



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

DESARROLLO DE SERVICIOS DE NEXT GENERATION TV PARA OPERADOR
CON CORE IMS/EPC MONTADO SOBRE LTE

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL ELECTRICISTA

SEBASTIÁN ANDRÉS RIVAS SÁENZ

PROFESOR GUÍA
ALFONSO EHIJO BENBOW

MIEMBROS DE LA COMISIÓN
JORGE SANDOVAL ARENAS
HÉCTOR AGUSTO ALEGRÍA

SANTIAGO DE CHILE
AGOSTO 2012

RESUMEN DE LA MEMORIA
PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL ELECTRICISTA
POR: SEBASTIÁN ANDRÉS RIVAS SÁENZ
FECHA: AGOSTO 2012
PROF. GUÍA: ALFONSO EHIJO B.

DESARROLLO DE SERVICIOS DE NEXT GENERATION TV PARA OPERADOR CON CORE IMS/EPC MONTADO SOBRE LTE

La llegada de redes de cuarta generación cambia el paradigma de creación y consumo de contenidos. Este cambio se sustenta principalmente en las mejoras en la velocidad de conexión que ofrecen estas redes y en el uso de arquitecturas de red de próxima generación (NGN). Ambos factores hacen posible el despliegue de servicios convergentes que se caracterizan por permitir la interacción de servicios tradicionales bajo una misma plataforma.

El objetivo del presente trabajo de memoria es la utilización de la arquitectura de red IMS y la red de acceso LTE para el desarrollo de servicios de televisión de próxima generación de carácter convergente y que ofrezcan interfaces gráficas interactivas y amigables, logrando así sacar mayor provecho de las características que ofrecen estas tecnologías de red.

Este trabajo es parte de un proyecto de desarrollo continuo del Departamento de Ingeniería Eléctrica de la Facultad de Ciencias Físicas y Matemáticas de la Universidad de Chile que busca la creación de ambientes docentes para el estudio de redes 4G que aún no existen comercialmente en el país.

Tras una presentación del contexto y antecedentes respecto a las características principales de los conceptos involucrados, se presenta el diseño de dos servicios de televisión. En primer lugar, se muestra un servicio de IPTV básico que permite la elección flexible de contenido y la interacción con otros usuarios a través de video conferencias, llamadas de voz y mensajería de texto. En segundo lugar, en base al ambiente de creación de servicios Mobicents, se muestra la construcción de una plataforma web para la elección del contenido de video que permite el envío de trailers a dispositivos móviles y la posibilidad de compra de películas.

Una vez terminados los diseños, se hace un especial énfasis en detallar la implementación de los servicios mencionados con el propósito de aportar en el proceso de construcción de herramientas para el estudio de redes y facilitar su posterior puesta en marcha. Finalmente, se muestran diversas capturas de tráfico para verificar el correcto funcionamiento de los servicios.

En base a estas pruebas, se verifica que el trabajo cumple con los objetivos propuestos, en cuanto se logra la implementación de servicios de televisión convergentes sobre redes LTE en un ambiente de laboratorio controlado y replicable para futuras experiencias.

Es relevante el hecho de haber logrado estos desarrollos en base a un ambiente de creación de servicios, pues entrega un punto de partida para continuar con la mejora de estos servicios o la elaboración de nuevas experiencias de laboratorio en trabajos futuros.

AGRADECIMIENTOS

Me gustaría dedicar este espacio a todos quienes me han acompañado y apoyado durante este proceso. Si bien, me gustaría mencionar a cada persona individualmente, lo más probable que no lo logre, por lo que debo remarcar que mencionados o no, agradezco enormemente a cada uno de ustedes.

En primer lugar quiero agradecer a mi familia. A mis padres y a su apoyo incondicional. A ellos les debo mi educación que con un gran esfuerzo han logrado brindarme. A mi hermano, que siempre un paso más adelante, ha logrado darme el mejor ejemplo. A mis abuelos, por el inmenso cariño que siempre me han dado. A mis primos, por todos los momentos que hemos vivido juntos.

Quiero agradecer a Marcela, quien siempre estuvo presente tanto en los éxitos como en las frustraciones y nunca ha dejado de apoyarme, a quien además, le debo gran parte de la labor artística de este trabajo.

Agradezco además a cada uno de mis amigos, tanto a los que siempre han estado junto a mí como también a otros que tuve la suerte de conocer en este proceso universitario. De este último grupo, quiero mencionar a Pablo Hevia-Koch, agradezco todo el tiempo que dedicó a ayudarme con algunas implementaciones de este trabajo y por enseñarme que hay cosas en la vida más importantes que el trabajo.

En el ambiente de trabajo, quiero hacer una especial mención al Dr. Mosiuoa Tsietsi. A su enorme paciencia y voluntad para constantemente responder cada una de mis dudas y ayudarme en este proceso a pesar de no habernos conocido previamente. Gran parte de los desarrollos de este trabajo no habrían sido posibles sin su asistencia.

No puedo dejar de mencionar a mis compañeros memoristas, Cristián Segura y John Oliva con quienes logramos conformar un excelente grupo de trabajo. Quiero además, extender mis agradecimientos al profesor Alfonso Ehijo por permitirme la oportunidad de participar de este proyecto que aborda la investigación de temas de primer interés en la industria de Telecomunicaciones y por brindarme el apoyo necesario para desarrollar tanto mi carrera profesional como mis habilidades personales. En este contexto, quiero también agradecer a los profesores Jorge Sandoval y Hector Augusto, por sus rigurosas correcciones que permitieron grandes mejoras a este trabajo.

Finalmente, quiero agradecer a la empresa ZTE por involucrarse en este proyecto de desarrollo al entregar la posibilidad de utilizar redes y equipos que aún no están disponibles comercialmente en el país. Permitiendo así, la realización de trabajos que hagan interactuar a personal del mundo académico y laboral.

CONTENIDO

CAPÍTULO I: INTRODUCCIÓN	8
1.1 MOTIVACIÓN	8
1.2 OBJETIVOS	9
1.3 HIPÓTESIS	9
1.4 DESCRIPCIÓN DE CONTENIDOS	10
CAPÍTULO II: ANTECEDENTES	11
2.1 CONTEXTUALIZACIÓN DE TECNOLOGÍA Y ENTIDADES DE ESTANDARIZACIÓN	11
2.2 REDES DE PRÓXIMA GENERACIÓN (NGN).....	12
2.3 IP MULTIMEDIA SUBSYSTEM (IMS)	14
2.3.1 <i>Arquitectura y Entidades Fundamentales</i>	15
2.3.2 <i>Servicios en IMS</i>	17
2.4 LONG TERM EVOLUTION (LTE)	18
2.4.1 <i>LTE como paso hacia 4G</i>	19
2.5 SERVICIOS DE TELEVISIÓN DE PRÓXIMA GENERACIÓN.....	19
2.5.1 <i>Servicio de IPTV</i>	19
2.6 PROYECTOS UTILIZADOS PARA LA IMPLEMENTACIÓN.	23
2.6.1 <i>Fokus OpenIMS Core</i>	23
2.6.2 <i>UCT Advanced IPTV Server</i>	24
2.6.3 <i>UCT IMS IPTV Charging System</i>	26
2.6.4 <i>VLC Media Player</i>	28
2.6.5 <i>Mobicents</i>	28
2.6.6 <i>Mobicents JAIN SLEE</i>	30
2.6.7 <i>Mobicents Media Server</i>	31
2.6.8 <i>UCT IMS Client</i>	32
CAPÍTULO III: METODOLOGÍA	33
3.1 DESCRIPCIÓN DE METODOLOGÍA	33
3.2 CONTEXTO DE TRABAJO	34
3.3 ARQUITECTURA DISPONIBLE	34
3.4 DISEÑO GENERAL	37
3.4.1 <i>Alcances del Diseño</i>	37
3.4.2 <i>Descripción de Prestaciones</i>	39
3.5 FUNCIONAMIENTO Y ARQUITECTURA	41
3.5.1 <i>Servicio de IPTV sobre IMS</i>	41
3.5.2 <i>Servicio Video Shopping</i>	43
3.6 IMPLEMENTACIÓN DE LOS SERVICIOS	51
3.7 IMPLEMENTACIÓN SERVICIO DE IPTV.....	52
3.8 IMPLEMENTACIÓN DEL SERVICIO DE VIDEO SHOPPING.....	56
3.9 INTERCONEXIÓN CON LABORATORIO LTE	60
CAPÍTULO IV: RESULTADOS	66

4.1	RESULTADOS DE LA IMPLEMENTACIÓN DEL SERVICIO DE IPTV	66
4.1.1	<i>Características Generales</i>	66
4.1.2	<i>Funcionamiento</i>	71
4.1	RESULTADOS DEL VIDEO SHOPPING SOBRE LTE.....	73
4.1.1	<i>Características Generales</i>	73
4.1.1	<i>Funcionamiento</i>	78
CAPÍTULO V: DISCUSIÓN		81
CAPÍTULO VI: CONCLUSIONES.....		84
CAPÍTULO VII: BIBLIOGRAFÍA		86
ANEXOS.....		90
8.1	ANEXO A: ACRÓNIMOS	90
8.2	ANEXO B: CREACIÓN DE UNA MÁQUINA VIRTUAL CONECTADA A UNA LAN	92
8.3	ANEXO C: INSTALACIÓN Y CONFIGURACIÓN DEL SERVICIO DE IPTV.....	97
8.3.1	<i>Preparación del computador y sistema operativo</i>	97
8.3.2	<i>Instalación y configuración de Componentes de IMS y Cliente Compatible</i>	99
8.3.3	<i>Instalación y configuración de Sistema de IPTV básico</i>	103
8.3.4	<i>Instalación y configuración de Sistema de Tarificación</i>	109
8.3.5	<i>Implementación distribuida y personalización del Servicio de IPTV</i>	113
8.4	ANEXO D: INSTALACIÓN DEL SERVICIO DE VIDEO SHOPPING	117
8.4.1	<i>Instalación de Mobicents</i>	118
8.4.2	<i>Instalación Converged-Demo Original</i>	120
8.4.3	<i>Instalación y Configuración del Servicio de Video Shopping</i>	125
8.4.4	<i>Instalación del Servidor Multimedia y del Cliente SIP</i>	132
8.4.5	<i>Interconexión con IMS</i>	139

