSYSTEM (IMS)

IMS merupakan sebuah arsitektur yang didefinisikan oleh 3rd Generation Partnership Project (3GPP) yang memungkinkan layanan *IP telephony* dan multimedia dapat diimplementasikan. IMS merupakan arsitektur yang *access independent* karena arsitektur IMS terpisah dari *access layer*, sehingga IMS dapat mendukung berbagai tipe akses yang berbeda seperti GSM, WCDMA, CDMA 2000, WLAN, *Wireline broadband*, dsb. Dengan adanya IMS operator telekomunikasi dapat mengambil manfaat yang ada pada internet serta menghadirkan banyak layanan baru dan juga memberikan layanan *voice* menggunakan jaringan *packet switch*, sehingga memungkinkan konvergensi antara *voice* dan *data*. [5]. IMS menggunakan standard IETF sebagai protokol signalingnya seperti *Session Initiation Protocol (SIP)*, *Session Description Protocol (SDP)*, maupun *Real Time Transport Protocol (RTP)*.

2.5.1 Arsitektur IMS



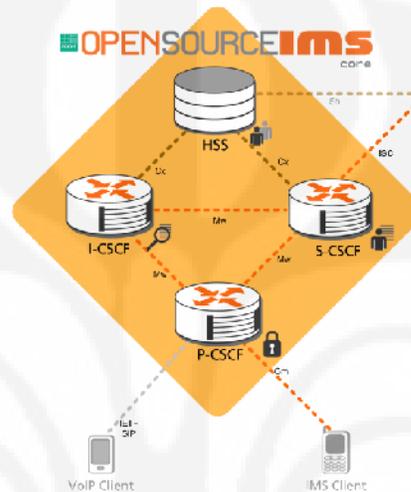
Gambar 2.4 Arsitektur IMS

sumber : Francisco J, Hens., Jose M, Caballero. (2008). Triple Play Building the converged network for IP, VoIP and IPTV. Wiley

Elemen – elemen dari arsitektur IMS menurut 3GPP ialah seperti berikut :

IMS Core

IMS Core merupakan elemen – elemen dasar yang berupa sekumpulan SIP dan database server yang menjalankan fungsinya masing-masing.



Gambar 2.5 IMS Core

sumber : <http://www.openimscore.org/>

Fungsi dari masing-masing elemen yang terdapat pada IMS Core ialah sebagai berikut :

Home Subscriber Server (HSS)

HSS merupakan database yang menyimpan data user yang berada dalam suatu domain tertentu, data tersebut seperti informasi profil dan lokasi user, data AAA, data S-CSCF yang dialokasikan terhadap user tersebut, dll. Jaringan yang memiliki lebih dari satu HSS memerlukan suatu entitas tambahan yang disebut dengan Subscription Location Function (SLF) yang berfungsi untuk mencari HSS mana yang memegang profil dari suatu user tertentu[4].

Call Session Control Function (CSCF)

Call Session Control Function (CSCF) merupakan sekelompok SIP Server yang memiliki fungsi masing-masing, CSCF dapat dibagi menjadi tiga bagian yaitu :

Proxy-CSCF

P-CSCF merupakan titik masuk menuju domain IMS. P-CSCF merupakan server yang peran utamanya adalah untuk meneruskan pesan SIP kepada P-CSCF atau S-CSCF yang lebih "dekat" dengan tujuan. Jika *subscriber* berada dalam keadaan *roaming* (di luar server asalnya) maka perangkat Bergeraknya akan melakukan akses ke IMS melalui P-CSCF terdekat dengan jaringan yang dikunjungi dan jaringan home akan melakukan fungsi IMS lainnya[2].

Interrogating-CSCF

Fungsi I-CSCF ialah menentukan S-CSCF mana yang akan menangani *subscriber*. I-CSCF mengumpulkan informasi seperti lokasi, kapabilitas, dan availabilitas S-CSCF untuk menentukan S-CSCF yang cocok bagi *subscriber*, setelah S-CSCF yang sesuai telah terpilih, I-CSCF akan menginformasikan S-CSCF yang harus dipilih oleh pengguna berdasarkan data di HSS[2].

Serving-CSCF

S-CSCF merupakan node pusat signaling IMS, fungsi utama S-CSCF adalah untuk menyediakan pelanggan IMS kepada akses menyeluruh terhadap layanan yang disediakan[2]. Selain itu S-CSCF berfungsi untuk melakukan SIP routing, sebagai contoh, ketika pelanggan melakukan panggilan berupa nomor telepon, maka S-CSCF akan melakukan translasi nomor tersebut menjadi SIP URI (Uniform Resource Identifier) melalui mekanisme ENUM. Fungsi lain dari S-CSCF ialah menerapkan kebijakan dari network operator, sehingga mencegah user untuk melakukan sesi atau panggilan yang tidak diizinkan [4].

Application Server (AS)

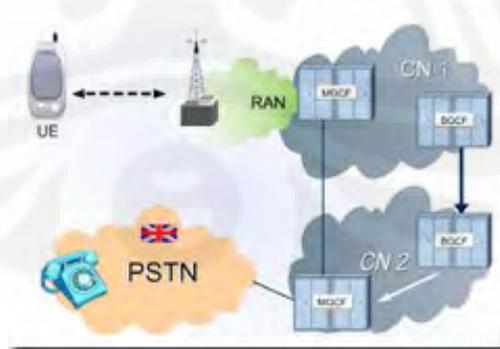
Application server merupakan suatu entitas SIP yang menjadi *host* dan menjalankan suatu layanan atau konten tertentu misalnya *presence* dan IPTV server. AS berinteraksi dengan S-CSCF serta I-CSCF menggunakan SIP, dan menggunakan protokol Diameter untuk berinteraksi dengan HSS[4].

Media Resource Function (MRF)

MRC menjalankan fungsi-fungsi yang berkaitan dengan media, seperti manipulasi media, memainkan nada, dll. MRC merupakan node yang tersebar. MRC terdiri dari MRF Controler (MRFC) dan MRF Processor (MRFP) [4].

Breakout Gateway Control Function (BGFC)

BGFC digunakan ketika sebuah panggilan atau *media session* membutuhkan koneksi ke jaringan PSTN atau *circuit switched*. BGFC akan menentukan dimana putusnya jalur *signaling (breakout)* menuju jaringan *circuit switched*. Jika *breakout* terjadi di jaringan yang sama dengan jaringan dimana BGFC berada, maka BGFC akan meneruskan SIP *signaling* ke MGCF yang terdapat pada jaringan tersebut untuk kemudian diubah menjadi ISUP/BICC *signaling* menuju ke jaringan *circuit switched*. Jika *breakout* terjadi di jaringan lain, maka BGFC akan meneruskan SIP *signaling* menuju BGFC di jaringan lain melalui I-CSCF [2].



