



**UNIVERSITAS INDONESIA**

**Implementasi Policy Control Rule Function Pada Jaringan**

**OpenIMSCore**

**SKRIPSI**

**KRISNA JUANTA  
0706267830**

**FAKULTAS TEKNIK  
PROGRAM TEKNIK ELEKTRO  
DEPOK  
JANUARI 2012**



**UNIVERSITAS INDONESIA**

**Implementasi Policy Control Rule Function pada Jaringan  
OpenIMSCore**

**SKRIPSI**

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

**KRISNA JUANTA  
0706267830**

**FAKULTAS TEKNIK  
PROGRAM STUDI TEKNIK ELEKTRO  
KEKHUSUSAN TEKNIK TELEKOMUNIKASI  
DEPOK  
DESEMBER 2012**

ii

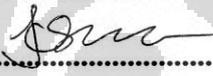
**Universitas Indonesia**

## HALAMAN PERNYATAAN ORISINALITAS

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

**Nama : Krisna Juanta**

**NPM : 0706267830**

**Tanda Tangan : **

**Tanggal : 26 Januari 2012**

## HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :  
Nama : Krisna Juanta  
NPM : 0706267830  
Program Studi : Teknik Elektro  
Judul Skripsi : Implementasi Policy Control Rule Function pada Jaringan OpenIMScore.

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 : Prof. Ir. Djamhari Sirat M.Sc., Ph.D. (.....)

Penguji : Ir. A. Endang Sriningsih M.T. Si (.....)

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

Ditetapkan di : Depok

Tanggal : 26 Januari 2012

## HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

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

Nama : Krisna Juanta  
NPM : 0706267830  
Program Studi : Teknik Elektro  
Departemen : Teknik Elektro  
Fakultas : Teknik  
Jenis Karya : Skripsi

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

### **Implementasi *Policy Control Rule Function* pada Jaringan OpenIMSCore**

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

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok

Pada Tanggal : Januari 2012

Yang menyatakan



( Krisna Juanta )

v

Universitas Indonesia

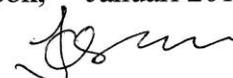
## KATA PENGANTAR

Puji serta syukur penulis panjatkan kehadirat Allah SWT karena dengan limpahan berkat, rahmat serta hidayah-Nya penulis dapat menyelesaikan seminar ini. Penulisan skripsi dilakukan sebagai salah satu syarat untuk menjadi Sarjana Teknik di Departemen Teknik Elektro FTUI. Saya menyadari bahwa tanpa bantuan banyak pihak, skripsi ini tidak mungkin terselesaikan. Oleh sebab itu, pada kesempatan ini saya ingin mengucapkan terima kasih kepada pihak-pihak yang membantu saya dalam segala hal mengenai penyusunan skripsi baik secara langsung maupun tidak langsung. Hal ini penulis tujukan kepada:

1. Allah SWT yang telah memberikan kekuatan kepada penulis untuk menyelesaikan seminar ini.
2. Orang tua dan keluarga penulis.
3. Prof. Ir. Djamhari Sirat M.Sc, Ph.D selaku dosen Pembimbing Akademik dan
4. Pembimbing seminar dan skripsi saya yang telah meluangkan waktunya, serta masukan-masukan selama bimbingan.
5. Teman-teman Kosan Lalolo 2 yang telah memberikan bantuan tenaga dan dukungan moril kepada penulis dalam menyelesaikan seminar ini.
6. Rekan-rekan yang telah berjuang bersama dalam mempelajari IP Multimedia Subsystem: Ardy Thiotrisno, Chandra Gunawan, Faisal Jamil, dan Rosa
7. Asisten laboratorium jaringan atas bantuan teknisnya
8. Seluruh keluarga besar Civitas Akademika Fakultas Teknik Universitas Indonesia khususnya karyawan sekretariat Departemen Teknik Elektro yang telah banyak memberikan bantuan dalam urusan administrasi.

Akhir kata, penulis berharap agar Allah SWT berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga laporan kerja praktik ini membawa manfaat.

Depok, Januari 2012



Krisna Juanta

## ABSTRAKSI

Nama : Krisna Juanta  
Program Studi : Teknik Elektro  
Judul : Implementasi Policy Control Rules Function pada jaringan OpenIMSCore

Skripsi ini membahas mengenai implementasi *Policy Control Rules Function* (PCRF) pada jaringan *OpenIMSCore*. PCRF berfungsi sebagai penentu kebijakan pada *OpenIMSCore* dalam pengaturan *Quality of Service* (QoS). QoS dapat menjamin layanan *OpenIMSCore* agar dapat beroperasi sesuai dengan yang diharapkan. Implementasi PCRF dilakukan untuk memodelkan suatu jaringan *OpenIMSCore* dalam pengaturan *bandwidth* yang diberikan kepada layanan yang ada. Instalasi PCRF membutuhkan komponen penting yaitu protocol *DIAMETER* yang memiliki peranan dalam *Authentication, Authorization, and Accounting* (AAA). PCRF terhubung langsung dengan P-CSCF sehingga pada P-CSCF juga harus dilakukan pengaturan protocol *DIAMETER* agar dapat tersambung dengan PCRF.

Kata kunci : *OpenIMSCore, Quality of Service, PCRF, delay, jitter.*

## ABSTRACT

Nama : Krisna Juanta  
Program Studi : Electrical Engineering  
Judul : Implementation of Policy Control Rules Function at  
OpenIMSCore network

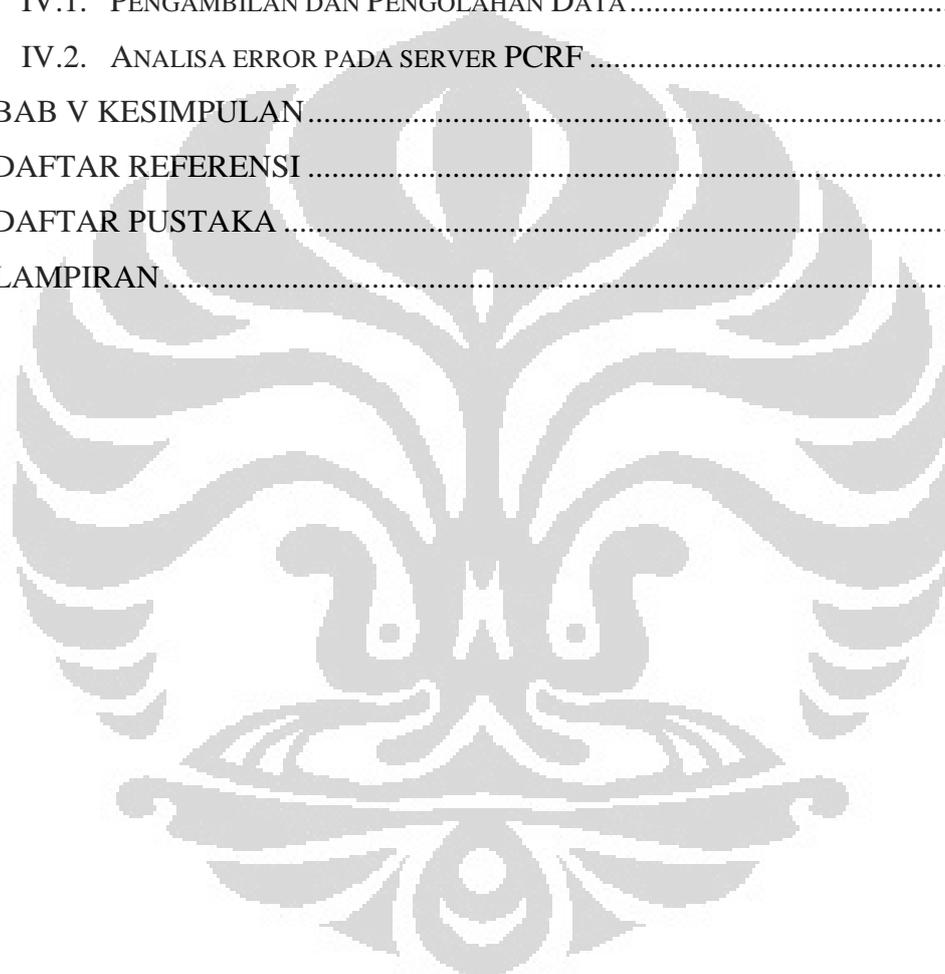
This paper discusses the implementation of the Policy Control Rules Function (PCRF) on the OpenIMSCore network. PCRF functions as a determinant in setting policy on OpenIMSCore Quality of Service (QoS). QoS can guarantee OpenIMSCore services in order to operate as expected. Implementation PCRF performed to model a OpenIMSCore network in bandwidth settings given to existing services. Installation requires a critical component of PCRF DIAMETER protocol that has a role in the Authentication, Authorization, and Accounting (AAA). PCRF is connected directly to the P-CSCF to the P-CSCF should also be performed DIAMETER protocol settings in order to connect to the PCRF.

Key words: OpenIMSCore, Quality of Service, PCRF, delay, jitter.

## DAFTAR HALAMAN

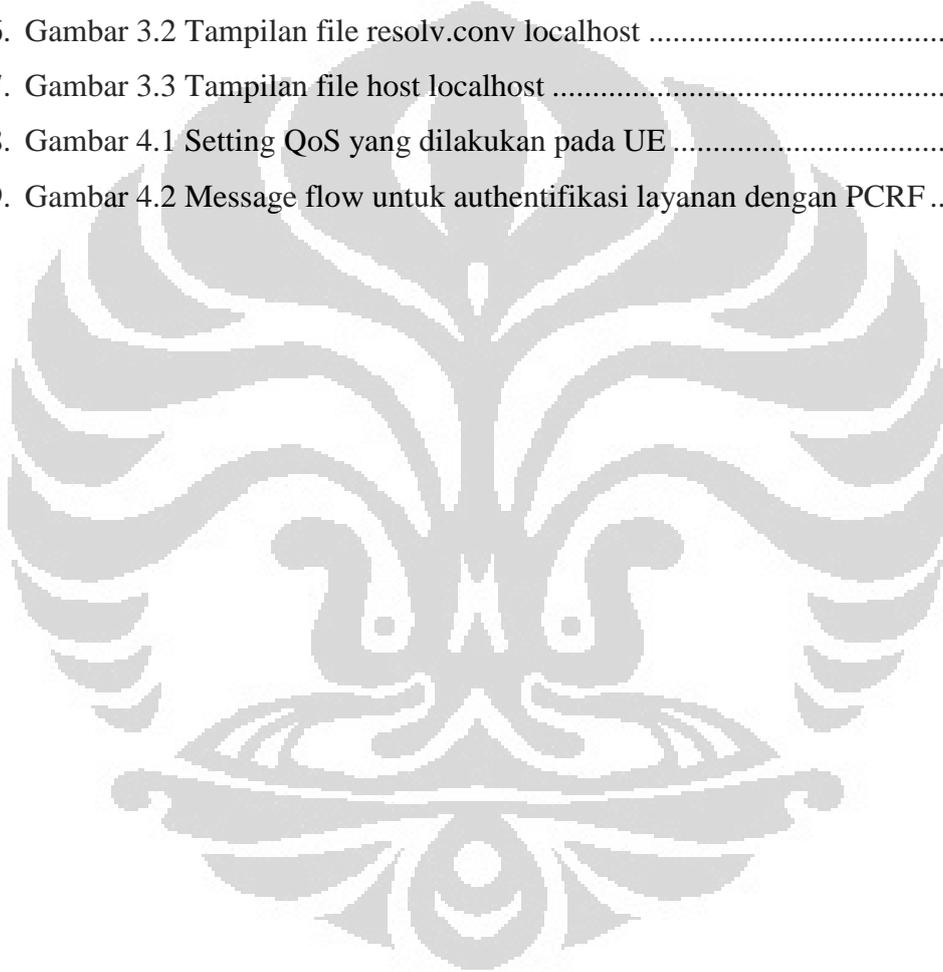
HALAMAN SAMPUL .....	I
HALAMAN JUDUL.....	II
HALAMAN PERNYATAAN ORISINALITAS.....	III
HALAMAN PENGESAHAN.....	IV
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI	
TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS .....	V
KATA PENGANTAR .....	VI
ABSTRAKSI .....	VII
DAFTAR HALAMAN .....	VIII
DAFTAR GAMBAR .....	XI
DAFTAR TABEL.....	XII
BAB I PENDAHULUAN .....	1
I.1.    LATAR BELAKANG .....	1
I.2.    PERUMUSAN MASALAH.....	3
I.3.    TUJUAN PENELITIAN .....	3
I.4.    BATASAN MASALAH .....	3
I.5.    SISTEMATIKA PENULISAN.....	4
BAB II OPENIMSCORE DAN QUALITY OF SERVICE .....	5
II.1.    IP MULTIMEDIA SUBSYSTEM (IMS) .....	5
II.1.1.    Arsitektur pada IMS.....	6
II.2.    PROTOCOL-PROTOCOL DI DALAM IMS .....	9
II.2.1.    Session Initiation Protocol (SIP) .....	9
II.2.2.    Session Description Protocol (SDP).....	10
II.2.3.    Real Time Protocol (RTP).....	11
II.2.4.    Real Time Control Protocol (RTCP).....	13
II.2.5.    Real Time Streaming Protocol (RTSP) .....	14
II.2.6.    Control Open Policy Service (COPS).....	15
II.3.    QUALITY OF SERVICE.....	15
II.3.1.    Parameter-Parameter Quality of Service .....	16
II.3.2.    Jenis-Jenis Quality Of Service .....	17