INDICE DE FIGURAS

FIGURA 2.1: RELEASES DE LA 3GPP	12
FIGURA 2.2: ARQUITECTURA NGN.....	13
FIGURA 2.3: ARQUITECTURA IMS.....	15
FIGURA 2.4: DIAGRAMA DE BLOQUES DE UNA RED DE IPTV.....	22
FIGURA 2.5: DIAGRAMA SIMPLIFICADO DE LA ARQUITECTURA DEL OPENIMSCORE	24
FIGURA 2.6:DIAGRAMA DE LA ARQUITECTURA DE UCT ADVANCED IPTV SERVER	26
FIGURA 2.7:DIAGRAMA DE LA ARQUITECTURA DE UCT IMS IPTV CHARGING SYSTEM.....	28
FIGURA 3.1: COBERTURA DE LTE DE LA FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS	35
FIGURA 3.2: EQUIPAMIENTO INSTALADO EN EL LABORATORIO.....	36
FIGURA 3.3: ARQUITECTURA DE LA RED LTE DISPONIBLE EN EL LABORATORIO	37
FIGURA 3.4: DIAGRAMA DE INTERCONEXIONES PARA EL SEVICIO DE IPTV SOBRE IMS.....	43
FIGURA 3.5: DIGRAMA DE FUNCIONAMIENTO DEL SERVICIO DE VIDEO SHOPPING.....	48
FIGURA 3.6: DIAGRAMA DE INTERCONEXIONES SERVICIO VIDEO SHOPPING.....	50
FIGURA 3.7: DIAGRAMA DE INTERCONEXIONES SERVICIO VIDEO SHOPPING.....	51
FIGURA 3.8: DIAGRAMA DE INTERCONEXIÓN RED LTE, NÚCLEO IMS Y COMPONENTES SERVICIO DE IPTV	62
FIGURA 3.9: DIAGRAMA DE INTERCONEXIÓN RED LTE Y COMPONENTES SERVICIO DE VIDEO SHOPPING	63
FIGURA 3.10: DIAGRAMA DE INTERCONEXIÓN RED LTE Y COMPONENTES SERVICIO DE VIDEO SHOPPING	64
FIGURA 4.1: INTERFAZ GRÁFICA CLIENTE IMS UTILIZADO	67
FIGURA 4.2: EJEMPLO DE VIDEO-ON-DEMAND EN EL SERVICIO DE IPTV.....	68
FIGURA 4.3: INICIO DE TRANSMISIÓN EN MODALIDAD LIVE STREAMING EN SERVICIO DE IPTV	69
FIGURA 4.4: ELECTRONIC PROGRAM GUIDE EN EL SERVICIO DE IPTV.....	70
FIGURA 4.5: MENSAJES SIP EN REGISTRO DE CLIENTE EN IMS	72
FIGURA 4.6: MENSAJES SIP EN SOLICITUD DE.....	73
FIGURA 4.7: PORTAL WEB DEL SERVICIO DE VIDEO SHOPPING.....	74
FIGURA 4.8: CREACIÓN DE USUARIOS ONLINE EN EL PORTAL WEB DEL SERVICIO DE VIDEO SHOPPING	75
FIGURA 4.9: CARRO DE COMPRAS EN EL PORTAL WEB DEL SERVICIO DE VIDEO SHOPPING	75
FIGURA 4.10: RECEPCIÓN DE VIDEO MEDIANTE CLIENTE SIP EN SERVICIO DE VIDEO SHOPPING	76
FIGURA 4.11: UTILIZACIÓN DE CLIENTE IMS PARA ACCEDER AL SERVICIO DE VIDEO SHOPPING	77
FIGURA 4.12: MENSAJES SIP EN SERVICIO DE VIDEO SHOPPING SIN IMS	78
FIGURA 4.13: MENSAJES RTSP EN SERVICIO DE VIDEO SHOPPING SIN IMS	79
FIGURA 4.14: MENSAJES SIP EN SERVICIO DE VIDEO SHOPPING CON IMS.....	80
FIGURA 8.1: SELECCIÓN DE NOMBRE Y SISTEMA OPERATIVO PARA MÁQUINA VIRTUAL.....	93
FIGURA 8.2: SELECCIÓN DE MEMORIA RAM PARA MÁQUINA VIRTUAL.....	93
FIGURA 8.3: CREACIÓN DE DISCO DURO PARA MÁQUINA VIRTUAL.....	94
FIGURA 8.4: SELECCIÓN DE TIPO DE ALMACENAMIENTO DE LA MÁQUINA VIRTUAL.....	94
FIGURA 8.5: SELECCIÓN DE CAPACIDAD DE DISCO DURO PARA MÁQUINA VIRTUAL.....	95
FIGURA 8.6: CONFIGURACIÓN DE TARJETA DE RED DE LA MÁQUINA VIRTUAL.....	96
FIGURA 8.7: HABILITACIÓN DE SET DE INSTRUCCIONES DE VIRTUALIZACIÓN.....	96
FIGURA 8.8: DATOS PARA CREACIÓN DEL APPLICATION SERVER.....	104
FIGURA 8.9: DATOS PARA CREACIÓN DEL TRIGGER POINT.....	105
FIGURA 8.10: DATOS PARA CREACIÓN DEL INITIAL FILTER CRITERIA.....	105
FIGURA 8.11: DATOS DE CONFIGURACIÓN EN SHARED IFC SETS.....	106
FIGURA 8.12: DATOS DE CONFIGURACIÓN EN SERVICE PROFILE.....	107
FIGURA 8.13: CONFIGURACIÓN CLIENTE IMS.....	109

FIGURA 8.14: MANAGEMENT-CONSOLE DEL SERVIDOR DE MOBICENTS.....	119
FIGURA 8.15: RESOURCE ADAPTERS NECESARIOS PARA LA INSTALACIÓN DEL SERVICIO.....	123
FIGURA 8.16: INTERFAZ WEB DEL EJEMPLO ORIGINAL CONVERGED-DEMO.....	125
FIGURA 8.17: INTERFAZ WEB DEL SERVICIO DE VIDEO SHOPPING.....	127
FIGURA 8.18: CONFIGURACIÓN DE PARAMETROS DE CONEXIÓN.....	129
FIGURA 8.19: CLIENTE JAIN-SIP-APPLET-PHONE.....	137
FIGURA 8.20: FUNCIÓN DE DIAGNÓSTICO EN EL CLIENTE SIP.....	138

INDICE DE TABLAS

TABLA 3.1: ELEMENTOS DE IMS Y SUS NOMBRES DE DOMINIO	54
TABLA 3.2: ELEMENTOS DEL SISTEMA DE IPTV Y SUS NOMBRES DE DOMINIO	55
TABLA 3.3: ELEMENTOS DEL SISTEMA DE IPTV Y SUS NOMBRES DE DOMINIO	55
TABLA 3.4: ELEMENTOS DEL SERVICIO DE VIDEO SHOPPING	59
TABLA 3.5: ELEMENTOS DEL SERVICIO DE VIDEO SHOPPING A TRAVÉS DE IMS	60
TABLA 3.6: DOMINIOS, PUERTOS, Y DIRECCIONES IP DE TODOS LOS ELEMENTOS QUE CONFORMAN LOS SERVICIOS	61

1 CAPÍTULO I: INTRODUCCIÓN

1.1 MOTIVACIÓN

Hace algún tiempo, los límites entre los distintos servicios ofrecidos por las operadoras de telecomunicaciones eran bastante estrictos y estaban relacionados con las distintas actividades y contenidos correspondientes a cada dispositivo asociado a los servicios. Era poco frecuente comunicar dispositivos de telefonía móvil con contenido web o con televisión, sin embargo, a medida que la tecnología lo posibilita, se generan distintos hábitos y hoy en día el consumo de contenido multimedia se extiende mucho más allá del alcance de la televisión o acceso a la red fija.

La principal razón por la cual ha sido posible lograr esta unificación de los servicios, es la convergencia de las tecnologías de acceso. El interés de las operadoras de desarrollar redes convergentes capaces de prestar servicios de próxima generación bajo una misma plataforma ha provocado una gran evolución en la arquitectura de las redes de acceso y los núcleos de telecomunicaciones. LTE (*Long Term Evolution*) es el más reciente estándar de tecnología de redes móviles de la organización 3GPP (*Third Generation Partnership Project*) que apunta a conseguir la arquitectura denominada “All-IP”. Con estos avances se puede aumentar el rendimiento de las redes notablemente y se consigue unificar la manera de prestar servicios, reduciendo así los costos.

La motivación principal del trabajo, es justamente desarrollar servicios compatibles en las arquitecturas de redes convergentes. En particular, se implementan servicios de televisión (VoD, Mobile TV, VideoStreaming, etc) con la estrategia de Multiscreen que están realizando los Operadores Convergentes, pero esta vez en un Laboratorio Docente. Para esto se utilizan las arquitecturas de núcleos IMS/EPC de manera de lograr prestar servicios específicamente compatibles con LTE. La importancia de este tipo de servicios nace naturalmente debido a los hábitos generados en los clientes al posibilitar convergencia en los servicios móviles. La inmensa cantidad de aplicaciones que cada vez requieren de mayor ancho de banda siempre termina limitada por los grandes requerimientos asociados a la transferencia de video. Si los operadores no logran la implementación de servicios de televisión de próxima generación capaces de reemplazar a los servicios actuales, no será posible el traspaso a redes convergentes.

Actualmente los desarrollos de LTE son temas de primera prioridad para las operadoras alrededor del mundo, pues será el estándar de comunicación global a corto plazo. En este sentido, la actividad del Departamento de Ingeniería Eléctrica de la Universidad de Chile ha sido pionera en desarrollos de LTE en Latino América al realizar las primeras pruebas e instalando el primer

Laboratorio para Entrenamiento en Tecnología LTE, posicionando a la Universidad como líder en el continente.

1.2 OBJETIVOS

El objetivo general de este trabajo de memoria es desarrollar servicios de televisión de próxima generación para núcleos IMS/EPC sobre la red LTE de la empresa ZTE, que se encuentra instalada en el Laboratorio de Telecomunicaciones del Departamento de Ingeniería Eléctrica de la Universidad de Chile.

Para lograrlo, se definen los siguientes objetivos específicos.

- Analizar y comprender el contexto actual de las distintas arquitecturas de telecomunicaciones y su funcionamiento.
- Estudiar el estado del arte de los servicios de televisión de próxima generación y lograr delimitar alcances de lo que se implementará.
- Realizar una investigación de las herramientas o proyectos que puedan aportar al desarrollo de servicios de televisión.
- Comprender la implementación actual ya existente en el Laboratorio de LTE y estudiar sus características para lograr compatibilizar el servicio a implementar con la red.
- Establecer el diseño del servicio a desarrollar logrando obtener un enfoque convergente de próxima generación que lo diferencie de los servicios tradicionales.
- Implementar el servicio diseñado en el Laboratorio de Telecomunicaciones del departamento detallando todas las actividades para la concreción de este objetivo.
- Realizar pruebas de funcionamiento y capturas de tráfico en distintos escenarios de prueba con el fin de lograr demostrar la efectividad.

1.3 HIPÓTESIS

La hipótesis principal de este trabajo de memoria es que se logrará desarrollar servicios de televisión de próxima generación para núcleos IMS/EPC sobre las redes LTE. Por otro lado, también se tiene como hipótesis que se tendrá disponibilidad para el uso del laboratorio de LTE.

1.4 DESCRIPCIÓN DE CONTENIDOS

A continuación se describe brevemente el contenido de los siguientes capítulos que componen este trabajo.

