UNIVERSITAS INDONESIA

IMPLEMENTASI QUALITY OF SERVICE PADA JARINGAN IMS DENGAN PRIORITAS PAKET

SKRIPSI

ARDY THIOTRISNO

0706267534

FAKULTAS TEKNIK

PROGRAM STUDI TEKNIK ELEKTRO

DEPOK

JUNI 2011

UNIVERSITAS INDONESIA

IMPLEMENTASI QUALITY OF SERVICE PADA JARINGAN IMS DENGAN PRIORITAS PAKET

SKRIPSI

Diajukan sebagai salah satu syarat memperoleh gelar sarjana

Disusun Oleh :

ARDY THIOTRISNO

0706267534

DEPARTEMEN TEKNIK ELEKTRO

FAKULTAS TEKNIK

UNIVERSITAS INDONESIA

JUNI 2011

HALAMAN PERNYATAAN ORISINALITAS

Skripsi ini adalah hasil karya saya sendiri,

dan semua sumber baik yang dikutip maupun dirujuk

Telah saya nyatakan dengan benar.

Nama	**: Ardy Thiotrisno**
NPM	**: 0706267534**
Tanda Tangan	**:**
Tanggal	**: 15 Juni 2011**

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh	:
Nama	: Ardy Thiotrisno
NPM	: 0706267534
Program Studi	: Teknik Elektro
Judul Skripsi	: Implementasi Quality of Service pada Jaringan IMS dengan Prioritas Paket

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 : Dr. Ir. Anak Agung Putri Ratna M.Eng ( )

Penguji : Ir. Endang Sriningsih MT, Si ( )

Ditetapkan di : Depok

Tanggal : 27 Juni 2011

KATA PENGANTAR

Segala puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa karena atas berkat rahmat dan karunia-Nya lah penulis dapat menyelesaikan penulisan skripsi ini. Penulisan skripsi dilakukan sebagai salah satu syarat untuk menjadi Sarjana Teknik di Departemen Teknik Elektro FTUI. Pada penulisan skripsi ini penulis menyadari bahwa tanpa bantuan banyak pihak, skripsi ini tidak mungkin terselesaikan. Oleh sebab itu, pada kesempatan ini penulis ingin mengucapkan terima kasih kepada pihak-pihak yang membantu dalam segala hal mengenai penyusunan seminar baik secara langsung maupun tidak langsung. Hal ini penulis tujukan kepada:

1. Ir. Djamhari Sirat M.Sc, Ph.D selaku dosen Pembimbing seminar yang telah meluangkan waktunya, serta masukan-masukan selama bimbingan dan pengerjaan skripsi.
2. Prof. Dr. Ir. Bagio Budiardjo M.Sc. dan Muhammad Salman S.T., M.IT sebagai dosen pembimbing DTE UI IMS *Research Group*.
3. Kedua orang tua penulis dan segenap keluarga yang selalu mendukung dan menyemangati kegiatan yang penulis lakukan.
4. Dipl.Ing Dragos Vingarzan selaku pengembang OpenIMSCore atas kerjasama, saran serta masukannya selama pengerjaan skripsi ini.
5. Pak Randi, Pak Angkoso, Pak David dan Pak Iwan (PT. telkom) yang telah membantu dalam *sharing* informasi mengenai IMS.
6. Rekan-rekan satu bimbingan: Chandra Gunawan, Faisal Jamil, Krisna Juanta, dan Rosa yang senantiasa membantu, memotivasi dalam membahas mengenai topik-topik yang berkaitan dengan IMS.
7. Asisten laboratorium jaringan: Alfa Sheffildi, Burhan Adi Wicaksana, dan Ruki Harwahyu atas bantuan teknisnya dalam membantu penulis dalam menyelesaikan skripsi ini.
8. Hartono dan Deolens atas informasi teknisnya mengenai aplikasi VoD di jaringan OpenIMSCore.
9. Sahabat-sahabat karib penulis: Dimas, Stephen, Sandi, Yuananda, Cello, Irwansah, Rhyando, Rizky ATA, Rudi dan Yuddy yang selalu menyemangati penulis dalam penulisan dan segala keceriaan yang telah dilalui bersama.

10. Anggota Keluarga Mahasiswa Buddhist Universitas Indonesia terkhusus untuk angkatan 2007 : Dewi Tiaw, Vinny, Adi, Yuli, Lita , Biyanto, Prisilia, Elaine, Devyana, Lidya, Jo, Berry, dan rekan-rekan lainnya atas semangat yang diberikan.
11. Rekan-rekan penulis di Laboratorium Elektronika : Edy, Danang, Chatra, Novri, Taufik, Rifky, Gavin, Osel, Jujud yang bersama-sama dengan penulis menjalani semester akhir perkuliahan di DTE UI atas segala kesenangan yang telah dilalui bersama-sama.
12. Rekan-rekan penulis : Wandy, Fia, dan Zunaidi Maruf yang bersedia meminjamkan laptopnya untuk membantu penulis melakukan percobaan, Benny, Bayu, Archie, Firdaus, Vanessa, Rizka, dan Tania yang bersama-sama ketika mengerjakan percobaan skripsi baik di Laboratorium Jaringan DTE UI ataupun di Mercator Office Engineering Center Universitas Indonesia.
13. Rekan-rekan SMA penulis : Adinda Lee dan Yumiko Wongso atas bantuan moralnya untuk menyelesaikan skripsi ini dan saudara Edwin Thioriks atas segala bantuan teknis yang sangat membantu pengerjaan skripsi ini.
14. Seluruh keluarga besar Civitas Akademika Fakultas Teknik Universitas Indonesia khususnya karyawan sekretariat Departemen Teknik Elektro yang telah banyak memberikan bantuan dalam urusan administrasi.

Penulis menyadari bahwa skripsi ini masih jauh dari kesempurnaan. Hal ini karena keterbatasan pengetahuan dan pengalaman yang dimiliki penulis. Dengan segala kerendahan hati, penulis mengharapkan kritik dan saran untuk memperbaiki skripsi ini pada khususnya dan kemampuan penulis pada umumnya. Semoga tugas akhir ini dapat berguna dan memberikan manfaat bagi kita semua.

Depok, 15 Juni 2011

Penulis

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI UNTUK KEPENTINGAN AKADEMIS

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

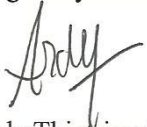
Nama	: Ardy Thiotrisno
NPM	: 0706267534
Program Studi	: Teknik Elektro
Departemen	: Teknik Elektro
Fakultas	: Teknik
Jenis Karya	: Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia hak bebas royalty noneksklusif (*Non-exclusive Royalty Free Right*) atas karya ilmiah saya yang berjudul :

Implementasi Quality of Service pada Jaringan IMS dengan Prioritas Paket

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalty Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (database), merawat, dan mempublikasikan tugas akhir saya selama tetap mencatumkan nama saya sebagai penulis/pencipta dan sebagai pemilik hak cipta.
Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok
Pada Tanggal : 15 Juni 2011
Yang menyatakan



(Ardy Thiotrisno)

ABSTRAKSI

Nama : Ardy Thiotrisno
Program Studi : Teknik Elektro
Judul : Implementasi Quality of Service pada Jaringan IMS dengan Prioritas Paket

Skripsi ini membahas mengenai implementasi *Quality of Service* di jaringan IMS. IMS memiliki mekanisme *Quality of Service* yang dapat menjamin layanan IMS untuk beroperasi sesuai dengan yang diharapkan. Implementasi *Quality of Service* dilakukan dengan memodelkan suatu jaringan IMS yang memberikan prioritas paket terhadap aplikasi VoIP dibanding VoD dengan menggunakan *open source router*.

Pengambilan data dilakukan dengan menggunakan program *Wireshark* untuk mengamati parameter *Quality of Service* seperti *delay, jitter,* dan *packet loss*. Setelah dilakukan implementasi *Quality of Service*, parameter-parameter QoS aplikasi VoD menunjukkan peningkatan *delay* sebesar 42 ms, *jitter* sebesar 47 ms, dan *packet loss* 39 % dibandingkan tidak dilakukan implementasi QoS untuk *bandwidth* 384 kbps. Aplikasi VoIP menunjukkan penurunan *delay* 42 ms, *jitter* 75 ms, dan *packet loss* 5% untuk *bandwidth* 16 kbps ketika dilakukan implementasi QoS dibandingkan dengan tidak adanya QoS di jaringan. Hal ini menunjukkan bahwa implementasi QoS di jaringan IMS telah berjalan sesuai teori.

Selain itu, dilakukan sebuah percobaan tambahan yang mengvariasikan durasi gangguan dari VoIP terhadap VoD. Hasil yang diperoleh menunjukkan performa VoD yang terpengaruh oleh variabel durasi gangguan sedangkan performa VoIP tidak terganggu oleh gangguan dari VoD baik dengan variasi durasi 45 detik dan 2 menit dimana *delay* VoIP bernilai tetap di angka 20 ms.

Kata kunci : IMS, Quality of Service, prioritas paket, delay, jitter, packet loss.

ABSTRACT

Name : Ardy Thiotrisno
Study Program : Electrical Engineering
Title : Implementing Quality of Service in IMS Network by prioritizing packet

This thesis described the implementation of Quality of Service in IMS network. IMS has Quality of Service mechanism which can guarantee the IMS services to operate as expected. Quality of Service is implemented by modeling an IMS network that gives packet priority VoIP application better than VoD with open source routers.

Data of this experiment was taken using Wireshark program to analyze Quality of Service parameters such as delay, jitter and packet loss. After impelementing Quality of Service, VoD's QoS parameters increase 42 ms in delay, 47 ms in jitter, and 39 % in packet loss compare with no QoS implementation for 384 kbps bandwidth usage. VoIP application show decreasing trend, 42 ms in delay, 75 ms in jitter, and 5 % in packet loss compare with no QoS implementation for 16 kbps bandwidth usage. It shows that the implementation of QoS in IMS network has suited with the theory.

Besides, there's an additional experiment with VoIP interruption duration variable. The results of this experiment show that VoD is affected by this duration variable while VoIP is not affected by this durational variable, VoIP delay is constant at 20 ms delay.

Keywords: IMS, Quality of Service, packet priority, delay, jitter, packet loss.

DAFTAR ISI

DAFTAR GAMBAR

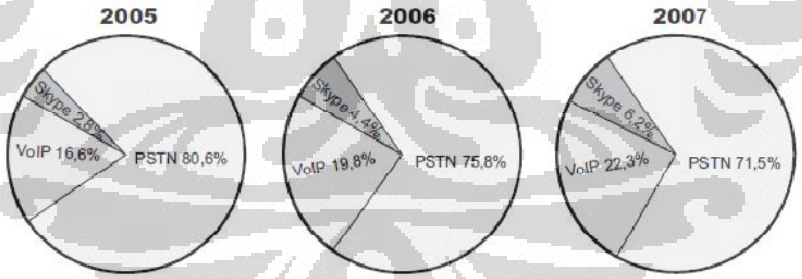
DAFTAR TABEL

BAB 1
PENDAHULUAN

1.1 Latar Belakang

Dunia telekomunikasi sebagai salah satu penyokong kehidupan masyarakat luas, mengalami perkembangan yang sangat pesat, hal ini didorong oleh perkembangan teknologi yang selalu menciptakan inovasi-inovasi baru, baik berupa perangkat telekomunikasi baru seperti *handphone* yang sudah menggantikan peran dari telepon rumah, maupun aplikasi-aplikasi baru dalam berkomunikasi seperti *instant messaging*, VoIP ataupun *video call*.

Teknologi-teknologi baru ini membawa perubahan pada kebutuhan serta tuntutan masyarakat yang tadinya hanya terbatas pada layanan komunikasi suara hingga kini berkembang dengan tambahan layanan video dan multimedia lainnya. Hal ini terbukti dengan program *Yahoo! Messenger* dengan *Instant messaging* dan *Skype* dengan layanan VoIP serta *Video Call* yang mampu menyerap pengguna aktif hingga 27 juta orang dari seluruh dunia. [1] Layanan lain yang cukup menarik perhatian masyarakat adalah layanan *Video on Demand*, hal ini dapat dilihat dari pengguna situs *youtube* yang mencapai 100 juta orang per bulan. [2]



Gambar 1.1 pertumbuhan pengguna VoIP dan Skype [3]

Internet yang menyediakan layanan secara *best effort* mampu menarik perhatian masyarakat karena mayoritas aplikasi yang tersedia di internet merupakan aplikasi yang gratis dan hal ini menjadi pemicu pertumbuhan pengguna internet. Pertumbuhan pengguna internet yang dapat dilihat pada gambar 1.1 yang menunjukkan penurunan persentasi jumlah pengguna PSTN dari tahun 2005 hingga tahun 2007 dan digantikan oleh penggunan VoIP dan *skype*.

Pertumbuhan jumlah pengguna internet telah menjadi salah satu ancaman terbesar bagi operator telekomunikasi. Operator telekomunikasi seolah-olah hanya berfungsi sebagai penyedia infrastruktur bukan sebagai penyedia layanan. Oleh karena itu, dibutuhkan suatu solusi untuk mengatasi masalah tersebut yang dikenal sebagai teori konvergensi. Melalui konsep yang ditawarkan oleh jaringan konvergen, pelanggan dapat mengakses berbagai macam aplikasi hanya dari satu perangkat elektronik sama seperti pengguna internet yang mampu mengakses berbagai layanan melalui satu divais seperti laptop ataupun sebuah komputer.

Saat ini, operator-operator penyedia jasa telekomunikasi telah berupaya untuk mewujudkan jaringan konvergen, dengan penggunaan jalur yang sama untuk penyediaan berbagai macam layanan seperti aplikasi suara, video dan data. Namun penggunaan jaringan secara bersama-sama ini masih memiliki masalah, dimana dalam pendudukan jalur di jaringan terjadi tumpang tindih antar paket data, suara maupun video sehingga penggunaan jaringan menjadi kurang efisien.

IP *Multimedia Subsystem* (IMS) merupakan salah satu usaha untuk mewujudkan konvergensi jaringan sekaligus mampu membantu operator telekomunikasi untuk bersaing dengan perkembangan dunia IT khususnya internet. Keunggulan operator telekomunikasi sebagai penyedia jasa telekomunikasi dibanding internet adalah bahwa operator telekomunikasi memiliki infrastruktur yang dapat menjamin kelancaran serta kualitas pelanggan dalam berkomunikasi jika dibandingkan dengan internet yang memberikan layanan kepada pelanggan dengan metode *best effort.* IMS memiliki mekanisme untuk memberikan *Quality of Service* yang dapat menjamin kepuasan bagi pengguna layanan. Di samping itu, IMS memiliki mekanisme *charging* yang dapat membantu operator dalam menetapkan tarif bagi pelanggannya.

Secara garis besar, *Quality of Service* dibagi ke dalam tiga metode yaitu, *best effort, Integrated Service (intserv),* dan *Differentiated Service (diffserv)*. Dari ketiga metode tersebut, metode yang sering digunakan adalah metode *Diffserv* karena hanya membutuhkan suatu *server* tambahan untuk mengatur *Quality of Service. Diffserv* tidak membutuhkan kesepakatan secara *end-to-end* seperti metode *intserv. Diffserv* mengimplementasi QoS dengan cara membeda-bedakan

paket kemudian memberikan prioritas yang berbeda-beda terhadap paket-paket tersebut.

Pada penelitian ini dibuat sebuah mekanisme *Quality of Service* yang didasarkan pada pemberian prioritas terhadap aplikasi-aplikasi yang digunakan di jaringan IMS. Salah satu contoh pengaplikasian prioritas ini adalah menempatkan layanan suara sebagai prioritas tertinggi kedua setelah *signaling* dibandingkan layanan multimedia lainnya seperti *video streaming* dan *online gaming* karena layanan suara rentan terhadap *delay* yang dapat mengganggu kualitas komunikasi yang diperoleh pelanggan.

1.2 Perumusan Masalah

Pada Skripsi ini, ingin diimplementasikan suatu jaringan berbasis IMS dengan *OpenIMSCore* sebagai *server*-nya. Jaringan ini nantinya akan mengimplementasikan *Quality of Service* dengan membuat suatu mekanisme prioritas terhadap layanan-layanan yang digunakan di IMS.

Masalah yang ingin dibahas dalam penelitian ini adalah:

1. Mengapa *Quality of Service* dibutuhkan pada jaringan IMS?
2. Bagaimana mekanisme manajemen *Quality of Service* di jaringan *OpenIMSCore*?
3. Bagaimana mengimplementasikan prioritas layanan pada jaringan *OpenIMSCore*?

1.3 Tujuan

Penulisan Skripsi dengan judul "Implementasi QoS pada Jaringan IMS dengan prioritas paket" ini bertujuan untuk:

1. Memodelkan suatu jaringan IMS dengan prioritas paket pada jaringan *OpenIMSCore*.
2. Membuktikan bahwa VoIP mendapatkan prioritas yang lebih baik dibandingkan VoD melalui implementasi *Quality of Service* di jaringan IMS.

1.4 Batasan Masalah

Pada Skripsi ini, penelitian dilakukan tahap pembangunan jaringan IMS dengan layanan VoIP dan VoD karena kedua layanan ini dianggap cukup mewakili layanan-layanan di jaringan IMS. Konfigurasi pada *server* dan *router* yang digunakan akan memberi VoIP performa yang lebih baik karena prioritas yang lebih tinggi dibandingkan dengan VoD di dalam jaringan IMS. Pada penelitian ini akan dilihat perubahan performansi aplikasi VoD dan VoIP baik sebelum dan setelah dilakukan implementasi QoS di jaringan IMS.

1.5 Sistematika Penulisan

Skripsi ini akan ditulis ke dalam lima bab, dimana masing-masing bab akan membahas hal-hal sebagai berikut.

- BAB 1 : Pendahuluan

  Bab 1 berisi mengenai gambaran umum mengenai pembahasan penelitian ini. Bagian ini mencakup latar belakang topik yang dibahas, perumusan masalah, tujuan penelitian, dan sistematika penulisan penelitian.
- BAB 2 : IP Multimedia Subsystem

  Bab 2 membahas mengenai konsep dari jaringan konvergen, IMS sebagai teknologi yang mendukung terciptanya jaringan konvergen, arsitektur jaringan IMS, dan pembahasan mengenai QoS di jaringan IMS.
- BAB 3 : Implementasi *Quality of Service* pada Jaringan *OpenIMSCore*

  Bab 3 membahas tentang proses perancangan jaringan untuk mengimplementasikan *Quality of Service* di jaringan *OpenIMSCore*, serta menjelaskan peralatan yang dibutuhkan dalam perancangan, topologi jaringan serta skenario percobaan yang akan dilakukan.
- BAB 4 : Hasil Percobaan dan Analisis Data

  Bab 4 membahas mengenai metode pengambilan data serta analisis dari data yang diperoleh.
- BAB 5 : Kesimpulan

  Bab 5 menjelaskan kesimpulan yang diperoleh dari penelitian ini.