II.3.3. <i>Quality of Service di Jaringan IMS</i> .....	22
II.4. POLICY CONTROL RULES FUNCTION (PCRF).....	22
BAB III IMPLEMENTASI <i>POLICY CONTROL RULE FUNCTION</i> .....	24
III.1. KONFIGURASI DAN INSTALASI IMS CORE .....	26
III.2. INSTALASI POLICY CONTROL RULE FUNCTION (PCRF) .....	28
III.3. SAMBUNGAN ANTARA PCRF DENGAN P-CSCF .....	29
BAB IV HASIL PERCOBAAN DAN ANALISA DATA .....	31
IV.1. PENGAMBILAN DAN PENGOLAHAN DATA.....	32
IV.2. ANALISA ERROR PADA SERVER PCRF .....	36
BAB V KESIMPULAN.....	40
DAFTAR REFERENSI .....	42
DAFTAR PUSTAKA .....	43
LAMPIRAN.....	44



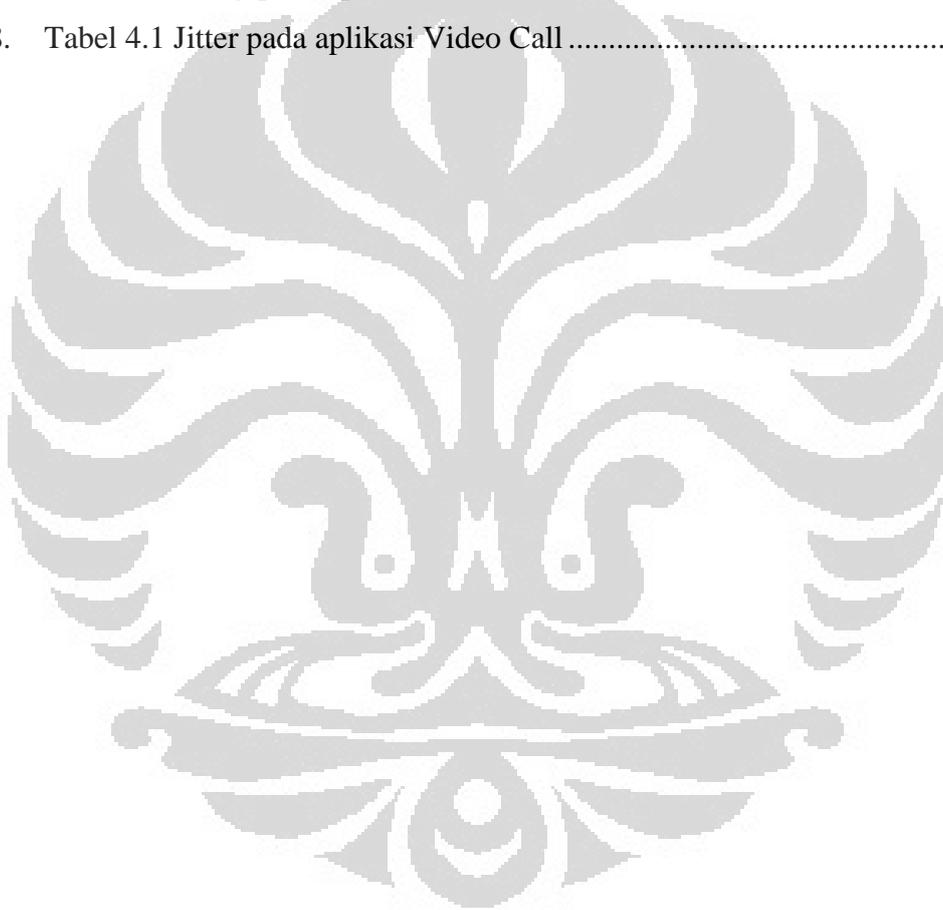
## DAFTAR GAMBAR

1. Gambar 2.1 Konvergensi Jaringan dengan Layanan.....	5
2. Gambar 2.2 Arsitektur IMS.....	6
3. Gambar 2.3 Enkapsulasi RTP .....	12
4. Gambar 2.4 RTP Packet Format.....	12
5. Gambar 3.1 Konfigurasi PCRF dengan OpenIMSCore .....	25
6. Gambar 3.2 Tampilan file resolv.conv localhost .....	27
7. Gambar 3.3 Tampilan file host localhost .....	27
8. Gambar 4.1 Setting QoS yang dilakukan pada UE .....	32
9. Gambar 4.2 Message flow untuk autentifikasi layanan dengan PCRF .....	37



## DAFTAR TABEL

1. Tabel 2.1 RTCP Packet Types .....	13
2. Tabel 2.2 Type message COPS.....	15
3. Tabel 2.3 Assured Forwarding (AF) Behavior Group .....	20
4. Tabel 2.4 QoS Priority Level .....	21
5. Tabel 4.1 Delay pada aplikasi VoIP.....	33
6. Tabel 4.1 Jitter pada aplikasi VoIP .....	33
7. Tabel 4.1 Delay pada aplikasi Video Call.....	34
8. Tabel 4.1 Jitter pada aplikasi Video Call .....	34



# BAB I

## PENDAHULUAN

### I.1. Latar Belakang

Perkembangan teknologi telekomunikasi yang sangat cepat membuat banyak operator telekomunikasi mencari strategi baru dalam memberikan pelayanan kepada pengguna jasa telekomunikasi. Penyedia layanan internet saat ini menyediakan beberapa layanan yang dapat bersaing dengan penyedia layanan telekomunikasi pada umumnya, seperti layanan *voice* dan *video*. Masuknya penyedia layanan internet ke dalam industri telekomunikasi menyebabkan menurunnya *voice business*, profit yang datar pada akses data. Selain itu, regulasi baru memicu kompetisi pada perusahaan telekomunikasi untuk menciptakan sebuah jaringan konvergen.

Dalam menghadapi masalah yang ada, operator telekomunikasi menerapkan konvergensi sebagai suatu strategi bisnis terbaru. Dengan menggunakan konvergensi memungkinkan perusahaan untuk menggabungkan beberapa layanan yang sebelumnya dilakukan secara terpisah menjadi satu, seperti *mobile/ fixed telephony*, data, dan *broadband*. Selain itu juga penggunaan teknologi konvergensi memberikan keuntungan lebih bagi penyedia layanan.

Konvergensi bertujuan untuk menyatukan antara layanan yang sudah ada dan layanan baru dengan menggunakan 1 (satu) *core network*, meminimalkan jumlah *network layer* protokol dan menggabungkan transport dari seluruh jenis trafik jaringan. Konvergensi layanan tidak membedakan penggunaan antara *mobile cellular* dengan *broadband*, sehingga dapat dikatakan bahwa dengan menggunakan konvergensi maka setiap orang dapat memiliki akses internet serta PSTN yang tidak terbatas. Selain itu, sesi multimedia dapat ditransfer melalui jaringan yang berbeda tanpa perubahan atau interupsi pada layanan yang ditawarkan.

*Next Generation Network (NGN)* adalah teknologi telekomunikasi masa mendatang yang menggambarkan arsitektur jaringan telekomunikasi

beberapa tahun kedepan. Prinsip yang digunakan pada *NGN* adalah dengan mentransfer semua informasi (suara, data, dan multimedia) dalam berupa paket-paket seperti yang digunakan pada layanan internet. Pada penggunaannya, jaringan *NGN* menggunakan *Internet Protocol (IP)* sehingga dapat dikatakan bahwa jaringan *NGN* merupakan “*all IP*”.

Akses broadband yang cepat dan tidak terbatas merupakan tujuan dari perubahan yang dilakukan oleh perusahaan telekomunikasi. *Session Initial Protocol (SIP)* juga merupakan faktor penting dimana *SIP* berfungsi untuk menyatukan alur multimedia yang berbeda dalam satu sesi serta menyederhanakan pengaturan layanan pengguna paralel atau berbeda-beda. Faktor penting lainnya terletak pada *IP Multimedia Subsystem (IMS) Core* yang memungkinkan berbagai jenis layanan bekerja dalam kesatuan core yang konvergen.

Dengan menggunakan *IMS* sebagai core network serta infrastruktur yang telah dimiliki oleh operator telekomunikasi merupakan sebuah keunggulan yang dimiliki oleh operator telekomunikasi untuk menjamin kelancaran serta kualitas pelanggan dalam berkomunikasi jika dibandingkan dengan penyedia layanan internet yang memberikan layanan kepada pelanggan dengan metode *best effort*. *IMS* memiliki kemampuan untuk memberikan *Quality of Service (QoS)* pada layanan yang diberikan untuk menjamin kepuasan bagi pengguna layanan tersebut. Selain itu, *IMS* juga memiliki mekanisme *charging* yang dapat membantu operator telekomunikasi dalam menetapkan tarif bagi pelanggan. *Quality of Service (QoS)* memiliki 3 metode pengaturan diantaranya, *Best Effort*, *Integrated Service (intserv)*, dan *Differentiated Service (diffserv)*.

Pengaturan *QoS* pada jaringan *OpenIMSCore* dapat dilakukan dengan menggunakan bantuan sebuah server bernama *Policy Control Rule Function (PCRF)*. Server ini memiliki banyak fungsi dalam pengaturan *QoS* seperti memberikan prioritas pada layanan, pengaturan bandwidth untuk user, dan lain sebagainya. Pada karya tulis ini akan dibahas mengenai implementasi *PCRF* pada jaringan *OpenIMSCore*. *PCRF* sangat

diperlukan oleh jaringan *OpenIMSCore* dalam memenuhi tugasnya untuk memberikan layanan terbaik bagi pelanggannya.

## **I.2. Perumusan Masalah**

Pada Skripsi ini, ingin diimplementasikan suatu jaringan berbasis SIP 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. Fungsi dari PCRF pada arsitektur jaringan *OpenIMSCore*
2. Bagaimana kinerja dari Policy Control Rule Function pada jaringan *OpenIMSCore* ?
3. Bagaimana mengimplementasikan PCRF pada jaringan *OpenIMSCore*?

## **I.3. Tujuan Penelitian**

1. Meng-implementasikan server *Policy Control Rules Function* (PCRF) pada jaringan *OpenIMSCore*.
2. Membandingkan layanan VoIP dan Video Call sebelum implementasi PCRF dengan sesudah implementasi *PCRF* di jaringan *OpenIMSCore*.

## **I.4. Batasan Masalah**

Pada Skripsi ini, penelitian dilakukan mulai dari tahap pembangunan jaringan IMS dengan layanan VoIP dan Video Call sampai dengan instalasi server *Policy Control Rule Function*. Penggunaan layanan VoIP dan Video Call sudah dianggap cukup mewakili layanan-layanan di jaringan *OpenIMSCore* dalam melakukan pengujian terhadap kinerja server PCRF. Pada penelitian ini akan dilihat perubahan performansi aplikasi VoIP dan Video Call baik sebelum dan setelah dilakukan implementasi PCRF di jaringan IMS.

## I.5. Sistematika Penulisan

Penulisan Tugas Akhir ini terdiri dari 4 bab dengan sistematika penulisan sebagai berikut:

### BAB I PENDAHULUAN

Bab ini membahas tentang latar belakang, tujuan, pembatasan masalah, metodologi penelitian dan sistematika penulisan.

### BAB II DASAR TEORI

Bab ini membahas dasar teori mengenai arsitektur IMS, protokol pada IMS core, *Policy Control Rule Function* (PCRF).

### BAB III IMPLEMENTASI *POLICY CONTROL RULES FUNCTION*

Bab ini membahas mengenai Implementasi *Policy Control Rules Function* (PCRF) pada jaringan *OpenIMSCore*.

### BAB IV PENGOLAHAN DATA

Pada bab ini akan dibahas mengenai metode pengambilan data serta pengolahan data hasil percobaan.