En el Capítulo II se exponen antecedentes generales que fue necesario estudiar previamente al trabajo de implementación. Se describen las redes de próxima generación, mostrando particularmente la arquitectura de IMS, sus componentes y la manera en que se ejecutan los servicios. Se describe la red de acceso LTE y sus ventajas para lograr prestar servicios que tengan fuerte demanda en recursos de red. Se expone el concepto de IPTV, describiendo sus características principales. Finalmente se hace una revisión de los aspectos generales de los proyectos de código libre que se utilizan para lograr el desarrollo de los servicios.

En el Capítulo III se describe la metodología de trabajo utilizada, explicando en términos genéricos el proceso completo que comprende este trabajo. A continuación, se define el contexto del trabajo, detallando la arquitectura disponible en el laboratorio de LTE, avances de trabajos de título previos y los objetivos del grupo de trabajo. Se define también, el diseño específico de los servicios a desarrollar, considerando limitaciones y características que sea deseado que estos posean.

En el Capítulo IV se presentan los resultados de la implementación diseñada en el capítulo anterior. En primer lugar se exponen características generales que se obtienen de los servicios y luego se muestran capturas de tráfico para explicar en detalle su funcionamiento.

En el Capítulo V se presenta una discusión de los alcances e impactos de los servicios que se ponen en marcha. Se hace especial énfasis en la utilización de un ambiente de desarrollo de servicios y cómo esto genera un punto de partida para trabajos futuros.

En el Capítulo VI se entregan las conclusiones del trabajo desarrollado. Se comparan los resultados con los objetivos que se proponen inicialmente.

Finalmente, se incluye un capítulo de Anexos, donde se pone a disposición una guía que detalla paso a paso la puesta en marcha de cada componente necesario para la implementación de la plataforma de pruebas utilizada en este trabajo.

2 CAPÍTULO II: ANTECEDENTES

En este capítulo se presenta un resumen de los temas que es necesario manejar para realizar el diseño e implementación de un servicio de televisión de próxima generación en un core IMS o EPC montado sobre LTE. En particular, se expone una contextualización de las arquitecturas involucradas y los protocolos asociados, describiendo componentes, funcionamiento e interrelación. Es importante mencionar que en este capítulo no se pretende entregar una visión detallada, sino sólo servir como iniciación al tema.

2.1 CONTEXTUALIZACIÓN DE TECNOLOGÍA Y ENTIDADES DE ESTANDARIZACIÓN

El desarrollo de las comunicaciones móviles es tradicionalmente entendido como una secuencia de generaciones según el cumplimiento de algún estándar. A partir de la segunda generación (2G), donde se evolucionó a sistemas digitales, se dio origen a dos estándares tecnológicos ampliamente utilizados hoy en día, GSM (Global System for Mobile Communications) y los que utilizan CDMA (Code Division Multiple Access). GSM se ha transformado en el estándar dominante en tecnología móvil celular, llegando incluso a cubrir el 80% del mercado móvil a nivel mundial con más de 5 millones de usuarios en 212 países al 2010 [1].

La ITU (International Telecommunication Union) es la agencia líder de las Naciones Unidas enfocada en dar solución a problemas de informática y tecnología de comunicaciones y es considerada como punto de foco por los gobiernos o empresas privadas para el desarrollo de redes o servicios [2]. En este contexto, la ITU dispuso el camino para avanzar hacia la tercera generación (3G) de sistemas móviles al introducir la iniciativa IMT-2000 (International Mobile Telecommunications). Esta iniciativa nació del deseo de proveer cobertura móvil universal, roaming desapercibido, sistemas móviles basados en el protocolo IP (Internet Protocol) transparente a los servicios y mayores tasas de transferencia de datos para permitir una mayor cantidad de servicios multimedia. En 1997, la ITU publicó la recomendación M.1225 [3] donde se solicitan propuestas de tecnologías de radio transmisión para cumplir con los requerimientos de IMT-2000. De estos requerimientos se puede destacar una tasa de transmisión de datos de 144kb/s en vehículos, 384kb/s a pie y de 2Mb/s en ambientes internos. En respuesta a este llamado, se formaron dos organizaciones para desarrollar estándares para la tercera generación de sistemas móviles, 3GPP (The Third Generation Partnership Project) [4] y 3GPP2 (The Third Generation Partnership Project 2) [5]. 3GPP se ha encargado de manejar la evolución de sistemas basados en GSM y 3GPP2 los sistemas basados en CDMA. Ambas entidades han desarrollado sus propias versiones de 3G e incluso se han transformado en desarrolladores de estándares que van más allá de 3G. IMS y LTE son parte de este proceso y se revisan en mayor

detalle en las siguientes secciones. La Figura 2.1 muestra en orden cronológico los releases de la 3GPP a la fecha. La información fue tomada de [4].

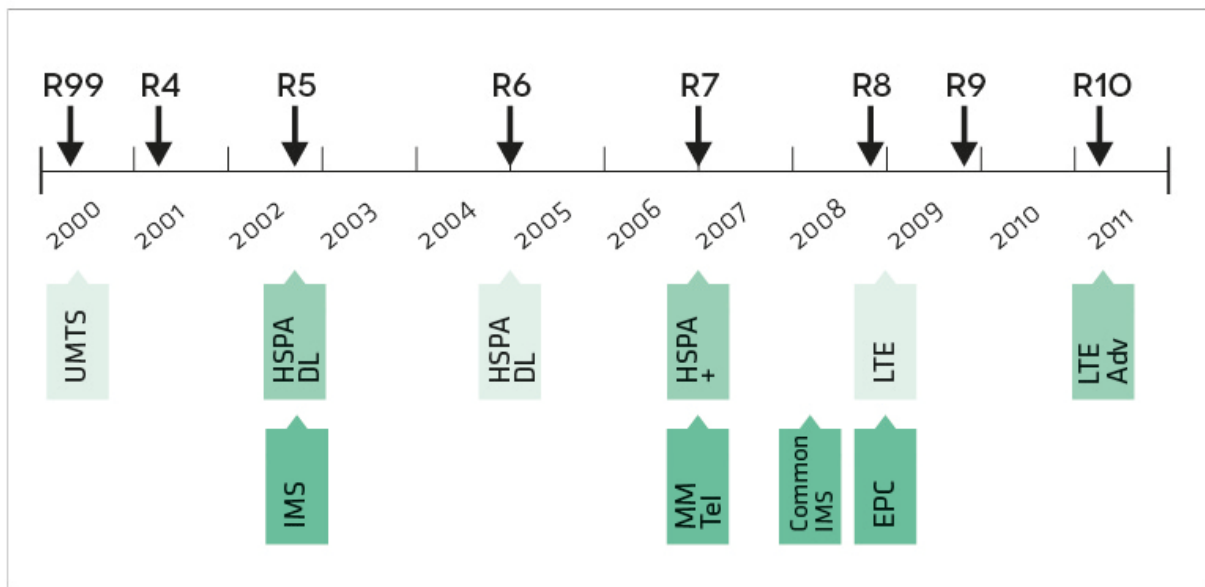


FIGURA 2.1: RELEASES DE LA 3GPP

2.2 REDES DE PRÓXIMA GENERACIÓN (NGN)

Una Red de Próxima Generación (NGN, NextGeneration Network), es un término genérico para definir la arquitectura de los núcleos de telecomunicaciones y las redes de acceso según ciertos requisitos. La idea general, es definir una red de telecomunicaciones convergente y de manera universal de forma que logre transportar toda la información y servicios de manera transparente.

La ITU-T a través de la recomendación Y.2001 [6] establece las características básicas que esta debiera tener:

1. Ser una red basada en la conmutación de paquetes.
2. Agnóstica al acceso permitiendo el uso de diferentes redes de entrada.
3. Soporte de movilidad generalizada.
4. Red multiservicios.
5. Capas con funciones autónomas (estrato de transporte y de servicios autónomos).
6. Equipo de usuario final fijo y móvil.

La NGN tiene como principal objetivo ofrecer a los suscriptores una serie de aplicaciones que provean calidad de servicio (QoS) en un ambiente on-demand, sin importar la arquitectura de acceso o el dispositivo que se utilice. Las funciones de esta red se dividen en estratos de servicio y transporte como lo especifica la recomendación Y.2001. Como se puede ver en la Figura 2.2, las funciones del usuario final se conectan a la NGN a través de la interfaz UNI (User-to-Network Interface), mientras que las otras redes se encuentran interconectadas a través de la interfaz NNI (Network-to-Network Interface). Por último, la interfaz ANI (Application-to-Network Interface) demarca la frontera con los proveedores de aplicaciones.

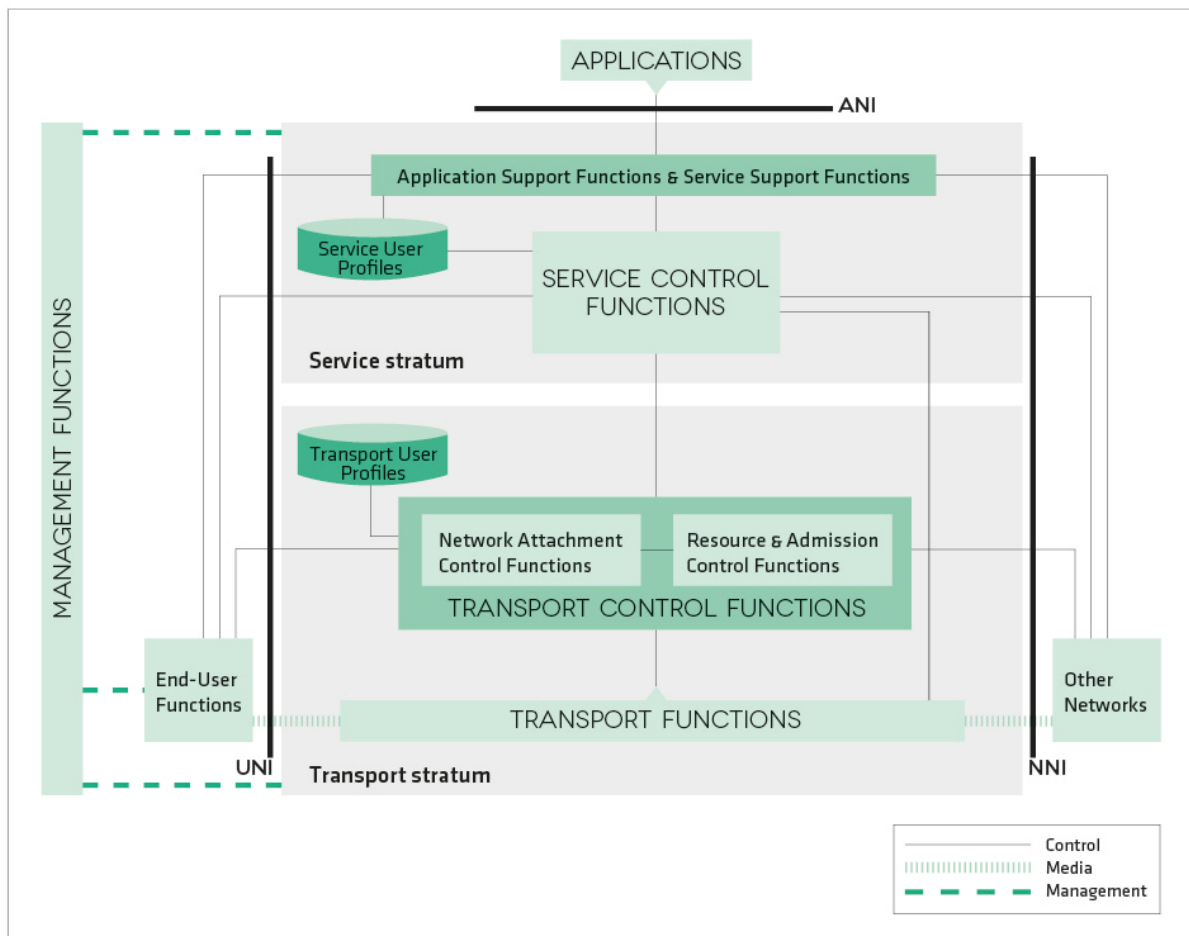


FIGURA 2.2: ARQUITECTURA NGN

Funciones del Estrato de Transporte: Las funciones del estrato de transporte permiten la conectividad entre todos los componentes y funcionalidades que están separadas físicamente. El estrato de transporte es el responsable de proveer Calidad de Servicio (QoS – *Quality of Service*) punto a punto. El estrato de transporte se divide en las redes de acceso y el núcleo de la red con una función de enlace entre las dos entidades.