BAB 2
IP MULTIMEDIA SUBSYSTEM

2.1 Konvergensi jaringan

Industri telekomunikasi saat ini mengalami perkembangan yang sangat pesat. Perkembangan ini mendorong lahirnya sebuah teknologi baru yang mampu menyatukan berbagai macam teknologi telekomunikasi yang sudah ada sekarang ini seperti PSTN, teknologi seluler, *broadband* dan lain-lain. Konvergensi jaringan mampu menciptakan suatu konvergensi layanan yang semakin memudahkan orang-orang dalam melakukan kegiatan berkomunikasi.

Konvergensi sendiri bertujuan untuk memungkinkan terintegrasinya berbagai jaringan telekomunikasi, mengurangi jumlah *protocol* di *network layer* dan menggabungkan *transport* dari berbagai macam trafik jaringan dengan menggunakan sebuah *core*. Kunci utama dari terwujudnya jaringan konvergen adalah layanan internet *broadband* yang mampu melakukan pengiriman aplikasi multimedia di semua jenis jaringan. *Session Initiation Protocol* (SIP) memegang peranan penting dalam penyediaan layanan multimedia dalam sebuah sesi komunikasi. *Core* inti yang digunakan dalam jaringan konvergen adalah *IP Multimedia Subsystem* (IMS).[4]

Jaringan konvergen mampu menyediakan berbagai macam jenis konvergensi seperti.

1. Bundling

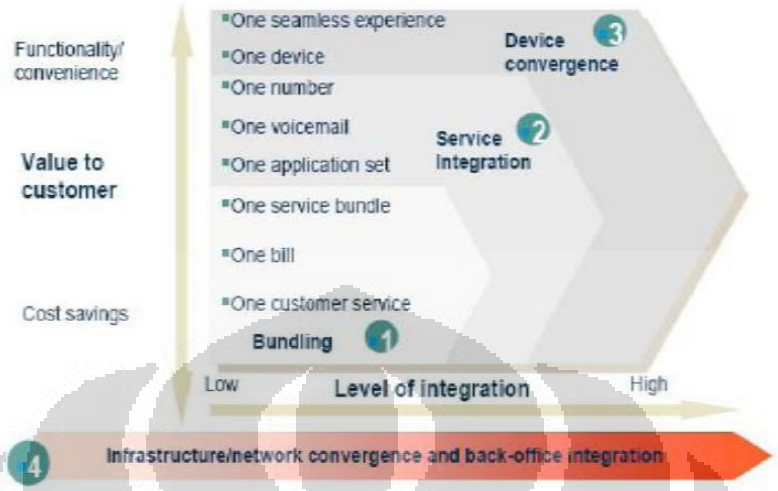
   Bundling merupakan suatu pengemasan di level provider dimana hanya terdapat satu *pelanggan identity*, satu paket layanan, dan satu tagihan.

2. Service convergence

   Service convergence adalah sebuah konsep konvergensi dimana suatu layanan dapat diakses dari berbagai macam divais.

3. Device convergence

   Device convergence adalah konsep konvergensi dimana satu buah divais mampu mengakses berbagai macam layanan, seperti *handphone* selain mampu melakukan panggilan suara dan mengirimkan sms, mampu memutar musik, video dan layanan lainnya.



Gambar 2.1 Konvergensi Telekomunikasi [5]
Soetikno Tendy. *IMS as Convergence Enabler*

2.2IP Multimedia Subsystem

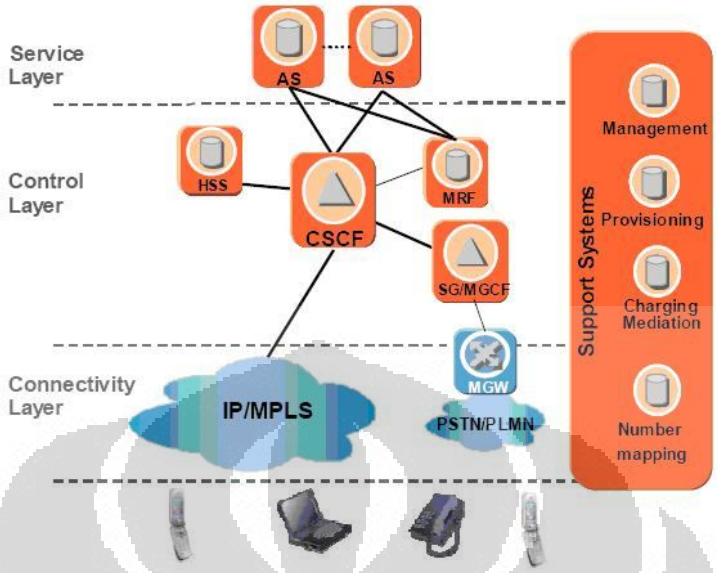
IP Multimedia Subsystem (IMS) adalah sebuah arsitektur jaringan yang digunakan untuk menyediakan layanan multimedia yang berbasis internet *protocol* (IP).[6] IMS merupakan standar yang dibuat oleh *3rd Generation Partnership Project* (3GPP) untuk mewujudkan *Fixed Mobile Convergence* (FMC) yang mampu mengintegrasikan hubungan antara teknologi seluler dengan jaringan PSTN. IMS berperan sebagai *session-control subsystem* yang didasari pada internet *protocol* (IP), *Session Initiation Protocol* (SIP), *Session Description Protocol* (SDP) dan beberapa *protocol* lainnya yang mampu mendukung layanan-layanan multimedia yang terhubung melalui berbagai macam jaringan akses.[3] Sebagai *session control*, IMS menyediakan beberapa fungsi seperti *subscriber profile management*, mekanisme *charging*, serta alokasi QoS pada media transmisi.

Ada berbagai macam aplikasi yang dapat digunakan oleh IMS *subscribers*, diantaranya adalah *video telephony, push to talk over cellular* (PoC), *instant messaging and presence* (IMP), *multiparty conferencing*, IPTV, *online gaming*, dan masih banyak yang lainnya. Untuk menyediakan aplikasi-aplikasi ini IMS

harus dapat menjalankan *advance session control* dengan cara yang berbeda-beda, yaitu:

- Menyesuaikan aplikasi-aplikasi berdasarkan *pelanggan* dan *access network*, dengan fungsi ini *network* dapat mengurangi *bitrate* sinyal mengurangi *bandwidth*, atau mengurangi ukuran dari *image* di aplikasi *video telephony* pada saat video ditampilkan pada layar yang kecil di *mobile phone*.
- Memodifikasi sesi sesuai dengan permintaan pengguna, dengan fungsi ini IMS dapat memenuh permintaan pelanggan yang ingin mengatur *mobile phone*-nya untuk berpindah dari mode *Wi-Fi* di kantor ke mode 3G di rumah.
- Memungkinkan pengguna untuk mengakses layanan yang pada *home network* ketika berada pada *visited network*, fungsi ini biasanya digunakan oleh pengguna IPTV yang ingin menonton *channel* yang disediakan oleh *home network* ketika dia sedang berada pada *visited network*.

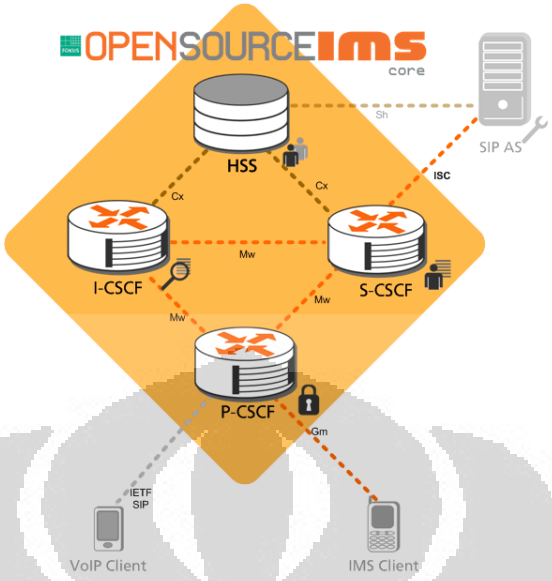
Gambar 2.1 menunjukkan arsitektur jaringan IMS secara umum yang terbagi ke dalam tiga *layer* yaitu, *Service Layer* yang terdiri dari beberapa *Application Server* (AS) yang bertugas untuk menyediakan konten dan aplikasi multimedia di dalam jaringan IMS, *Control Layer* terdiri dari *server-server* yang mengatur media session di dalam jaringan IMS, dan *Connectivity Layer* terdiri dari perangkat-perangkat seperti *switch* dan *router* yang menjadi bagian dari jaringan akses *client* ke jaringan IMS.



Gambar 2.2 Arsitektur Jaringan IMS [7]

Divais elektronik yang telah berbasis IP dapat langsung terhubung dengan IMS namun untuk jaringan yang belum berbasis IP seperti jaringan PSTN ataupun teknologi seluler GSM dibutuhkan suatu *Media Gateway* (MGW) sebelum terhubung dengan jaringan IMS. Selain itu terdapat *Media Gateway Control Function* (MGCF) yang melakukan kontrol terhadap *Gateway* yang masuk ke jaringan IMS. *Media Resource Function* (MRF) berfungsi sebagai jalur informasi dari sumber lain jika dibutuhkan.

Server Open IMS Core yang digunakan berbasis *open source* yang dijalankan di *Operating system Ubuntu* Linux. Gambar 2.3 menunjukkan komponen-komponen *Open IMS Core* yang terdiri dari *Home Subscriber Server* (HSS), *Proxy CSCF* (P-CSCF), *Interrogating CSCF* (I-CSCF), dan *Serving CSCF* (S-CSCF).



Gambar 2.3 IMS Core [8]

2.2.1 Home Subscriber Server (HSS)

Home Subcriber Server (HSS) merupakan *server* yang berfungsi untuk menyimpan semua data yang dimiliki oleh pelanggan, baik itu *private identity* maupun *public identity*. HSS memiliki fungsi yang sama dengan *Home Location Register* (HLR) pada teknologi seluler. Selain itu, HSS memiliki data mengenai layanan apa saja yang dapat diakses oleh *subscriber* serta memiliki informasi mengenai suatu *subscriber* apakah sedang teregistrasi atau tidak dan juga mengetahui lokasi dari *subscriber*.

Tugas lain dari HSS dalam jaringan IMS adalah menentukan *Serving CSCF* yang ingin diakses oleh subscriber. Hubungan antara HSS dan SCSCF melalui *Cx reference point* menggunakan *protocol Diameter* yang didesain untuk fungsi otentikasi, otorisasi, dan akuntasi.

Sebuah jaringan IMS dapat terdiri lebih dari satu HSS, dalam hal jika jumlah pelanggan yang terlalu tinggi untuk ditangani oleh satu HSS.[9] Jaringan dengan satu HSS tidak memerlukan *Subscription Locator Function* (SLF). Di sisi lain, jaringan dengan lebih dari satu HSS memerlukan SLF. The SLF adalah *database* yang sederhana memetakan alamat pengguna pada semua HSS. Sebuah simpul berupa *query* SLF, dengan alamat pengguna sebagai *input*, memperoleh HSS yang berisi semua informasi yang terkait dengan pengguna sebagai *output*.

2.2.2 Call Session Control Function

Call session control function (CSCF) merupakan *proxy server* yang membantu proses inisiasi pelanggan, melakukan proses setup layanan, mengatur sesi dan mengirimkan pesan di dalam jaringan IMS. CSCF sendiri terbagi lagi menjadi tiga bagian yang memiliki fungsi yang lebih spesifik yaitu.

1. Proxy Call Session Control Function (P-CSCF)

   P-CSCF adalah titik pertama yang menghubungkan antara pelanggan dengan IMS. P-CSCF berfungsi untuk meneruskan SIP *message* dari pelanggan ke dalam jaringan. Karena P-CSCF merupakan titik ujung pada saat informasi masuk dan keluar dari jaringan IMS, maka P-CSCF juga memiliki fungsi untuk melakukan autorisasi terhadap identitas pelanggan apakah pelanggan tersebut memiliki hak untuk mengakses jaringan atau sudah terdaftar di dalam jaringan.

2. Interogatting Call Session Control Function (I-CSCF)

   Pesan dari P-CSCF akan diteruskan menuju I-CSCF yang berfungsi untuk memilih S-CSCF yang sesuai dengan permintaan awal dari pelanggan. Sebelum menuju ke S-CSCF, I-CSCF akan mengakses HSS terlebih dahulu untuk meminta alamat dari S-CSCF yang dituju.

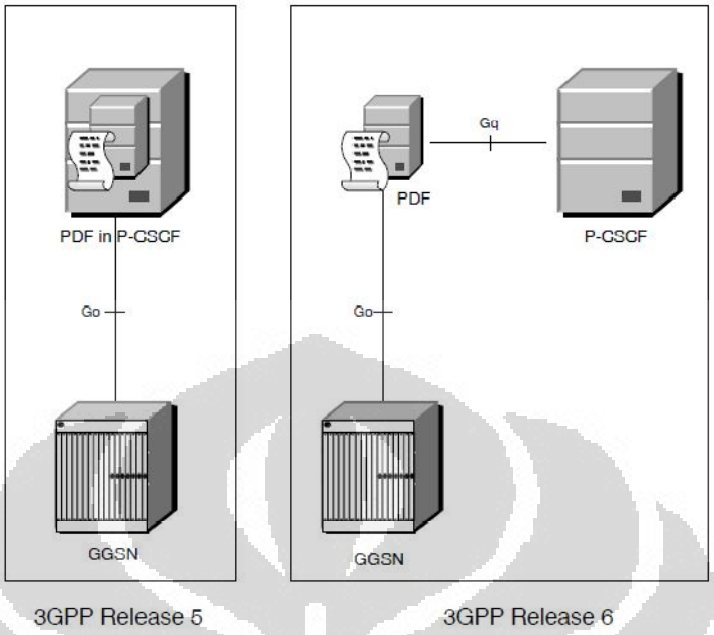
3. Serving Call Session Control Function (S-CSCF)

   S-CSCF berfungsi untuk menyediakan layanan bagi *pelanggan*, setelah I-CSCF memilih S-CSCF yang akan digunakan S-CSCF akan langsung bertukar informasi dengan P-CSCF dan I-CSCF dihilangkan dari jalur *signaling*.

2.2.3 Policy Control Rule Function (PCRF) & Policy and Charging Enforcement Point (PCEF)

IMS menggunakan *Policy Decision Function* (PDF) dan *Gateway GPRS Support Node* (GGSN) untuk mengatur *policy* baik yang berkaitan dengan *Quality of Serving* maupun *charging* yang diterapkan di dalam jaringan IMS. PDF berperan sebagai *Packet Data Protocol* (PDP) dan GGSN berperan sebagai *Policy Enforcement Point* (PEP) yang dapat digantikan dengan *Edge Router*.[10] PDF bertugas untuk menginformasikan GGSN mengenai karakteristik atau otoritas seorang pelanggan dalam suatu sesi multimedia.[11] Sebagai contoh, PDF mengirimkan informasi mengenai seorang *pelanggan* dengan kemampuan menggunakan layanan *audio streaming* dengan *bandwidth* maksimum 20 kbit/s. GGSN kemudian menggunakan informasi ini untuk menjalankan *packet filter* pada *routing logic* yang dimiliki oleh GGSN atau *Edge Router*. Jika *pelanggan* tersebut mencoba untuk menjalankan suatu aplikasi diluar kemampuannya maka *packet filter* tidak akan mengijinkan *traffic* dari aplikasi tersebut untuk masuk ke dalam jaringan.

Standar yang dikeluarkan oleh 3GPP *release 5*, PDF dapat ditempatkan di dalam *node* yang sama dengan P-CSCF sedangkan pada 3GPP *release 6*, PDF dan P-CSCF secara jelas telah berada pada *node* yang berbeda hal ini dapat dilihat pada Gambar 2.4. Standar 3GPP *release 7,* PDF mengalami perubahan dengan ditambahkannya kemampuan untuk mengatur *charging policy* IMS dan berganti nama menjadi *Policy Control Rule Function* (PCRF) sedangkan GGSN berganti menjadi *Policy and Charging Enforcement Point* (PCEF).



Gambar 2.4 Media authorization 3GPP release 5 dan 6 [11]

2.3 Protocol-protocol di dalam IMS

2.3.1 Session Initiation Protocol (SIP)

IMS menggunakan *Session Initiation Protocol* (SIP) sebagai *protocol* yang berfungsi untuk mengontrol sesi multimedia yang dilakukan oleh pelanggan dengan pelanggan lainnya ataupun antara pelanggan dengan media *server*. Berikut metode SIP yang digunakan dalam suatu sesi multimedia di dalam jaringan IMS.

- *Invite*, memanggil pelanggan untuk membangun suatu sesi komunikasi
- *Ack*, menandakan bahwa *client* telah menerima final respon dari sebuah *invite request*
- *Option*, digunakan untuk menanyakan suatu *server* tentang kemampuan yang dimilikinya
- *Bye*, menyampaikan pesan kepada *server* untuk mengakhiri suatu sesi
- *Cancel*, digunakan untuk mengakhiri suatu *request* yang sedang menunggu keputusan
- *Register*, digunakan oleh *client* untuk mendaftarkan identitas pribadi

Response messages berisi status kode dan keterangan tentang kondisi dari *request* tersebut. Nilai-nilai dari kode status yang serupa dengan penggunaan pada HTTP, dibagi dalam enam kategori:

- **1xx**: *Provisional, request* telah diterima dan sedang melanjutkan proses.
- **2xx**: *Success*, tindakan dengan sukses diterima, dipahami, dan disetujui.
- **3xx:** *Redirection*, tindakan lebih lanjut diperlukan untuk memproses permintaan ini.
- **4xx**: *Client Error*, permintaan berisi sintak yang salah dan tidak bias dikenali oleh *server* sehingga *server* tidak dapat memprosesnya.
- **5xx**: *Server Error, server* gagal untuk memproses suatu permintaan yang sah.
- **6xx**: *Global Failure*, permintaan tidak dapat dipenuhi oleh *server* manapun.