### BAB V KESIMPULAN

Menjelaskan kesimpulan dari hasil penelitian yang telah dilakukan.

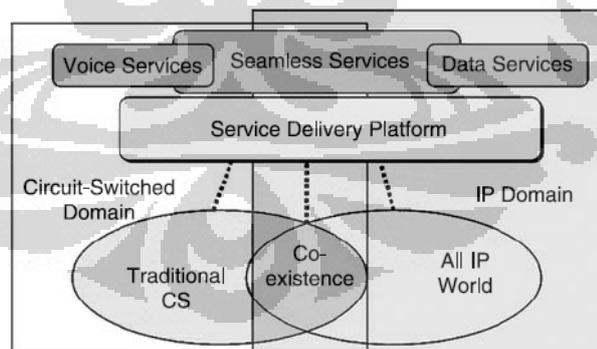
## BAB II

### OpenIMSCore dan Quality of Service

#### II.1. IP Multimedia Subsystem (IMS)

Beberapa tahun terakhir, peningkatan konvergensi pada dunia telekomunikasi dan layanan internet semakin cepat. IP Multimedia Subsystem (IMS) merupakan salah satu layanan yang dapat digunakan pada berbagai type jaringan dengan kondisi berbeda. Jaringan-jaringan tersebut dapat berupa teknologi jaringan akses tetap dan bergerak seperti *Wireless local Area Network (WLAN)*, *Universal Mobile Telecommunications System (UMTS)*, *Digital Subscriber Line (DSL)*, dan lain sebagainya.

IMS dikembangkan oleh *3<sup>rd</sup> Generation Partnership Projects (3GPP dan 3GPP2)* dan mencakup arsitektur layanan yang memungkinkan efisiensi dari layanan multimedia dengan integrasi yang tinggi, menggabungkan web browsing, e-mail, *instant messaging*, VoIP, *video conferencing*, *telephony*, transfer aplikasi multimedia, dan lain sebagainya dibandingkan teknologi jaringan yang lain. Penggabungan antara Internet dengan Layanan telekomunikasi (seperti pada Gambar II.1) membutuhkan dukungan integrasi layanan dari *Service Delivery Platform (SDP)*.



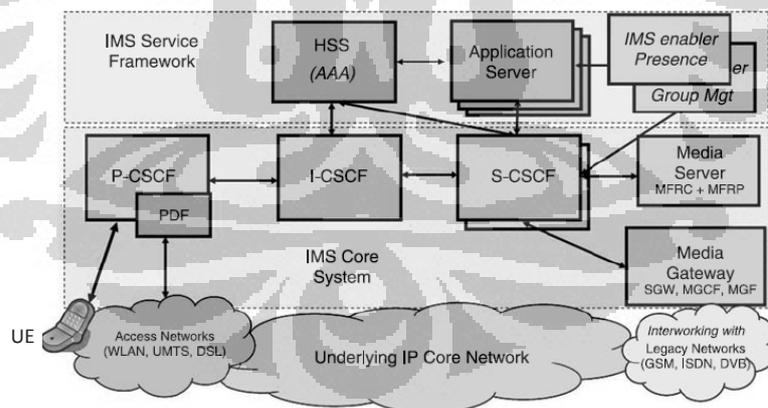
Gambar 2. 1 Konvergensi Jaringan dengan Layanan

IMS di bentuk dengan menggunakan Internet Protokol yang disusun oleh *Internet Engineering Task Force (IETF)*, dimana hal itu di lakukan berdasarkan permintaan dari 3GPP untuk mendukung kebutuhan

telekomunikasi, akuntabilitas, kualitas layanan (QoS), dan lain sebagainya. IMS memiliki banyak elemen fungsional (termasuk SIP server dan proxy) dan membuat, modifikasi dan meng-akhiri sesi multimedia. Prinsip penggunaan IMS adalah untuk memudahkan dalam meng-integrasikan berbagai layanan, walaupun dari pihak lain dan memungkinkan integrasi sebagai layanan yang sah dan di disain untuk interaksi dengan domain *circuit switch*. IMS mengatur QoS charging dalam satu sesi, yang berarti setiap user dalam satu sesi dapat memiliki charging yang berbeda. Hal ini yang menyebabkan IMS sebagai teknologi masa depan yang memiliki layanan komperhensif dan aplikatif yang berorientasi kepada jaringan.

### II.1.1. Arsitektur pada IMS

IMS memiliki beberapa komponen penting, diantaranya IMS Core, IMS Application Platform, IMS *Client* dan *Gateway*. IMS Core merupakan inti dari arsitektur IMS yang terdiri dari *Call Session Control Function (CSCF)*, *Home Subscriber Server (HSS)*, dan *Application Server (AS)*. Komponen-komponen pada IMS dapat dilihat seperti pada Gambar II.2.



Gambar 2. 2 Arsitektur IMS

#### 1. User Equipment (UE)

Merupakan perangkat yang digunakan oleh user IMS dimana perangkat tersebut harus memiliki IMS protokol untuk *signalling*, atau yang biasa disebut dengan SIP, serta memiliki layanan multi media di dalamnya. Koneksi yang dimiliki oleh UE

harus berupa teknologi akses seperti GPRS, WLAN, DSL, dan lain sebagainya.

## 2. P-CSCF

*Proxy-Call Session Control Function (P-CSCF)* adalah titik awal sambungan antara UE dengan IMS Core. Seluruh SIP signaling traffic dari atau ke UE akan melewati P-CSCF. Alamat dari P-CSCF dimiliki oleh UE untuk aktivasi *Packet Data Protocol (PDP)* yang berfungsi sebagai sambungan antara mobile device dengan jaringan pada server yang memungkinkan keduanya saling berkomunikasi. P-CSCF memiliki fungsi untuk otorisasi, mengatur QoS, sambungan emergency, monitoring, serta identifikasi I-CSCF.

## 3. I-CSCF

*Interrogating-Call Session Control Function (I-CSCF)* merupakan *contact point* sebuah jaringan pada operator untuk melayani sambungan yang masuk ke jaringan operator tersebut atau sambungan ke jaringan lain dalam suatu service area. Sebuah jaringan pada operator mungkin saja memiliki lebih dari satu I-CSCF. Fungsi dari I-CSCF adalah sebagai penghubung dengan S-CSCF ketika registrasi SIP, menghitung charging, dan bertindak sebagai *Topology Hiding Inter-working Gateway (THIG)*

## 4. S-CSCF

*Serving-Call Session Control Function (S-CSCF)* bertugas sebagai pengontrol sesi layanan serta mengatur sesi yang dibutuhkan oleh network operator untuk mendukung perlengkapan yang dimiliki. Pada operator jaringan, setiap S-CSCF dapat memiliki fungsi yang berbeda-beda. S-CSCF berfungsi untuk registrasi user serta berinteraksi dengan *service platform* untuk mendukung sebuah service yang disediakan. S-CSCF akan menentukan data yang dibutuhkan dari *Application Server* terkait dengan permintaan sesi SIP untuk mengakses sebuah layanan. Tindakan yang dilakukan oleh S-CSCF

berdasarkan data yang didapat dari *Home Subscriber Server (HSS)*.

#### 5. PDF

*Policy Decision Function (PDF)* adalah sebuah *logical policy decision* yang menggunakan mekanisme IP standar guna mengimplementasikan *Service Based Local Policy (SBLP)* pada layer *IP bearer*. Ini membuat policy decision bergantung pada informasi sesi yang diterima oleh P-CSCF.

#### 6. HSS

*Home Subscriber Server (HSS)* merupakan database utama dari IMS yang berisikan data *IMS User* termasuk informasi penting didalamnya, status dari user serta profil dari *application server*. Kegunaannya mirip dengan *HLR (Home Location Register)* pada sistem 2G, akan tetapi HSS memiliki kelebihan dengan menggunakan dua titik referensi berbasis Diameter (standard interfaces).

#### 7. Application Server (AS)

*Application Servers* merupakan server yang menyediakan service pada IMS. 3GPP tidak menentukan standar mengenai pembuatan dari sebuah aplikasi, yang diatur hanyalah signaling serta *administration interface*, serta protokol-protokol yang berhubungan. Hal tersebut memungkinkan pengembang membuat berbagai macam program yang dapat digunakan pada SIP AS seperti, *legacy Intelligent Network servers*, OSA/ Parlay servers/gateways, atau berbagai jenis program VoIP SIP seperti SIP Servlets, Call Programming Language (CPL), dan lain sebagainya. SIP AS dapat bekerja berdasarkan request yang diberikan oleh S-CSCF dimana S-CSCF menghubungkan langsung beberapa sesi ke SIP AS berdasarkan penyeleksian yang telah dilakukan dengan menggunakan informasi pada HSS.

## II.2. Protocol-protocol di dalam IMS

### II.2.1. 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 bisa 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.

### II.2.2. 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.[6]

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 “\*”.[11]

#### **Session Description**

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

#### **One or more time description**

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

#### **Time description**

t =\* (time the *session* is active)

r            =\* (zero or more repeat times)

**Media description**

m            = (media name and transport address)

i            =\* (media title)

### II.2.3. 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.[6] RTP didesain untuk membawa informasi waktu bersama dengan data yang dikirimkan sehingga RTP digunakan untuk membawa data *audio* maupun *video streams*. RTP terdiri dari suatu data dan *control part* yang disebut RTCP.[5]

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. *Timestamp mere-time* data yang diterima dengan *source timing* dan dengan akurasi yang cukup baik bagi sebagian besar aplikasi multimedia *streaming*.

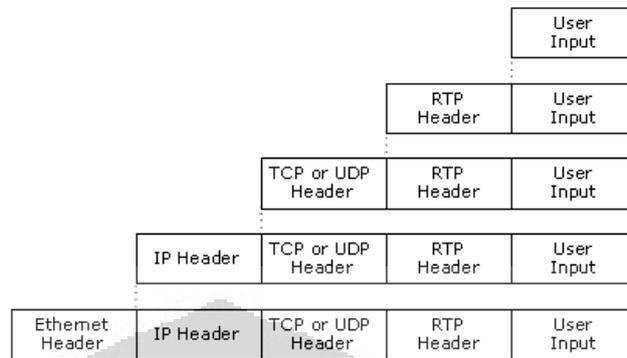
Pada IP network dapat terjadi kehilangan paket atau *packet loss*, untuk itu *sequence number* dimasukkan ke dalam RTP. Penerima paket dapat mengetahui berapa paket yang hilang pada saat pengiriman melalui *sequence number*.

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 *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

**Universitas Indonesia**

pelanggan tidak terlalu lama menunggu (*long pause*) atau *delay*, dikarenakan permintaan *retransmission*.



Gambar 2. 3 Enkapsulasi RTP

Proses enkapsulasi RTP pada Gambar II. 3 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 4 menunjukkan *header RTP* dan Tabel 2.1 menunjukkan jenis-jenis *payload RTP*.

Version	P	X	CC	M	PT	Sequence Number
Timestamp						
SSRC identifier						
CSRC identifiers						
...						
Profile-specific extension header ID					Extension header length	
Extension header						
...						

Gambar 2. 4 RTP Packet Format

- *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 mengindikasikan waktu relatif ketika *payload* disample.
- *Synchronization Source Identifier (SSRCI)*: mengidentifikasi *sender* dari paket RTP.
- *CSRC Contributing Source Identifier*: *field* ini hanya ada jika paket RTP telah dikirim oleh *mixer*.

#### II.2.4. 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. Protokol ini memungkinkan partisipasi 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. 1 RTCP Packet Types [6]

<i>Type</i>	<i>PT id</i>	<i>Purpose</i>
<i>SR</i>	<i>200</i>	<i>Sender report: quality statistics from active senders.</i>
<i>RR</i>	<i>201</i>	<i>Receiver report: quality statistics from receivers that not sender</i>

<b><i>SDES</i></b>	<b><i>201</i></b>	<b><i>Source description item: information on transmitter</i></b>
<b><i>BYE</i></b>	<b><i>203</i></b>	<b><i>Indicates end of participation</i></b>
<b><i>APP</i></b>	<b><i>204</i></b>	<b><i>Application-specific function</i></b>

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

#### **II.2.5. 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*. 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:

- ***Setup***, yaitu *server* mengalokasikan sumber daya kepada sesi klien.
- ***Play***, yaitu *server* 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.

### II.2.6. 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. 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 II. 2 menunjukkan jenis *message* di dalam protocol COPS.

Tabel 2. 2 type message COPS [3]

No.	Type	Form PEP	Form 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

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.