Funciones del Estrato de Servicio: Estas funciones proveen servicios basados y no basados en la sesión, incluyendo suscripción o notificación de información de presencia y un método de mensajes para intercambio de mensajería instantánea. Las funciones del estrato de servicio también entregan todas las funcionalidades de red asociadas a los servicios existentes de la PSTN o ISDN y las capacidades e interfaces para los equipos antiguos de clientes.

Funciones de Administración: Las funciones de administración permiten a los operadores de una NGN administrar la red y entregar servicios que cumplan con la calidad, seguridad y fiabilidad esperadas. Estas funciones se localizan de forma distribuida a cada entidad e interactúan con la administración de elementos de red, administración de red y entidades administradoras de servicios. Las funciones de administración incluyen funciones de precios y facturas. Estas funciones interactúan entre ellas dentro de la NGN para recolectar la información de cobranza y para permitir que los operadores de la red NGN cuenten con los recursos necesarios para cobrar apropiadamente a los clientes. Los cargos pueden realizarse tanto en aplicaciones fuera de línea como servicios en línea.

Funciones de Usuario Final: Las interfaces hacia el usuario final son tanto físicas como funcionales (control). En una red NGN se deben soportar todas las categorías de equipos de clientes, desde los teléfonos utilizados en una red PSTN a complejas redes corporativas. Pudiendo además el terminal de usuario ser fijo o móvil.

Tanto la estructura de la Figura 2.2 como la descripción de las funciones de cada bloque fueron tomadas de [17].

2.3 IP MULTIMEDIA SUBSYSTEM (IMS)

La arquitectura de red IP Multimedia Subsystem (IMS) (Figura 2.3) fue desarrollada por la 3GPP (3rd Generation Partnership Project) como parte del trabajo de estandarización asociado a la Tercera Generación de Telefonía Celular (3G). En su primera versión (Release 5) se diseñó para la evolución de telefonía móvil 2G a 3G, soportando redes GSM y GPRS y siendo añadidos además el soporte de contenidos multimedia basados en SIP (Session Initiation Protocol). Para el Release 6 se añadió el soporte para acceso vía redes inalámbricas (WLAN, WiMAX). Finalmente, en el Release 7 se incluyó el soporte para redes fijas (xDSL, cable modem, ethernet). Entre las aplicaciones más importantes que se puede soportar esta red están:

- Voz y videotelefonía
- Servicios presenciales
- Mensajería instantánea
- Mensajería unificada

- IPTV
- Audio y videoconferencia
- Servicios Pushtotalk

2.3.1 ARQUITECTURA Y ENTIDADES FUNDAMENTALES

IMS corresponde a una arquitectura basada en estándares y compuesta por 3 capas que soporta un amplio rango de servicios basados en IP para tecnologías móviles y fijas, por lo tanto IMS aparece como un escenario ideal para nuevos modelos de negocios y oportunidades a través de conectividad SIP y movilidad continua para sus usuarios. Su arquitectura está compuesta por tres capas.

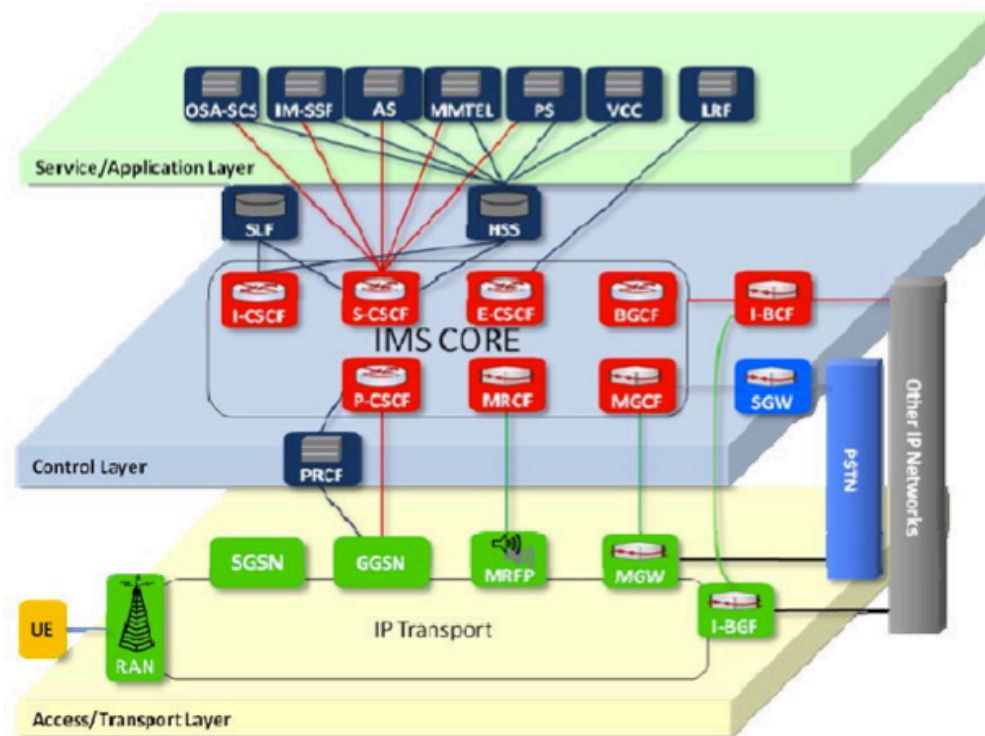


FIGURA 2.3: ARQUITECTURA IMS

Tanto la Figura 2.3 como la descripción de componentes se ha tomado de [17].

2.3.1.1 CAPA DE APLICACIÓN/SERVICIO

Esta capa maneja toda la lógica de servicios y aplicaciones. Las principales componentes se describen a continuación:

Application Server (AS): Corresponde a la entidad SIP que ejecutará los servicios de valor agregado. Los SIP Application Servers pueden simplificar enormemente la construcción de aplicaciones a través del uso de Service Creation Environments (SCEs) que permiten a los desarrolladores concentrarse en el sistema de negocio abstrayéndose de toda la infraestructura.

El resto de componentes se requieren para el servicio de telefonía multimedia IMS.

MMTEL (IMS Multimedia Telephony Service): Fija la cantidad de recursos necesarios para establecer interoperabilidad entre telefonía multimedia y servicios anexos.

OSA-SCS (Open Service Access-ServiceCapability Server): Permite a los suscriptores acceso seguro a las nuevas aplicaciones.

VCC (Voice Call Continuity): Ofrece continuidad de llamadas entre redes fijas y móviles.

LRF (Location Retrieval Function): Determina la locación del usuario.

2.3.1.2 CAPA DE CONTROL

Es principalmente compuesto por el denominado “núcleo IMS” el cual contiene los siguientes elementos de señalización y control.

HSS (Home Subscriber Service): Base de datos de usuarios dentro de la red. Contiene perfiles, credenciales de autenticación, locación física de usuarios y datos relevantes para cada usuario hacia un particular AS.

SLF (Subscriber Locator Function): Base de datos que mapea los usuarios a un HSS.

CSCF (Call Session Control Function): Proxies SIP que autentifican y rutean a los usuarios dentro de las distintas redes. Se clasifican en:

- *P-CSCF (Proxy CSCF)*: Es el punto de entrada, valida mensajes SIP para rutearlos hacia el resto de los CSCF y establece QoS.

- *I-CSCF (Interrogating CSCF)*: Determina a que S-CSCF enviar la información, de acuerdo al usuario en cuestión.
- *S-CSCF (Serving CSCF)*: Es el elemento central de la capa de control, pues realiza el registro de usuario y el control de sesión.

BGCF (Breakout Gateway Control Function): Rutea las llamadas hacia las redes conmutadas, ejemplo PSTN:

IBCF (Interconnection Border Control Function): Punto intermediario entre redes IMS o con alguna otra red basada en SIP. Provee información como la topología de red.

2.3.1.3 CAPA DE TRANSPORTE Y ACCESO

Contiene los elementos de transporte de elementos IP. Sus principales componentes son:

MGW (Media Gateway): Convierte RTP (Real Time Protocol) en PCM (Pulse Code Modulation).

GGSN (Gateway GPRS Support Node): provee conectividad a redes externas de datos, ya sea IMS o Internet.

RAN (Radio Access Network): Establece el tipo de red de acceso ya sea: GSM, UMTS o GSM/EDGE.

User Equipment (UE): Terminal de usuario que permite acceder a servicios y aplicaciones de la red IMS (teléfono celular, laptop, pda, teléfono 3G, softphone, etc.).

2.3.2 SERVICIOS EN IMS

En las especificaciones de 3GPP se establece que la intención no es estandarizar los servicios IP multimedia en sí. La razón es que si así fuera, entonces se inhibiría la habilidad de los operadores de proveer diferenciación en los productos y se desincentivaría la innovación. Lo que se busca estandarizar son las capacidades de los servicios a través de la definición de bloques que permitan acceso a las prestaciones de red. En este sentido, se puede decir que IMS estandariza las

entidades que permiten desarrollar y entregar servicios (*Service Enablers*). Las capacidades de los servicios se pueden hacer visibles a otras entidades en la red tales como los servidores de aplicación (AS) a través de interfaces de aplicación. De esta manera, se pueden usar las interfaces para manipular prestaciones de los servicios y así lograr interacción de manera abierta y segura sin necesidad de tener acceso directo a los demás servicios.