2.3.2 Session Description Protocol (SDP)

Session Description Protocol (SDP) merupakan protocol yang menggambarkan sesi multimedia yang akan dibangun di dalam jaringan IMS. SDP dibutuhkan karena SIP tidak memberikan informasi kepada *server* mengenai jenis sesi multimedia yang diinginkan oleh pelanggan. SDP mendeskripsikan beberapa hal, antara lain sebagai berikut. [3]

1. Nama dan tujuan *session*
2. Waktu aktif *session* tersebut
3. Tipe media yang akan digunakan
4. *Address, port,* dll.

Berikut adalah isi dari pesan SDP, *field-field* yang bersifat optional akan diberi tanda *. [12]

***Session* Description**

v= (protocol version)
o= (qwner/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*.

2.3.3 Real Time Protocol (RTP)

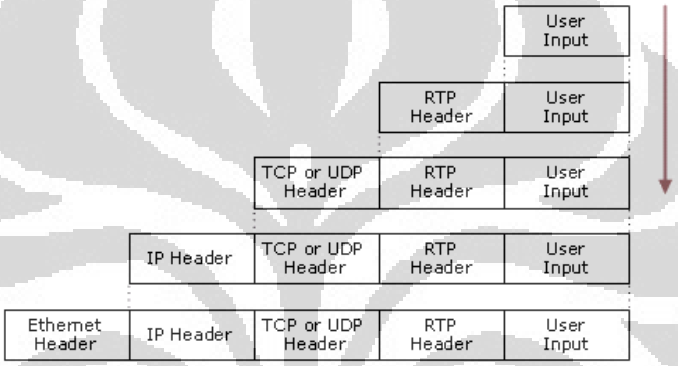
Real Time Protocol (RTP) merupakan protokol yang digunakan untuk membawa data *real time* baik data yang dikirimkan secara *unicast* ataupun *multicast* melalui jaringan berbasis IP. [3] RTP didesain untuk membawa informasi waktu bersama dengan data yang dikirimkan sehingga RTP digunakan untuk membawa data *audio* maupun *video sreams*. RTP terdiri dari suatu data dan *control part* yang disebut RTCP. [13]

RTP menyediakan servis pengiriman data *end-to-end real-time*. Servis ini meliputi *payload type identification, sequence numbering,* dan *time stamping*. Jaringan tempat pengiriman paket kemungkinan besar akan menyebabkan *delay* dengan variasi waktu tertentu. RTP didesain untuk membawa informasi waktu bersama dengan data. T*imestamp* m*ere- time* data yang diterima dengan *source timing* dan dengan akurasi yang cukup baik bagi sebagian besar aplikasi multimedia *streaming.*

Pada IP network dapat terjadi kehilangan paket atau *packet loss,* untuk itu *sequence number* dimasukkan ke dalam RTP. Penerima paket dapat mengetahui berapa paket yang hilang pada saat pengiriman melalui *sequence number.*

Fungsi penting lainnya dari RTP adalah *payload identification.* Pada implementasinya data dengan *payload* yang berbeda akan diperlakukan berbeda pada penerima sesuai dengan tipe paket. RTP akan mengabarkan penerima tentang tipe data yang dikirim pada *payload* tersebut dan bagaimana data tersebut di*encoded*.

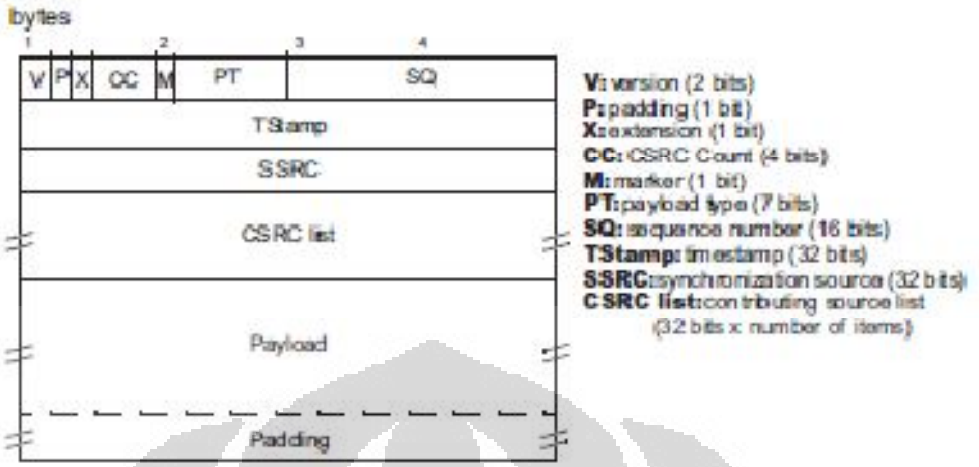
Informasi RTP dienkapsulasi dalam packet UDP. Jika paket RTP hilang atau didrop di jaringan, maka RTP tidak akan melakukan *retransmission* (sesuai standard *protocol* UDP). Hal ini agar pelanggan tidak terlalu lama menunggu (*long pause*) atau *delay*, dikarenakan permintaan *retransmission*.



Gambar 2.5 Enkapsulasi RTP [14]

Proses enkapsulasi RTP pada Gambar 2.5 terdapat IP header. IP header digunakan untuk mengindikasikan ukuran *header IP* dimana ukuran terkecilnya adalah 5 (0x05) yakni 20 *byte* dan ukuran *header IP* maksimum adalah 60 *byte*, yang diindikasikan dengan nilai 15 (0x0F). [11] IP *header* merupakan penanda terhadap prioritas paket diberikan yaitu *Differentiated Service Code Point* (DSCP).

Sebuah Pesan RTP mengandung RTP *header* yang diikuti dengan RTP *payload*. Gambar 2.6 menunjukkan *header* RTP dan Tabel 2.1 menunjukkan jenis-jenis *payload* RTP.



Gambar 2.6 RTP Packet Format [3]

- *Version* (V): *Field* ini merupakan versi dari RTP.
- *Padding* (P): *Field* ini digunakan jika media *stream* di-enkripsi.
- *Extension* (X): *Field* ini merupakan ekstensi tambahan yang mengikuti *header* yang dibuat oleh tipe *payload* tertentu.
- CSCR *count* (CC): *Field* ini memuat nomor CSRC.
- *Marker* (M): berfungsi menandai awal dari *frame* baru pada video.
- *Payload Type* (PT): merupakan 7-*bit field* yang menandai *codec* yang digunakan.
- *Sequence Number*: berukuran 16 *bit* yang berfungsi untuk mendeteksi hilangnya paket karena jumlah *sequence number* akan bertambah untuk setiap paket RTP yang dikirim.
- *Timestamps*: berupa 32 *bit* yang mengindikasi waktu relatif ketika *payload* di*sample*.
- *Synchronization Source Identifier* (SSRCI): mengindentifikasi *sender* dari paket RTP.
- CSCR *Contributing Source Identifier*: *field* ini hanya ada jika paket RTP telah dikirim oleh *mixer*.

Tabel 2.1 Payload RTP [3]

PT	Name	Encoder	Media
0	PCMU	PCM with μ-law scaling as per ITU-T G.711	Audio
3	GSM	GSM standard ETS 300 961	Audio
4	G723	ITU-T Recommendation G.723.1	Audio
5	DVI4	IMA ADPCM wave type, 8000 Hz clock	Audio
6	DVI4	IMA ADPCM wave type, 16000 Hz clock	Audio
7	LPC	Linear predictive encoding contributed by R. Frederick	Audio
8	PCMA	PCM with A-law scaling as per ITU-T G.711	Audio
9	G722	ITU-T Recommendation G.722	Audio
10	L16	Uncompressed audio, 16 bits, 44 100 Hz and 2 channels	Audio
11	L16	Uncompressed audio, 16 bits, 44 100 Hz and 1 channel	Audio
12	QCELP	EIA/TIA standard IS-733	Audio
13	CN	Comfort noise as per RFC 3389	Audio
14	MPA	ISO/IEC 11172-3 and 13818-3 MPEG-1 /MPEG-2 audio	Audio
15	G728	ITU-T Recommendation G.728	Audio
16	DVI4	IMA ADPCM wave type, 11 025 Hz clock	Audio
17	DVI4	IMA ADPCM wave type, 22 050 Hz clock	Audio
18	G729	ITU-T Recommendation G.729	Audio
Dynamic	G726-40	ITU-T Recommendation G.726, 40 kit/s ADPCM	Audio
Dynamic	G726-32	ITU-T Recommendation G.726, 32 kit/s ADPCM	Audio
Dynamic	G726-24	ITU-T Recommendation G.726, 24 kit/s ADPCM	Audio
Dynamic	G726-16	ITU-T Recommendation G.726, 16 kit/s ADPCM	Audio
Dynamic	G729D	ITU-T Recommendation G.729 annex D	Audio
Dynamic	G729E	ITU-T Recommendation G.729 annex E	Audio
Dynamic	GSM-EFR	GSM standard ETS 300 726	Audio
Dynamic	L8	Uncompressed audio, 8 bit samples	Audio
Dynamic	RED	REDundant audio payload as per RFC 2198	Audio
Dynamic	VDVI	Variable rate version of DVI4	Audio
25	CelB	Sun Microsystems CELL-B encoding	Video
26	JPEG	ISO Standards 10918-1 and 10918-2	Video
28	nv	'nv' encoding, version 4 developed at Xerox PARC	Video
31	H261	ITU-T Recommendation H.261	Video
32	MPV	ISO/IEC 11172-3 and 13818-3 MPEG-1/MPEG-2 video	Video
33	MP2T	MPEG-2 transport streams for either audio or video	Audio/Video
34	H263	ITU-T Recommendation H.263 of 1996	Video
Dynamic	H263-1998	ITU-T Recommendation H.263 of 1996	Video

2.3.4 Real Time Control Protocol (RTCP)

RTCP merupakan protokol komplemen dari RTP yang mempunyai fungsi utama untuk memberikan *feedback* tentang kualitas transmisi. RTCP dimultiplex bersama dengan RTP dimana RTP menggunakan nomor port genap dan RTCP menggunakan nomor port ganjil pada saat dikirimkan di dalam UDP. [12] 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. Tipe paket RTCP dapat dilihat dari Tabel 2.2

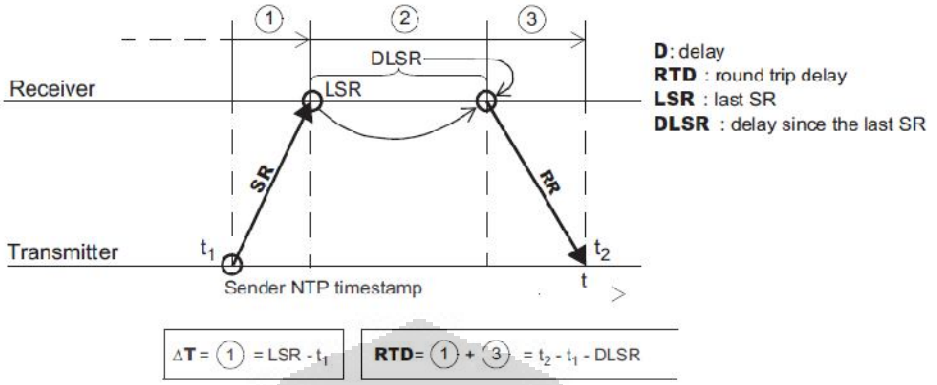
Tabel 2.2 RTCP Packet Types [3]

Sumber : Triple Play Building The Converges Network For IP, VoIP, and IPTV

Type	PT id	Purpose
SR	200	Sender report: quality statistics from active senders
RR	201	Receiver report: quality statistics from receivers that are not senders
SDES	202	Source description items: information on the transmitter
BYE	203	Indicates end of participation
APP	204	Application-specific functions

Fungsi utama dari RTCP adalah sebagai QoS *monitoring*. RTCP paket yang paling berperan untuk melakukan hal ini adalah *Sender Report* (SR) dan *Receiver Report* (RR).

RTP menyimpan informasi mengenai jumlah paket pada *sender's packet count filed* di SR dan jumlah *byte* dari payload yang dikirim pada *sender's byte count filed* di SR untuk memonitoring *packet loss*. RTCP dapat mengkalkulasikan jumlah paket yang hilang pada *cumulative number of packet lost field*.



Gambar 2.7 Perhitungan One Way *Delay* dan Round Trip *Delay* [3]

Gambar 2.7 menunjukkan proses perhitungan *One Way Delay* dengan melihat perbedaan waktu dari *originating packet timestamp* dengan waktu tibanya paket SR. Nantinya penerima juga akan mengirimkan RR ke pengirim sehingga memungkinkan untuk pengkalkulasian *Round Trip Delay*.

2.3.5 Real Time Streaming Protocol (RTSP)

Real Time Streaming Protocol (RTSP) adalah protokol pada *layer application* digunakan oleh program *streaming* multimedia untuk mengatur pengiriman data secara *real-time*.[12] Keuntungan RTSP adalah bahwa protokol ini menyediakan koneksi yang memiliki status antara *server* dan klien, yang dapat mempermudah *client* ketika ingin melakukan *pause* atau mencari posisi *random* dalam *stream* ketika memutar kembali data. Biasanya diterapkan pada pengiriman *video on demand.*

RTSP memiliki empat buah perintah. Perintah ini dikirim dari client ke sebuah *server streaming* RTSP. Perintah-perintah tersebut adalah sebagai berikut: [3]

- ***Setup,*** yaitu s*erver* mengalokasikan sumber daya kepada sesi klien.
- ***Play***, yaitu s*erver* mengirim sebuah *stream* ke sesi klien yang telah dibangun dari perintah *setup* sebelumnya.
- ***Pause,*** yaitu *server* menunda pengiriman stream namun tetap menjaga sumber daya yang telah dialokasikan.

- ***Teardown***, yaitu *server* memutuskan koneksi dan membebas tugaskan sumber daya yang sebelumnya telah digunakan.

2.3.6 Control Open Policy Service (COPS)

Control Open Policy Services (COPS) merupakan protocol yang menyediakan permintaan dan respon terhadap *policy* yang diterapkan di jaringan serta informasi mengenai status resources di jaringan. [16] COPS bekerja diantara *Policy Decision Function* (PDF) sebagai pembuat *policy* dan *Policy Enforcement Function* (PEF) atau *Edge Router* sebagai pelaksana *policy* yang bekerja dengan meng-*filter* paket dan menerapkan prioritas terhadap paket-paket yang masuk ataupun keluar dari jaringan.

Proses otorisasi terhadap level QoS pelanggan terjadi di dalam pertukaran *message* protocol COPS. Protocol COPS memiliki beberapa jenis *message* untuk melakukan pekerjaannya. Tabel 2.3 menunjukkan jenis *message* di dalam protocol COPS.

Tabel 2.3 type message COPS [16]

	Type	From PEP	From PDP	Description
1	REQ	√		Request
2	DEC		√	Decision
3	RPT	√		Report State
4	DRQ	√		Delete Request State
5	SSQ		√	Synchronize State Request
6	OPN	√		Client-Op
7	CAT		√	Client-Accept
8	CC	√	√	Client-Close
9	KA	√	√	Keep-Alive
10	SSC	√		Synchronize Complete

Source: Based on information in RFC 2748.

COPS memiliki dua buah permodelan, yaitu *Outsourcing* model dan *Provisioning* model. Hal yang membedakan kedua permodelan ini adalah pembuat *policy* di dalam jaringan. Pada model *Outsourcing*, PDP akan berlaku sebagai pembuat policy sedangkan pembuat *policy* di model *Provisioning* adalah PEP.

2.4 Quality of service

*Quality of Servic*e (QoS) merupakan mekanisme jaringan yang memungkinkan aplikasi-aplikasi atau layanan dapat beroperasi sesuai dengan yang diharapkan.[17] Kinerja jaringan komputer dapat bervariasi akibat beberapa masalah, seperti halnya masalah *bandwidth*, *latency* dan *jitter*, yang dapat membuat efek yang cukup besar bagi banyak aplikasi. Komunikasi suara (seperti VoIP atau *IP Telephony*) serta *video streaming* dapat membuat pengguna frustrasi ketika paket data aplikasi tersebut dialirkan di atas jaringan dengan *bandwidth* yang tidak cukup, dengan *latency* yang tidak dapat diprediksi, atau *jitter* yang berlebih.

Mekanisme *Quality of Service* (QoS) mampu memprediksi *bandwidth*, *latency*, dan *jitter* dan mencocokkannya dengan kebutuhan aplikasi yang digunakan di dalam jaringan tersebut yang ada. *Quality of Service* dapat diterapkan pada jaringan melalui mekanisme prioritas pada paket yang masuk ke jaringan, dimana setiap paket yang masuk ke jaringan akan diidentifikasi terlebih dahulu baik berdasarkan aplikasi maupun *protocol*, kemudian paket-paket mendapatkan prioritas berdasarkan *policy* yang berlaku di jaringan.

2.4.1 Parameter-Parameter Quality of Service

Pada implementasinya, Quality of Service memiliki beberapa parameter yang cukup penting bagi kualitas layanan yang diterima pelanggan, yaitu sebagai berikut.

1. *Delay*

Delay adalah waktu yang dibutuhkan oleh sebuah paket data terhitung dari saat pengiriman oleh *transmitter* sampai saat diterima oleh *receiver*. Beberapa jenis *delay* diantaranya adalah sebagai berikut:

- *Propagation delay* (*delay* yang terjadi akibat transmisi melalui jarak antar pengirim dan penerima)
- *Serialization delay* (*delay* pada saat proses peletakan bit ke dalam *circuit*)
- *Processing delay* (*delay* yang terjadi saat proses *coding*, *compression*, *decompression* dan *decoding*)
- *Packetization delay* (*delay* yang terjadi saat proses paketisasi *digital voice sample)*
- *Queuing delay* (*delay* akibat waktu tunggu paket sampai dilayani)
- *Jitter buffer* (*delay* akibat adanya *buffer* untuk mengatasi *jitter*)

2. *Jitter*

Jitter merupakan variasi *delay* yang terjadi akibat adanya selisih waktu atau interval antar kedatangan paket di penerima. Untuk mengatasi *jitter* maka paket data yang datang dikumpulkan dulu dalam *jitter buffer* selama waktu yang telah ditentukan sampai paket dapat diterima pada sisi penerima dengan urutan yang benar.

3. *Packet loss*

Packet loss adalah banyaknya paket yang hilang selama proses transmisi ke tujuan. *Packet loss* terjadi ketika *peak load* dan *congestion* (kemacetan transmisi paket akibat padatnya *traffic* yang harus dilayani) dalam batas waktu tertentu, maka *frame* (gabungan data *payload* dan *header* yang di transmisikan) data akan dibuang sebagaimana perlakuan terhadap *frame* data pada jaringan berbasis IP.