### II.3. Quality of service

*Quality of Service* (QoS) merupakan mekanisme jaringan yang memungkinkan aplikasi-aplikasi atau layanan dapat beroperasi sesuai

dengan yang diharapkan.[7] Jaringan telekomunikasi berbasis IP memiliki beberapa parameter penting yang perlu diperhatikan, yaitu *bandwidth*, *latency* dan *jitter*. Parameter-parameter tersebut sangat berperan dalam berbagai aplikasi yang disediakan oleh layanan telekomunikasi berbasis IP (seperti IMS). *Bandwidth* yang tidak cukup, *latency* yang tidak dapat diprediksi, atau *jitter* yang berlebih dapat menyebabkan peangguna aplikasi komunikasi suara (seperti VoIP atau *IP Telephony*) serta *video streaming* merasa kecewa atas kualitas layanan yang mereka dapatkan.

Penggunaan mekanisme *Quality of Service* (QoS) memberikan kemampuan kepada penyedia layanan dalam memprediksi *bandwidth*, *latency*, dan *jitter* serta menyesuaikannya 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.

### II.3.1. 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.

### II.3.2. Jenis-Jenis Quality Of Service

Terdapat 3 tingkat QoS yang umum dipakai, yaitu *best-effort service*, *integrated service* dan *differentiated service*. [10]

#### 1. *Best Effort*

*Best-effort service* merupakan jenis servis dimana media *server* melakukan semua usaha agar dapat mengirimkan sebuah paket ke suatu tujuan. Penggunaan *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

**Universitas Indonesia**

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.[11] 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.[10] 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 Forwarding* merupakan kelas PHB yang memiliki prioritas paling utama karena mempunyai karakteristik *low delay*, *low loss* dan *low jitter*. Karakteristik tersebut cocok untuk aplikasi *voice*, *video call* dan servis-servis yang *realtime*. Nilai DSCP yang direkomendasikan untuk kelas

*Expedited Forwarding* (EF) adalah 101110<sub>B</sub> 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. 3.

Tabel 2. 3 Assured Forwarding (AF) Behaviour Group [12]

	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<sub>B</sub>. 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 4 menunjukkan level prioritas berdasarkan aplikasi di dalam metode *diffserv*.

Tabel 2. 4 QoS Priority Level

<i>QoS Level</i>	<i>Priority Level</i>	<i>Example Services</i>	<i>Session information</i>
<i>EF</i>	<i>1</i>	<i>IMS Signaling</i>	<i>QoS level BW Capacity</i>
	<i>2</i>	<i>VoIP</i>	
	<i>3</i>	<i>Video Conferencing</i>	
<i>AF4</i>	<i>4</i>	<i>Audio and Video Streaming</i>	<i>Qos level BW availability</i>
<i>AF3</i>	<i>5</i>	<i>Transactional</i>	

		<i>services</i>	<i>PEC/ PEV</i>
<i>AF2</i>	<i>6</i>	<i>Web Browsing</i>	<i>SFB</i> <i>Service BW</i>
<i>AF1</i>	<i>7</i>	<i>Telnet</i>	
<i>BE</i>	<i>8</i>	<i>e-mail</i>	
	<i>9</i>	<i>Web Browsing</i>	

### II.3.3. Quality of Service di Jaringan IMS

Pengaturan QoS di IMS memerlukan interaksi yang cukup kompleks antara pelanggan dengan komponen IMS. 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*.

### II.4. Policy Control Rules Function (PCRF)

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*. [2] PDF bertugas untuk menginformasikan GGSN mengenai karakteristik atau otoritas seorang pelanggan dalam suatu sesi multimedia. [5] 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).

Perkembangan paling signifikan yang terdapat pada 3GPP *Release 7* adalah *policy and charging* yang mendefinisikan arsitektur konvergensi telekomunikasi terbaru yang meng-optimalkan interaksi antara fungsi dari *Policy* dan *Rules*. Perkembangan tersebut meliputi sebuah *node* jaringan terbaru pada PCRF yang merupakan rangkaian dari *Policy*.

## BAB III

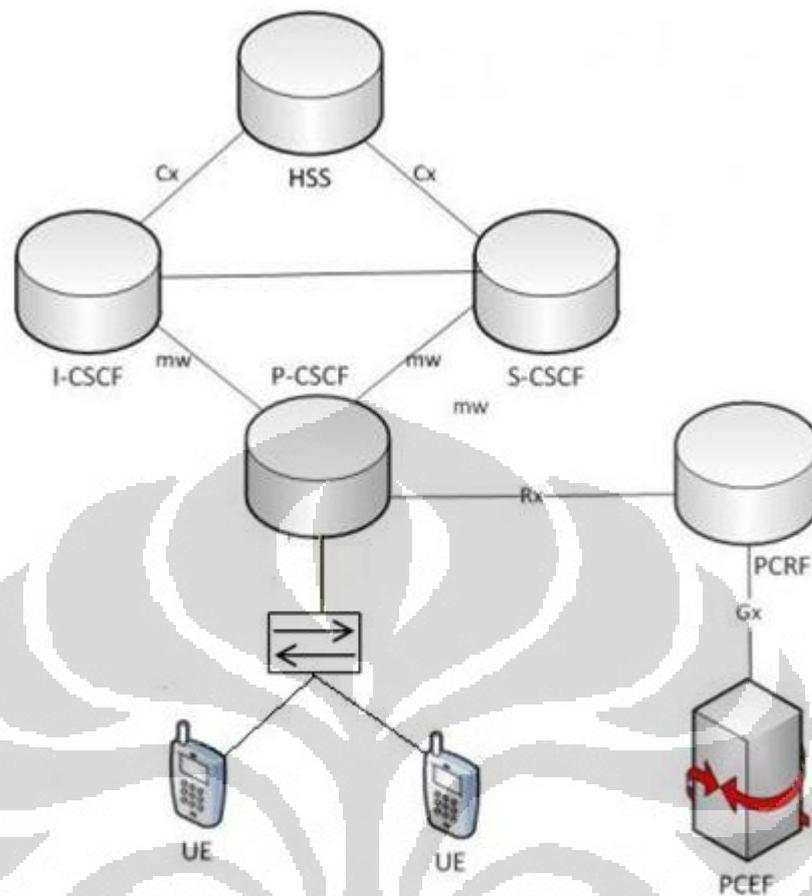
### Implementasi *Policy Control Rule Function*

Berdasarkan teori yang dibahas pada bab 2, implementasi *Quality of Service* dapat dijalankan di jaringan *OpenIMSCore* dengan menggunakan bantuan sebuah protokol yang bernama Diameter. Protokol *Diameter* berperan untuk memberikan *Authentication, Authorization and Accounting* (AAA) protokol untuk aplikasi yang berkaitan dengan network access or IP mobility. Konsep dasar dari protokol tersebut adalah untuk memberikan layanan kepada teknologi akses terbaru.

*Diameter* dikembangkan agar dapat memberikan layanan AAA baik dalam satu jaringan maupun berbeda jaringan. *Diameter* sendiri merupakan pengembangan dari protokol yang sudah ada sebelumnya (seperti *RADIUS*). Protokol yang ada pada *Diameter* memberikan beberapa fasilitas seperti:

- Connection and session management
- User authentication and capabilities negotiation
- Reliable delivery of attribute value pairs (AVPs)
- Agent support for proxy, redirect, and relay servers
- Extensibility, through addition of new commands and AVPs
- Basic accounting services[9]

Protokol *Diameter* akan secara terus menerus memberikan perintah untuk saling memberikan informasi antara *Diameter Clients* dan *Server*. Aplikasi yang menggunakan protokol *Diameter* dalam pengaturan *QoS* di *OpenIMSCore* adalah PCRF. Server PCRF akan terhubung langsung dengan P-CSCF yang merupakan titik awal sambungan antara UE dengan *OpenIMSCore*. Konfigurasi sambungan antar PCRF dengan *OpenIMSCore* tampak seperti pada Gambar 3.1. Server PCRF akan terhubung secara langsung dengan server P-CSCF yang akan berfungsi sebagai pengatur kebijakan *QoS*, khususnya *bandwidth* pada jaringan *OpenIMSCore*.



Gambar 3. 1 Konfigurasi PCRF dengan OpenIMSCore

Percobaan implementasi *PCRF* pada jaringan *OpenIMSCore* yang dilakukan pada penelitian ini akan membandingkan kualitas layanan yang disediakan oleh *OpenIMSCore*, terutama layanan VoIP dan Video Call. Hal yang akan diamati adalah perbandingan parameter-parameter *Quality of Service* seperti *delay*, *jitter*, dan *packet loss* yang terjadi pada jaringan, sebelum dan sesudah diimplementasikan *PCRF* di dalam jaringan *Open IMS Core*.

Pada percobaan ini, terdapat beberapa komponen penting yaitu, sebuah server *OpenIMSCore*, dua buah laptop sebagai *IMS client*, dan satu buah *switch merk cisco*. Secara garis besar, skenario percobaan yang akan dijalankan adalah sebagai berikut:

1. Seorang client akan menghubungi client lainnya yang masih dalam satu jaringan yang tidak menggunakan *PCRF*. Saat sambungan telah terjadi,

packet-packet yang dikirimkan akan direkam menggunakan *wireshark* yang berfungsi sebagai packet tracer.

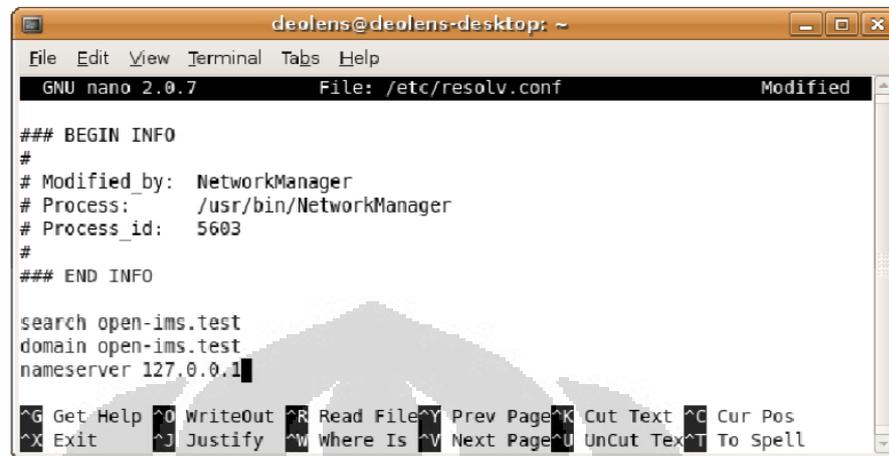
2. Mengimplementasikan server PCRF pada jaringan *OpenIMSCore* sebagai *policy control*.
3. Client pertama akan menghubungi client lainnya pada satu jaringan yang telah dilengkapi oleh server PCRF. Kemudian packet-packet yang dikirimkan saat melakukan sambungan akan direkam menggunakan aplikasi *wireshark*.
4. Aplikasi *wireshark* akan merekam semua pengiriman packet data yang terjadi pada UE. Setelah semua packet data terekam dengan menggunakan *wireshark*, dapat diketahui delay serta jitter yang terjadi saat sambungan dengan menggunakan fitur *stream analysis* untuk packet RTP.

### III.1. Konfigurasi dan Instalasi IMS Core

Untuk melakukan simulasi IMS Bench SIPp maka dibutuhkan sebuah core IMS. Pada simulasi ini digunakan Open IMS Core yang sifatnya open source dengan menggunakan IP yang berbeda untuk setiap server yang ada. Langkah-langkah yang perlu diperhatikan dalam melakukan instalasi *OpenIMSCore* instalasinya adalah sebagai berikut:

1. Update repositori ubuntu. Hal tersebut sangat diperlukan agar seluruh package-package yang dibutuhkan dalam melakukan instalasi tersedia.
2. Download dan install Subversion package yang terdapat pada [openimscore.org](http://openimscore.org). File-file tersebut merupakan komponen penting yang dikembangkan oleh *Franhoufer* untuk melakukan instalasi server *OpenIMSCore* sehingga diperlukan repository dari update terakhir.
3. Saat melakukan instalasi dibutuhkan beberapa package tambahan seperti *sun-java6-jdk mysql-server, libmysqlclient15-dev, libxml2, libxml2-dev, bind9*, dan lain sebagainya. File-file tersebut dibutuhkan untuk melakukan subtraksi file instalasi *OpenIMSCore* yang telah di download sebelumnya serta membuat *database*.
4. Tentukan domain serta name server pada file “*resolv.conf*” dan file “*hosts*” sehingga dapat diakses oleh jaringan yang digunakan. Jika domain serta server name tidak ditentukan (diatur menjadi *loopback*)

maka webserver tidak dapat diakses oleh jaringan. Selain itu juga, server CSCF serta HSS tidak dapat diakses oleh UE.