Una de las maneras de lograr la anterior, es la implementación de servicios a través de servidores de aplicación. Un servidor de aplicación en IMS es un servidor que ofrece servicios IP multimedia de valor agregado. Se definen tres tipos de servidores de aplicación, Servidores de Aplicación SIP (SIP-AS), Servidores de Capacidad de Servicios de la organización *Open Service Access* (OSA SCS) y *IP Multimedia Service Switching Function* (IM-SSF). Los SIP-AS alojan servicios desarrollados utilizando el protocolo SIP. El OSA-SCS es un servidor de aplicación SIP, pero que implementa una interfaz a un servidor de aplicación de la OSA en un dominio no alojado en la red. La OSA define una arquitectura que permite a los desarrolladores de servicios de telecomunicaciones utilizar funcionalidades nativas a la red del operador a través de APIs. El objetivo de la OSA es permitir el desarrollo de aplicaciones futuras no conocidas hoy en día al proveer esta interfaz que soporta la manipulación de prestaciones de red. La API permite a los desarrolladores el acceso a prestaciones como el control de llamadas, mensajería, movilidad, y cobrabilidad entre otras. La APIs de la OSA son independientes al lenguaje de programación y hoy en día ya existen implementaciones en Java SE, Java EE y Parlay X (servicios Web). Por último la IM-SSF provee un puente entre el IMS y los servicios desarrollados con CAMEL (*Customized Applications for Mobile Network Enhanced Logic*) para las redes 2G a través de la interfaz CAP (*Camel Application Part*). CAMEL es un ambiente de servicios que permite a operadores móviles antiguos (legacy) crear servicios específicos. El IM-SSF realiza la traducción de los llamados entre SIP y CAP.

2.4 LONG TERM EVOLUTION (LTE)

Long Term Evolution (LTE), es un estándar desarrollado por la 3GPP para comunicaciones inalámbricas de datos en alta velocidad para teléfonos o terminales móviles. Es una evolución de las tecnologías de red GSM/EDGE y UMTS/HSPA, aumentando la capacidad y velocidad usando nuevas técnicas de modulación.

Lo expuesto a continuación es sólo un resumen de algunas de las características relevantes de LTE relacionadas con el trabajo desarrollado. Para mayor información consultar [7] y [8].

2.4.1 LTE COMO PASO HACIA 4G

Con la evolución a 4G se pretende lograr que los operadores inalámbricos aprovechen el aumento en las tasas de transferencia de datos para aumentar la cantidad y el tipo de contenido disponible en los aparatos móviles. Las redes 4G podrán ser soluciones compatibles con IP para desplegar voz, datos y contenido multimedia a usuarios móviles cuándo y dónde sea. Con banda ancha móvil más rápida se pueden soportar servicios de más alto nivel, incluyendo aplicaciones de negocios, streaming de audio y video, mensajería de video, videoconferencias, TV móvil o juegos online. Como un paso hacia 4G, la organización 3GPP comenzó su investigación del estándar LTE como una tecnología viable el 2004. Se espera que LTE ofrezca un número de atributos distintivos por sobre otras tecnologías inalámbricas. Estas ventajas incluyen atributos de mejor rendimiento tales como mayores tasas de transmisión máximas (peak), baja latencia y mejor eficiencia en el uso de espectro radioeléctrico.

2.5 SERVICIOS DE TELEVISIÓN DE PRÓXIMA GENERACIÓN

2.5.1 SERVICIO DE IPTV

2.5.1.1 DESCRIPCIÓN DE IPTV

IPTV (Internet Protocol Television) es una tecnología que día a día crece más en importancia, ya comenzando a interrumpir en los modelos de negocios existentes de los operadores de televisión de pago tradicionales. La definición oficial determinada por la ITU (ITU-T FG IPTV) [13] es la siguiente : “IPTV es definido como servicios multimedia tales como televisión/audio/texto/gráficos/datos transmitidos sobre una red IP gestionada para entregar los niveles requeridos de calidad y experiencia de servicio, así como también seguridad, interactividad y confiabilidad.”

IPTV es un término usado para referirse a la transmisión de canales de televisión tradicional, películas, y video-on-demand sobre redes de datos privadas. Se tiene entonces que desde la mirada del usuario final, IPTV solamente se presenta como otro sistema más de televisión de pago. Bajo la perspectiva de un proveedor de servicios, IPTV comprende la adquisición, procesamiento, y transmisión segura y confiable de contenido de video sobre una infraestructura de red basada en IP. El tipo de proveedores de servicios involucrados en el desarrollo de servicios de IPTV abarca desde los operadores de cable y TV satelital, hasta las grandes compañías de telefonía y operadores de redes privadas en diferentes partes del planeta [14]. Algunas características destacadas de IPTV son:

Soporte para TV interactiva: Las habilidades de comunicación bidireccional de los sistemas de IPTV permiten a los proveedores de servicios entregar un amplio rango de aplicaciones de TV

interactiva. Los tipos de servicios entregados a través de IPTV pueden ser televisión en vivo con sistemas de votación online, solicitud de contenido audiovisual específico (video on demand), juegos multimedia interactivos y navegación por Internet.

Cambiar horarios de transmisión: IPTV en combinación con un grabador de video digital permite cambiar los horarios en que el usuario final ve los contenidos de la programación. Es decir un sistema en que se graba y almacena contenido de IPTV para verlo posteriormente. Esta idea no esta limitada a que sea un PVR (Personal Video Recorder) local, sino esto se puede lograr utilizándose una memoria caché centralizada y administrada por los proveedores de IPTV en donde los usuarios pueden solicitar contenidos en vivo transmitidos en el pasado con una antigüedad temporal solo limitada por el tamaño de este almacenamiento.

Personalizable: Un sistema de IPTV soporta comunicación bidireccional permitiendo que el usuario final pueda decidir que y cuando quiere ver televisión.

Bajo requerimiento de ancho de banda: En vez de transmitir cada canal a cada usuario, las tecnologías de IPTV permiten a los proveedores de servicio sólo transmitir el contenido que el usuario ha solicitado ver y no todos los canales. Esta característica permite a los operadores de red optimizar el uso del ancho de banda de sus redes.

Accesibles en múltiples dispositivos: La visualización del contenido de IPTV no esta limitado a los televisores, los consumidores pueden usar sus PCs o aparatos celulares para acceder a los servicios de IPTV.

2.5.1.2 DIFERENCIAS ENTRE IPTV Y TV POR INTERNET

Existen algunas diferencias entre IPTV y TV por Internet o IP Video. Aunque ambos términos son muy similares, existe una clara distinción en como son usadas en el mercado. IPTV es usado para referirse a la comercialización por parte de proveedores de servicios hacia sus suscriptores de canales de televisión y contenidos multimedia con un aspecto similar a la televisión tradicional de pago. TV por Internet se usa comúnmente para referirse a páginas web o portales que ofrecen programas de televisión o películas on-demand [14]. Aunque ambos sistemas se basan en las mismas tecnologías, sus enfoques en como distribuyen el contenido de Video sobre IP difieren de las siguientes maneras:

Plataformas diferentes: Como su nombre lo sugiere, Internet TV utiliza la red de Internet pública para transmitir el contenido de video a los usuarios finales. IPTV, por el contrario, usa una segura red privada dedicada para transmitir el contenido de video a los consumidores. Estas redes privadas son manejadas y operadas por el proveedor del servicio de IPTV.

Alcance geográfico: Las redes pertenecen y son controladas por las compañías operadoras de telecomunicaciones siendo no accesibles por los usuarios de Internet y siendo localizadas en un lugar geográfico determinado. Internet, por el contrario, no posee limitaciones geográficas donde los servicios pueden ser accedidos desde cualquier parte del planeta.

Dueños de la infraestructura de red: Cuando el video es enviado sobre la red pública de Internet, algunos de los paquetes IP usados para transportar el video sufren retardos o se pierden completamente al atravesar las numerosas redes que componen la Internet. Como resultado, los proveedores de contenido de video sobre Internet no pueden garantizar una experiencia de ver televisión que se pueda comparar con la ver televisión tradicional. De hecho, el video transmitido sobre Internet puede a veces verse muchas veces con cortes o con una baja resolución de imagen. El contenido de video es generalmente transmitido al usuario final a través del mejor esfuerzo posible. En comparación a esto, IPTV es transmitido sobre una infraestructura de red, que pertenece típicamente al proveedor de servicios. Ser dueño de la infraestructura de red permite a las empresas proveedoras de telecomunicaciones poder gestionar sus sistemas de modo de hacer posible la entrega de video de alta calidad.

Mecanismo de acceso: Un Set-Top Box digital es generalmente usado para acceder y decodificar el contenido de video transmitido a través de un sistema de IPTV y reproducido típicamente en un televisor estándar, mientras que un PC es casi siempre utilizado para acceder a los servicios de Internet TV. El tipo de software usado en el PC va a depender del tipo de contenido de Internet TV que se desea. Por ejemplo, la descarga de contenidos desde un portal de Internet TV a veces requiere la instalación de un reproductor especializado para ver el contenido. Un sistema robusto de Digital Rights Management (DRM) es necesario también para este mecanismo de acceso.

Costos: Un porcentaje significativo del contenido de video que es entregado sobre la red de Internet pública está disponible gratuitamente. Aunque un creciente número de empresas de medios de comunicación han empezado a introducir servicios de Internet TV de pago. La estructura de costos aplicada a los servicios de IPTV es similar al modelo de suscripción mensual adoptada por los proveedores tradicionales de TV de pago. Con el tiempo, varios analistas esperan que la TV por Internet e IPTV converjan en un solo servicio de entretenimiento.

2.5.1.3 INFRAESTRUCTURA GENERAL DE UNA RED DE IPTV

La Figura 2.4 muestra los requerimientos funcionales de alto nivel típicos de un sistema de IPTV end-to-end [14].