Gambar 2.6 Breakout Gateway Control Function (BGFC)

sumber : <http://uchaan.com/ims/mgcf/>

2.5.2 Policy and Charging Control

IMS dapat mengontrol media plane melalui signaling plane menggunakan arsitektur yang disebut *Policy and Charging Control (PCC)*. Arsitektur PCC terdiri dari beberapa fungsi yaitu *Policy and Charging Rules Function (PCRF)*, *Policy and Charging Enforcement Function (PCEF)*, *Subscription Profile Repository (SPR)*, *Online Charging System (OCS)*, dan *Offline Charging System (OFCS)*. PCRF mengambil *subscription information* dari SPR dan menggunakan informasi tersebut untuk mengatur QoS pada *access gateway* menggunakan PCEF, jika menggunakan akses selular 3GPP, maka *access gateway* nya adalah GGSN [4]

2.6 VIDEO CALL

Video call atau dikenal juga dengan *video telephony* maupun *videophone* merupakan sebuah bentuk komunikasi *multimedia* antara dua user di tempat yang berbeda secara realtime dimana kedua user tersebut dihubungkan oleh suatu jaringan. Video call merupakan salah satu bentuk layanan multimedia, menurut 3rd Generation Partnership Project (3GPP) *multimedia service* merupakan layanan yang menangani beberapa jenis media, yang beberapa diantaranya harus disinkronisasikan [3]. Dalam video call jenis media yang digunakan adalah video dan audio.

Untuk menjamin kualitas layanan *video call* sehingga layak bagi user, parameter-parameter QoS haruslah dijaga sesuai dengan standard. Dalam aplikasi *video call*, video dan audio *streams* bekerja dengan mode *full duplex*, besarnya *delay* sama dengan delay pada aplikasi VoIP, namun sinkronisasi harus dilakukan antara audio dan video stream tersebut agar terjadi *lip-synch*. Tabel 2.1 menunjukkan rekomendasi dari 3GPP mengenai ekspektasi unjuk kerja oleh *end user* terhadap *realtime / conversational service*

Tabel 2.1 ITU-T G1010 Performance target for audio and video application

Medium	Application	Degree of symmetry	Data rate	Key performance parameters and target values		
				End to end One-way Delay	Delay Variation within a call	Information loss
Audio	Conversational voice	Two-way	4-25 kb/s	<150 msec preferred <400 msec limit Note 1	< 1 msec	< 3% FFR
Video	Vidconferencing	Two-way	32-384 kb/s	< 150 msec preferred <400 msec limit lip-synch < 100 msec		< 1% FER
Data	Telemetry - two-way control	Two-way	<28.8 kb/s	< 250 msec	N.A	Zero
Data	realtime games	Two-way	< 60 kb/s Note 2	< 75 msec preferred	N.A	< 3% FER preferred, < 5% FFR limit Note 2
Data	Telnet	Two-way (asymmetric)	< 1 KB	< 250 msec	N.A	Zero

2.7 PARAMETER QUALITY OF SERVICE (QoS)

Parameter QoS yang menentukan kualitas keseluruhan dari aplikasi video call yaitu sebagai berikut :

2.7.1 Delay

End to end delay disebut juga sebagai waktu tunda merupakan waktu total yang diperlukan sebuah paket mulai dari ketika di proses di sisi pengirim sampai akhirnya diterima dan dapat didengarkan di sisi penerima. *End to end delay* ini merupakan agregasi dari *delay-delay* yang terjadi dari berbagai proses yang terjadi sepanjang perjalanan dari terminal user satu ke user lainnya. Proses-proses yang memberikan kontribusi terhadap *end to end delay* dari *transmitter* sampai ke *receiver* ialah sebagai berikut

- Coding / Decoding delay
Codec yang canggih dapat mengkompresi data suara sampai sekecil-kecilnya namun harus membayarnya dengan delay yang besar
- Packet assembly delay

Kebanyakan voice encoder menghasilkan frame yang sangat pendek. Encoder G.711 menghasilkan 8 bit frame setiap 125 μ s sedangkan G.729 menghasilkan 80 bit frame setiap 10 ms. Untuk mengefisiensikan transmisi maka setiap IP diisi dengan frame sebanyak banyaknya, agar rasio overhead vs payload bytenya semakin besar, namun hal itu menyebabkan delay yang terjadi semakin besar

- **Serialization delay**
Serialization delay merupakan waktu yang dibutuhkan untuk menempatkan satu paket pada transmission line. Sebagai contoh, dibutuhkan 12.5 ms bagi 100 byte frame untuk dapat dikirimkan dengan menggunakan 64 kbit/s line
- **Queuing delay**
Queuing delay terjadi ketika jaringan mengalami kongesti. Dalam jaringan terdapat beberapa delay lain seperti switching/routing delay atau propagation delay, namun efeknya tidak terlalu signifikan jika dibandingkan queuing delay.
- **Dejittering delay**
Delay ini terjadi pada buffer, paket-paket yang sampai disimpan terlebih dahulu di dalam buffer sebelum dimainkan

2.7.2 Jitter

Merupakan variasi waktu tunda atau variasi *delay* yang terjadi akibat adanya selisih waktu atau interval antar-kedatangan paket di penerima. 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.

2.7.3 Packet Loss

Merupakan hilangnya paket data yang sedang dikirimkan. Hilangnya data ini bisa disebabkan karena adanya permasalahan di perangkat-perangkat jaringan seperti *router* yang terlalu sibuk dan jalur komunikasi yang terlalu padat yaitu

ketika terjadi *peak load* dan *congestion* (kemacetan transmisi akibat padatnya *traffic* yang harus dilayani) dalam batas waktu tertentu, sehingga *frame* (terdiri dari *header* dan *payload*) suara yang ditransmisikan akan dibuang.



BAB 3

INSTALASI DAN KONFIGURASI JARINGAN IMS

Jaringan IMS pada penelitian ini merupakan testbed yang berbasis OpenIMSCore yang dikembangkan oleh FOKUS Franhoufer. Pada jaringan ini digunakan sebuah PC dengan *operating sistem* berbasis Linux dengan menggunakan distro Ubuntu 9.10 Karmic Koala yang difungsikan sebagai Core IMS. Core tersebut terdiri dari PCSCF, ICSCF, SCSCF, serta HSS. Selain *core*, diperlukan *client* yang berfungsi sebagai *user agent* (UE). *Client* yang digunakan adalah Mercurio bronze, dipilih karena client tersebut merupakan *native* IMS client. Client tersebut diinstal pada computer dengan OS *Windows*. Sedangkan untuk mensimulasikan bandwidth jaringan digunakan dua buah *router* Vyatta.