2.4.2 Jenis-Jenis Quality Of Service

Terdapat 3 tingkat QoS yang umum dipakai, yaitu *best-effort service, integrated service* dan *differentiated service*. [18]

1. Best Effort

   Best-effort service merupakan jenis servis dimana media *server* melakukan semua usaha agar dapat mengirimkan sebuah paket ke suatu tujuan. Penggunakan *best-effort service* tidak memberikan jaminan bahwa paket dapat sampai ke tujuan yang dikehendaki. Sebuah aplikasi dapat mengirimkan data dengan besar yang bebas kapan saja tanpa harus meminta ijin atau mengirimkan pemberitahuan ke jaringan.

   Beberapa aplikasi dapat menggunakan *best-effort service*, sebagai contohnya FTP dan HTTP yang dapat mendukung *best-effort service* tanpa mengalami permasalahan. Aplikasi-aplikasi yang sensitif terhadap *network delay*, fluktuasi *bandwidth*, dan perubahan kondisi jaringan, penerapan *best-effort service* merupakan suatu tindakan yang kurang tepat.

2. Integrated Service

   Model *integrated service* menyediakan aplikasi dengan tingkat jaminan layanan melalui negosiasi parameter-parameter jaringan secara *end-to-end*. Aplikasi-aplikasi akan meminta tingkat layanan yang dibutuhkan untuk dapat beroperasi dan bergantung pada mekanisme QoS untuk menyediakan sumber daya jaringan yang dimulai sejak permulaan transmisi dari aplikasi-aplikasi tersebut.

   Aplikasi tidak akan mengirimkan trafik, sebelum menerima tanda bahwa jaringan mampu menerima beban yang akan dikirimkan aplikasi dan

juga mampu menyediakan QoS yang diminta secara *end-to-end*. Untuk itulah suatu jaringan akan melakukan suatu proses yang disebut *admission control*. *Admission control* adalah suatu mekanisme yang mencegah jaringan mengalami *over-loaded*. Jika QoS yang diminta tidak dapat disediakan, maka jaringan tidak akan mengirimkan tanda ke aplikasi agar dapat memulai untuk mengirimkan data. Aplikasi yang telah memulai pengiriman data, maka sumber daya pada jaringan yang sudah dipesan aplikasi tersebut akan terus dikelola secara *end-to-end* sampai aplikasi tersebut selesai.

3. Differentiated Service

Model terakhir dari QoS adalah model *differentiated service*. *Differentiated service* menyediakan suatu set perangkat klasifikasi dan mekanisme antrian terhadap protokol-protokol atau aplikasi-aplikasi dengan prioritas tertentu di atas jaringan yang berbeda. *Differentiated service* bergantung pada kemampuan *edge router* untuk memberikan klasifikasi terhadap paket-paket yang berbeda tipe yang melewati jaringan. Trafik jaringan dapat diklasifikasikan berdasarkan alamat jaringan, *protocol* dan *port*, *ingress interface*, atau klasifikasi lainnya selama masih didukung oleh *standard access list* atau *extended access list*.

Manajemen QoS yang efektif memerlukan kemampuan untuk menyediakan *guaranteed or contracted quality* untuk jenis servis yang dipilih.[10] Dari ketiga jenis QoS yang sering diterapkan di jaringan, metode Diffserv merupakan metode yang mampu untuk menyediakan *guaranteed quality* sehingga metode ini akan diimplementasikan pada jaringan IMS.

Diffserv model menyediakan dua buah fungsi yaitu *classification* dan *marking*. *Classification* mengidentifikasi setiap *flow* di dalam jaringan kemudian menentukan *Class of Service* (COS) tertentu. *Flow* kemudian ditandai menggunakan *Type of Service* (ToS) atau *Differentiated Service Code Point* (DSCP) sebagai IP *header*. Setelah sebuah paket data telah ditandai (*marked*), paket kemudian diolah di *Router* sesuai dengan jalur yang telah ditentukan dan prioritas yang berbeda-beda. Di *Router* sendiri terdapat beberapa teknik seperti *Dropping* (untuk low priority paket) dan *shaping* (membatasi bandwidth untuk setiap flow).

Paket yang diterima di Router akan dilihat nilai 6 bit DSCP per paket data.[19] DSCP kemudian memberi perlakuan istimewa pada paket tersebut. Perlakuan istimewa ini dinamakan *Per-Hop Behavior* (PHB). Saat ini IETF (*Internet Engineering Task Force*) mempunyai standar klasifikasi PHB yaitu,

a. Expedited Forwarding (EF)

Expedited Forrwarding merupakan kelas PHB yang memiliki prioritas paling utama karena mempunyai karakteristik *low delay*, *low loss* dan *low jitter*. [20] Karakteristik tersebut cocok untuk aplikasi *voice, video call* dan servis-servis yang *realtime*. Nilai DSCP yang direkomendasikan untuk kelas *Expedited Forwarding* (EF) adalah 101110_B atau 46 dalam desimal.

b. Assured Forwarding(AF)

Assured Forwarding merupakan kelas PHB yang memberikan jaminan kualitas layanan selama *traffic* tidak melewati *subscribed rate* yang diijinkan. Jika *traffic* melewati *subscribed rate*, maka *traffic* akan di-*drop* ketika terjadi konghesi pada jaringan. AF dibagi ke dalam empat buah kelas, dimana di tiap kelas terdapat *drop precedence* (*high, medium, low*). Kombinasi ini menghasilkan dua belas DSCP dari AF11 hingga AF 43 yang dapat dilihat pada Tabel 2.4.

Tabel 2.4 Assured Forwarding (AF) Behaviour Group [20]

	Class 1	Class 2	Class 3	Class 4
Low Drop	AF11 (DSCP 10)	AF21 (DSCP 18)	AF31 (DSCP 26)	AF41 (DSCP 34)
Med Drop	AF12 (DSCP 12)	AF22 (DSCP 20)	AF32 (DSCP 28)	AF42 (DSCP 36)
High Drop	AF13 (DSCP 14)	AF23 (DSCP 22)	AF33 (DSCP 30)	AF43 (DSCP 38)

c. Best Effort (BE)

Best Effort merupakan default PHB yang memiliki karakteristik best effort dan memiliki nilai DSCP 000000_B.

Masing-masing PHB ini dikarakteristikkan dari *resources* yang mereka miliki (seperti ukuran *buffer* dan *bandwidth*), prioritas relatif terhadap PHB lainnya atau karakteristik pengamatan yang mereka miliki (seperti *delay* dan *loss*).

Klasifikasi trafik multimedia digolongkan dalam kelas *diffserv* meliputi VoIP dan *video conferencing* yang digolongkan kelas EF, data UDP sebagai kelas AF dan data TCP (FTP) sebagai kelas BE. Dari keterangan di atas dapat dijelaskan beberapa hal yang menjadi karakteristik DiffServ, yaitu:

- *Header* pada IP termasuk DSCP menunjukkan tingkat layanan yang diinginkan.
- DSCP memetakan paket ke PHB tertentu untuk diproses oleh *router* yang kompatibel.
- PHB menyediakan tingkat layanan tertentu (seperti *bandwidth*, *queueing*, dan *dropping decisions*) yang sesuai dengan *network policy*. Misal untuk paket-paket yang sangat *sensitive* terhadap timbulnya *error*, seperti pada aplikasi keuangan, paket-paket tersebut dikodekan dengan sebuah DSCP yang mengindikasikan layanan dengan *bandwidth* tinggi dan lintasan *routing* yang bebas *error* (0-*frame-loss*). Sedangkan pada aplikasi-aplikasi seperti *email* dan *web-browsing* data dapat dikodekan dengan sebuah DSCP yang mengindikasikan layanan dengan *bandwidth* yang lebih rendah. Selanjutnya *router* akan memilih jalur yang dipergunakan dan meneruskan paket-paket tersebut sesuai dengan yang telah ditentukan oleh *network policy* dan PHB. Kelas trafik yang tertinggi akan memperoleh pelayanan yang terbaik, baik dalam hal antrian maupun *bandwidth*, sedangkan kelas trafik dibawahnya akan memperoleh layanan yang lebih rendah. Tabel 2.5 menunjukkan level prioritas berdasarkan aplikasi di dalam metode *diffserv*.

Tabel 2.5 QoS Priority Level [21]

QoS Level	*Priority Level*	*Example services*	*Session Information*
EF	1	IMS signaling	QoS Level BW capacity QoS Level BW availability PEC/PEV SFB Service BW
	2	VoIP	
	3	Video Conferencing	
AF4	4	Audio and video streaming	
AF3	5	Transactional services	
AF2	6	Web browsing	
AF1	7	Telnet	
BE	8	E-mail	
	9	Web browsing	

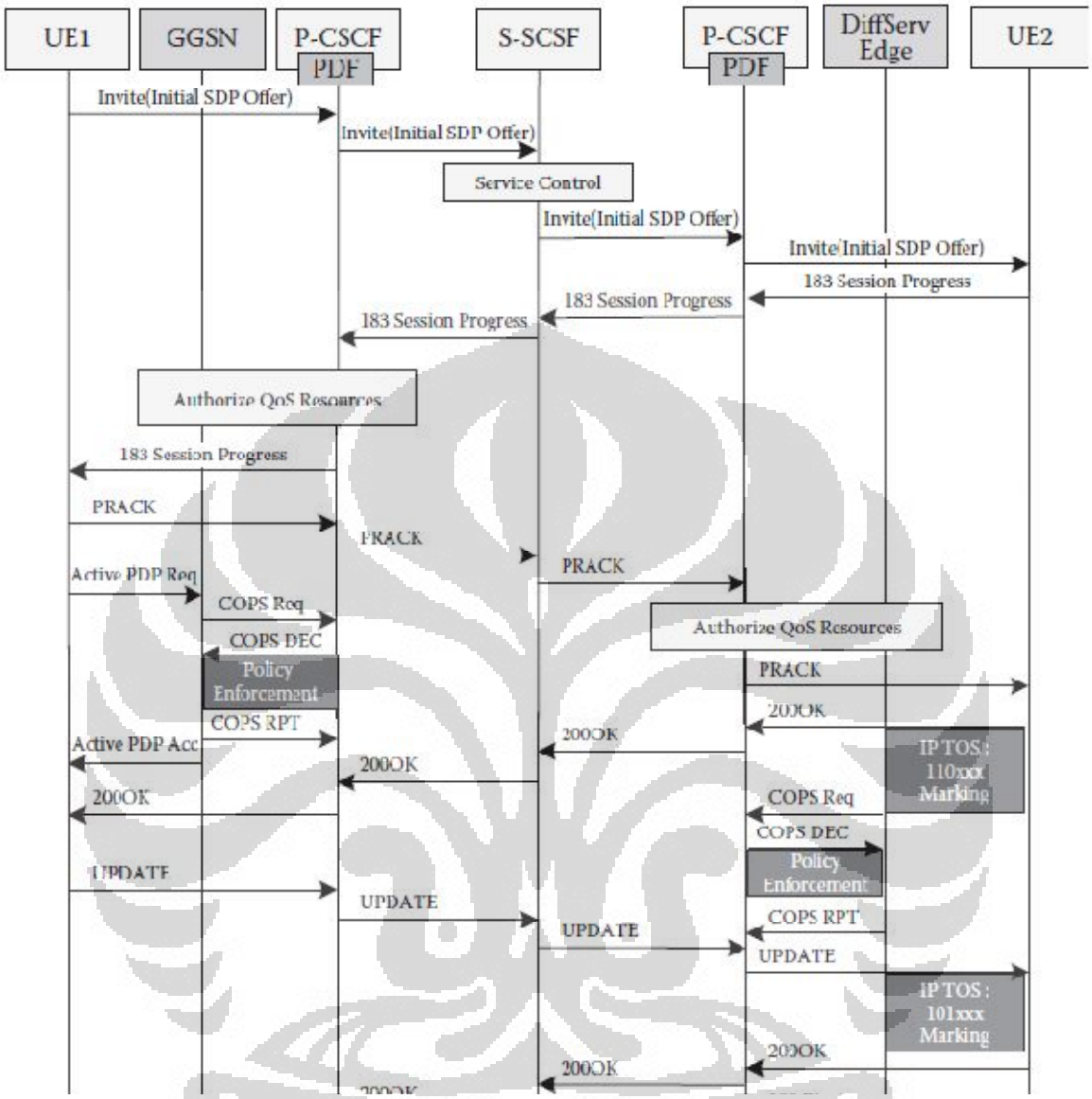
2.4.3 Quality of Service di Jaringan IMS

Pengaturan QoS di IMS memerlukan interaksi yang yang cukup kompleks antara pelanggan dengan komponen IMS.[10] Pada umumnya interaksi ini dilakukan untuk menyediakan sebuah jalur komunikasi yang sesuai dengan kebutuhan aplikasi yang ingin dijalankan. Pengaturan QoS pada jaringan IMS menggunakan tiga buah *protocol* yaitu SIP, *Diameter*, dan *Common Open Policy Service* (COPS).

Implementasi QoS di IMS melibatkan dua komponen untuk melakukan kontrol yaitu, *Policy Decision Function* (PDF) dan *Gateway GPRS Support Node* (GGSN) yang dapat digantikan dengan sebuah *router*.

2.4.4 Manajemen Arsitektur Quality of Service

Jaringan IMS yang mampu menyediakan berbagai layanan kepada *pelanggan* memiliki pengaturan yang berbeda-beda sesuai dengan servis layanan yang ingin digunakan. Pelanggan akan meminta kebutuhan QoS yang diinginkan secara *real time* ke *host* IMS. Gambar 2.8 akan memperlihatkan alur dari negosiasi QoS di dalam IMS.

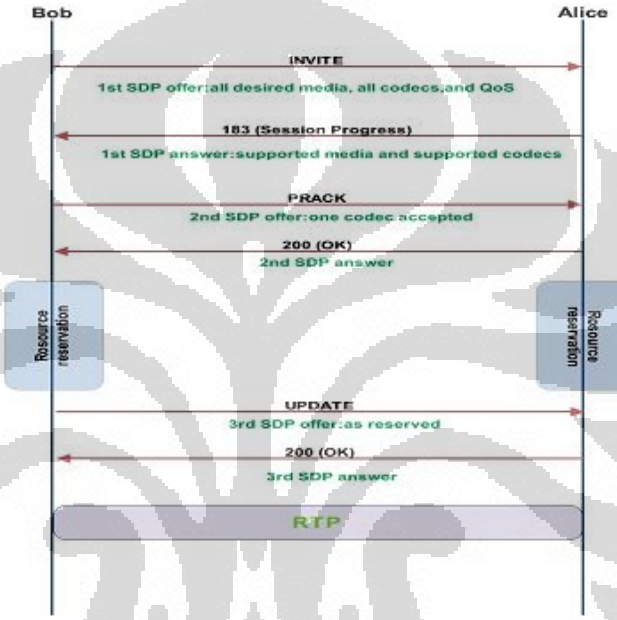


Gambar 2.8 Alur manajemen QoS dalam sebuah sesi [22]

Alur manajemen QoS memiliki beberapa *reference point* yang terlibat dalam hubungan antar titik di dalam pengaturan QoS di dalam jaringan IMS.

1. Gm Reference point

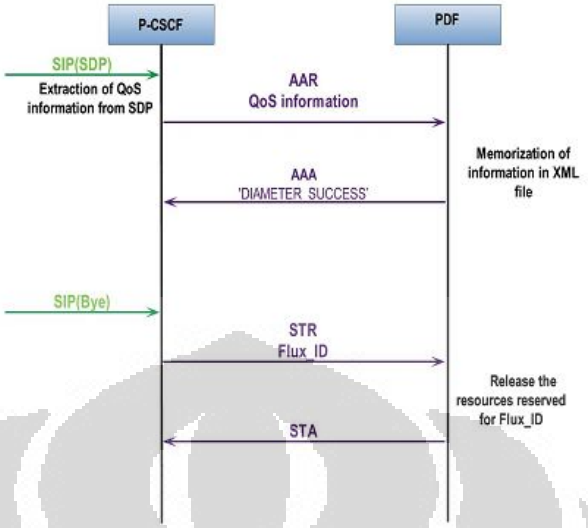
   Gm Reference point adalah sebuah *reference point* berbasis SIP antara *pelanggan* dengan P-CSCF. Fungsi *Gm reference point* adalah sebagai proses registrasi pada *session control* di IMS yang dapat dilihat pada Gambar 2.9.



Gambar 2.9 Negosiasi SDP parameter antara 2 pelanggan [10]

2. Gq Reference point

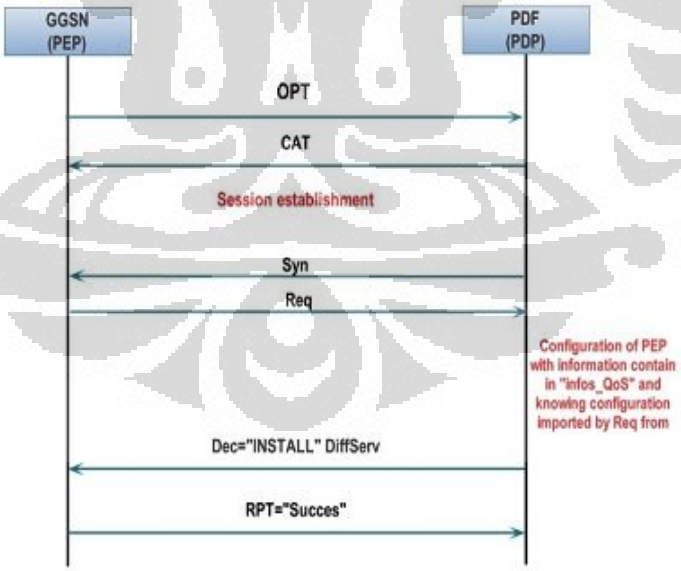
   Gq reference point merupakan kanal yang digunakan untuk membawa informasi penting dari *pelanggan* dan servis yang diminta oleh *pelanggan* untuk memulai komunikasi antara P-CSCF dengan *Policy Decision Function* (PDF) pada awal sebuah sesi hal ini ditunjukkan oleh Gambar 2.10.



Gambar 2.10 Hubungan antara P-CSCF dengan PDF [10]

3. Go Reference point

   Go reference point mendefinisikan informasi yang dikirimkan antara PDF dengan *policy enforcement node* baik itu GGSN ataupun *edge router*. Gambar 2.11 menunjukkan peran dari *protocol* COPS.