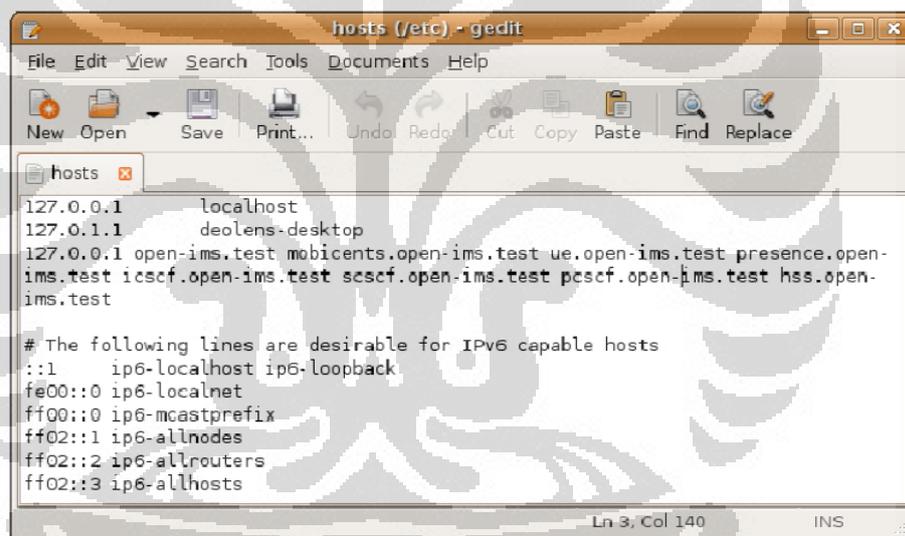
```

deolens@deolens-desktop: ~
File Edit View Terminal Tabs Help
GNU nano 2.0.7 File: /etc/resolv.conf Modified

### BEGIN INFO
#
# Modified_by: NetworkManager
# Process: /usr/bin/NetworkManager
# Process_id: 5603
#
### END INFO

search open-ims.test
domain open-ims.test
nameserver 127.0.0.1
  
```

Gambar 3. 2 Tampilan file resolv.conf localhost



```

hosts (/etc) - gedit
File Edit View Search Tools Documents Help
New Open Save Print... Undo Redo Cut Copy Paste Find Replace

hosts
127.0.0.1 localhost
127.0.1.1 deolens-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

# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts
Ln 3, Col 140 INS
  
```

Gambar 3. 3 Tampilan File Host localhost

5. Lakukan restart bind9 agar seluruh setting yang telah dilakukan dapat berjalan dengan baik.
6. Komponen-komponen OpenIMSCore harus dijalankan pada masing-masing terminal untuk setiap CSCF dan HSS.

### III.2. Instalasi Policy Control Rule Function (PCRF)

Setelah berhasil dalam melakukan instalasi OpenIMSCore, langkah selanjutnya adalah implementasi PCRF. Terdapat beberapa hal yang perlu diperhatikan dalam melakukan implementasi PCRF agar PCRF dapat berjalan sebagaimana mestinya, diantaranya:

1. Download the extended ser\_ims untuk setting QoS dari alamat [http://svn.berlios.de/svnroot/repos/openimscore/ser\\_ims/trunk](http://svn.berlios.de/svnroot/repos/openimscore/ser_ims/trunk). hal ini diperlukan agar seluruh setting QoS pada OpenIMSCore yang akan dilakukan tidak mengganggu konfigurasi utama dari OpenIMSCore.
2. Melakukan instalasi uctpcrf tarball yang dapat di-download dari [https://developer.berlios.de/project/showfiles.php?group\\_id=784](https://developer.berlios.de/project/showfiles.php?group_id=784). uctpcrf merupakan file instalasi serta konfigurasi untuk server PCRF dan PCEF. Seluruh file konfigurasi yang akan dipergunakan dalam melakukan setting QoS ada pada direktori ini.
3. Setelah berhasil melakukan instalasi PCRF perlu diperhatikan beberapa package tambahan seperti *libxerces-java*, *libcommons-httpclient-java*, dan *tomcat5.5*. *Package-package* tersebut nantinya diperlukan ketika menjalankan *web setting* dari PCRF. *Web setting* tersebut berfungsi sebagai *interface* dalam pengaturan QoS di PCRF.
4. Konfigurasi DNS Server untuk menentukan alamat dari PCRF dan PCEF. Alamat tersebut dipakai oleh P-CSCF dalam menentukan lokasi PCRF yang akan dituju. Setelah berhasil melakukan konfigurasi dari DNS *server* jangan lupa melakukan *Restart* “bind9” agar konfigurasi tersebut dapat berjalan.
5. Set the JAVA\_HOME environment variable
6. Pengaturan Web Interface.
  - Web management folder yang ada pada direktori uctpcrf perlu di-copy ke komputer *server* sehingga web setting interfaces tersebut hanya dapat dijalankan oleh computer server.
  - Mengatur policy dari uct\_web.pcm serta properties dari PCRF dan PCEF sehingga dikonfigurasi yang ada dapat diatur melalui web management dari PCRF.

7. Copy konfigurasi file yang ada pada direktori `uctpcrf/config` seperti `pcscf.qos.cfg`, `pcscf.qos.rtp.cfg`, `pcscf.qos.xml`, `pcscf.qos.sh`, dan lain sebagainya. File-file tersebut memiliki fungsi agar P-CSCF dapat menemukan lokasi dari PCRF serta mengaktifkan protokol *DIAMETER* yang ada pada *OpenIMSCore*.
8. Menjalankan seluruh server *OpenIMSCore* pada masing-masing terminal. Khusus untuk menjalankan P-CSCF menggunakan file konfigurasi `pcscf.qos.sh` karena pada file tersebut telah terdapat seluruh setting yang diperlukan untuk menghubungkan antara P-CSCF dengan PCRF. Untuk server lain pada *OpenIMSCore* jalankan seperti biasanya.
9. Jalankan PCRF and PCEF harus dijalankan pada masing-masing terminal.
10. Run the Web Management Interface in a suitable web browser ([http://localhost:8180/uct\\_web\\_pcm/](http://localhost:8180/uct_web_pcm/)) untuk melakukan pengaturan QoS *Bandwidth* pada server PCRF.

### III.3. Sambungan antara PCRF dengan P-CSCF

Setelah melakukan instalasi *OpenIMSCore* serta server PCRF dan PCEF sambungkan server tersebut melalui node *Rx interfaces*. *Rx interfaces* akan menghubungkan P-CSCF dengan PCRF selama *call initiation* dan *renegotiation* untuk pengaturan *policy*. Melalui *Rx interface* yang ada pada *IMSCore*, PCRF dapat melakukan beberapa fungsi sebagai berikut:

- Melakukan konfirmasi *policy* terhadap request yang dikirim oleh UE
- Menentukan jalur layanan untuk sambungan UE sesuai dengan QoS yang didapatkan.
- Mengetahui informasi *charging* ketika diperlukan.

Untuk melakukan sambungan antara PCRF dengan P-CSCF perlu dilakukan konfigurasi tambahan untuk mengaktifkan *Rx interfaces* yang ada pada *IMSCore*, dengan menambahkan perintah dibawah ini pada file konfigurasi P-CSCF:

- `loadmodule "/opt/OpenIMSCore/ser_ims/modules/cdp/cdp.so"`

- `modparam("cdp", "config_file", "/opt/OpenIMSCore/pcscf.xml")`
- `loadmodule "/opt/OpenIMSCore/ser_ims/modules/cdp_avp/cdp_avp.so"`
- `modparam("pcscf", "forced_qos_peer", "pcrf.open-ims.test")`
- `modparam("pcscf", "qos_release7", 1)`
- `modparam("pcscf", "pcc_ipv4_for_signaling", "192.168.10")`
- `modparam("pcscf", "pcc_port_for_signaling", 4060)`

Perintah tersebut diatas berfungsi untuk meng-aktifkan module `cdp.so` dan `cdp_avp.so` pada server PCRF. Module-module tersebut merupakan konfigurasi protocol *DIAMETER*. Tanpa menggunakan module ini, maka P-CSCF tidak akan mengirimkan *message* ke PCRF karena protocol yang ada tidak mendukung tindakan tersebut. Selain meng-aktifkan *module* tersebut di atas, perintah tersebut juga meng-aktifkan *Rx interfaces* untuk *Policy Control dan Charging (PCC)* sehingga protocol *DIAMETER* antara P-CSCF dan PCRF dapat saling berkoordinasi. Sehingga perintah-perintah tersebut di atas diperlukan agar P-CSCF dapat mengetahui lokasi alamat dari PCRF.

## BAB IV

### HASIL PERCOBAAN DAN ANALISA DATA

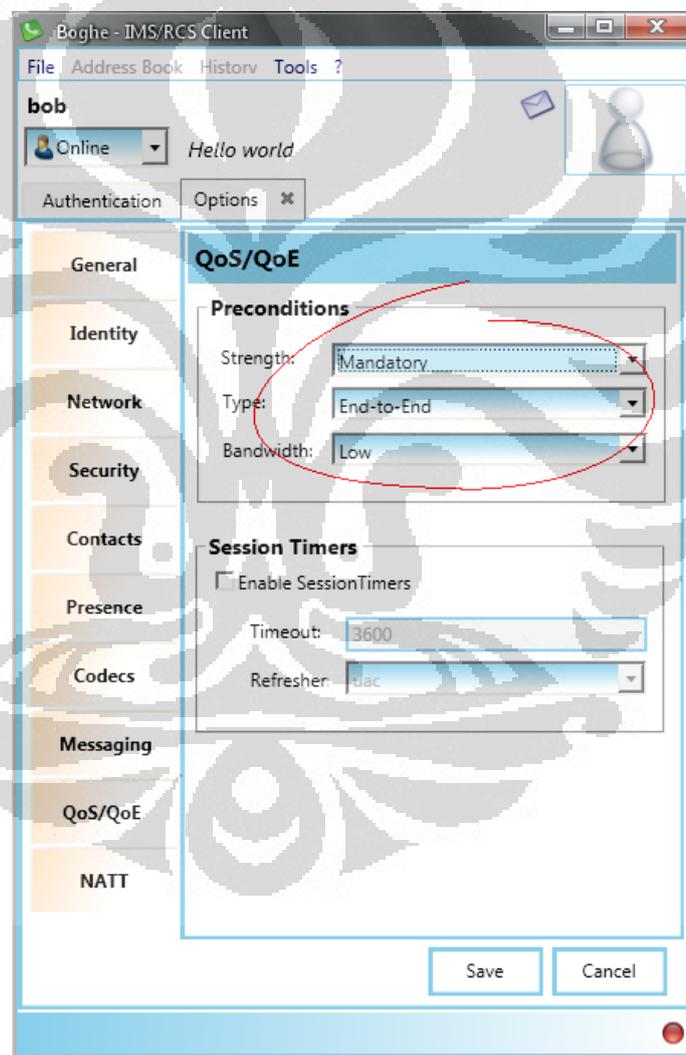
Pada bab ini akan dibahas mengenai pengukuran parameter-parameter QoS yang terdapat pada server PCRF untuk melihat perubahan performansi layanan multimedia. Pengukuran dilakukan dengan beberapa perubahan pengaturan *bandwidth* pada *user client*. Pengukuran dilakukan untuk mengamati perubahan performansi pada layanan *VoIP* dan *Video Call* ketika menggunakan server PCRF yang terhubung pada *OpenIMSCore*. Pengukuran untuk perubahan performansi pada aplikasi *VoIP* dan *Video Call* dengan memberikan variasi *bandwidth* yang disediakan oleh aplikasi *Boghe IMS Client*. Dengan melakukan variasi *bandwidth* tersebut diharapkan dapat terlihat perbedaan pada kualitas layanan antara menggunakan server PCRF dengan tanpa server.

Pengukuran dilakukan untuk mengamati perubahan kuantitatif dari parameter-parameter QoS yaitu *delay*, *jitter*, dan *packet loss* yang diamati dengan menggunakan bantuan program *packet tracer* *wireshark* pada komputer klien. Program *wireshark* akan merekam paket-paket yang ada di jaringan kemudian melakukan perhitungan terhadap paket-paket tersebut. Pengukuran yang dilakukan pada penelitian ini difokuskan untuk mengamati paket-paket yang menggunakan protokol *Real Time Protocol* (RTP) karena kedua jenis layanan yang akan dicoba menggunakan protokol tersebut.