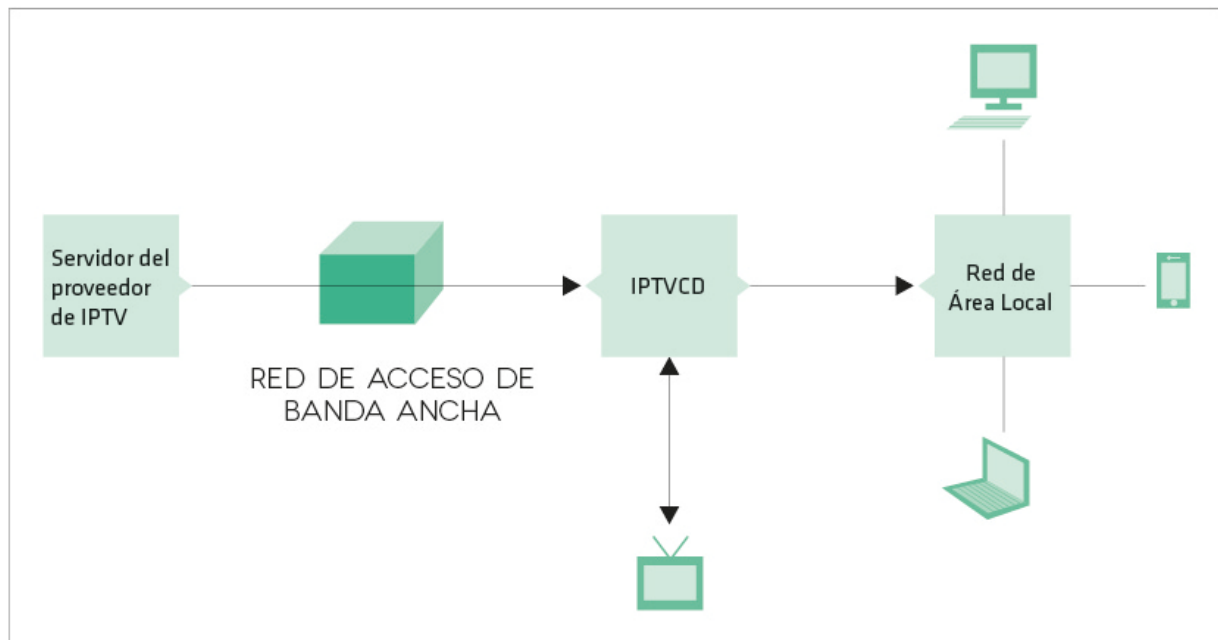


FIGURA 2.4: DIAGRAMA DE BLOQUES DE UNA RED DE IPTV

IPTV Data Center: También conocido como “Headend”, el IPTV Data Center recibe el contenido desde una diversidad de fuentes incluyendo todo tipo de generadores de contenido, productoras, canales de televisión gratuitos y de pago. Una vez recibida la información, diferentes equipos de hardware como encoders y servidores de video, así como también IP routers y hardware dedicado a la seguridad son usados para preparar el contenido de video que será transmitido sobre la red IP. Adicionalmente, es necesario un sistema de manejo de suscriptores para gestionar los datos de los clientes del servicio de IPTV y realizar labores de tarificación. La localización física del IPTV Data Center estará dada por la infraestructura de red usada por el proveedor de servicios.

Broadband Delivery Network: La entrega de servicios de IPTV requiere de conexiones uno a uno. En el caso de grandes despliegues de IPTV, el número de conexiones uno a uno aumenta significativamente y la demanda en términos de ancho de banda para la infraestructura de la red puede ser bastante grande. Los avances tecnológicos en el último par de años permiten que los proveedores de telecomunicaciones puedan enfrentarse a la demanda. Infraestructuras híbridas de fibra óptica y redes coaxiales son particularmente convenientes para transportar contenido de IPTV.

IPTVCDs: Dispositivos de consumo de IPTV (IPTVCDs) son los componentes clave para permitir que la gente pueda acceder a servicios de IPTV. El IPTVCD conecta al usuario a la red IP y es responsable de decodificar y procesar el contenido de video entrante. Los IPTVCDs ayudan con tecnologías avanzadas que minimizan o eliminan completamente los efectos de los problemas en la red cuando se procesa el contenido de IPTV. Los IPTVCDs más populares son Gateways residenciales, IP Set-Top-Boxes, consolas de videojuegos y Media Servers los cuales aumentan cada vez en su sofisticación.

Home Network: Una red hogareña conecta dispositivos digitales en una pequeña área geográfica. Esto mejora la comunicación y permite el intercambio de grandes volúmenes de contenidos digitales entre miembros de una familia. El propósito de una red hogareña es proveer acceso a la información, como es voz, audio, datos y entretenimiento entre diferentes dispositivos digitales a lo largo de una casa. Con redes hogareñas locales, los consumidores pueden ahorrar dinero y tiempo porque los periféricos como impresoras y scanners, como también las conexiones de banda ancha, pueden ser fácilmente compartidos.

2.6 PROYECTOS UTILIZADOS PARA LA IMPLEMENTACIÓN.

En esta sección se describen a grandes rasgos algunos de los proyectos de código libre o bien disponibles gratuitamente que comprenden parte del ambiente de prueba a construir.

2.6.1 FOKUS OPENIMS CORE

Open IMS Core [21] es un proyecto de software open source que se inicio a fines del 2006 y que es manejado por el instituto de investigación Fraunhofer FOKUS (Fraunhofer - Institute for Open Communication Systems) en Alemania. El principal objetivo del instituto FOKUS es potenciar el desarrollo de nuevas tecnologías de comunicación de próxima generación, entregando un marco de trabajo basado en software *open-source* que permita establecer prototipos de tecnologías acorde al mercado y los requerimientos del usuario final.

Una temática que particularmente aborda FOKUS son las tecnologías emergentes. Dentro de este marco se ha diseñado e implementado en base a herramientas *open-source* la plataforma de pruebas denominada “Open IMS Core” que permite testear la tecnología y probar servicios. Dicho desarrollo está disponible para que cualquier persona pueda experimentar y trabajar con fines experimentales y/o docentes. En ningún caso constituye un kit de desarrollo para la implementación de plataformas comerciales. Más aún, debe tenerse en cuenta que dada la cantidad de organizaciones involucradas en IMS, posteriormente podrían existir patentes de por medio que deberán ser respetadas.

IMS aún se encuentra en etapas de prueba con un número creciente de operadores interesados en la tecnología. Mientras que en el área de VoIP existen muchos proyectos *open-source* asociados a clientes SIP, proxies y herramientas (alrededor del estándar SIP IETF), prácticamente no existen proyectos *open-source* con foco en IMS. Luego, el objetivo principal de esta iniciativa es llenar ese vacío para entregar una herramienta que permita probar y desarrollar servicios IMS y permitir el estudio del core de dicha arquitectura.

Según lo anterior, el Open IMS Core es una implementación de las entidades CSCFs y de un HSS de bajo peso que en conjunto forman los elementos *core* de todas las arquitecturas IMS/NGN especificadas dentro de 3GPP, 3GPP2, ETSI TISPAN y la iniciativa PacketCable. Las cuatro componentes (S-CSCF, P-CSCF, I-CSCF, HSS) están basadas principalmente en los proyectos SIP Express Router (SER) y MySQL, los cuales también son *softwares* de código libre.

La Figura 2.5 [21] muestra un diagrama simplificado de su arquitectura.

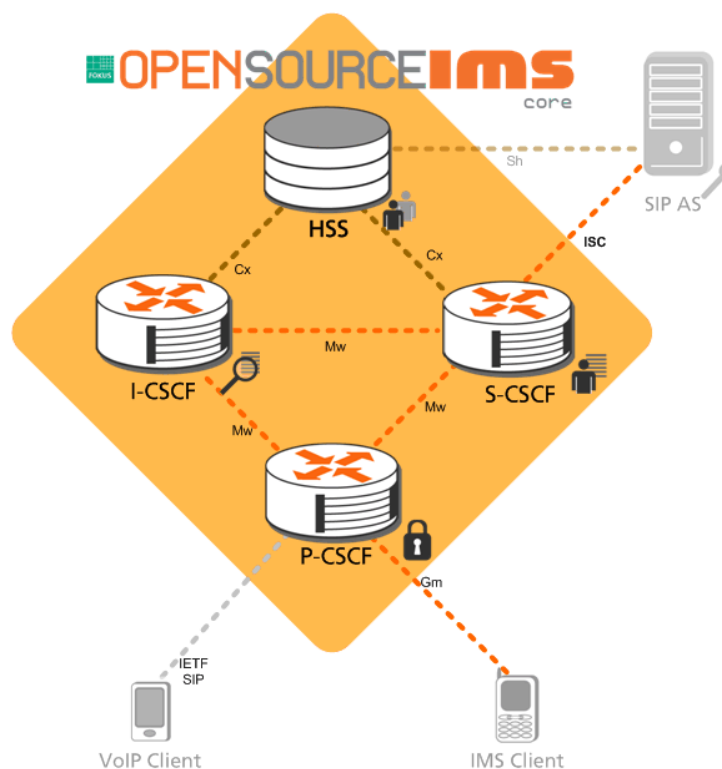


FIGURA 2.5: DIAGRAMA SIMPLIFICADO DE LA ARQUITECTURA DEL OPENIMSCORE

2.6.2 UCT ADVANCED IPTV SERVER

UCT Advanced IPTV Server [22] es un proyecto que pretende implementar un servicio de IPTV basado en IMS que cumpla con los estándares tradicionales asociados a este servicio. Su desarrollo es parte de una investigación de IPTV-IMS aún en curso en el Grupo de Investigación en Comunicaciones (*Communications Research Group*) en la Universidad de Cape Town, Sudáfrica. En particular, fue diseñado específicamente a raíz del proyecto UCT IMS Client, un cliente IMS también desarrollado en este grupo de investigación.

En términos generales, UCT Advanced IPTV Server es un servidor de aplicación (AS) SIP que permite emular el servicio de transmisión de televisión tradicional y el servicio de VoD cumpliendo con ciertos estándares de calidad de servicio, control y cobrabilidad. Al utilizar exclusivamente señalización SIP, se le permite encajar perfectamente dentro de la arquitectura IMS existente. Este enfoque reduce enormemente la complejidad del *software* al utilizar ciertas prestaciones de IMS para enriquecer el servicio. Este es el caso de la autenticación de usuarios, que es manejada por los servicios de aprovisionamiento tradicionales presentes en IMS. Por otro lado, cada sesión multimedia que establezca el servidor con algún cliente es manejada por IMS como una llamada en un solo sentido, lo que le permite al servidor manejar un gran número de transmisiones simultáneas.

Para que el servidor sea capaz de responder a las solicitudes de los clientes, se utiliza el mecanismo de la IETF de redirección de contenido RFC 4483 (*Content Indirection*) [25]. La idea principal del mecanismo es redirigir las peticiones dentro de los mensajes SIP (Mensajes tipo *Request*) hacia direcciones RTSP donde se aloja el contenido multimedia requerido. Para que efectivamente ocurra la transmisión, se debe utilizar un servidor adicional de Streaming multimedia donde se aloja el contenido en estas direcciones RTSP. En este sentido, el funcionamiento del servicio involucra tres etapas; En primer lugar el cliente IMS genera un mensaje SIP *INVITE* para alguno de los canales disponibles y se lo envía al servidor de redirección. Luego el servidor consulta una tabla que mapea los canales requeridos con direcciones RTSP y devuelve la dirección correspondiente al cliente dentro de un mensaje SIP *OK*. Finalmente, el cliente IMS inicia una sesión RTSP con el servidor multimedia.

Las llamadas entre el servidor de aplicación y el cliente deben ser ruteadas por el IMS, pero el cliente internamente debe ser capaz de interpretar la dirección RTSP que recibe y hacer la llamada al servidor multimedia. El servicio es compatible con cualquier cliente SIP capaz de realizar dicha función y que además pueda recibir y decodificar el estándar de video H.263-1998 y el estándar de audio MPEG1.

En la Figura 2.6 [27] se muestra un diagrama simplificado de la arquitectura del sistema.