Gambar 2.11 Hubungan antara PDF dengan GGSN [10]

BAB 3
IMPLEMENTASI QUALITY OF SERVICE PADA JARINGAN OPEN IMS CORE

Berdasarkan teori yang dibahas pada bab 2, implementasi *Quality of Service* dapat dijalankan di jaringan *OpenIMSCore* dengan menggunakan sebuah server tambahan yaitu PCRF dan PCEF sebagai *policy control*, namun karena program dari UCT PCRF telah dikomersialkan oleh pihak *University of Cape Town*, maka instalasi PCRF tidak dapat dilakukan. Implementasi QoS dilakukan dengan menggantikan PCRF dan PCEF dengan sebuah *open source router* yang mampu memberikan prioritas terhadap paket berdasarkan IP *address* dari pengguna aplikasi.

Percobaan implementasi *Quality of Service* yang dilakukan pada penelitian ini adalah dengan menjalankan dua buah aplikasi yaitu *Voice over IP* (VoIP) dan *Video on Demand* (VoD) yang berjalan secara bersamaan di jaringan *Open IMS Core*. Hal yang akan diamati adalah perbandingan parameter-parameter *Quality of Service* seperti *delay, jitter*, dan *packet loss* yang terjadi pada jaringan, sebelum dan sesudah diterapkan *policy control* di dalam jaringan *Open IMS Core*. *Policy control* yang diterapkan didalam penelitian ini adalah pemberian prioritas yang lebih tinggi terhadap data suara pada aplikasi *Voice over IP* dibandingkan dengan data video dari aplikasi *Video on Demand*.

Pada percobaan ini, terdapat beberapa komponen penting yaitu, sebuah server *Open IMS Core*, tiga buah laptop sebagai IMS *client*, *Application Server*, *Media Server*, dan dua buah router. Secara garis besar, skenario percobaan yang akan dijalankan adalah sebagai berikut.

1. Seorang *client* dari jaringan berbeda sedang menggunakan layanan *Video on Demand* dari server *Open IMS Core*, kemudian dua orang *client* berbeda ingin menggunakan layanan VoIP di jaringan *Open IMS Core*.
2. Menerapkan *policy control* di *open source router vyatta*.
3. Menggunakan *wireshark* sebagai *packet analyzer* untuk melihat komunikasi di dalam jaringan serta melihat nilai dari parameter-parameter QoS di jaringan, baik sebelum dan sesudah diterapkannya *policy control* di jaringan.

3.1 Instalasi OpenIMSCore

Pada penelitian ini, digunakan *OpenIMSCore* yang bersifat *open source* yang terdiri dari *Home Subscriber Server* (HSS) dan *Call Session Control Function* (CSCF). OpenIMSCore berfungsi untuk menerima, memeriksa apakah *client* tersebut sudah terdaftar dan mempunyai hak untuk mengakses layanan yang dia minta atau tidak, meneruskan request dari *client* ke *aplication server,* dan selanjutnya mengembalikan jawaban dari *application server* ke *client*. Instalasi dilakukan melalui *terminal ubuntu* dengan langkah-langkah sebagai berikut:

1. Update Repositori *Ubuntu*

   Update repositori diperlukan agar, packages *ubuntu* yang digunakan di *server* merupakan *packages* yang terbaru. Hal ini dilakukan karena setelah instalasi *ubuntu*, banyak *packages* yang belum terinstall.

2. Download dan Install Subversion Package dan packages lain yang dibutuhkan

   Subversion package merupakan salah satu *package* yang diperlukan untuk meng-install *OpenIMSCore*. Selain *Subversion, packages* lain yang dibutuhkan untuk instalasi *OpenIMSCore* adalah sebagai berikut : *libxml2-dev, libmysqlclient-dev, bind9, bison, flex, mysql-server, ant, make, sun-java6-jre, sun-java6-jdk, g++, ipsec-tools, libcurl4-openssl-dev*

3. Membuat folder OpenIMSCore

   Membuat folder *OpenIMSCore* di dalam *folder* /opt/. kemudian membuat *folder ser_ims* sebagai direktori instalasi CSCF dan *folder* FHoSS sebagai direktori instalasi HSS.

4. Download source FHoSS dan CSCF

   Untuk mendownload file HSS dan CSCF jalankan perintah berikut di *terminal* namun sebelumnya masuk ke direktori /opt/OpenIMSCore/.

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

 # svn checkout http://svn.berlios.de/svnroot/repos/openimscore/FHoSS/trunk FHoSS

5. Instalasi CSCF

   Instalasi CSCF di *folder ser_ims* dilakukan dengan perintah "*make install-libs all*"

6. Install FHoSS

   Instalasi HSS dengan meng-*compile* data yang telah di-*download* dengan perintah "*ant compile deploy*"

7. Setup database Mysql

   # 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/pelanggandata.sql

8. Copy configuration file ke folder OpenIMSCore

   Copy file CSCF ke *folder OpenIMSCore*

9. Menambahkan isi file named.conf

   # sudo gedit /etc/bind/named.conf

   Kemudian tambahkan

   zone "open-ims.test" {

   type master;

   file "/etc/bind/open-ims.dnszone";

   };

10. Copy file open-ims.dnszone ke direktori /etc/bind/

11. Edit file /etc/resolv.conf

    Mengganti *domain* dan *search* menjadi *open-ims.test* dan *nameserver* menjadi *localhost*.

12. Edit file /etc/hosts

    # sudo gedit /etc/hosts

    127.0.0.1 *localhost*

    127.0.1.1 *netlab-desktop*

127.0.0.1 *open-ims.test mobicents.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*

13. Restart bind server
14. Pengecekan konfigurasi

    #dig open-ims.test

    Pada bagian *answer* harus tertulis 127.0.0.1
15. Menjalankan HSS dan CSCF pada tab terminal yang berbeda-beda:
16. Daftar element dan port number

Element	Port Number
P-CSCF	4060
I-CSCF	5060
S-CSCF	6060
Diameter	3868, 3869, 3870

17. Merubah OpenIMSCore ke spesifik IP

    Untuk merubah *OpenIMSCore* dari *localhost* ke spesifik IP maka kita harus merubah isi *file resolv.conf, hosts*, dan *open-ims.dnszone*, dengan perintah "*sudo gedit*" untuk masing-masing *file* tersebut kemudian mengganti semua IP *localhost* 127.0.0.1 menjadi spesifik IP yang digunakan. Pada skripsi ini digunakan IP 192.168.102.229

    Kemudian ulangi langkah untuk me-*restart* bind server dan mengeset kembali *mysql database*, lalu menjalankan HSS dan CSCF.

18. Akses HSS via web browser

    web browser ketikkan alamat 192.168.102.229:8080

    Kemudian masukkan *username* "*hssAdmin*" dan *password* "*hss*"

3.2 Account IMS client

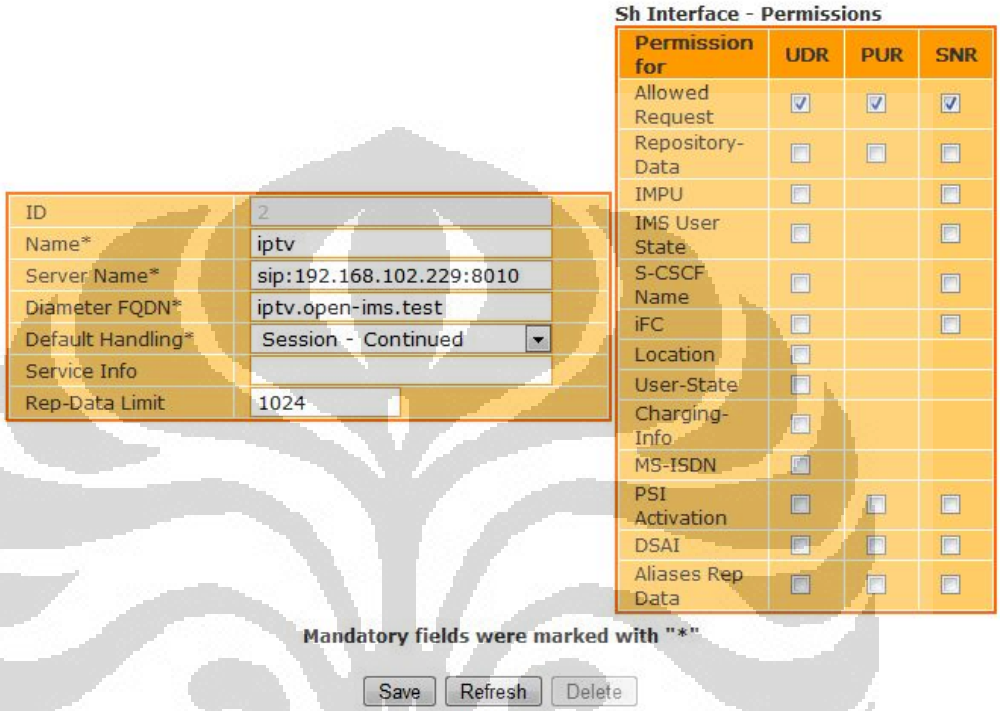
Pada awal instalasi OpenIMSCore, terdapat dua buah *default user* yang dapat digunakan, yaitu Alice dan Bob. Pembuatan *user* baru dilakukan dengan mengakses HSS via *web browser*. Pada halaman utama FHoSS, pilih *USER IDENTITIES*. Kemudian pada bagian IM *Subscription*, masukkan nama dari *user* yang dibuat dan memilih *Capabilities Set* dan *Preferred* S-CSCF. Setelah mengisi informasi di IM Subscription, bagian *Create & Bind new IMPI* akan meminta informasi mengenai *identity*, *secret key*, dan memilih jenis *Authentication Schemes* yang ingin digunakan *user*. Informasi-informasi yang disimpan ke dalam HSS merupakan *Private User Identity*.

Selain *Private User Identity*, seorang *user* juga membutuhkan *Public User Identity*. Informasi yang diperlukan adalah informasi mengenai *service profile*, *charging info set*, IMPU *type* dan *display name*. *Public User identity* merupakan informasi yang digunakan saat seorang user berada di *visited network*, sehingga pada bagian ini juga diperlukan penambahan informasi mengenai nama dari *Visited Network*. Setelah itu *account* telah dapat digunakan untuk mengakses jaringan IMS melalui ims *client* baik di *Operating System ubuntu* maupun *windows*.

3.3 Application Server

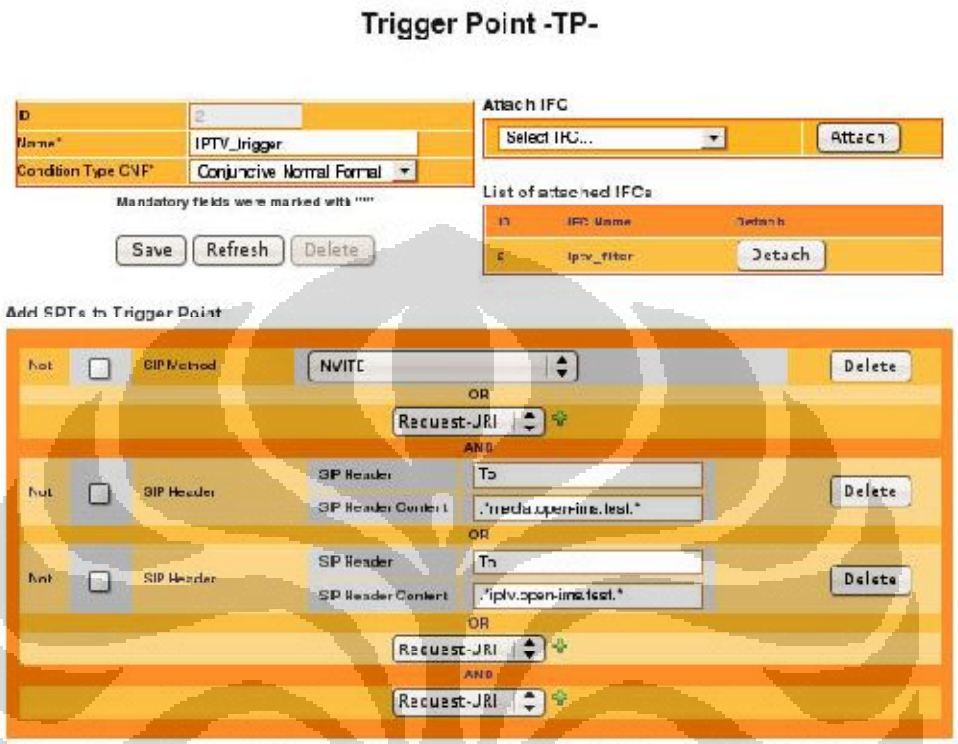
Application Server berfungsi untuk me-*routing*-kan *request* dari *client* ke alamat RTSP dari *media server* yang dituju. Oleh karena itu *aplication server* yang digunakan pada VoD disebut *indirection application server*. Langkah-langkah yang dilakukan untuk menginstall sebuah *application server* diawali dengan men-*download debian package application server* dari https://developer.berlios.de/project/showfiles.php?group_id=7844. Lalu install *debian package* tersebut.

Konfigurasi terhadap *application server* melalui FhoSS pada Gambar 3.1 dilakukan dengan memberi nama terhadap *application server* yang dibuat serta mengisi bagian *Server Name*, *Diameter* FQDN dan memberi *permission* kepada *interface* Sh.



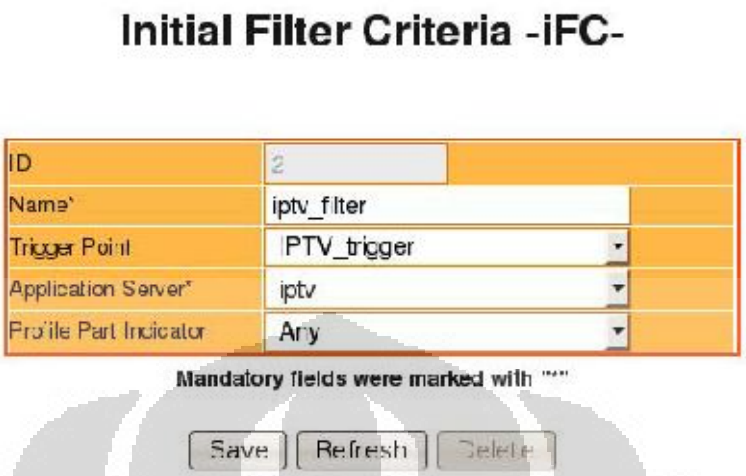
Gambar 3.1 Application Server IPTV

Gambar 3.2 menunukkan proses pembuatan *trigger point* untuk *application server*.



Gambar 3.2 Trigger Point IPTV

Application server dan ***trigger point*** dihubungkan dengan sebuah *initial filter criteria*. *Initial Filter Criteria (iFC)* berfungsi untuk membaca setiap permintaan terhadap layanan IPTV yang langsung diarahkan ke server IPTV (*UCT Indirection Application Server*) untuk diproses lebih lanjut. Gambar 3.3 menunjukkan sebuah *Initial Filter Criteria*.

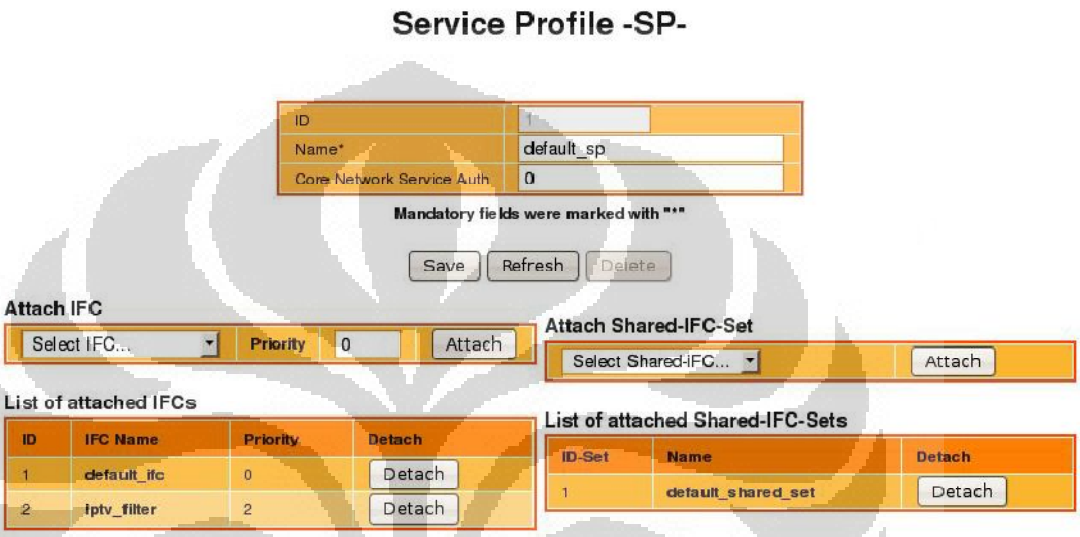


Gambar 3.3 Initial Filter Criteria

Initial Filter Criteria (iFC) yang telah dibuat, perlu untuk ditambahkan ke dalam *Shared iFC Sets*. Proses ini ditunjukkan melalui Gambar 3.4.

Gambar 3.4 Shared iFC Sets

Melalui Shared iFC, filter-filter yang telah dibuat sebelumnya seperti *iptv_filter* dimasukkan ke dalam "*List of attached IFCs*" sehingga dapat diakses oleh semua user. *Initial Filter Criteria* (*i*FC) dan *Shared i*FC dibaca di dalam sebuah *service Profile*. Gambar 3.5 menunjukkan *i*FC dan *shared i*FC di dalam sebuah *service profile*.



Gambar 3.5 Service Profile

Key value file ini merupakan *file* yang menghubungkan *request* dari *client* dengan alamat RTSP dari *media server* tujuan sehingga *key value file* harus sesuai dengan alamat *media server*. *Key value file* berisi informasi mengenai alamat RTSP dari *media server*. *Key value file* berada di dalam direktori */usr/share/uctiptv_advanced/* dan dapat diubah dengan perintah *sudo gedit.*

```
<key-value_pairs>
   <key-value_pair>
         <key>channel1</key>
         <value>rtsp://media_server_address.domain:8000/request
         ed_channel</value>
   </key-value_pair>
</key-value_pairs>
```

Setelah mengubah *media server address*, *Application Server* dapat dijalankan dengan *command* di terminal.

uctiptv_as /usr/share/uctiptv_advanced/key_value_file

Gambar 3.6 menunjukkan tampilan dari *Application Server*.