Pada penelitian ini menggunakan *codec* H.263 untuk video pada layanan *Video Call* yang membutuhkan *bandwidth* 20-25 kbps. Sedangkan untuk layanan voice menggunakan *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 ( $260 \text{ bits}/20 \text{ ms}=13.000 \text{ bits/s}=13 \text{ kbits/s}$ ).

#### IV.1. Pengambilan dan Pengolahan Data

Pengukuran performansi layanan VoIP dan Video Call dilakukan dengan menggunakan aplikasi Boghe IMS Client pada dua variasi bandwidth yaitu bandwidth low dan high dengan tambahan server PCRF serta tanpa tambahan server PCRF. Setiap variasi bandwidth dilakukan pengukuran sebanyak 3 kali pengulangan dan kemudian hasil yang didapat akan dirata-ratakan untuk mendapatkan hasil yang lebih akurat. Setting bandwidth dilakukan pada sisi client (*Boghe IMS Client*) sesuai dengan gambar dibawah ini



Gambar 4. 1 Setting Qos yang dilakukan pada UE

Hasil dari setiap pengukuran yang didapatkan kemudian dihitung nilai rata-rata dari masing-masing parameter QoS. Tabel di bawah ini

Universitas Indonesia

menunjukkan hasil pengambilan data parameter QoS pada layanan *VoIP* serta *Video Call*.

**Tabel 4. 1** *Delay pada aplikasi VoIP*

Percobaan ke-	tanpa server PCRF		dengan server PCRF	
	BW Low (ms)	BW High (ms)	BW Low (ms)	BW High (ms)
1	19,99889	19,99869	19,96144	19,96806
2	19,92312	19,99149	19,97106	19,96875
3	19,99168	19,92327	19,97231	19,97016
4	19,96856	19,96658	19,96357	19,96251
5	19,96963	19,96862	19,98532	19,96863
6	19,97023	19,97209	19,96589	19,96235
7	19,98663	19,98369	19,96324	19,97155
8	19,97856	19,97898	19,97367	19,96333
9	19,99769	19,99334	19,96243	19,9619
10	19,98537	19,98864	19,96535	19,96577
Rata-rata	19,97704	19,97616	19,96827	19,96630

**Tabel 4. 2** *Jitter pada aplikasi VoIP*

Percobaan ke-	tanpa server PCRF		dengan server PCRF	
	BW Low (ms)	BW High (ms)	BW Low (ms)	BW High (ms)
1	14,98793	14,96821	16,85369	16,83512
2	14,97962	14,97268	16,95687	16,82689
3	14,98598	14,97325	16,90342	16,84152
4	14,97768	14,98654	17,05385	16,84731
5	14,98158	14,97894	17,07871	16,85371
6	14,97963	14,99154	17,10358	16,85691
7	14,98061	14,98398	17,02898	16,85051
8	14,97866	14,97642	16,97925	16,84091
9	14,98256	14,98902	16,95439	16,84411
10	14,97671	14,98146	17,00412	16,86011
Rata-rata	14,98141	14,97940	16,98377	16,84469

Tabel 4. 3 Delay pada aplikasi Video Call

Percobaan ke-	tanpa server PCRF		dengan server PCRF	
	BW Low (ms)	BW High (ms)	BW Low (ms)	BW High (ms)
1	19,97596	19,96892	19,96532	19,95562
2	19,97656	19,97358	19,96839	19,96025
3	19,97737	19,96825	19,94095	19,99293
4	19,97804	19,96958	19,92166	19,98691
5	19,97874	19,98924	19,89729	20,02565
6	19,96945	19,97891	19,90948	20,04422
7	19,98015	19,96857	19,86074	19,62875
8	19,98086	19,97824	19,93385	19,95153
9	19,97156	19,98795	19,87292	19,10185
10	19,98227	19,98657	19,88511	20,11884
Rata-rata	19,97710	19,97698	19,91896	19,87666

Tabel 4. 4 Jitter pada aplikasi Video Call

Percobaan ke-	tanpa server PCRF		dengan server PCRF	
	BW Low (ms)	BW High (ms)	BW Low (ms)	BW High (ms)
1	10,92596	11,89082	17,17856	16,44259
2	10,99568	11,89492	17,18025	16,44890
3	11,26958	11,89995	17,18029	16,43193
4	11,08736	11,60436	17,18143	16,43048
5	11,37917	11,90892	17,18575	16,40916
6	11,25098	11,71349	17,18316	16,41982
7	11,32279	11,31805	17,18229	16,39853
8	11,09465	11,92262	17,18489	16,41449
9	11,26641	11,42718	17,18662	16,40383
10	11,07822	11,23175	17,18402	16,42515
Rata-rata	11,16708	11,68121	17,18258	16,42219

Dari hasil data yang didapat terlihat bahwa antara delay yang didapat ketika tidak menggunakan server PCRF pada layanan VoIP

**Universitas Indonesia**

terlihat memiliki sedikit perbedaan. Ketika server PCRF belum dipergunakan rata-rata delay yang terjadi antar bandwidth *low* dan *high* hanya sedikit berbeda yaitu 0,00085 ms. Hal tersebut menunjukkan bahwa ketika server PCRF belum dipergunakan dan *precondition* yang diperlukan untuk pengaturan QoS (*strength* dan *type*) tidak diatur, maka *setting* bandwidth yang kita atur tidak akan memberikan pengaruh terhadap kualitas layanan yang dipergunakan.

Ketika server PCRF sudah di jalankan terlihat bahwa perbedaan delay yang sedikit lebih besar antara bandwidth *low* dan *high* yaitu 0,00197 ms sedikit lebih besar dibandingkan dengan ketika server PCRF belum dijalankan. Hal tersebut menunjukkan bahwa *setting bandwidth* yang dilakukan pada *user client* memiliki pengaruh terhadap layanan yang diberikan. Hasil *delay* yang didapat setelah menjalankan server PCRF-pun terlihat sedikit lebih baik dibandingkan dengan ketika server PCRF belum dijalankan.

Hasil dari jitter yang didapat juga tidak menunjukkan banyak perbedaan ketika setting QoS belum dipergunakan pada *user client* yaitu 0,00201 ms pada layanan VoIP. Hal tersebut juga menunjukkan bahwa *setting bandwidth* yang dilakukan pada *user client* tidak memberikan dampak sama sekali terhadap QoS dari layanan yang dipergunakan. Sedangkan ketika server PCRF sudah dijalankan menunjukkan hasil yang cukup berbeda antara *bandwidth low* dengan *bandwidth high* sehingga dapat dikatakan *setting bandwidth* sudah berjalan sesuai dengan yang diinginkan.

Berbeda halnya dengan hasil dari *delay* yang didapat, hasil yang didapat untuk *jitter* menunjukkan perbedaan yang signifikan ketika server PCRF sudah dijalankan. Seperti yang sudah diketahui bahwa *jitter* menunjukkan variasi *delay* yang terjadi pada layanan yang dipergunakan. Perbedaan *jitter* yang cukup jauh tersebut menunjukkan bahwa penggunaan PCRF memberikan delay yang lebih bervariasi dibandingkan ketika belum menggunakan server PCRF. Hal tersebut tentu saja menunjukkan QoS yang diberikan oleh server PCRF tidak cukup baik.

Sedangkan hasil didapat dari penggunaan layanan *video call* menunjukkan hasil yang hampir menyerupai hasil yang didapat dari percobaan *VoIP*. Pada percobaan ini juga terlihat error yang menyerupai error yang didapat pada percobaan dengan menggunakan layanan *VoIP* dimana nilai *jitter* yang didapat seharusnya lebih baik ketika server PCRF sudah dijalankan.

#### IV.2. Analisa error pada server PCRF

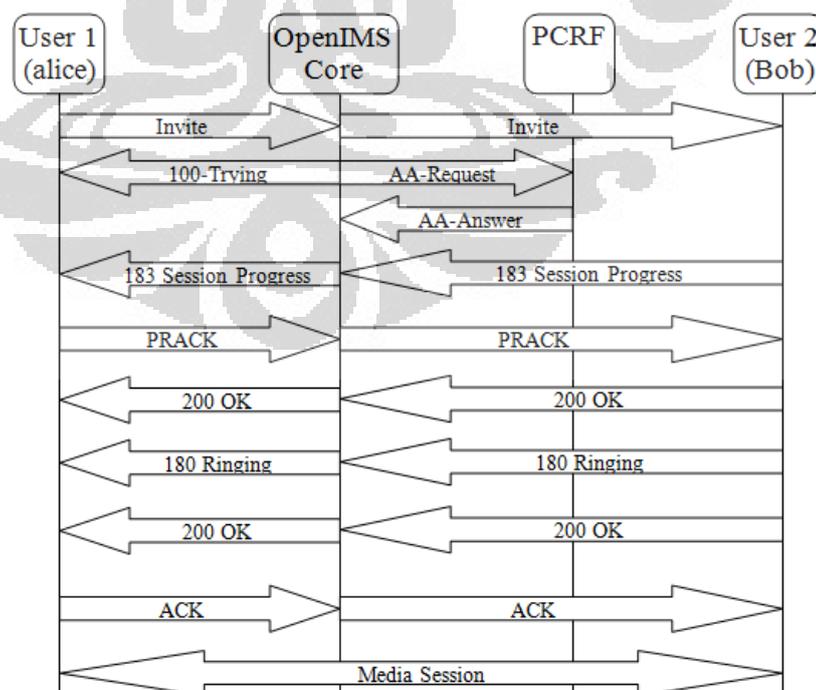
Dari hasil percobaan yang telah dilakukan dengan menggunakan layanan *VoIP* dan *Video Call*, terlihat bahwa terjadi error yang menyebabkan hasil dari implementasi server PCRF tidak memberikan QoS yang jauh berbeda dengan tanpa server PCRF. Parameter QoS dengan menggunakan server PCRF untuk *delay* memberikan hasil yang sedikit lebih baik dibandingkan dengan hasil yang didapat tanpa menggunakan server PCRF, akan tetapi hasil *jitter* yang didapat menunjukkan penurunan kualitas QoS dan pada terminal PCRF terlihat log error sebagai berikut:

```
# INFO: Received AAR DIAMETER message from P-CSCF
# DEBUG: Session ID: pscf.open-ims.net;2882060697;1
# DEBUG: Orig Realm: open-ims.net
# DEBUG: Dest Realm: open-ims.net
# DEBUG: Media Component Number: 1
# DEBUG: Flow Number: 1
# DEBUG: Flow Description: permit out ip from 192.168.102.2 to
192.168.102.3
# ADEBUG: Source IP: 192.168.102.2
# DEBUG: Dest IP: 192.168.102.3
# Exception in thread "Thread-4" java.util.NoSuchElement
# Exception
# at
# java.util.StringTokenizer.nextToken(StringTokenizer.java:332)
# at
```

```
# de.fhg.fokus.pcrf_diam.rx.AARCommandListener.recvMessage(AARComm
andListener.java:218)
# at
# de.fhg.fokus.diameter.DiameterPeer.DiameterWorker.run(DiameterWorker.j
ava:100).
```

Dari error tersebut di atas terlihat bahwa protocol *DIAMETER* pada PCRF tidak dapat merespon request yang dikirimkan oleh P-CSCF. Request tersebut berupa *Authentication, Authorization Request (AAR)* yang dikirimkan oleh *Rx interfaces* yang ada pada P-CSCF. Setelah diamati lebih lanjut, diketahui bahwa *file error* yang ditunjukkan pada *message error* di atas terletak pada instalasi PCRF. Hal tersebut menyebabkan *port* dari *protocol DIAMETER* pada sisi PCRF tidak dapat berjalan sesuai dengan fungsinya.

Skenario yang terjadi adalah ketika salah satu *user* akan melakukan panggilan maka *user client* tersebut mengirimkan message request ke P-CSCF yang kemudian akan diteruskan menuju ke I-CSCF. Selain meneruskan *message request* tersebut ke I-CSCF, P-CSCF juga akan mengirimkan AAR ke PCRF.