3.1 KONFIGURASI OPENIMSCORE

3.1.1 Download dan Compile Source Code

OpenIMSCore merupakan testbed yang dikembangkan oleh FOKUS Franhoufer yang bersifat *open source*, oleh karena itu pada saat skripsi ini ditulis OpenIMSCore hanya bisa dijalankan pada *operating system* linux. Untuk mendownload *source code* nya, dibutuhkan program subversion untuk memudahkan kita dalam mengambil *source code* tersebut dari servernya. Untuk mengistalnya dapat menggunakan synaptic package manager ataupun menggunakan command line interface (CLI) pada terminal dengan memasukan perintah :

```
~$ sudo apt-get install subversion
```

Kemudian download source code OpenIMSCore dengan menggunakan subversion dengan alamat :

http://svn.berlios.de/svnroot/repos/openimscore/ser_ims/trunk

<http://svn.berlios.de/svnroot/repos/openimscore/FHoSS/trunk>

Kedua folder file tersebut disimpan dalam direktori `/opt/OpenIMSCore/`. Jika koneksi internet menggunakan *proxy*, maka sebelumnya atur terlebih dahulu setingan *proxy* beserta *port* nya pada `/etc/subversion/servers`



```

GNU nano 2.8.3 File: /etc/subversion/servers
## ssl-authority-files is a semicolon-delimited list of files,
## each pointing to a PEM-encoded Certificate Authority (CA)
## SSL certificate. See details above for overriding security
## due to SSL.
[global]
# http proxy exceptions = *.exception.com, www.internal.salt.org
http.proxy-host = 152.118.24.10
http.proxy-port = 8080
# http-proxy-username = defaultusername
# http-proxy-password = defaultpassword
# http-compression = no
# http-auth-types = basic;digest;negotiate
# No http-timed-out, so just use the builtin default.
# No non-debug-mask, so non-debugging is disabled.
# ssl-authority-files = /path/to/CACert.pem;/path/to/CAcert2.pem
# Password / passphrase caching parameters:
# store-passwords = no
# store-plaintext-passwords = no
# store-ssl-client-cert-pg = no
# store-ssl-client-cert-pg-plaintext = no
  
```

Gambar 3.1 Konfigurasi proxy subversion

Untuk mencompile serta menjalankan OpenIMSCore diperlukan beberapa *package / software* pendukung yaitu sebagai berikut:

- sun-java6-jdk
- mysql-server
- libmysqlclient15-dev
- libxml2 libxml2-dev
- bind9
- ant
- flex
- bison

Setelah semua kebutuhan terpenuhi, *compile source code* yang berada pada folder `ser_ims` serta `FHoSS/deploy`

```
/opt/OpenIMSCore/ser_ims$ sudo make install-libs all
```

```
/opt/OpenIMSCore/FHoSS$ ant compile deploy
```

3.1.2 Konfigurasi Database

Untuk membangun database yang diperlukan masukan perintah berikut

```
~$ mysql -uroot -p < /opt/OpenIMSCore/ser_ims/cfg/icscf.sql
```

```
~$ mysql -uroot -p < /opt/OpenIMSCore/FHoSS/scripts/hss_db.sql
```

```
~$ mysql -uroot -p < /opt/OpenIMSCore/FHoSS/scripts/userdata.sql
```

3.1.3 Konfigurasi DNS

Copy file open-ims.dnszone yang berada pada direktori /opt/OpenIMSCore/ser-ims/cfg ke folder /etc/bind dengan menggunakan perintah berikut

```
~$ sudo cp /opt/OpenIMSCore/ser_ims/cfg/open-ims.dnszone /etc/bind/
```

Kemudian edit file /etc/hosts serta /etc/resolve.conf sesuai dengan alamat ip yang dikehendaki, Untuk konfigurasi awal, alamat IP yang digunakan ialah IP loopback 127.0.0.1 localhosts dan domainnya adalah open-ims.test



```

rheaverze@ubuntu: ~
File Edit View Terminal Help
GNU nano 2.0.9 File: /etc/hosts Modified

127.0.0.1 localhost
127.0.1.1 ubuntu
127.0.0.1 open-ims.test mobiterms.open-ims.test ue.open-ims.test presence.open-ims.test
      icscf.open-ims.test scscf.open-ims.test pcscf.open-ims.test hss.open-ims.test

# The following lines are desirable for IPv6 capable hosts
::1 localhost ip6-localhost ip6-loopback
fe80::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts

^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell

```

Gambar 3.2 Konfigurasi hosts

File tersebut berfungsi untuk memberikan host – host yang tercantum pada file tersebut dengan suatu alamat IP



```

rheaverze@ubuntu: ~
File Edit View Terminal Help
GNU nano 2.0.9 File: /etc/resolv.conf Modified

# Generated by NetworkManager
domain open-ims.test
search open-ims.test
nameserver 127.0.0.1

^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell

```

Gambar 3.3 Konfigurasi resolve.conf

Kemudian edit file `/etc/bind/named.conf.local` sehingga menjadi seperti yang tampak dalam gambar berikut



```

rheaverze@ubuntu: ~
File Edit View Terminal Help
GNU nano 2.0.9 File: /etc/bind/named.conf.local

//
// Do any local configuration here
//

zone "open-ims.test" {
type master;
file "/etc/bind/open-ims.dnszone";
};

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";

[ Read 13 lines ]
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell

```

Gambar 3.4 Konfigurasi BIND

Setelah semua konfigurasi berhasil, maka core siap diaktifkan, jalankan file `pcscf.sh`, `icscf.sh`, `scscf.sh` Serta file `./startup.sh` yang terletak dalam folder `/opt/OpenIMSCore/deploy`, keempatnya dijalankan secara bersamaan pada terminal masing-masing. HSS dapat diakses melalui webbrowser dengan alamat `http://localhost:8080` dengan menggunakan

username: hssAdmin

password: hss

Setelah dapat mengakses HSS kita dapat melakukan *provisioning* terhadap user, seperti menambah maupun menghapus user yang ada, charging, konfigurasi *application server*, dll. Ada dua user yang terdaftar pada HSS secara default yaitu Alice (`sip:alice@open-ims.test`) serta Bob (`sip:bob@open-ims.test`). Untuk menambahkan user baru gunakan pilihan *create* pada *IMS Subscription*



Gambar 3.5 FHoSS webinterface

Untuk mengganti alamat IP dan nama domain sesuai dengan yang kita inginkan, gunakan *script* configurator.sh

```
/opt/OpenIMSCore$ ./configurator.sh pcscf.cfg icscf.cfg icscf.xml scscf.cfg scscf.xml
ser_ims/cfg/icscf.sql FHoSS/deploy/DiameterPeerHSS.xml
FHoSS/deploy/hss.properties FHoSS/scripts/hss_db.sql FHoSS/scripts/userdata.sql
```

3.2 KONFIGURASI CLIENT

Client yang digunakan dalam penelitian ini adalah Mercuro bronze, mercuro dipilih karena merupakan *client native* IMS dan sudah mendukung video, selain itu banyak fasilitas lain yang sudah didukung *client* ini seperti fasilitas *presence*, *instant messaging*, dll. *Client* ini berbentuk softphone dan dapat diinstall di *windows*.