Gambar 3.6 Application Server

3.4 RTSP Media Server

Fungsi dari media server adalah menyediakan *content* bagi *client*. *Media server* dikhususkan untuk menjalankan layanan *Video on Demand* (VoD). Cara kerja VoD adalah dengan menyediakan beberapa *video* yang telah siap tayang pada *media server*. Ketika datang permintaan pada akan suatu *video*, maka secara otomatis *server* akan mengirim langsung menuju *client* yang meminta layanan VoD melalui UCT IMS *Client*.

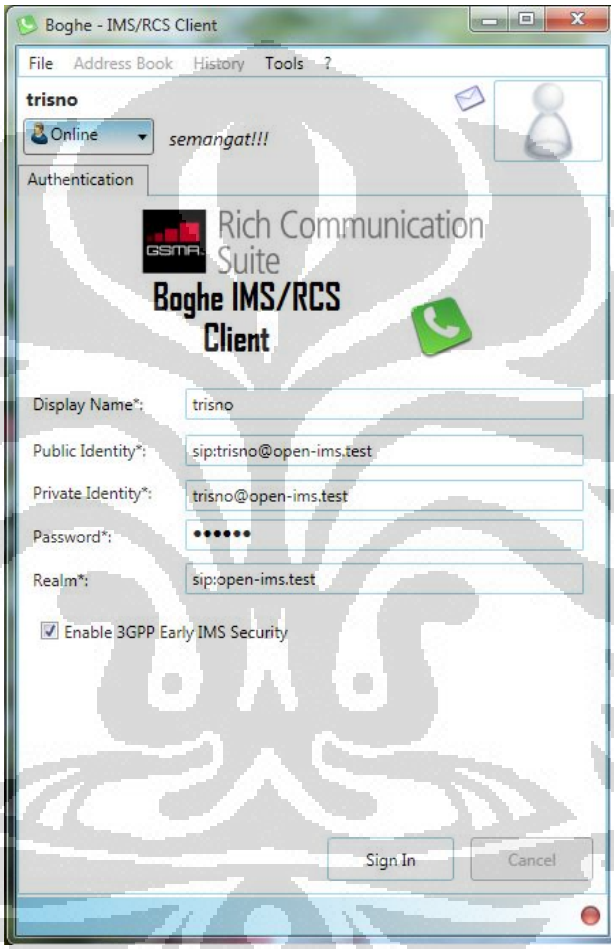
Program yang digunakan sebagai *media server* adalah VLC *media player* yang berjalan di *Operating System Ubuntu* 8.04. VLC *media player* dapat bekerja sebagai *media server* setelah dilakukan konfigurasi terhadap file video yang akan di-*streaming*-kan. *File* konfigurasi yang dibuat merupakan *file* dengan format vlm. *File* konfigurasi terdiri dari informasi mengenai nama *shortcut* video, format video dan direktori dari video yang dimasukkan ke dalam *media server*.

Pada skripsi ini, video yang digunakan di-*streaming*-kan dari alamat RTSP 192.168.102.229 port 8000. *Media server* dijalankan dengan perintah di terminal:

 # vlc -vvv --ttl 12 --intf telnet --rtsp-host 192.168.102.229:8000 --vlm-conf vod.vlm

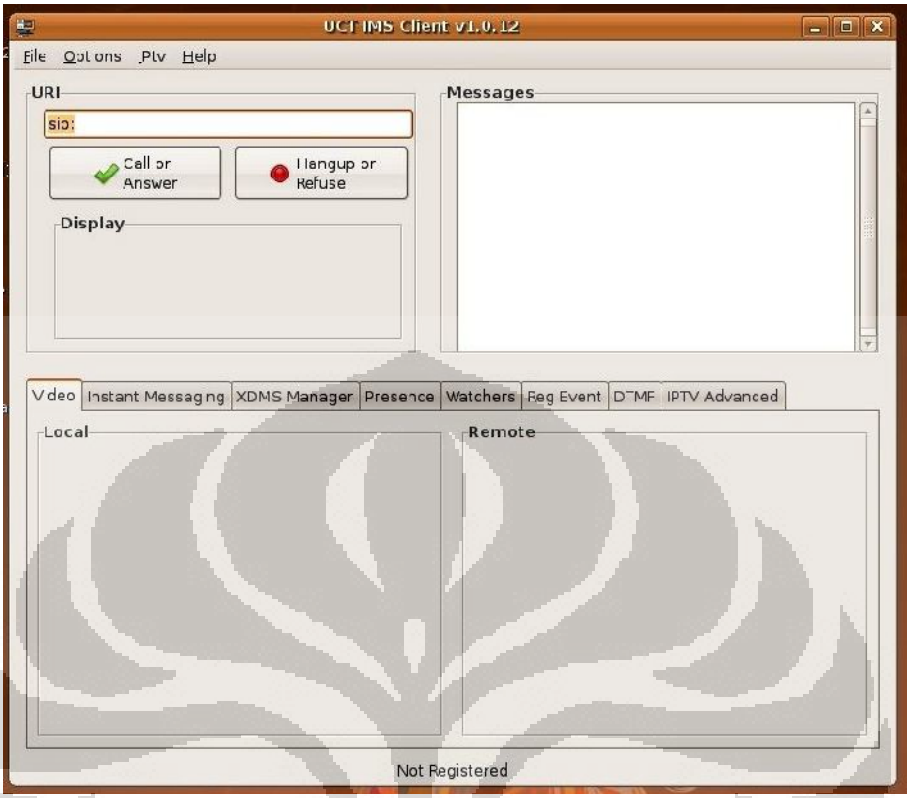
3.5 IMS Client

IMS *Client* dapat dijalankan baik di windows maupun di *operating system ubuntu*. Pada OS Windows dapat digunakan program Boghe IMS *Client* yang dapat menjalankan aplikasi *instant messaging*, VoIP, dan *video call*. Boghe IMS *Client* dapat di-*download* dari http://boghe-ims-client.software.informer.com/ Gambar 3.11 menunjukkan tampilan dari BogheIMS *Client.*



Gambar 3.7 Boghe IMS Client

IMS *Client* di OS *Ubuntu* merupakan IMS Client yang dikembangkan oleh *University of Cape Town*, yaitu *uctimsclient*. *Uctimsclient* selain mampu menjalankan aplikasi IM, VoIP, video call, juga mampu menjalankan aplikasi *Video On Demand*. Untuk meng-*install* UCT IMS *Client* ini kita dapat men-*download debian package*-nya dari http://uctimsclient.berlios.de/. Setelah berhasil melakukan instalasi, UCT IMS *Client* dapat dijalankan melalui perintah di terminal ubuntu. Gambar 3.12 menampilkan halaman muka dari *uctimsclient.*



Gambar 3.8 UCT IMS Client

3.6 Vyatta Router

Vyatta merupakan sebuah *Operating System* (OS) yang bersifat *open source* dan berfungsi sebagai *router* yang dalam sebuah jaringan. Penggunaan *vyatta* dijalankan pada media *command line interface* (CLI). Melalui CLI inilah, dapat dilakukan konfigurasi mengenai pengaturan yang ingin diterapkan ke jaringan yang terhubung dengan *router* baik terhubung ke *server* ataupun *client.* Konfigurasi yang perlu dilakukan adalah pengaturan mengenai interface router dan *routing protocol* yang digunakan. Pada penelitian ini, akan ditambahkan beberapa fitur yang mampu dijalankan di *router vyatta* untuk mengimplementasikan penerapan *Quality of Service* pada jaringan *OpenIMSCore*, seperti *traffic shaper* dan *Bandwidth management*.

Vyatta dapat dijalankan melalui sebuah *live CD*, kemudian memasukkan *username* :*vyatta* dan *password* :*vyatta* dan mengetikkan perintah *install-system*. Lalu ikuti petunjuk yang tampak pada monitor untuk menyelesaikan proses

instalasi. Setelah itu, keluarkan *live CD* dan *reboot computer* yang digunakan sebagai *router*.

Pada percobaan ini, digunakan dua buah *router* vyatta untuk menerapkan QoS di jaringan *OpenIMSCore* sebagai pengganti *edge router* yang ada di jaringan IMS yang sebenarnya. Kedua buah router dihubungkan dengan menggunakan *Routing Information Protocol* (RIP).

Konfigurasi node yang digunakan untuk menerapkan QoS di penelitian ini adalah sebagai berikut.

~#vyatta@vyatta set *traffic-policy shaper* final (nama node yang ingin dibuat)

~#vyatta@vyatta set *traffic-policy shaper* final *bandwidth* xxxkbit

~#vyatta@vyatta set *traffic-policy shaper* final *class 10 match* voip ip *destination address* xx.xx.xx.xx/24

~#vyatta@vyatta set *traffic-policy shaper* final *default bandwidth* 100%

~#vyatta@vyatta *commit*

~#vyatta@vyatta *save*

Setelah membuat konfigurasi *node*, langkah selanjutnya yang dilakukan adalah menerapkan konfigurasi tersebut ke *interface* yang di miliki *router*. Perintah yang dijalankan untuk menerapkan konfigurasi ke *interface* tertentu adalah sebagai berikut.

~#vyatta@vyatta *set interface Ethernet eth0 traffic-policy out* final

~#vyatta@vyatta *commit*

~#vyatta@vyatta *save*

3.7 Wireshark

Wireshark adalah salah satu program tool yang berfungsi sebagai *network analyzer*. *Wireshark* merupakan program yang menggunakan *Graphical Pelanggan Interface* (GUI) atau tampilan grafis yang membantu pelanggan untuk mengamati paket di jaringan. *Wireshark* mampu menangkap semua paket yang melewati jaringan. Paket yang akan diamati pada penelitian ini adalah paket RTP yang bersifat *real time*. Setelah meng-*filter* paket-paket RTP, *Wireshark* dapat melakukan pengukuran parameter-parameter QoS seperti *delay, jitter* dan *packet*

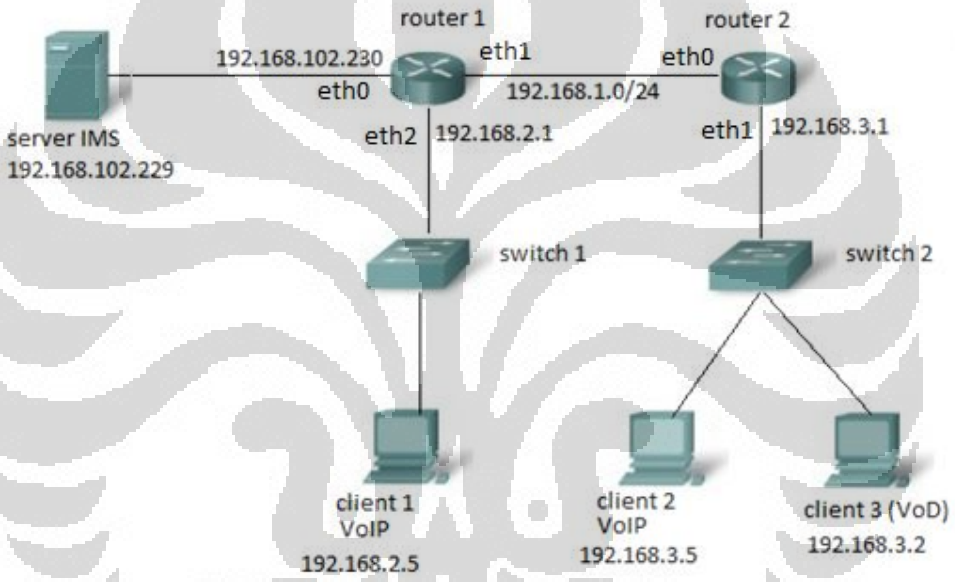
loss. *Wireshark* dapat dijalankan baik di OS *ubuntu* maupun *windows*. Untuk instalasi di *ubuntu*, jalankan perintah

#sudo apt-get install wireshark

sedangkan untuk OS *windows, wireshark* dapat di *download* di situs www.wireshark.org

3.8 Topologi Jaringan

Penelitian implementasi QoS dengan prioritas paket di dalam jaringan IMS akan digunakan topologi jaringan seperti Gambar 3.13.



Gambar 3.9 Topologi jaringan

Pada jaringan ini digunakan satu buah *server* Open IMS Core, dua buah *router vyatta*, dua buah *switch* dan tiga buah PC sebagai *client* dengan IP, *netmask* dan *default gateway* sesuai dengan gambar di atas.

Pada jaringan ini akan dilakukan ujicoba untuk mengamati performansi VoD dengan ada atau tidaknya implementasi QoS di jaringan dengan menjalankan skenario sebagai berikut.

1. Menjalankan aplikasi VoD pada client 3 dengan implementasi QoS di jaringan.
2. Menjalankan VoIP pada client 1 dan 2 pada saat client 3 sedang menjalankan aplikasi *Video on Demand* dengan implementasi QoS di jaringan.

3. Menjalankan aplikasi VoD pada client 3 tanpa implementasi QoS di jaringan.
4. Menjalankan VoIP pada client 1 dan 2 pada saat client 3 sedang menjalankan aplikasi Video on Demand tanpa implementasi QoS di jaringan.

Ujicoba dilakukan pada variasi *bandwidth* 384 kbps, 512 kbps, dan 640 kbps. Setelah itu, dilakukan ujicoba pada *bandwidth* 16 kbps, 32 kbps dan 56 kbps untuk melihat ada atau tidaknya pengaruh implementasi QoS di jaringan terhadap performansi VoIP.

BAB 4
HASIL PERCOBAAN DAN ANALISIS DATA

Bab 4 akan dibahas mengenai pengukuran parameter-parameter QoS dengan beberapa skenario untuk melihat perubahan performasi layanan multimedia. Secara garis besar pengukuran dilakukan sebanyak dua kali yaitu pengukuran untuk mengamati perubahan performansi aplikasi *Video On Demand* dengan variasi *bandwidth* 384 kbps, 512 kbps dan 640 kbps serta pengukuran untuk mengamati perubahan performansi aplikasi VoIP dengan variasi *bandwidth* 16 kbps, 32 kbps, dan 56 kbps. Dengan melakukan perubahan variasi pada *bandwidth* total jaringan diharapkan dapat terlihat dengan lebih jelas perbedaan kualitas layanan setelah dilakukan prioritas paket di dalam jaringan.

Pengukuran dilakukan untuk mengamati perubahan kuantitatif dari parameter-parameter QoS yaitu *delay*, *jitter,* dan *packet loss* dengan menggunakan program *packet tracer wireshark* untuk merekam paket-paket yang ada di jaringan kemudian melakukan perhitungan terhadap paket-paket tersebut. Pada pengukuran yang dilakukan pada penelitian ini hanya difokuskan untuk mengamati paket dengan protocol *Real Time Protocol* (RTP) karena kedua jenis aplikasi yang digunakan merupakan aplikasi yang bersifat *real-time*.

Pada penelitian ini digunakan video dengan *codec* MPEG-II dengan resolusi 320x240 *pixel* dan *datarate* 514 kbps. Sedangkan untuk *voice* digunakan *codec* GSM 6.10 sesuai dengan *default codec* dari program Boghe IMS *Client* yang digunakan dalam penelitian ini. *Codec* GSM 6.10 memiliki karakteristik *bitrate* 13 kbps dengan 160 sampel dan sampel rate 8 Khz. *Encoder* memproses *block* suara sebesar 20 ms yang tiap *block* berisi 260 *bit* sehingga dihasilkan kecepatan 13 Kbps (260 *bit*s/20 ms=13.000 *bit*s/s= 13k*bit*s/s).

4.1 Pengambilan dan Pengolahan Data Performansi Video on Demand

Pengukuran performansi aplikasi *Video on Demand* dilakukan pada tiga variasi *bandwidth* yaitu 384 kbps, 512 kbps, dan 640 kbps. Untuk setiap variasi *bandwidth* akan dijalankan empat kali pengukuran yaitu

1. Pengukuran performansi VoD dengan implementasi QoS sebelum disela oleh VoIP.
2. Pengukuran performansi VoD dengan implementasi QoS setelah disela oleh VoIP.
3. Pengukuran performansi VoD tanpa implementasi QoS sebelum disela oleh VoIP.
4. Pengukuran performansi VoD tanpa implementasi QoS setelah disela oleh VoIP.

Setiap Pengukuran akan diulang sebanyak lima kali kemudian dihitung nilai rata-rata untuk masing-masing parameter QoS. Tabel 4.1 menunjukkan hasil pengambilan data untuk *bandwidth* 384 kbps dengan implementasi QoS dan Tabel 4.2 menunjukkan data pengukuran VoD tanpa implementasi QoS.

Tabel 4.1 Pengukuran VoD dengan Implementasi QoS

No.	QoS					
	belum diganggu voip			setelah diganggu voip		
	delay (ms)	jitter (ms)	packet loss (%)	delay (ms)	jitter (ms)	packet loss (%)
1	39.95	26.78	34.56	42.96	47.63	39.1
2	39.89	27.61	34.6	42.84	48.44	38.9
3	39.89	26.77	34.5	42.76	46.4	39.1
4	39.92	29.61	34.5	42.9	48.33	39
5	40.5	29.27	35.4	42.86	47.015	39.1
mean	**40.03**	**28.008**	**34.712**	**42.864**	**47.563**	**39.04**

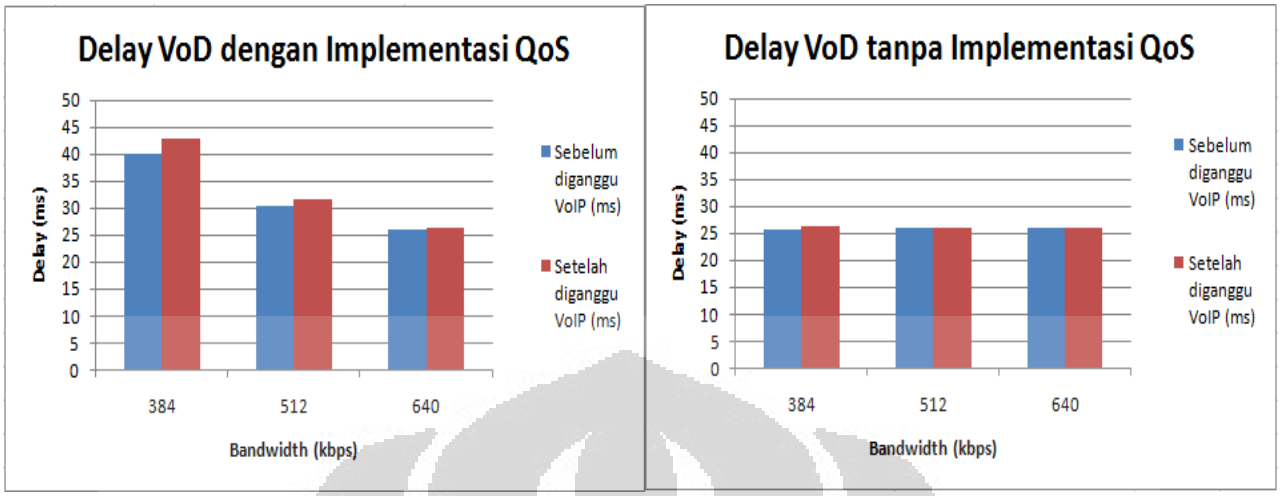
Tabel 4.2 Pengukuran VoD tanpa Implementasi QoS

No.	Tanpa QoS					
	belum diganggu voip			setelah diganggu voip		
	delay (ms)	jitter (ms)	packet loss (%)	delay (ms)	jitter (ms)	packet loss (%)
1	25.71	17.11	0	25.93	19.08	0
2	26.44	17.68	0	26.53	19.09	0
3	26.07	16.8	0	25.79	18.91	0
4	24.89	16.94	0	26.77	19.13	0
5	25.55	17.23	0	25.94	18.59	0
mean	**25.732**	**17.152**	**0**	**26.192**	**18.96**	**0**

Hal yang serupa juga dilakukan pada percobaan dengan *bandwidth* 512 kbps dan 640 kbps, kemudian membuat tabel seperti Tabel 4.1 dan 4.2 yang dapat dilihat di lampiran. Dari tabel-tabel tersebut akan dibuat tabel baru yang semakin memperjelas perubahan *delay, jitter*, dan *packet loss* VoD sebelum dan sesudah dilakukan implementasi QoS di jaringan.