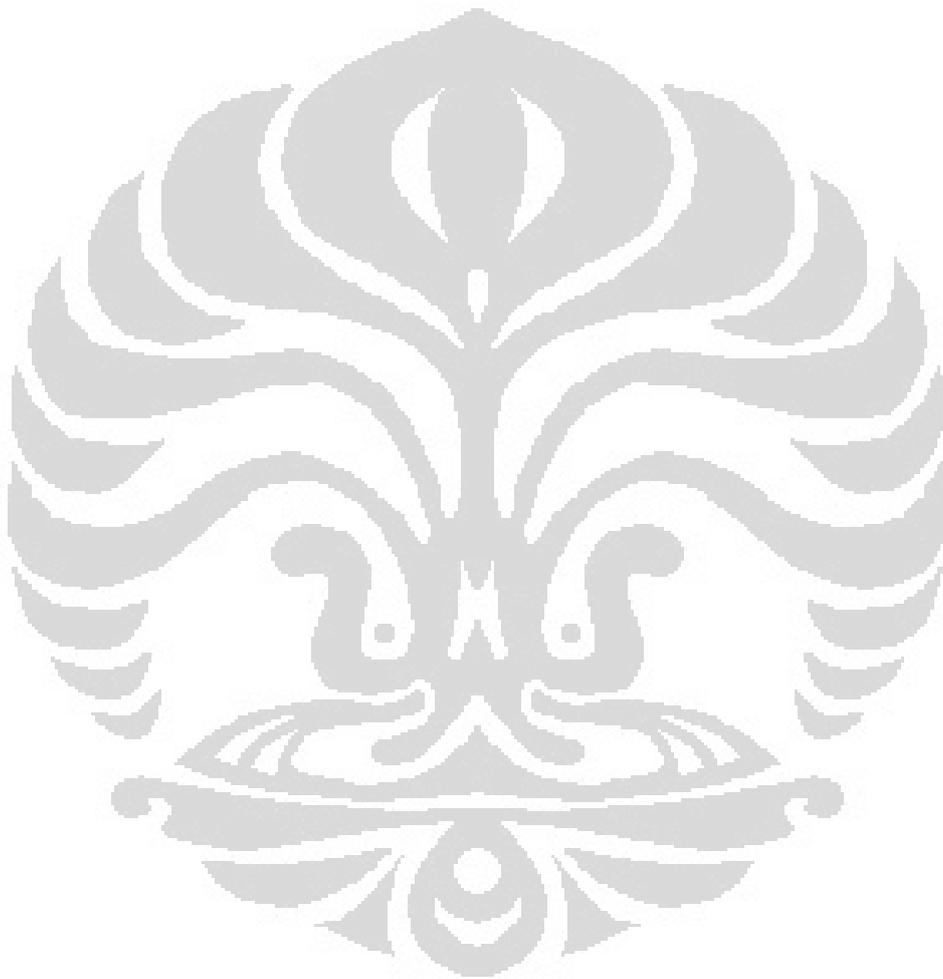
Gambar 4. 2 Message flow untuk autentifikasi layanan dengan PCRF

AAR yang dikirim dari P-CSCF ke PCRF bertujuan untuk melakukan autentifikasi QoS yang berhak didapatkan oleh UE. AAR *message* tersebut membawa informasi penting berupa alamat IP UE, *Specific Action AVP (Attribute Value Pair)*, *Subscription ID*, *Supported Features*, dan lain sebagainya.[1] Setelah menerima AAR *message* dari P-CSCF, maka PCRF akan memeriksa *request* yang diminta dengan PCEF apakah *request* tersebut sesuai dengan kualitas layanan yang berhak diterima oleh UE yang melakukan *request* serta UE yang akan dihubungi. Setelah mengetahui QoS yang berhak didapat oleh *user client* maka PCRF akan mengirimkan *Authentication, Authorization Answers (AA-Answer)* ke P-CSCF. Namun terjadi *error* ketika PCRF akan memberikan jawaban ke P-CSCF.

*Error* yang terjadi tersebut menyebabkan PCRF tidak dapat mengirimkan AA-Answer ke P-CSCF. Dikarenakan tidak mendapatkan jawaban AA-Answer *message* setelah melewati batas *responding time* yang diberikan maka P-CSCF akan mengirimkan *Session Termination Request (STR) request message* ke PCRF. AA-Answer berisikan informasi mengenai *Origin Host*, *Origin Realm*, *Result Code*, *Access-Network-Charging-Identifier*, *Access-Network-Charging-Address*, dan lain sebagainya. [1] *Request message* tersebutlah yang menyebabkan *error* ketika user akan melakukan panggilan sehingga ketika *user client* melakukan sambungan server PCRF tidak berjalan sesuai dengan fungsinya. Hal tersebut dapat terlihat dari *delay* yang terjadi ketika UE melakukan sambungan baik *VoIP* maupun *Video Call*, *delay* yang didapat antara sebelum menjalankan server PCRF dan setelah menjalankan server PCRF menunjukkan hasil yang hampir sama.

Gagalnya server PCRF dalam memberikan AA-Answer *message* dapat disebabkan oleh file *cdp* yang tidak dapat men-generate AA-Answer *message* tersebut sehingga P-CSCF tidak dapat menerima AA-Answer *message*. Tidak berhasilnya *cdp* men-generate AA-Answer *message* tidak dapat meng-copy *Origin Host* dan *Origin Realm*, yang

diperlukan untuk men-*generate* AA-*Answer message*, dari AAR *message*.



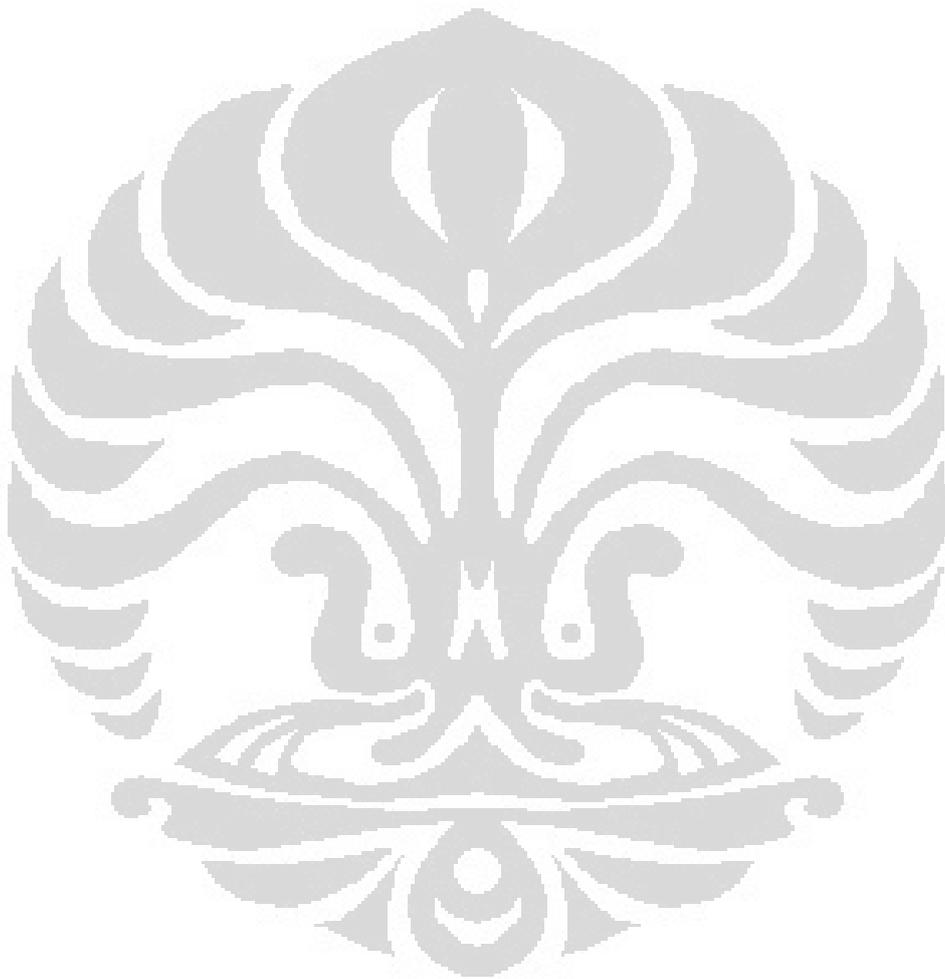
## **BAB V**

### **KESIMPULAN**

Hasil percobaan yang dilakukan pada skripsi yang berjudul “Implementasi *Policy Control Rule Function* pada Jaringan *OpenIMSCore*” dapat disimpulkan beberapa hal, diantaranya:

- 1) *Policy Control Rule Function* (PCRF) pada *OpenIMSCore* berperan sebagai pengatur kebijakan terhadap aplikasi layanan telekomunikasi yang diberikan oleh *OpenIMSCore*.
- 2) PCRF berperan secara langsung dalam pengaturan *Quality of Service* (QoS) pada layanan *OpenIMSCore*.
- 3) Delay yang didapat dari hasil percobaan VoIP dan Video Call pada *OpenIMSCore* yang dilakukan dengan menggunakan PCRF terlihat sedikit perbedaan dengan waktu delay yang didapat dari percobaan tanpa menggunakan PCRF yaitu sekitar 0,00932 ms untuk layanan VoIP dan 0,07923 ms untuk layanan *video call*.
- 4) Hasil percobaan pengukuran jitter yang didapatkan menunjukkan bahwa dengan menggunakan server PCRF maka variasi delay yang didapatkan menjadi semakin besar dengan kenaikan rata-rata sekitar 12,9 % untuk layanan VoIP dan 47,1% untuk layanan *video call*.
- 5) Saat dilakukan percobaan terjadi error yang disebabkan adanya *request message* yang tidak sesuai dengan alur yang seharusnya. Pada *request message* tersebut, P-CSCF mengirimkan *Session Termination Request* dikarenakan PCRF tidak dapat men-*generate AA-Answer message* sebagai balasan dari *request* yang dikirimkan oleh P-CSCF sebelumnya.

- 6) Error yang terjadi pada poin 5 dapat diperbaiki dengan melakukan perbaikan pada protokol *DIAMETER* yang ada pada instalasi PCRF.



## Daftar Referensi

- [1] 3GPP TS 29.214 V8.8.0
- [2] B. Raouyane, M. Bellafkih, D. Ranc,” *Qos Management in IMS : Diffserv Model,*” *IEEE ICC 2009 proceedings*, 2009.
- [3] Based on information in RFC 2748
- [4] Camarillo, Gonzalo & Miguel A. Garcia-Martin. *The 3G IP Multimedia Subsystem (IMS): Merging the Internet and the Cellular Worlds, Second Edition*. JohnWiley & Sons, 2006
- [5] Deolens. *ANALISA LAYANAN VIDEO ON DEMAND PADA ARSITEKTUR IP MULTIMEDIA SUBSYSTEM*. Universitas Indonesia. 2010.
- [6] Hens, F.J & Caballero, J.M.(2008). *Triple Play : Building the converged network for IP, VoIP and IPTV*. John Wiley & Sons
- [7] [http://id.wikipedia.org/wiki/Quality\\_of\\_Service](http://id.wikipedia.org/wiki/Quality_of_Service)
- [8] <http://sulistyonugroho.wordpress.com/2010/10/09/quality-of-service-dalam-data-komunikasi/>
- [9] [http://www.ittelkom.ac.id/library/index.php?view=article&catid=10%3Ajaringan&id=380%3Adifferentiated-servicesdiffserv&option=com\\_content&Itemid=15asi/](http://www.ittelkom.ac.id/library/index.php?view=article&catid=10%3Ajaringan&id=380%3Adifferentiated-servicesdiffserv&option=com_content&Itemid=15asi/)
- [10] “*Introduction to the Diameter Protocol in 3GPP context*” Whitepaper by Traffix Systems
- [11] M.Handley, V.Jacobson,& C.Perkins.*SDP:Session Initiation Protocol*. July 2006.RFC 4566.
- [12] [http://en.wikipedia.org/wiki/Differentiated\\_services](http://en.wikipedia.org/wiki/Differentiated_services)

## DAFTAR PUSTAKA

1. Coppeland, Rebecca. *Converging NGN Wireline and Mobile 3G Networks with IMS*. CRC Press. 2009: USA
2. Deolens. *ANALISA LAYANAN VIDEO ON DEMAND PADA ARSITEKTUR IP MULTIMEDIA SUBSYSTEM*. Universitas Indonesia. 2010.
3. [http://id.wikipedia.org/wiki/Quality\\_of\\_Service](http://id.wikipedia.org/wiki/Quality_of_Service)
4. <http://sulistyonugroho.wordpress.com/2010/10/09/quality-of-service-dalam-data-komunikasi/>
5. [http://en.wikipedia.org/wiki/IP\\_Multimedia\\_Subsystem](http://en.wikipedia.org/wiki/IP_Multimedia_Subsystem)
6. <http://joudane.wordpress.com/2008/07/15/ims-ip-multimedia-subsystem/>
7. <http://openimscore.org/>
8. “Introduction to the Diameter Protocol in 3GPP context” Whitepaper by Traffix Systems
9. [http://uctimsclent.berlios.de/policy\\_control\\_howto](http://uctimsclent.berlios.de/policy_control_howto)