Gambar 3.6 Konfigurasi Client

Agar dapat berkomunikasi dan melakukan registrasi maka client tersebut harus dikonfigurasi terlebih dahulu, konfigurasi dapat dilakukan pada *options* yang terdapat pada *tools*. Pada tab *identities* terdapat *field* mengenai public user identity, privat user identity, password, realm. Agar dapat terautentikasi isian-isian tersebut harus cocok dengan yang terdapat pada HSS. Sedangkan Proxy CSCF merupakan alamat PCSCF IMS yang dituju, oleh karena itu diisi dengan alamat IP 192.168.100.5:4060. 4060 merupakan port *default* dari PCSCF. Tab codec mengatur codec apa saja yang dapat digunakan.



Gambar 3.7 user interface client

3.3 KONFIGURASI ROUTER

Router yang digunakan adalah router Vyatta, Vyatta merupakan *software-based router* yang open source dan berbasis Linux. *Installer package* nya dapat diinstal di <http://www.vyatta.com>. Untuk menginstallnya boot komputer menggunakan CD *installer* Vyatta lalu masukan perintah

```
~$ install-system
```

Setelah terinstall, lalu dilakukan konfigurasi *IP address* masing–masing *interface* nya. Selain berfungsi untuk melakukan fungsi *routing* serta pembagian *network*, router tersebut digunakan untuk mengatur besarnya bandwidth jaringan antara *core* dengan jaringan *user* menggunakan *policy control*, dengan menggunakan

policy control besarnya bitrate yang keluar meninggalkan router dapat diatur. Besarnya bitrate divariasikan untuk mensimulasikan besarnya bitrate yang digunakan ISP.

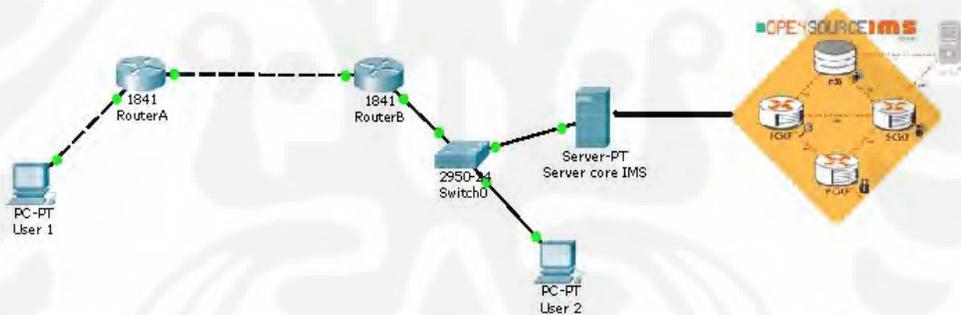


BAB 4

PENGUKURAN DAN ANALISIS

4.1 SKENARIO PENGUJIAN

Ujicoba konsep kerja IMS serta unjuk kerja layanan video call menggunakan jaringan IMS ini dilakukan antara dua user dengan menggunakan softphone *Mercuro bronze edition* untuk saling berkomunikasi. User 1 (sip:kharisma@open-ims.test) berada pada jaringan 192.168.1.0/24 dengan alamat IP 192.168.1.3 sedangkan User 2 (sip:hartono_hung@open-ims.test) dengan alamat IP 192.168.100.3 berada satu jaringan dengan server *core* IMS yaitu 192.168.100.0/24, sedangkan server *core* IMS menempati alamat IP 192.18.100.5. Untuk mensimulasikan besarnya bandwidth jaringan maka digunakan dua buah router yang memisahkan jaringan user 1 dengan user 2 menggunakan jaringan 200.200.200.0/24.



Gambar 4.1 Topologi jaringan

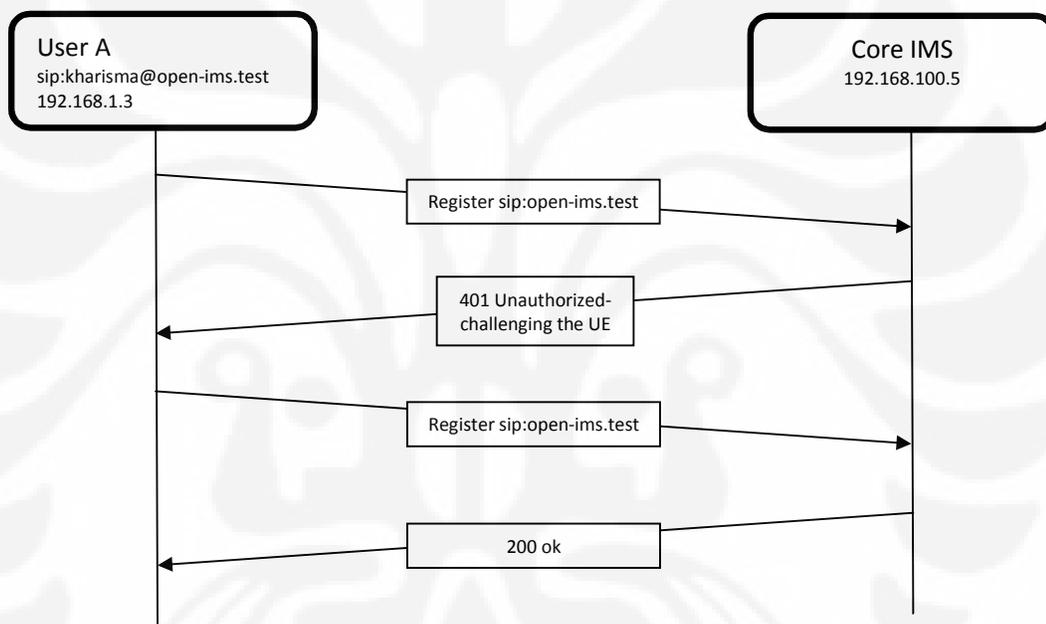
Besarnya bandwidth jaringan divariasikan mulai dari 96 kbps, hingga 512 kbps, *codec* audio yang digunakan adalah G.711 mLaw sedangkan *codec* video dibandingkan antara H.263 dan H.263+. Parameter performansi yang diukur adalah *delay*, dan *packet loss*. Selain parameter QoS, SIP *message flow* dalam melakukan registrasi maupun *call establishment* akan dilihat lalu kemudian dianalisa.