Tabel 4.3 *Delay* pada Aplikasi VoD

Bandwidth	QoS		tanpa QoS	
	Sebelum diganggu VoIP (ms)	Setelah diganggu VoIP (ms)	Sebelum diganggu VoIP (ms)	Setelah diganggu VoIP (ms)
384	40.03	42.864	25.732	26.192
512	30.40	31.574	26.042	26.096
640	26.09	26.26	26.09	26.10

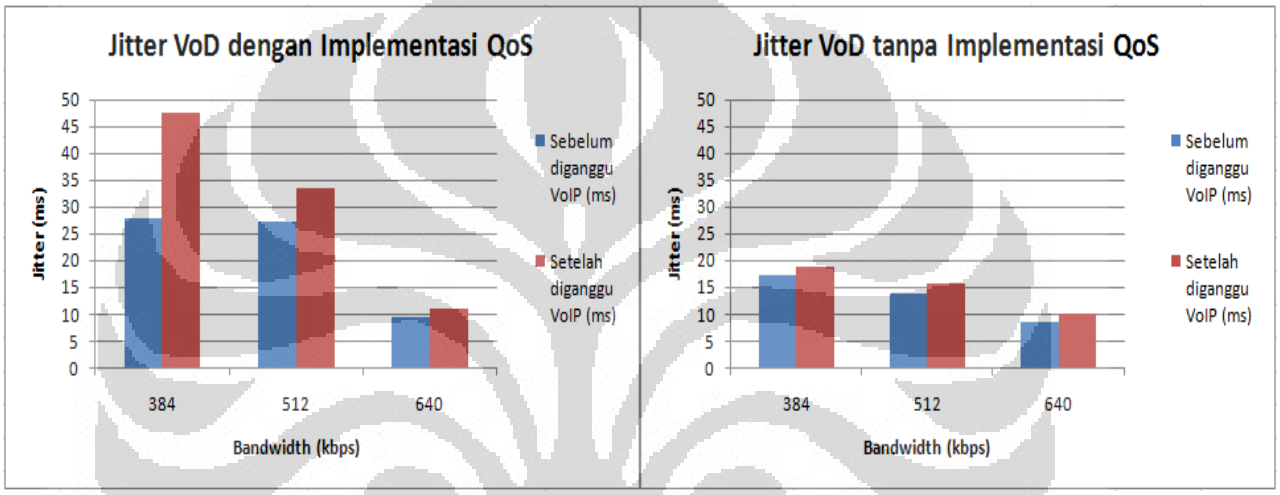


Gambar 4.1 perbandingan *delay* VoD

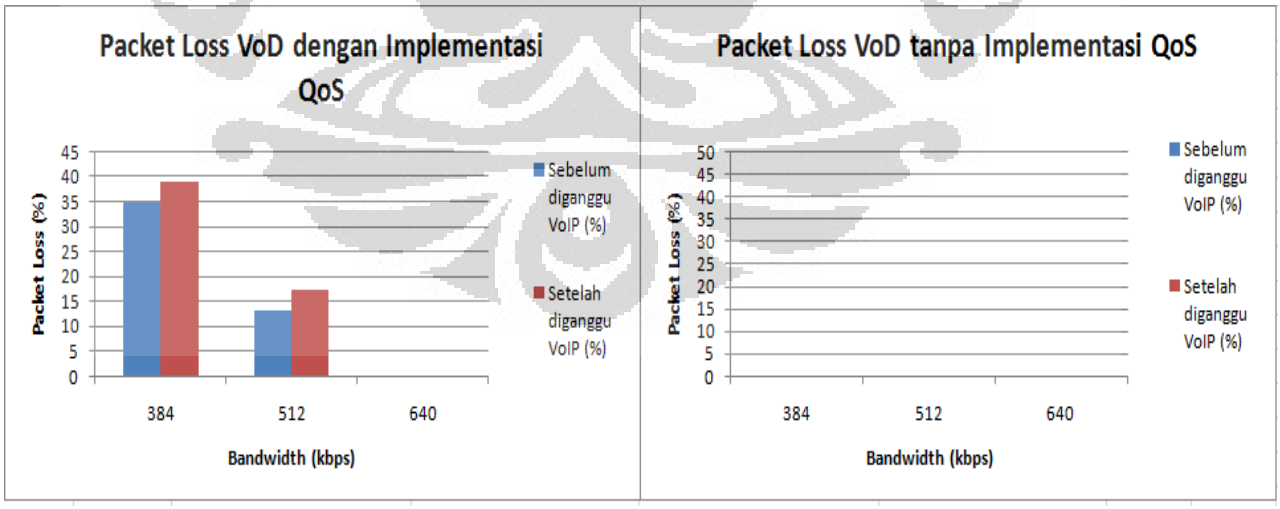
Dari Tabel 4.3 dan Gambar 4.1, terlihat perubahan *delay* yang signifikan pada saat dilakukan implementasi QoS dan tanpa implementasi QoS di jaringan. Perbedaan *delay* terlihat pada *bandwidth* 384 kbps dan 512 kbps, sedangkan pada bandwidth 640 kbps *delay* dari aplikasi VoD tidak mengalami perubahan karena telah berada dalam kondisi saturasi dimana penambahan *bandwidth* tidak menurunkan nilai *delay*. Hal yang dapat diamati dari Gambar 4.1 adalah pada saat dilakukan implementasi QoS di jaringan, *delay* VoD mengalami peningkatan sebesar 2 ms pada saat diganggu oleh *traffic* VoIP yang juga menggunakan jalur di jaringan. Implementasi QoS yang dilakukan mengkategorikan VoD sebagai *default traffic* dan VoIP sebagai *traffic* yang lebih diprioritaskan, sehingga *traffic* VoD akan terganggu. Perbedaan yang terjadi cukup kecil yaitu 2 ms karena *traffic* VoIP tidak membutuhkan *bandwidth* yang besar seperti *traffic* VoD.

Gambar 4.1 menunjukkan bahwa pada saat tidak dilakukan implementasi QoS di jaringan, *traffic* VoD tidak mendapatkan perlakuan berbeda dengan *traffic* VoIP sehingga mendapatkan nilai *delay* yang konstan berada di angka 26 ms. Lain halnya pada saat dilakukan implementasi QoS di jaringan, *traffic* VoIP akan diprioritaskan dibandingkan dengan *traffic* sehingga *delay* VoD bernilai lebih besar.

Trend yang sama juga ditunjukkan dari Gambar 4.2 yang menunjukkan perbandingan *jitter* pada saat ada implementasi QoS dengan tidak ada implementasi QoS. Perbandingan nilai *jitter* untuk penggunaan *bandwidth* 384 kbps terdapat perbedaan sebesar 29 ms pada saat VoD diganggu oleh VoIP dan untuk bandwidth 512 kbps terjadi perbedaan nilai *jitter* sebesar 18 ms sedangkan pada *bandwidth* 640 nilai *jitter* tidak mengalami banyak perubahan baik pada saat dilakukan implementasi QoS di jaringan ataupun tanpa implementasi QoS.



Gambar 4.2 Perbandingan *jitter* VoD



Gambar 4.3 Perbandingan *packet loss* VoD

Gambar 4.3 yang menunjukkan perbandingan *packet loss* mempertegas pernyataan yang telah disebutkan sebelumnya yaitu, pada saat tidak dilakukan implementasi QoS di jaringan, *traffic* VoD tidak mendapatkan perlakuan berbeda dengan *traffic* VoIP sehingga VoD tidak mengalami *packet loss*. Lain halnya pada saat dilakukan implementasi QoS di jaringan, *traffic* VoIP akan diprioritaskan dibandingkan dengan *traffic* sehingga terjadi *packet loss* pada aplikasi VoD. Penggunaan bandwidth 640 kbps tidak mengalami *packet loss* pada saat dilakukan implementasi QoS di jaringan terjadi karena bandwidth yang tersedia telah mencukupi kebutuhan jalur dari VoD yang masuk ke jaringan.

4.2 Pengambilan dan Pengolahan Data Performansi VoIP

Pengukuran performansi aplikasi VoIP dilakukan pada tiga variasi *bandwidth* yaitu 16 kbps, 32 kbps, dan 56 kbps dan setiap variasi *bandwidth* akan dijalankan empat kali pengukuran yaitu

1. Pengukuran performansi VoIP dengan implementasi QoS saat VoIP disela oleh VoD.
2. Pengukuran performansi VoIP dengan implementasi QoS saat VoIP menyela VoD.
3. Pengukuran performansi VoIP tanpa implementasi QoS saat VoIP disela oleh VoD.
4. Pengukuran performansi VoIP tanpa implementasi QoS saat VoIP menyela VoD.

Setiap Pengukuran akan diulang sebanyak lima kali kemudian dihitung nilai rata-rata untuk masing-masing parameter QoS. Berikut hasil pengambilan data untuk *bandwidth* 16 kbps ditunjukkan pada Tabel 4.4 dan 4.5.

Tabel 4.4 Pengukuran VoIP dengan Implementasi QoS

No.	Qos					
	voip diganggu vod			voip menggangu vod		
	delay (ms)	jitter (ms)	packet loss (%)	delay (ms)	jitter (ms)	packet loss (%)
1	145.44	203.61	86,,3	124.84	167.62	83.4
2	133.1	187.07	85.2	131.11	184.74	85
3	132.81	186.76	85	132.59	189.74	85.3
4	130.66	184.56	85	131	186.3	85.2
5	138.05	194.085	84.87	130.62	187.19	84.9
mean	**135.5**	**190.5**	**85.07**	**129.88**	**182.1**	**84.73**

Tabel 4.5 Pengukuran VoIP tanpa Implementasi QoS

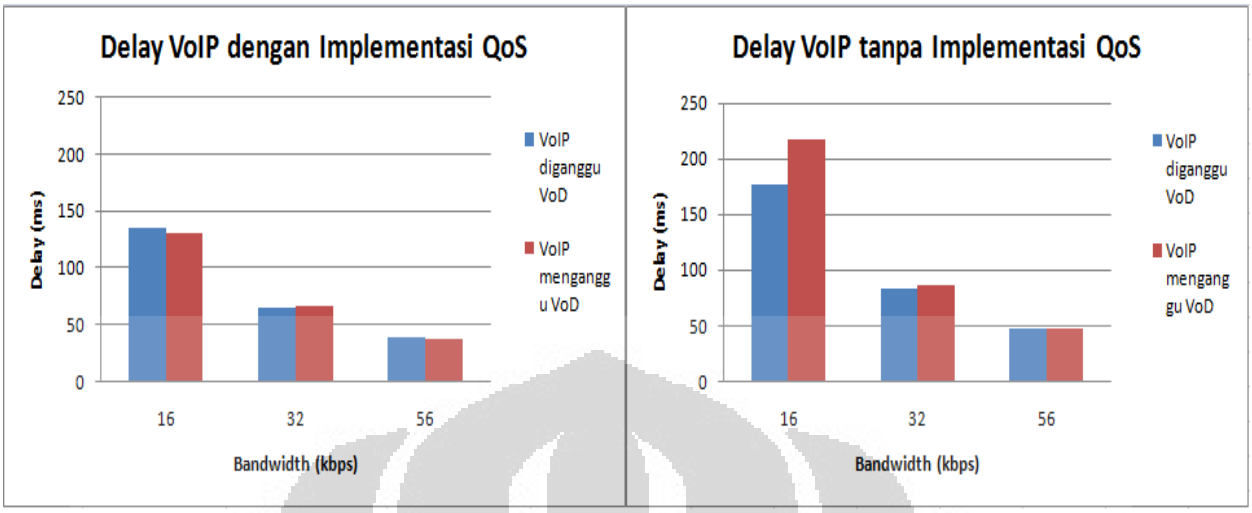
No.	tanpa Qos					
	voip diganggu vod			voip menggangu vod		
	delay (ms)	jitter (ms)	packet loss (%)	delay (ms)	jitter (ms)	packet loss (%)
1	174.03	265.62	90.6	215.92	293.74	91.4
2	185.43	262.4	90.4	218.95	359.82	91.3
3	172.13	268.88	89.1	215.76	351.51	91.1
4	178.78	265.64	89.75	216.45	347.48	91.2
5	174.66	267.26	89.565	217.94	327.53	91.4
mean	**177.19**	**265.64**	**90.03**	**216.88**	**335.02**	**91.27**

Hal yang serupa juga dilakukan pada percobaan dengan *bandwidth* 32 kbps dan 56 kbps. Hasil dari pengukuran *delay* di *bandwidth* 16, 32, dan 56 kbps disatukan menjadi tabel 4.6.

Tabel 4.6 *Delay* pada aplikasi VoIP

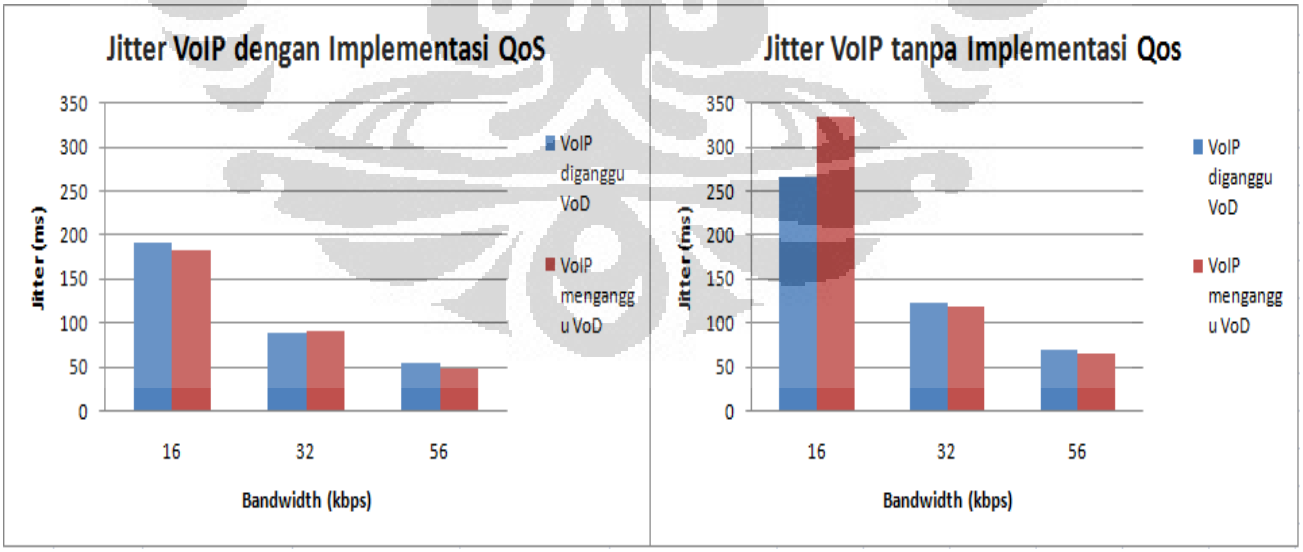
Bandwidth (kbps)	QoS		tanpa QoS	
	VoIP disela VoD	VoIP menyela VoD	VoIP disela VoD	VoIP menyela VoD
16	135.5	129.88	177.19	216.88
32	64.45	65.80	83.69	85.89
56	38.65	37.40	46.89	47.81

Dari Tabel 4.6 diperoleh Gambar 4.4.

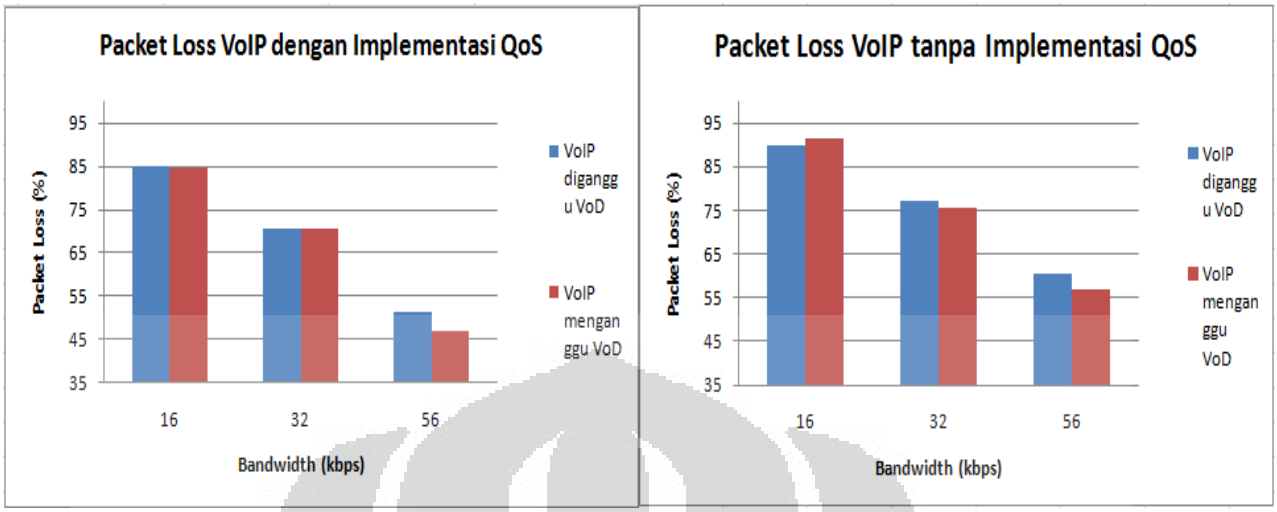


Gambar 4.4 Perbandingan *Delay* VoIP

Gambar 4.4 menunjukka bahwa performansi VoIP menjadi lebih baik ketika dilakukan implementasi QoS di jaringan IMS. Penurunan *delay* VoIP terjadi ketika dilakukan implementasi QoS di jaringan, perubahan terlihat pada *bandwidth* 16 kbps, 32 kbps dan 56 kbps. Hal ini sesuai dengan tujuan awal penelitian yaitu, memberikan prioritas yang lebih tinggi kepada VoIP dibandingkan dengan VoD pada saat dilakukan implementasi QoS di jaringan.