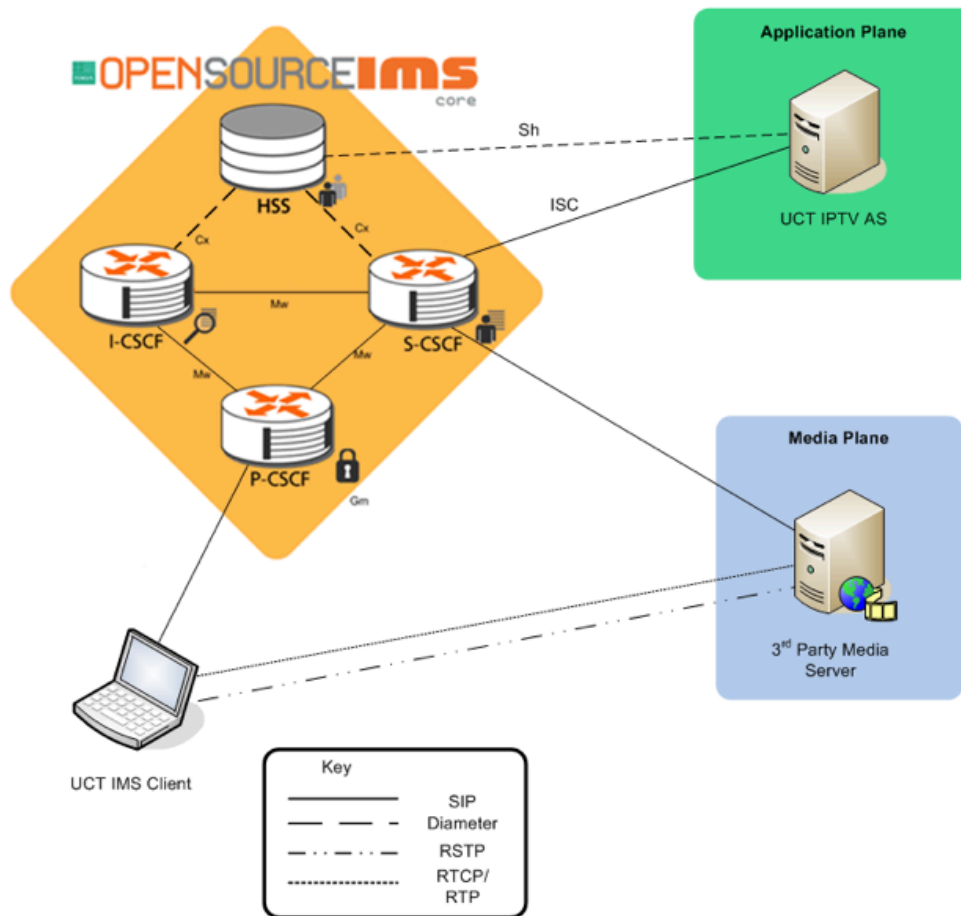


FIGURA 2.6: DIAGRAMA DE LA ARQUITECTURA DE UCT ADVANCED IPTV SERVER

El proyecto es de código libre y se pone a disposición en [27] en conjunto con una guía de configuración para su funcionamiento con el proyecto FOKUS Open IMS Core

2.6.3 UCT IMS IPTV CHARGING SYSTEM

UCT IMS IPTV Charging System [24] es un proyecto que pretende implementar una plataforma de cobros para servicios IMS que cumpla con los estándares tradicionales asociados a este servicio. Su desarrollo es parte de una investigación de IMS aún en curso en el Grupo de Investigación en Comunicaciones (*Communications Research Group*) en la Universidad de Cape Town, Sudáfrica. En particular, fue diseñado específicamente a raíz del proyecto UCT Advanced IPTV Server, el cual fue descrito en la sección anterior y aún se encuentra en desarrollo.

Como se establece en la estandarización de la arquitectura de cobros en IMS [26], se debe soportar cobros en línea (*online*) y fuera de línea (*offline*). En los cobros online debe existir una interacción en tiempo real entre los mecanismos de cobro y el consumo de recursos. Por otro lado, en los cobros offline, el control del cobro no depende del consumo de servicios en vivo. En

este sentido, en el proyecto UCT IMS IPTV Charging System se implementan entidades; CTF (*Charging Trigger Function*), CDF (*Charging Data Function*) y OCF (*Online Charging Function*).

CTF es la entidad encargada de provocar los avisos de cobro y se comunica con las otras dos entidades mediante mensajes del protocolo Diameter. Esta entidad muchas veces se integra directamente en el servidor de aplicación que presta el servicio, por lo que este proyecto desarrolla un nuevo servidor de IPTV que integra esta entidad. Las entidades OCF y CDF se encargan del manejo de cobros online y offline respectivamente. OCF lo hace a través de una interfaz denominada *Ro*, mientras que OCF lo hace a través de un punto de referencia denominado *Rf*.

El sistema de cobro utiliza el siguiente mecanismo: En primer lugar el cliente IMS envía un mensaje SIP de tipo *INVITE* para solicitar un canal de la plataforma IPTV (según se describe en la sección anterior). Luego, el servidor de IPTV consulta la tabla que mapea los canales con direcciones RTSP. Cuando esto ocurre, el CTF provoca un aviso al OCF o CDF según sea la manera de cobrar del servicio (online o offline). Con esto el cobro comienza, y entonces el servidor IPTV efectivamente devuelve al cliente la dirección RTSP dentro de un mensaje SIP de tipo *OK*. Con esta dirección, el cliente comienza la sesión contactando al servidor multimedia ubicado en la dirección RTSP que se le entregó. El cobro finalmente se detiene cuando el cliente envía un mensaje SIP de tipo *BYE* al servidor de IPTV. En el caso del cobro online, que descuenta fondos por segundo de consumo, si durante una sesión multimedia se acaban los fondos, entonces el OCF dispara una alerta e inmediatamente el servidor termina la sesión.

En la Figura 2.7 [28] se muestra un diagrama simplificado de la arquitectura del sistema.

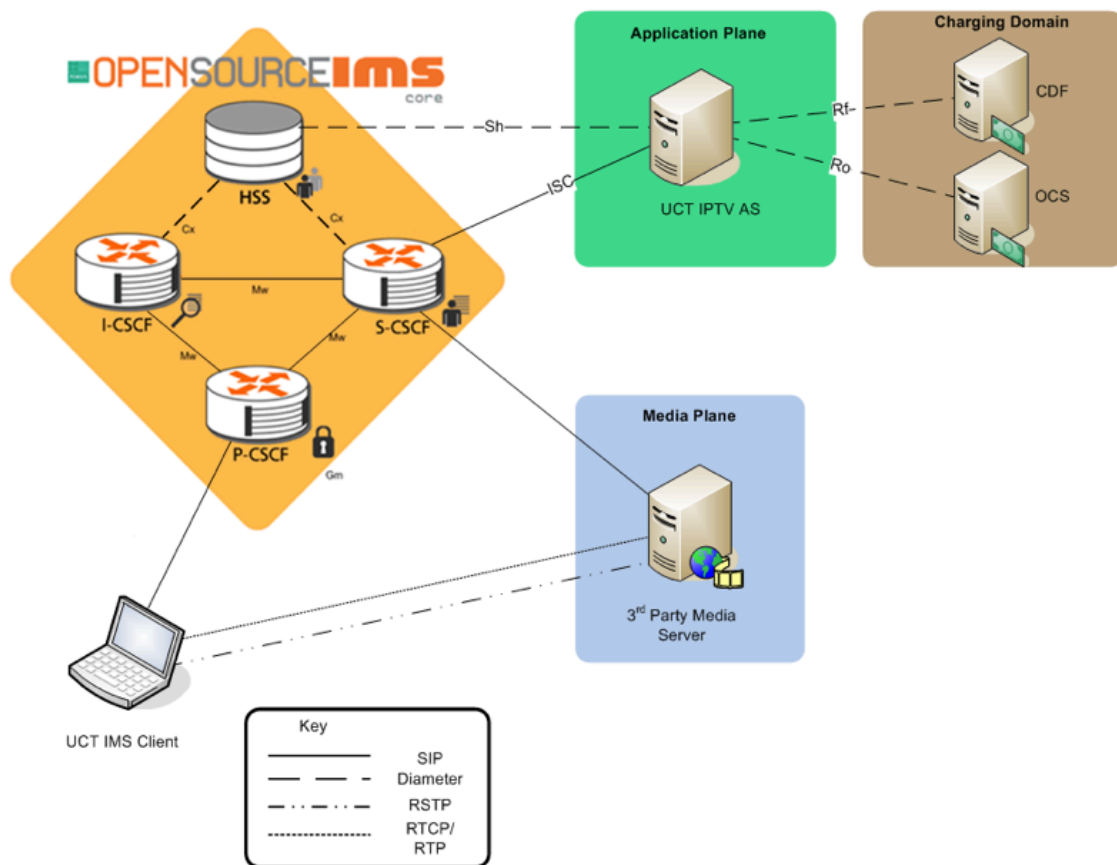


FIGURA 2.7: DIAGRAMA DE LA ARQUITECTURA DE UCT IMS IPTV CHARGING SYSTEM

Este proyecto también es de código libre y se pone a disposición en [28] donde al igual que el proyecto descrito anteriormente se incluye una guía de configuración para su funcionamiento con el proyecto FOKUS Open IMS Core

2.6.4 VLC MEDIA PLAYER

VLC Media Player [35] es el proyecto principal del grupo de desarrollo Open Source *VideoLAN Organization* [34] que se dedica al desarrollo de software para reproducir y transmitir video en múltiples formatos.

VideoLAN Client (VLC) inicialmente pretendía ser el cliente para recibir video y VideoLAN Server serviría para transmitirlo. Actualmente ambos proyectos convergen en VLC Media Player que cumple con ambas funciones. VLC Media Player es ahora un reproductor multimedia multiplataforma altamente portable que es capaz de interpretar casi todos los codecs existentes y a la vez un potente servidor de streaming con soporte para diversos protocolos tanto para transmisión en tiempo real o bajo demanda.

2.6.5 MOBICENTS

Mobicents [30] es un proyecto que persigue transformarse en la plataforma líder de comunicaciones en la nube, en el contexto de los software de código libre. Mobicents provee de un entorno de ejecución y un conjunto fácil de comprender de herramientas para desarrollar, desplegar y manejar servicios que integran voz, video y mensajería a través de una serie de redes de comunicación.

De esta manera se posibilita la creación de aplicaciones de comunicación que sean escalables y confiables de una manera más simple y rápida. En el enfoque de las redes NGN, Mobicents encaja como la base para el desarrollo de un SDP (*Service Delivery Platform*), promoviendo un ambiente estándar para el mercado de los desarrolladores de aplicaciones.

Mobicents dispone de los siguientes sub proyectos, a los cuales llaman *enablers* para denotar que *permiten* el desarrollo diversas aplicaciones en distintos contextos.