4.2 SIP MESSAGE FLOW

Pengujian ini bertujuan untuk melihat *SIP message flow* dalam melakukan registrasi serta *call establishment*.

4.2.1 Registrasi

User A (sip:kharisma@open-ims.test) dengan IP address 192.168.1.3 melakukan proses registrasi ke core IMS dengan IP address 192.168.100.5. Paket-paket SIP yang masuk dan keluar dilihat melalui *software* wireshark di komputer user A.



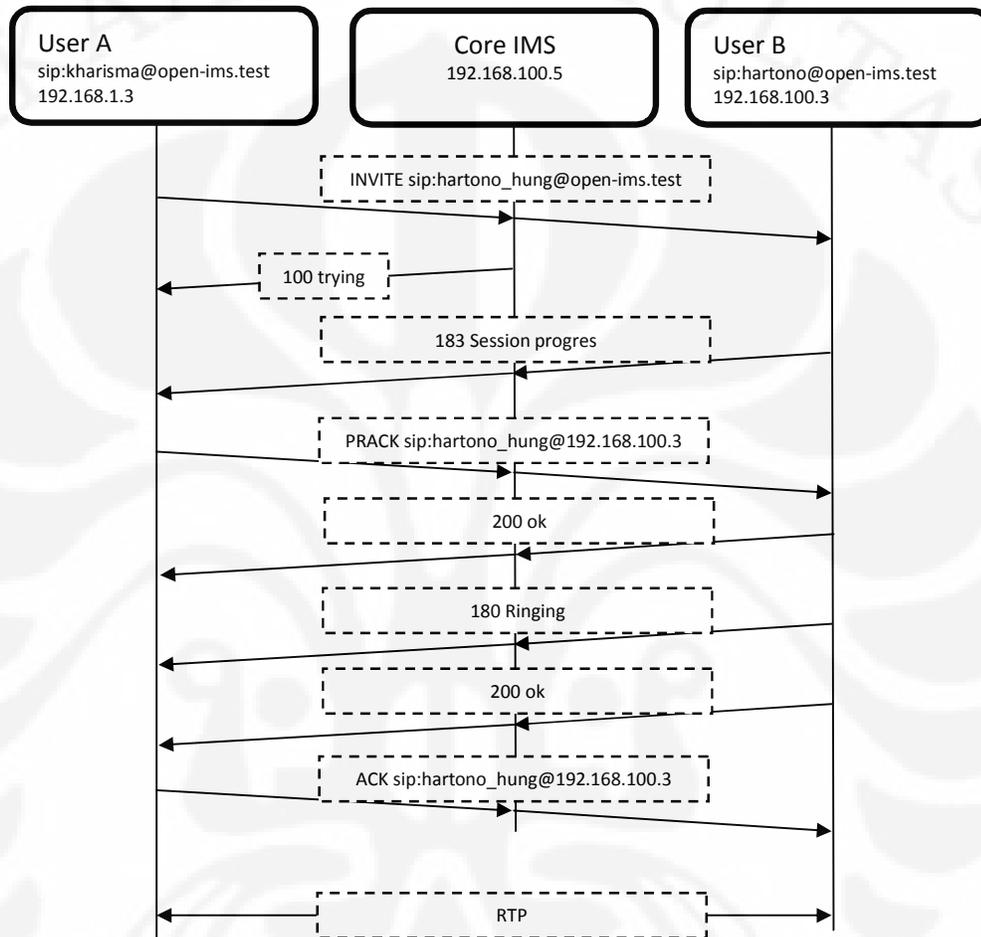
Gambar 4.2 *SIP message flow* pada proses registrasi

Dari hasil yang didapat terlihat bahwa proses registrasi membutuhkan dua kali *register message*. Hal tersebut karena IMS dalam penerapannya menggunakan sistem keamanan *HTTP Digest* dan *3GPP AKA* secara bersamaan untuk mengatasi kelemahan autentikasi menggunakan SIP yang kurang *secure*.

4.2.2 Session Establishment

User A (sip:kharisma@open-ims.test) dengan IP address 192.168.1.3 menghubungi User B (sip:hartono@open-ims.test) dengan IP ip address

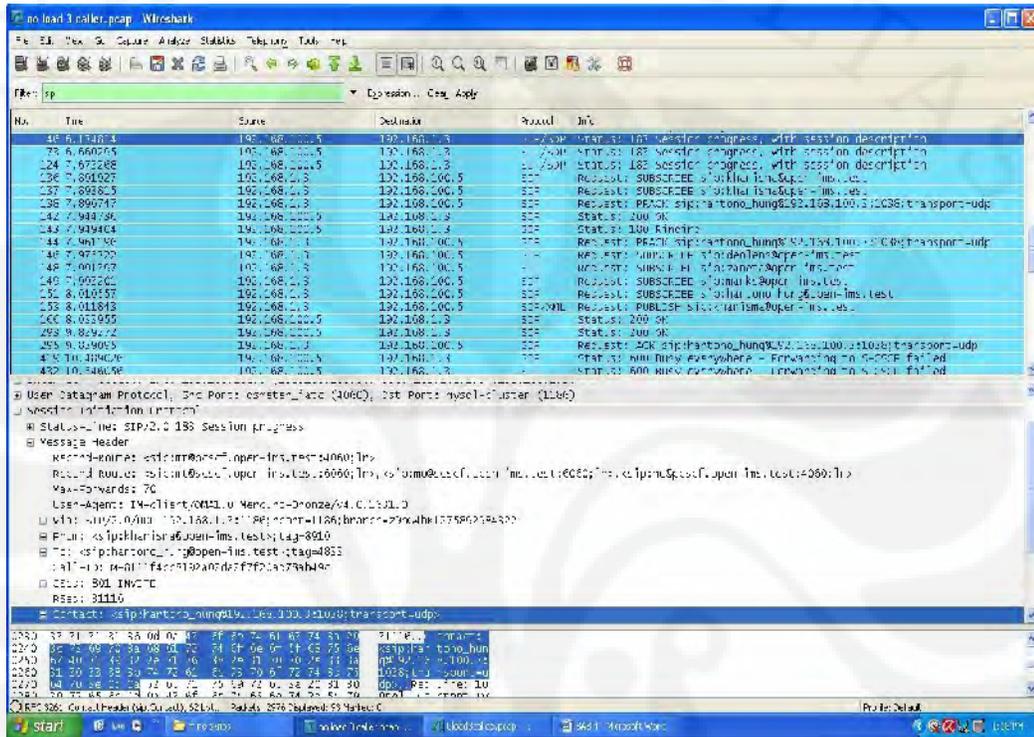
192.168.100.3. kedua user tersebut sebelumnya telah melakukan registrasi terlebih dahulu. Paket-paket udp yang mengandung SIP *message* dilihat dengan menggunakan wireshark baik pada user A maupun user B