## LAMPIRAN

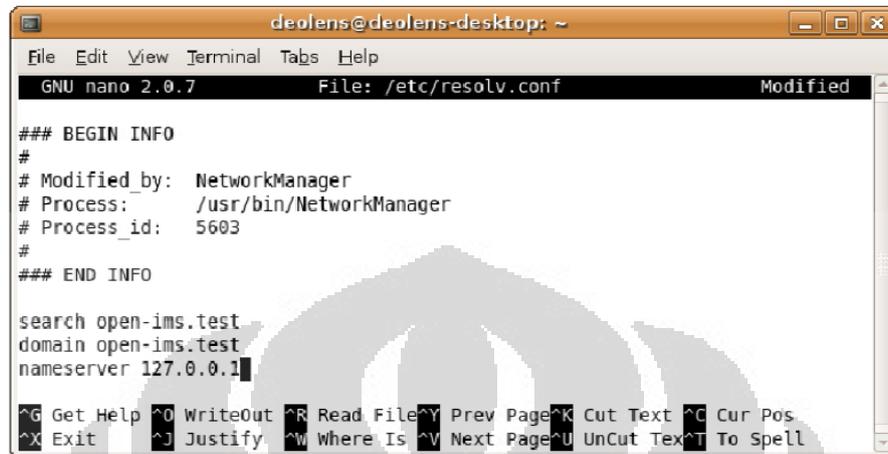
### Instalasi OpenIMSCore

1. Update repositori ubuntu  
# sudo apt-get update
2. Download dan install Subversion package  
# sudo apt-get install subversion
3. Membuat folder OpenIMSCore  
# sudo mkdir /opt/OpenIMSCore/  
# sudo chown -R username /opt/OpenIMSCore/
4. Membuat folder ser\_ims dan FHOSS di dalam folder OpenIMSCore  
# cd /opt/OpenIMSCore  
# mkdir ser\_ims  
# mkdir FHOSS
5. Download source code CSCF dan FHOSS Untuk CSCF:  
# svn checkout  
[http://svn.berlios.de/svnroot/repos/openimscore/ser\\_ims/trunk](http://svn.berlios.de/svnroot/repos/openimscore/ser_ims/trunk) ser\_ims  
Untuk HSS:  
# svn checkout  
<http://svn.berlios.de/svnroot/repos/openimscore/FHOSS/trunk> FHOSS
6. Install package-package yang dibutuhkan  
# sudo apt-get install sun-java6-jdk mysql-server libmysqlclient15-dev  
libxml2 libxml2-dev bind9 ant flex bison
7. Copy open-ims DNS file ke folder bind  
# sudo cp /opt/OpenIMSCore/ser\_ims/cfg/open-ims.dnszone/etc/bind/
8. Ubah file named.conf.local  
# sudo nano /etc/bind/named.conf.local  
Lalu tambahkan kata-kata berikut ke dalam file tersebut:  
zone "open-ims.test" {  
type master;  
file "/etc/bind/open-ims.dnszone";  
};

## 9. Ubah file resolv.conf

```
# sudo nano /etc/resolv.conf
```

Lalu ubah file tersebut menjadi seperti ini:



```
deolens@deolens-desktop: ~
File Edit View Terminal Tabs Help
GNU nano 2.0.7 File: /etc/resolv.conf Modified

### BEGIN INFO
#
# Modified_by: NetworkManager
# Process: /usr/bin/NetworkManager
# Process_id: 5603
#
### END INFO

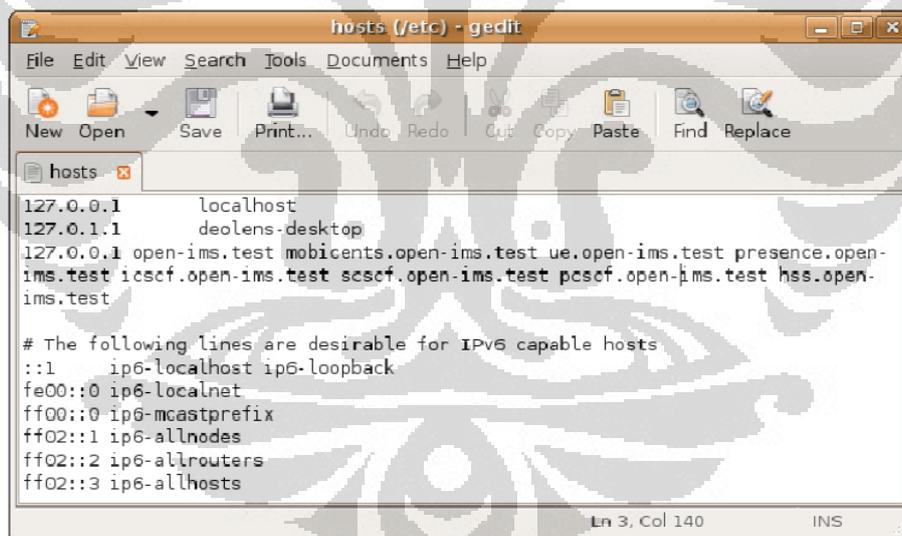
search open-ims.test
domain open-ims.test
nameserver 127.0.0.1

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

Gambar IV. 1. Tampilan file resolv.conf localhost

## 10. Ubah file hosts

```
# sudo nano /etc/hosts
```



```
hosts (/etc) - gedit
File Edit View Search Tools Documents Help
New Open Save Print... Undo Redo Cut Copy Paste Find Replace

hosts
127.0.0.1 localhost
127.0.1.1 deolens-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 pcsf.open-ims.test hss.open-
ims.test

# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts

Ln 3, Col 140 INS
```

Gambar IV. 2. Tampilan File Host localhost

## 11. Restart bind server

```
# sudo /etc/init.d/bind9 restart
```

## 12. Cek apakah step 7 sampai 11 sudah benar

Ada 2 cara untuk memeriksanya yaitu:

```
# ping pcsf.open-ims.test
```

Lihat apakah ping mendapat response, atau tidak

```
# dig open-ims.test
```

Pada answer section harus tertulis 127.0.0.1

### 13. Setup database

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

```
# mysql -uroot -p <
```

```
/opt/OpenIMSCore/FHoSS/scripts/hss_db.sql
```

```
# mysql -uroot -p <
```

```
/opt/OpenIMSCore/FHoSS/scripts/userdata.sql
```

### 14. Compile source code

Compile CSCF:

```
# cd /opt/OpenIMSCore/ser_ims
```

```
# sudo make install-libs all
```

Compile HSS:

```
# cd /opt/OpenIMSCore/ser_ims
```

```
# ant compile deploy
```

### 15. Copy configuration file ke folder OpenIMSCore

```
# cp /opt/OpenIMSCore/ser_ims/cfg/* /opt/OpenIMSCore/
```

### 16. Jalankan OpenIMSCore

Jalankan setiap CSCF pada satu tab terminal:

```
# cd /opt/OpenIMSCore/
```

```
# ./pscfc.sh
```

```
# ./icscf.sh
```

```
# ./scscf.sh
```

Jalankan FhoSS:

```
# cd /opt/OpenIMSCore/FhoSS/deply/
```

```
# export JAVA_HOME=/usr/lib/jvm/java-6-sun
```

```
# ./startup.sh
```

## LAMPIRAN

### Instalasi Policy Control Rules Function

1. Download seluruh file yang diperlukan oleh ser\_ims dari the 08\_cdp\_sessions branch menggunakan perintah dari terminal sebagai berikut:

```
#svn checkout http://svn.berlios.de/svnroot/repos/openimscore/ser\_ims/branches/08\_cdp\_sessions ser_ims_qos
```

Atau dengan mendownload file ser\_ims20081023.r0595.tgz secara manual dan kemudian meng-ekstrak serta meng-copy folder 08\_cdp\_sessions ser\_ims\_qos ke dalam folder ser\_ims\_qos

```
#cp 08_cdp_sessions ser_ims_qos /opt/OpenIMSCore/ser_ims_qos/
```

2. Download uctpcrf tarball dan extract dalam folder /opt/OpenIMSCore/  

```
#tar -xvzf /opt/OpenIMSCore/uctpcrf.tar.gz
```
3. Install libxerces-java libcommons-httpclient-java tomcat5.5 dengan command line:  

```
#sudo apt-get install libxerces-java libcommons-httpclient-java tomcat5.5
```
4. Configure DNS Server  
Tambahkan DNS entries untuk the PCRF and PCEF  
If you are running the DNS on your own machine add these lines to /etc/bind/open-ims.dnszone:

```
pcrf      1D IN A      192.168.6.128
```

```
pcef      1D IN A      192.168.6.128
```

kemudian restart bind9:

```
#sudo /etc/init.d/bind9 restart
```

- Periksa apakah pcrf dan pcef sudah dapat memberikan respon dengan perintah nslookup:

```
# nslookup pcrf.open-ims.test
```

```
# nslookup pcef.open-ims.test
```

- Compile source CSCF pada folder /opt/OpenIMSCore/ser\_ims\_qos

```
#cd /opt/OpenIMSCore/ser_ims_qos
```

```
#sudo make install-libs all
```

- Atur JAVA\_HOME environment variable yang dipergunakan:

```
#export JAVA_HOME="/usr/lib/jvm/java-1.6.0-sun"
```

- Compile the PCRf and PCEf. In /opt/OpenIMSCore/uctpcrf:

```
#cd /opt/OpenIMSCore/uctpcrf
```

```
#sudo ./compile
```

- Setup Web Interface

- Move the web management folder to the document root of the Tomcat server

```
#cd /opt/OpenIMSCore/uctpcrf
```

```
#sudo cp -r uct_web_pcm /var/lib/tomcat5.5/webapps
```

- Move the uct\_web\_pcm.policy config file to the policy folder of the Tomcat server

```
#cd /opt/OpenIMSCore/uctpcrf/config
```

```
#sudo cp uct_web_pcm.policy /etc/tomcat5.5/policy.d
```

12. Ubah file permissions pada files domain\_policies.xml, default\_domain\_properties.xml, pcef.properties and pcrf.properties dengan menggunakan perintah:

```
#chmod uog+rw /opt/OpenIMSCore/uctpcrf/policies/ domain_policies.xml
```

```
#chmod uog+rw /opt/OpenIMSCore/uctpcrf/policies/default_domain_policies.xml
```

```
#chmod uog+rw /opt/OpenIMSCore/uctpcrf/config/pcef.properties
```

```
#chmod uog+rw /opt/OpenIMSCore/uctpcrf/config/pcrf.properties
```

13. Restart Tomcat server

```
#sudo /etc/init.d/tomcat5.5 restart
```

14. Configure ip address serta domain untuk pcrf dan pcef yang akan dipergunakan dengan configurator.sh, dan file configurator pada folder ser\_ims\_qos.

```
#cd /opt/OpenIMSCore
```

```
#sudo ser_ims/cfg/configurator.sh uctpcrf/config/pcef.sh
```

```
uctpcrf/config/pcrf.sh uctpcrf/config/ims.sh
```

```
uctpcrf/config/DiameterPeerPCEF.xml uctpcrf/config/
```

```
DiameterPeerPCRF.xml uctpcrf/config/DiameterPeerPCSCF.xml uctpcrf/
```

```
config/pcscf.qos.cfg uctpcrf/config/pcscf.qos.rtp.cfg uctpcrf/config/
```

```
pcscf.qos.rtp.sh uctpcrf/config/pcscf.qos.sh uctpcrf/config/pcscf.qos.xml
```

```
uctpcrf/config/rtpproxy.sh uctpcrf/config/runDiameterClient.sh uctpcrf/
```

```
config//xcapTest.sh uctpcrf/config/uct_web_pcm.policy uctpcrf/
```

```
config/ims.qos.rtp.sh                                uctpcrf/config/ims.qos.sh
uctpcrf/config/pcrf.properties uctpcrf/config/pcef.properties
```

Domain Name:open-ims.test

IP Adress:192.168.102.105

15. Copy file configuration yang baru ke folder OpenIMSCore:

```
#cd /opt/OpenIMSCore/uctpcrf/config
```

```
#cp pscf.qos.cfg pscf.qos.rtp.cfg pscf.qos.xml pscf.qos.sh ims.sh
ims.qos.sh ims.qos.rtp.sh /opt/OpenIMSCore
```

16. Nyalakan PCRF serta ims core dengan menggunakan perintah:

```
#cd /opt_OpenIMSCore
```

```
#sudo ./ims.qos.sh
```

```
#cd uctpcrf
```

```
#./pcrf
```

```
#sudo ./pcef
```

17. Jalankan Web Management Interface dengan menggunakan web browser pada alamat:

[http://localhost:8180/uct\\_web\\_pcm/](http://localhost:8180/uct_web_pcm/)