Gambar 4.5 Perbandingan *Jitter* VoIP



Gambar 4.6 Perbandingan *Packet loss* VoIP

Gambar 4.5 dan Gambar 4.6 menunjukkan hal yang sama dengan grafik *delay* pada gambar 4.4 yaitu, pada saat tidak ada QoS di jaringan maka performansi VoIP akan berkurang.

4.3 Percobaan Tambahan

Percobaan tambahan merupakan percobaan yang dilakukan untuk mempertegas hasil percobaan yang telah diperoleh sebelumnya. Percobaan yang dilakukan mengvariasikan durasi gangguan dari VoIP terhadap VoD dan sebaliknya. Percobaan tambahan dilakukan sebanyak dua kali, yaitu VoD yang berjalan selama 3 menit, diganggu oleh VoIP dengan durasi 45 detik dan 2 menit. Percobaan yang kedua dilakukan dengan VoIP diganggu oleh VoD selama 45 detik dan 2 menit. Hasil dari kedua percobaan ini ditampilkan melalui Tabel 4.7 dan 4.8.

Tabel 4.7 Percobaan Tambahan pertama

No.	VoD 3 menit					
	diganggu VoIP selama 45 detik			diganggu VoIP selama 2 menit		
	Delta (ms)	Jitter (ms)	Packet Loss (%)	Delta (ms)	Jitter (ms)	Packet Loss (%)
1	294.59	222.72	91.10	327.24	273.98	92.10
2	293.47	216.40	91.10	323.48	248.54	91.90
3	293.90	235.81	91.10	328.46	255.60	92.10
4	294.03	219.56	91.10	327.85	264.79	92.10
5	292.38	220.24	91.13	325.94	253.65	92.10
mean	**293.67**	**222.95**	**91.11**	**326.59**	**259.31**	**92.06**

Tabel 4.8 Percobaan Tambahan kedua

No.	VoIP					
	diganggu VoD selama 45 detik			diganggu VoD selama 2 menit		
	Delta (ms)	Jitter (ms)	Packet Loss (%)	Delta (ms)	Jitter (ms)	Packet Loss (%)
1	19.99	11.07	0	20.00	11.13	0
2	19.98	11.08	0	20.00	11.12	0
3	19.99	11.08	0	20.00	11.13	0
4	19.97	11.06	0	20.03	11.11	0
5	19.98	11.07	0	20.00	11.13	0
mean	**19.98**	**11.07**	**0**	**20.00**	**11.12**	**0**

4.4 Analisa Data

4.4.1 Analisa Delay

Pada penelitian ini dilakukan pengambilan data untuk tiga parameter QoS yaitu, *delay*, *jitter* dan *packet loss*. Hasil percobaan, baik pada pengukuran untuk performansi VoD maupun VoIP, diperoleh hubungan yang berbanding terbalik antara bandwidth dan *delay*. *Delay* terjadi karena adanya keterbatasan dari saluran transmisi yang tidak dapat memenuhi kebutuhan *traffic* dari aplikasi yang menggunakan jaringan. Hal yang ingin dicapai pada penelitian ini adalah performansi VoIP yang tetap terjaga ketika ada *traffic* VoD yang juga sedang menggunakan jaringan ketika dilakukan implementasi QoS di jaringan IMS.

Pada pengukuran performansi VoD, terlihat bahwa *delay* dari *traffic* VoD cenderung bernilai tetap pada angka 26 ms pada saat tidak ada implementasi QoS di jaringan IMS, namun pada saat dilakukan implementasi QoS di jaringan IMS terlihat bahwa *delay* dari *traffic* VoD mengalami peningkatan. Perbedaan *delay* terlihat pada ketiga variasi *bandwidth* yang digunakan. Perbedaan delay yang terjadi untuk *bandwidth* 384 kbps sebesar 29 ms ketika dilakukan implementasi QoS. Implementasi QoS yang dilakukan dengan mengklasifikasikan *traffic* VoD sebagai default *traffic* di jaringan dan tidak masuk ke dalam kelas apapun yang telah dikategorikan sebelumnya pada saat dilakukan konfigurasi QoS di jaringan IMS. Perubahan *delay* juga terjadi pada saat *traffic* VoD disela oleh *traffic* VoIP dimana terjadi peningkatan *delay* sebesar 2 ms ketika VoD disela oleh VoIP. Hal ini dikarenakan kebutuhan *bandwidth* VoIP yang cukup kecil jika dibandingkan dengan kebutuhan *bandwidth* VoD.

Traffic VoIP tidak mengalami perubahan nilai *delay* ketika *bandwidth* jaringan yang digunakan sebesar 384 kbps, 512 kbps, dan 640 kbps, Sehingga penulis melakukan percobaan dengan *bandwidth* 16 kbps, 32 kbps dan 56 kbps untuk melihat perubahan *delay* pada *traffic* VoIP. Hasil percobaan pada bandwidth 16 kbps menunjukkan bahwa implementasi QoS cukup berpengaruh terhadap *delay*. Pada saat *traffic* VoD masuk ke dalam jaringan terjadi peningkatan nilai *delay* VoIP hingga 135 ms ketika dilakukan implementasi QoS dan peningkatan hingga 177 ms ketika tidak dilakukan implementasi QoS di jaringan IMS. Peningkatan *delay* ini menunjukkan bahwa *traffic* VoIP saling berbagi jalur dengan *traffic* VoD. Namun *traffic* VoIP tidak 100% mendapatkan penggunaan jaringan, VoIP hanya akan mendapatkan prioritas ketika terjadi kongesti di jaringan.

Pada percobaan tambahan, *delay* performansi VoD menjadi lebih buruk ketika VoD diganggu VoIP dengan durasi yang lebih lama, yaitu 326 ms dengan durasi VoIP 2 menit. Performansi VoIP tidak terganggu oleh gangguan dari VoD baik selama 45 detik maupun 2 menit yang menghasilkan *delay* 19.98 ms dan 20 ms.

4.4.2 Analisa Jitter

Parameter QoS kedua yang dibahas pada penelitian ini adalah *jitter*. *Jitter* merupakan variasi *delay* di jaringan, *jitter* sering juga diartikan sebagai variasi kedatangan paket. Pada layanan multimedia yang bersifat *real time* diperlukan nilai *jitter* yang stabil agar kualitas layanan tetap terjaga. Jika *jitter* bernilai besar maka variasi *delay* yang terjadi di jaringan juga besar, hal ini menujukkan bahwa semakin besar nilai *jitter* maka kualitas layanan menjadi semakin buruk.

Nilai *jitter* dari hasil percobaan VoD menunjukkan *trend* yang sama dengan *delay* hasil percobaan VoD, dimana pada saat dilakukan implementasi QoS di jaringan nilai *jitter* dari VoD lebih tinggi. Perbedaan jitter terjadi pada ketiga variasi bandwidth yang digunakan, perbedaan paling besar terjadi pada *bandwidth* 384 kbps, pada saat implementasi QoS dilakukan nilai jitter sebesar 28.008 ms sebelum VoD diganggu oleh VoIP dan nilai jitter sebesar 47.563 ms setelah VoD diganggu oleh VoIP. Hasil percobaan jitter VoD ketika QoS tidak diimplementasikan di jaringan menunjukkan hasil yang tidak berbeda jauh baik pada saat VoD sebelum diganggu oleh VoIP dengan VoD setelah oleh VoIP.

Percobaan performansi VoIP menghasilkan jitter yang tidak berbeda jauh baik pada saat VoIP diganggu oleh VoD maupun sebaliknya yaitu 88 ms dan 90 ms untuk bandwidth 32 kbps pada saat implementasi QoS dijalankan. Hasil percobaan pada bandwidth 32 kbps ketika implementasi QoS di jaringan tidak dijalankan adalah 122 ms untuk skenario VoIP diganggu oleh VoD dan 119 ms untuk skenario VoIP menganggu VoD. Hasil yang diperoleh menunjukkan bahwa traffic VoIP mendapatkan performa yang lebih baik ketika dijalankan implementasi QoS di jaringan IMS.

Pada percobaan tambahan, *jitter* performansi VoD menjadi lebih buruk ketika VoD diganggu VoIP dengan durasi yang lebih lama, yaitu 259 ms dengan durasi VoIP 2 menit. Performansi VoIP tidak terganggu oleh gangguan dari VoD baik selama 45 detik maupun 2 menit yang menghasilkan *jitter* yang sama yaitu 11 ms.

4.4.3 Analisa Packet Loss

Parameter QoS ketiga yang dibahas di penelitian ini adalah *packet loss*. *Packet loss* dipengaruhi oleh dua hal yang cukup berhubungan yaitu *bitrate* dari *codec* dari aplikasi yang digunakan dan *bandwidth* yang tersedia di jaringan. *Bitrate* yang tinggi membutuhkan penggunaan *bandwidth* yang besar juga agar tidak terjadi *packet loss*.

Pada pengukuran untuk performansi VoD ketika dilakukan implementasi QoS di jaringan IMS didapatkan nilai *packet loss* yang meningkat ketika disela oleh VoIP baik pada *bandwidth* 384 kbps yaitu 34 % sebelum VoD diganggu oleh VoIP dan 39% setelah VoD digangu oleh VoIP. Pengukuran di *bandwidth* 512 kbps menghasilkan *packet loss* 13% sebelum VoD diganggu VoIP dan *packet loss* 17 % setelah VoD diganggu VoIP, sedangkan untuk *bandwidth* 640 kbps tidak terjadi *packet loss* karena penggunaan *bandwidth* VoD yang sudah tercukupi. Ketika tidak ada QoS yang berlaku di jaringan, VoD sama sekali tidak mengalami *packet loss* baik pada *bandwidth* 384 kbps, 512 kbps dan 640 kbps. Hal ini menunjukkan bahwa *traffic* VoIP tidak mengganggu penggunaan *bandwidth* dari VoD.

Percobaan pengukuran performansi VoIP, pada penggunaan *bandwidth* 16 kbps tidak terlihat perbedaan *packet loss* yang cukup signifikan pada saat ada QoS ataupun tidak ada QoS dikarenakan *bandwidth* tersebut tidak dapat melayani VoIP dengan baik, hal ini terjadi ketika percobaan berlangsung dimana *call* sulit untuk di *establish* serta kualitas suara yang buruk.

Pada *bandwidth* 32 kbps terjadi perubahan nilai *packet loss* yaitu pada saat ada QoS terjadi *loss* sebesar 70% baik pada saat VoIP diganggu dan menggangu VoD sedangkan pada saat tidak ada QoS terjadi *loss* sebesar 77% pada saat VoIP diganggu VoD dan 75 % saat VoIP mengganggu VoD.

Pada *bandwidth* 56 kbps baru terlihat perbedaan yang lebih besar pada saat ada QoS dengan tidak adanya QoS di jaringan yaitu perbedaan sebesar 10% baik pada saat VoIP disela maupun menyela VoD.

Pada percobaan tambahan, *packet loss* performansi VoD menjadi lebih buruk ketika VoD diganggu VoIP dengan durasi yang lebih lama, yaitu 92 % dengan durasi VoIP 2 menit. Performansi VoIP tidak terganggu oleh gangguan dari VoD baik selama 45 detik maupun 2 menit yang menghasilkan *packet loss* 0 %.

BAB 5
KESIMPULAN

Pada skripsi yang berjudul "Implementasi QoS pada Jaringan IMS dengan prioritas paket" dapat disimpulkan beberapa hal yaitu sebagai berikut.

1. Permodelan Jaringan IMS dengan prioritas paket telah berhasil dilakukan dengan menggunakan open source router Vyatta.
2. Hasil percobaan pengukuran performansi aplikasi VoD, terlihat bahwa pada saat dilakukan implementasi QoS di jaringan, parameter-parameter QoS dari VoD menunjukkan peningkatan *delay* sebesar 42 ms, *jitter* sebesar 47 ms, dan *packet loss* 39 % untuk *bandwidth* 384 kbps pada saat VoD diganggu oleh VoIP.
3. Hasil percobaan pengukuran performansi aplikasi VoIP, terjadi penurunan *delay* 42 ms, *jitter* 75 ms, dan *packet loss* 5% untuk *bandwidth* 16 kbps pada saat dilakukan implementasi QoS. Hal ini menunjukkan bahwa implementasi QoS di jaringan IMS telah berjalan sesuai teori.
4. Hasil dari percobaan tambahan di bandwidth 56 kbps menunjukkan bahwa variabel durasi gangguan dari VoIP mempengaruhi performa VoD, dimana *delay* VoD ketika diganggu VoIP dengan durasi 45 detik sebesar 293 ms dan durasi gangguan VoIP 2 menit memiliki delay sebesar 326 ms. Performa VoIP tidak terganggu oleh gangguan dari VoD baik dengan variasi durasi 45 detik dan 2 menit dimana delay VoIP sebesar 20 ms.
5. *Quality of Service* di jaringan IMS pada percobaan ini diimplementasikan dengan cara memberikan prioritas terhadap paket-paket yang masuk ke dalam IMS berdasarkan aplikasi ataupun layanan yang digunakan.
6. Peran PCRF dan PCEF sebagai *server* yang membuat dan menerapkan *policy control* di dalam jaringan IMS dapat digantikan dengan menggunakan open source router yang mampu menerapkan QoS.

Daftar Referensi

[1] http://insideit.wordpress.com/2011/01/13/rekor-pengguna-skype-27-juta-orang/

[2] http://www.tempointeraktif.com/hg/it/2009/03/05/brk,20090305-163243,id.html

[3] Hens, F.J & Caballero, J.M.(2008). Triple Play : Building the converged network for IP, VoIP and IPTV. John Wiley & Sons

[4] Soetikno Tendy. *IMS as Convergence Enabler*.

[5] Juliet Bates, *et al.*, *Converged Multimedia Network* (John Wiley &Sons,2006)

[6] http://en.wikipedia.org/wiki/IP_Multimedia_Subsystem

[7] http://joudane.wordpress.com/2008/07/15/ims-ip-multimedia-subsystem/

[8] http://openimscore.org/

[9] Pelangi, Zanetta. IMPLEMENTASI VOIP MENGGUNAKAN OPEN IMS CORE SEBAGAI INFRASTRUKTUR JARINGAN. Universitas Indonesia. 2010.

[10] B. Raouyane, M. Bellafkih, D. Ranc," Qos Management in IMS : Diffserv Model," *IEEE ICC 2009 proceedings*, 2009.

[11] Camarillo, Gonzalo & Miguel A. Garcia-Martin. *The 3G IP Multimedia Subsystem (IMS): Merging the Internet and the Cellular Worlds, Second Edition*. JohnWiley & Sons, 2006.

[12] M.Handley, V.Jacobson,& C.Perkins.*SDP:Session Initiation Protocol*. July 2006.RFC 4566.

[13] Deolens. ANALISA LAYANAN VIDEO ON DEMAND PADA ARSITEKTUR IP MULTIMEDIA SUBSYSTEM. Universitas Indonesia. 2010.

[14] http://blognaivaniy.blogspot.com/2010/01/real-time-communication-rtc.html

[15] http://id.wikipedia.org/wiki/Protokol_Internet

[16] Copeland, Rebecca. Converging NGN Wireline and Mobile 3G Networks with IMS. CRC Press. 2009.

[17] http://id.wikipedia.org/wiki/Quality_of_Service

[18] http://sulistyonugroho.wordpress.com/2010/10/09/quality-of-service-dalam-data-komunikasi/

[19] http://www.ittelkom.ac.id/library/index.php?view=article&catid=10%3Ajaringan&id=380%3Adifferentiated-services-diffserv&option=com_content&Itemid=15

[20] http://en.wikipedia.org/wiki/Differentiated_services

[21] Miguel Navarro, Yezid Donoso, Viviana Rodriguez ," An IMS Architecture with QoS Parameter for Flexible Convergent Services," *IEEE ICC 2010 proceedings*, 2010.

[22] Syed A.Ahson, Mohammad Ilyas. IP Multimedia Subsystem (IMS) Handbook. CRC Press. 2009

LAMPIRAN

Tabel Jitter VoD

Bandwidth	QoS		tanpa QoS	
	Sebelum diganggu VoIP (ms)	Setelah diganggu VoIP (ms)	Sebelum diganggu VoIP (ms)	Setelah diganggu VoIP (ms)
384	28.008	47.563	17.152	18.96
512	27.24	33.643	13.768	15.632
640	9.38	10.94	8.48	10.05

Tabel Packet Loss VoD

Bandwidth	QoS		tanpa QoS	
	Sebelum diganggu VoIP (%)	Setelah diganggu VoIP (%)	Sebelum diganggu VoIP (%)	Setelah diganggu VoIP (%)
384	34.712	39.04	0	0
512	13.16	17.314	0	0
640	0	0	0	0

Tabel Jitter VoIP

Bandwidth (kbps)	QoS		tanpa QoS	
	VoIP diganggu VoD	VoIP menganggu VoD	VoIP diganggu VoD	VoIP menganggu VoD
16	190.5	182.1	265.64	335.02
32	88.83	90.15	122.96	119.29
56	54.57	48.06	69.4	65.14

Tabel Packet Loss VoIP

Bandwidth (kbps)	QoS		tanpa QoS	
	VoIP diganggu VoD	VoIP menganggu VoD	VoIP diganggu VoD	VoIP menganggu VoD
16	85.07	84.73	90.03	91.27
32	70.42	70.44	77.1	75.55
56	51.38	46.92	60.52	56.72