Gambar 4.3 SIP *message flow* pada proses *call establishment*

Proses *call establishment* antara user A dengan user B berlangsung dengan alur seperti yang terlihat dari gambar 4.3, user A melakukan inisiasi dengan memberikan pesan INVITE terhadap user B melalui core, dalam pesan INVITE tersebut terdapat SDP yang berisi mengenai nama *client*, codec yang digunakan, maupun *connection information*. Jika session dapat berlangsung maka pesan 180 Ringing akan memberitahu *client* user B jika ada panggilan masuk. Setelah user B mengangkat *client* nya maka sesi RTP yang membawa paket gambar dan suara

dapat berlangsung. Pertukaran paket RTP ini langsung dilakukan antara user A dengan user B tanpa melewati *core*, hal tersebut karena user A telah mengetahui IP address user B yang terdapat pada pesan SIP 183 Session progress sebelumnya, yang tercantum pada *field contact*



Gambar 4.4 *contact* pada pesan 183 session progress

4.3 RESPONSIVITAS CORE IMS

Dalam pengujian ini akan dilihat waktu yang diperlukan oleh IMS *core* dalam melayani rekues dari *subscriber*. Yang akan diukur adalah *registration time*, *initial ringing time*, serta *call setup*.

4.3.1 Registration Time

Pada pengujian ini akan dilihat waktu yang dibutuhkan untuk melakukan registrasi. Perhitungan ini dilihat dari selisih waktu antara pesan 200 ok dengan pesan INVITE melalui paket-paket UDP yang ditangkap oleh Wireshark.

Tabel 4.1 *Registration time*

no	Registration time (s)
1	0.5176
2	0.347
3	0.477
4	0.399
5	0.346
6	0.347
7	0.399
8	0.345
9	0.241
10	0.335
11	0.236
12	0.444
13	0.466
14	0.346
15	0.268
rata-rata	0.367573

Dari hasil yang ditunjukkan table 1 waktu rata-rata yang dibutuhkan untuk melakukan registrasi adalah sebesar 0.368 s.

4.3.2 Ringing Time

Ringing time adalah selisih waktu antara pesan 180 Ringing dengan pesan INVITE. Untuk mendapatkannya dapat dihitung melalui *timestamp* yang terdapat pada setiap paket yang ditangkap wireshark. User A akan menginisiasi panggilan terhadap user B kemudian paket UDP yang mengandung SIP antara user A dengan user B akan dilihat pada sisi user A.

Tabel 4.2 Ringing Time

No	ringing time (s)
1	1.956756
2	1.912614
3	1.883654
rata-rata	1.92

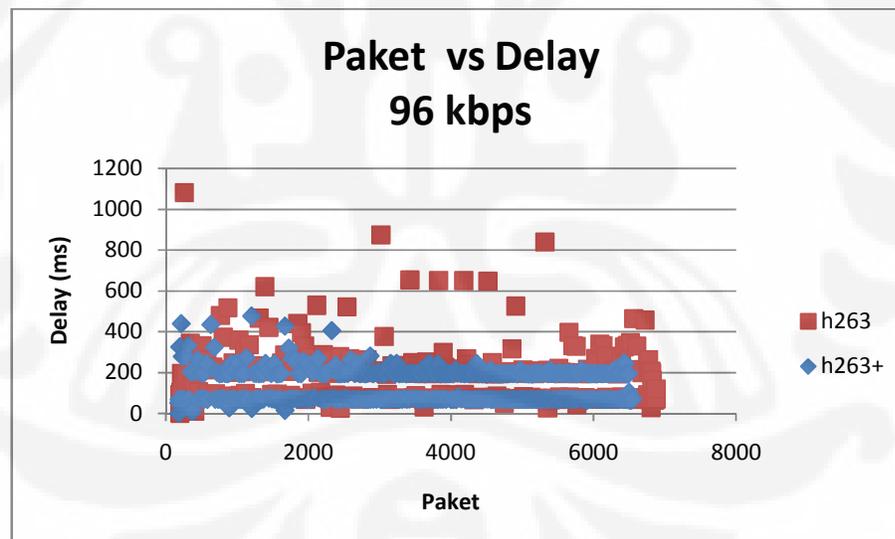
Dari hasil tiga kali percobaan waktu rata-rata *ringing time* adalah sebesar 1.92 s

4.4 PERFORMANSI LAYANAN VIDEO CALL

Pengujian ini bertujuan untuk melihat kualitas dari layanan *video call* yang bekerja dengan menggunakan jaringan IMS. Bandwidth jaringan divariasikan mulai dari 96 kbps, 128 kbps, 256 kbps, 384 kbps, 512 kbps, *codec* video yang digunakan ialah H.263 dan H.263+. Pengujian dilakukan dengan cara memonitor RTP *stream* dari user A dan user B.

4.4.1 Jaringan Dengan Bandwidth 96 kbps

Pada pengujian ini bandwidth jaringan antara user A dan user B diatur sebesar 96 kbps. *Delay* rata-rata yang didapat dengan menggunakan *codec* H.263 adalah sebesar sebesar 173.54 ms dengan *delay* maksimum mencapai 1081.32 ms, sedangkan besarnya *packet loss* mencapai 79%. Untuk *codec* H.263+ *delay* rata-rata nya sebesar 133.14 ms dengan *delay* maksimum sebesar 439.32 ms, dengan *packet loss* sebesar 71%. Perbandingan grafik *packet* dengan *delay* keduanya dapat dilihat pada gambar 4.5



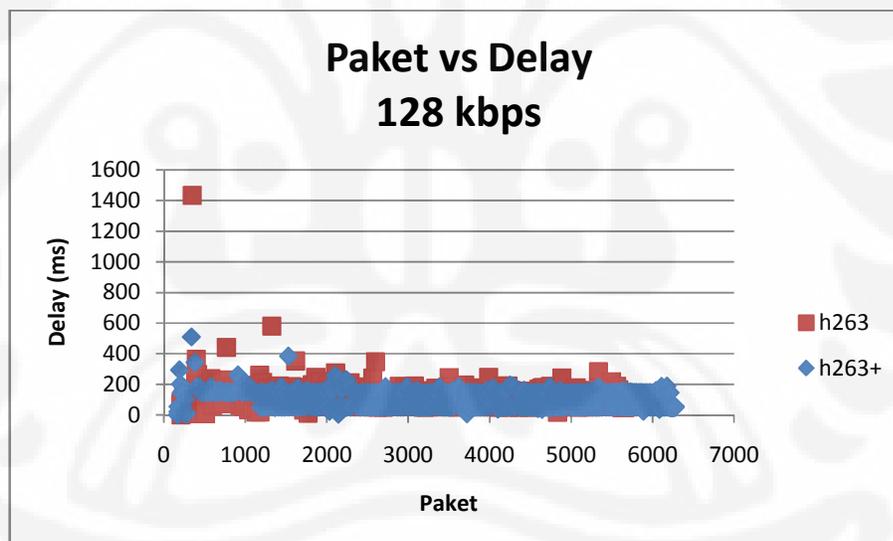
Gambar 4.5 grafik Paket vs *Delay* pada bandwidth 96 kbps

Dari pengukuran *delay* dan *packet loss* pada jaringan 96 kbps baik menggunakan *codec* H.263 maupun H.263+ terlihat bahwa layanan *video call* tidak memenuhi rekomendasi ITU-T dimana *delay* maksimal yang diizinkan adalah 150 ms serta *packet loss* < 1% . Perbincangan tidak dimungkinkan karena *delay* yang terjadi

sangat besar. Besarnya *packet loss* yang terjadi diakibatkan oleh banyaknya paket yang di *drop* pada saat jaringan mengalami kemacetan, hal tersebut karena rata-rata *bandwidth* tiap paket yang dikirimkan adalah sebesar 187.9 kbps untuk H.263+ dan 264.48 kbps untuk H.263 sedangkan *bandwidth* jaringan hanya sebesar 96 kbps.

4.4.2 Jaringan Dengan Bandwidth 128 kbps

Pada pengujian ini bandwidth jaringan antara user A dan user B diatur sebesar 128 kbps. *Delay* rata-rata yang didapat dengan menggunakan *codec* H.263 adalah sebesar 112.45 ms dengan *delay* maksimum mencapai 580.15 ms, sedangkan besarnya *packet loss* mencapai 64.8%. Untuk *codec* H.263+ *delay* rata-rata nya sebesar 102.63 ms dengan *delay* maksimum sebesar 508.46 ms, dengan *packet loss* sebesar 58.6%. Perbandingan grafik *packet* dengan *delay* keduanya dapat dilihat pada gambar 4.6



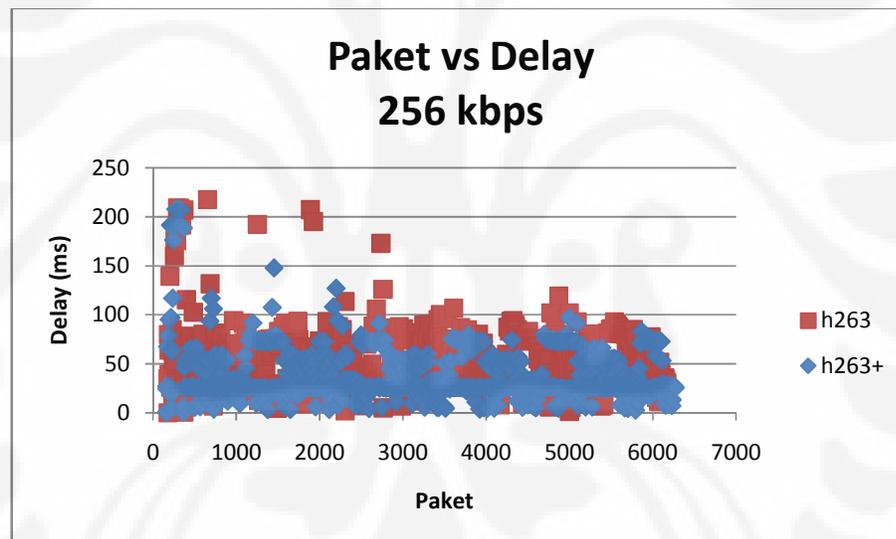
Gambar 4.6 grafik Paket vs *Delay* pada bandwidth 128 kbps

Dari pengukuran *delay* dan *packet loss* pada jaringan 128 kbps baik menggunakan *codec* H.263 maupun H.263+ terlihat bahwa layanan *video call* belum bisa diimplementasikan karena walaupun *delay* rata-rata dari kedua *codec* tersebut dibawah 150 ms, namun *packet loss* yang terjadi masih sangat besar untuk keduanya sehingga tidak memenuhi rekomendasi ITU-T dimana *packet loss* harus < 1%, hal tersebut dikarenakan bandwidth rata-rata tiap paket adalah sebesar

172.59 ms dan 240.26 ms masing-masing untuk H.263+ dan H.263 sedangkan bandwidth jaringan hanya 128 kbps. Begitu pula *delay*, besarnya *delay* dipengeuhi oleh *available bandwidth* dari jaringan.

4.4.3 Jaringan Dengan Bandwidth 256 kbps

Pada pengujian ini bandwidth jaringan antara user A dan user B diatur sebesar 256 kbps. *Delay* rata-rata yang didapat dengan menggunakan *codec* H.263 adalah sebesar sebesar 46.03 ms dengan *delay* maksimum mencapai 207.28 ms, sedangkan besarnya *packet loss* mencapai 12.4%. Untuk *codec* H.263+ *delay* rata-rata nya sebesar 37 ms dengan *delay* maksimum sebesar 191.45 ms, dengan *packet loss* sebesar 2.3%. Perbandingan grafik *packet* dengan *delay* keduanya dapat dilihat pada gambar 4.7

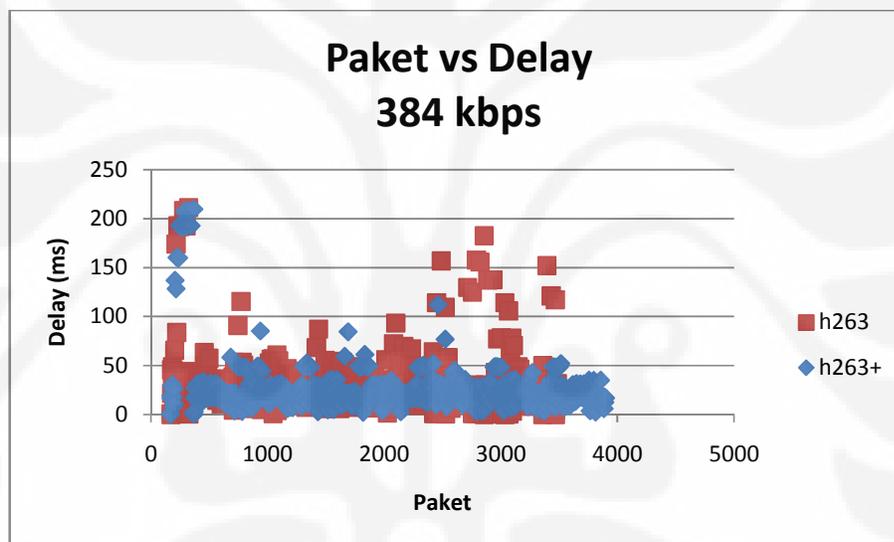


Gambar 4.7 grafik Paket vs *Delay* pada bandwidth 256 kbps

Dari pengukuran *delay* dan *packet loss* pada jaringan 256 kbps baik menggunakan *codec* H.263 maupun H.263+ terlihat bahwa layanan *video call* belum bisa diimplementasikan karena walaupun *delay* rata-rata dari kedua *codec* tersebut dibawah 150 ms, namun *packet loss* yang terjadi masih diatas 1 %. Rata-rata konsumsi bandwidth tiap paket untuk H.263+ dan H.263 masing-masing sebesar 192.85 kbps dan 231.93 kbps. Dari hasil percobaan, walaupun memungkinkan untuk melakukan percakapan, namun kualitas *video* yang dihasilkan dapat dikatakan belum cukup baik dan belum layak untuk diimplementasikan.

4.4.4 Jaringan Dengan Bandwidth 384 kbps

Pada pengujian ini bandwidth jaringan antara user A dan user B diatur sebesar 384 kbps. *Delay* rata-rata yang didapat dengan menggunakan *codec* H.263 adalah sebesar sebesar 30.92 ms dengan *delay* maksimum mencapai 210.98 ms, sedangkan besarnya *packet loss* mencapai 0%. Untuk *codec* H.263+ *delay* rata-rata nya sebesar 23.9 ms dengan *delay* maksimum sebesar 209.52 ms, dengan *packet loss* sebesar 0.9%. Perbandingan grafik *packet* dengan *delay* keduanya dapat dilihat pada gambar 4.8



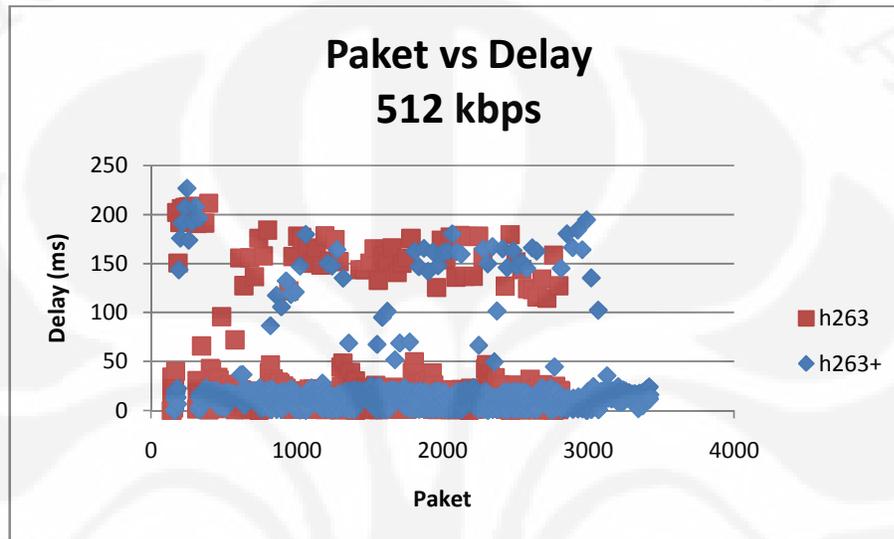
Gambar 4.8 grafik Paket vs *Delay* pada bandwidth 384 kbps

Dari pengukuran *delay* dan *packet loss* pada jaringan 384 kbps baik menggunakan *codec* H.263 maupun H.263+, keduanya mencapai hasil yang memuaskan yaitu dimana rata-rata *delay* nya dibawah 150 ms dan *packet loss* dibawah 1 %. Rata-rata konsumsi bandwidth tiap paket untuk H.263+ dan H.263 masing-masing sebesar 307.84 kbps dan 272.55 kbps

4.4.5 Jaringan Dengan Bandwidth 512 kbps

Pada pengujian ini bandwidth jaringan antara user A dan user B diatur sebesar 512 kbps. *Delay* rata-rata yang didapat dengan menggunakan *codec* H.263 adalah sebesar sebesar 25.98 ms dengan *delay* maksimum mencapai 211.33 ms,

sedangkan besarnya *packet loss* mencapai 0%. Untuk *codec* H.263+ *delay* rata-rata nya sebesar 23.16 ms dengan *delay* maksimum sebesar 196.25 ms, dengan *packet loss* sebesar 0%. Perbandingan grafik *packet* dengan *delay* keduanya dapat dilihat pada gambar 4.9



Gambar 4.9 grafik Paket vs *Delay* pada bandwidth 512 kbps

Rata-rata konsumsi bandwidth tiap paket untuk H.263+ dan H.263 masing-masing sebesar 298.64 kbps dan 228.78 kbps. Kualitas layanan *video call* yang ditunjukkan pada jaringan dengan bandwidth 512 kbps ini tidak memiliki perbedaan yang signifikan dengan kualitas pada jaringan 384 kbps. Dari hasil yang pengujian, layanan *voice call* ini dapat diimplementasikan dengan baik pada jaringan ini.

BAB 5

KESIMPULAN

1. Jaringan IMS yang dibangun berjalan sesuai dengan konsep kerjanya.
2. Registration time jaringan IMS yang dibangun ini adalah sebesar 0.367 s, sedangkan
3. Ringing time jaringan IMS yang dibangun ini adalah sebesar 1.92 s
4. Kualitas layanan *video call* dipengaruhi oleh besarnya *bandwidth* yang tersedia, dari hasil pengukuran *delay* dan *packet loss* dengan memvariasikan *bandwidth* jaringan terlihat bahwa layanan ini dapat diimplementasikan pada jaringan dengan bandwidth 384 kbps keatas, dimana besarnya *delay* dan *packet loss* tidak lebih dari standard yang dikeluarkan oleh ITU-T.

DAFTAR ACUAN

- [1] Johnston, A.B. (2004). SIP Understanding The Session Initiation Protocol. ARTECH HOUSE, INC
- [2] Bates, Juliet., Gallon, C., Bocci M., Walker, Stuart., Taylor, Tom. (2006). Converged Multimedia Networks. Wiley
- [3] Rodriguez. Iria. (2007). QoS Estimation during Session Initiation of Video Streaming Session. Wien
- [4] Chakraborty, S., Frankkila, T., Peisa. Janne., Synnergen, P. (2007). IMS Multimedia Telephony over Cellular Systems. Wiley
- [5] IMS Subsystem – NGN. <http://knol.google.com/k/ims-subsystem-ngn>
- [6] Francisco J, Hens., Jose M, Caballero. (2008). Triple Play Building the converged network for IP, VoIP and IPTV. Wiley
- [7] M. Handley, V. Jacobson, and C. Perkins. SDP: Session Description Protocol, July 2006. RFC 4566

LAMPIRAN

Grafik konsumsi bandwidth tiap paket