- Mobicents JAIN-SLEE
- Mobicents SIP Servlets
- Mobicents Media Server
- Mobicents Diameter
- Mobicents SIP Presence Service
- Mobicents SS7

Mobicents JAIN-SLEE es un servidor de aplicación orientado a eventos altamente escalable que contiene un robusto modelo de componentes y un ambiente de ejecución tolerante a fallas. Mobicents SIP Servlets es una plataforma abierta para el rápido desarrollo y despliegue de aplicaciones SIP. Mobicents Media Server es un servidor multimedia para voz sobre IP (VoIP) y aplicaciones para IMS o arquitecturas de redes convergentes. Mobicents Diameter es un conjunto de implementaciones de la familia de protocolos Diameter, tanto del lado del cliente como del servidor. Permite el rápido desarrollo de componentes de IMS como los CSCFs, HSS o servidores de aplicación. Mobicents SIP Presence Service provee de un servidor de presencia para redes basadas en SIP, un XDMS (XML Document Management Server) y un servidor de listas de recursos que cumplen con los estándares tradicionales (IETF, OMA y 3GPP), todo implementado sobre JAIN-SLEE. Mobicents SS7 es un conjunto de APIs que implementan el sistema de señalización SS7.

Cada sub proyecto puede descargarse en conjunto con el servidor de aplicación JBOSS que actúa como contenedor de los binarios que vienen previamente cargados. También es posible la descarga del código fuente, lo que permite compilarlo y cargarlo manualmente. Además, para entender más fácilmente el funcionamiento de cada subproyecto, se incluyen algunos ejemplos explicativos que utilizan las herramientas disponibles para construir aplicaciones típicas u otras prestaciones útiles.

En este trabajo son sólo de interés Mobicents JAIN-SLEE y Mobicents Media Server y en particular uno de los ejemplos contenidos en este último.

2.6.6 MOBICENTS JAIN SLEE

JAIN-SLEE es la especificación para la lógica de servicios y la arquitectura para un ambiente de ejecución basados en Java (SLEE – *Service Logic and Execution Environment*) que utiliza las APIs de redes inteligentes (JAIN – *Java APIs for Intelligent Networks*). Se creó en el *Java Community Process* (JCP), una comunidad que formaliza el proceso de definición de futuras versiones y características de la Plataforma Java a través requerimientos de estandarización llamados *Java Specification Request* (JSR). Esta especificación corresponde al JSR 240 [31] y en su definición participaron diversos individuos y compañías, incluyendo a Red Hat.

Un SLEE es un servidor de aplicación, o contenedor de servicios, que define un modelo de componentes para estructurar la lógica de los servicios de comunicación como un conjunto de componentes reutilizables, que además permite combinarlos con la intención de crear servicios aún más sofisticados. Este modelo fue diseñado y optimizado para aplicaciones orientadas a eventos. Por otro lado, el modelo de componentes SLEE también define interfaces de gestión utilizadas para administrar el contenedor y los componentes de servicios ejecutándose en él.

Las APIs de JAIN implementan por si solas un conjunto de protocolos como SIP, MGCP y otros. Por lo tanto, al basar la construcción del SLEE en JAIN, se permite el soporte para múltiples protocolos de internet y su interoperabilidad con otros sistemas más grandes que también estén basados en protocolos abiertos. SLEE aprovecha las abstracciones que provee JAIN al definir un marco para el desarrollo de adaptadores de recursos (RAs – *Resource Adapters*). Estos adaptadores residen afuera del ambiente SLEE, pero realizan la importante labor de adaptar los mensajes de protocolos externos (como los mensajes SIP) a eventos que pueden ser consumidos por el SLEE y vice-versa. Para la creación de servicios, JAIN-SLEE utiliza el concepto de bloques de construcción (SBBs – *Service Building Blocks*) para definir las componentes de aplicación atomizadas y reutilizables que se mencionaban anteriormente. Los SBB pueden suscribirse a la recepción de eventos y generar respuestas, función que es asistida por un enrutador de eventos.

Mobicents JAIN-SLEE [32] es la primera y única plataforma de código libre certificada para los estándares JAIN SLEE 1.0 y 1.1 y fue publicada bajo la licencia LGPL. Es el proyecto base o núcleo para el funcionamiento de la mayoría de los otros proyectos.

2.6.7 MOBICENTS MEDIA SERVER

En el mundo moderno de VoIP, los proveedores ofrecen servicios altamente customizados que combinan audio, video o mensajería. Estos servicios requieren una capacidad de procesamiento multimedia dedicada, por lo que las redes de VoIP separan las funciones de procesamiento en nodos individuales responsables sólo de esta tarea, excluyéndolos de la inteligencia que se encarga de controlar las llamadas. Este nodo que procesa los flujos de multimedia en la red toma el nombre de Servidor Multimedia o *Media Server*.

Al mismo tiempo, aún existen servicios multimedia en sistemas de comunicación tradicionales (*legacy systems*). Esto significa que el nodo de manejo multimedia debe ser capaz de cubrir la brecha entre estos sistemas y las redes VoIP modernas. A este nodo típicamente se le llama *Media Gateway* pero muchas veces un servidor multimedia también cumple con esta función.

El Mobicents Media Server [33] es la componente multimedia de la plataforma Mobicents y fue publicada bajo la licencia LGPL versión 2.1. Según lo que se describió anteriormente, es un Servidor Multimedia de código libre, desarrollado en Java, que soporta interfaces IP y tradicionales. En este sentido, actúa tanto como un Servidor Multimedia y un Media Gateway.

El Mobicents Media Server puede operar en conjunto con el servidor de aplicación Mobicents JAIN-SLEE, pero además está provisionado con la interfaz estándar de las Telco, el protocolo de control para las Media Gateway MGCP (Media Gateway Control Protocol), lo que permite su uso en conjunto con cualquier otro controlador de llamadas si se quisiera. Además de este protocolo de control, el servidor multimedia puede definir diversos otros protocolos relevantes, entre los cuales se puede mencionar MEGACO, MSCML y JSR 309. MEGACO es un protocolo de la entidad de estandarización IETF definido para controlar un servidor multimedia. MSCML (*Media Server Control Markup Language*) también es un protocolo de la IETF que define un lenguaje que puede ser utilizado en conjunto con SIP para controlar un servidor multimedia. MSCML generalmente se utiliza para servicios de conferencias multimedia. La especificación JSR 309 define una API genérica y fácil de utilizar, para el control multimedia que logra esconder la complejidad de los protocolos de control de llamadas a los desarrolladores. El servidor tiene soporte para los codecs de audio G.711a, G.711u, GSM, G.729 y speex, además del códec de video H.261.

Finalmente, es importante destacar que el servidor multimedia provee una interfaz SIP que puede ser utilizada para implementar una interfaz con el S-CSCF.

Es de particular interés un ejemplo de un servicio de compras online (Shopping-Demo) contenido en este proyecto. En las siguientes secciones se describirá como se utilizó dicho ejemplo para el desarrollo de la plataforma para video.

2.6.8 UCT IMS CLIENT

UCT IMS Client [23] es un cliente para la arquitectura IMS específicamente diseñado para ser usado en conjunto con el proyecto FOKUS Open IMS Core. El cliente fue desarrollado por el Grupo de Investigación en Comunicaciones (*Communications Research Group*) en la Universidad de Cape Town, Sudáfrica. Actualmente el proyecto sigue en desarrollo activo, y se va dando solución a problemas que se van detectando.

El cliente soporta autenticación AKA y emula la señalización en IMS lo mejor posible. Es posible realizar llamadas de voz y video con diversos codecs, permite mensajería instantánea, es compatible con servidores de presencia, incorpora un cliente XCAP (XML Configuration Access Protocol) y es capaz de interpretar mensajes SIP con contenido de redirección e interpreta adecuadamente mensajes RTSP, lo que lo hace compatible con el servidor UCT Advanced IPTV Server. Esta última característica es esencial y es la principal razón por la que se escoge este cliente por sobre otras alternativas, pues será requerida para poder utilizar tanto los servicios basados en el servidor de IPTV de la Universidad de Cape Town, como los basados en Mobicents. Considerando su compatibilidad con servidores de IPTV, este cliente además incorpora la posibilidad de visualizar una guía de programación (EPG – *Electronic Programming Guide*).

El proyecto es de código libre y está publicado bajo la Licencia Pública General GNU versión 3 (GPL v.3.0) [29].

3 CAPÍTULO III: METODOLOGÍA

3.1 DESCRIPCIÓN DE METODOLOGÍA

Para lograr el objetivo principal de este trabajo, que es el desarrollo de servicios de telecomunicaciones, se define una serie de objetivos específicos que definen un esquema de trabajo. Los objetivos específicos definen la siguiente metodología genérica para un proyecto de desarrollo e implementación.

En primer lugar, se hace una revisión de conceptos generales para la comprensión del contexto en el que se enmarca el trabajo. A continuación, ya con una mayor interiorización en el tema, se comienza la búsqueda de alternativas de solución para realizar el desarrollo. Luego, una vez seleccionadas estas alternativas se hace un estudio más profundo de las herramientas disponibles para evaluar cómo se hará uso de estas para construir el desarrollo en base a las alternativas de solución. Este estudio conlleva a la realización de un diseño de implementación que detalle la estructura general que se desea lograr. Se procede con la puesta en marcha de dicho diseño, realizando una división por etapas que garantice el adecuado funcionamiento de manera secuencial y así evitar que se dificulte la detección de errores. Finalmente, se concluye con una evaluación del resultado con pruebas que permitan corroborar la correcta implementación del diseño. Todo este proceso se realiza de manera iterativa, donde evidentemente se puede retroceder en ciertos pasos. Por ejemplo, en caso de que se determine que no es posible la realización de un diseño adecuado dadas las herramientas disponibles y las alternativas de solución seleccionadas, se debe volver a la investigación para encontrar mejores alternativas.

En este trabajo las alternativas de solución corresponden a los proyectos de código libre que se seleccionaron y las herramientas disponibles corresponden al laboratorio de LTE. En este sentido, los objetivos específicos corresponden justamente a los pasos de este proceso.

En el capítulo anterior, se hizo una revisión de los conceptos generales y un resumen de las características principales de los alternativas de solución seleccionadas. En las secciones siguientes se continua con la descripción de los siguientes pasos de la metodología. Por otro lado, en los Anexos, se incluye la descripción paso a paso y por etapas constructivas de la puesta en marcha de los servicios desarrollados

3.2 CONTEXTO DE TRABAJO

El trabajo de memoria se encuentra enmarcado en el convenio de cooperación tecnológica entre la empresa ZTE y la Facultad de Ciencias Físicas y Matemáticas de la Universidad de Chile, logrando así disponer del primer laboratorio de tecnología LTE en Latinoamérica [15]. El laboratorio permitirá que alumnos, profesores e investigadores tengan acceso a capacitarse y efectuar trabajos y ensayos en tecnologías de última generación a nivel mundial. También considera este acuerdo, que los distintos profesionales de las empresas de telecomunicaciones se incorporen y participen activamente en jornadas de entrenamiento y pruebas prácticas en el laboratorio.

Se aprovecha esta instancia para seguir con la línea propuesta por trabajos anteriores, enriqueciendo aún más los objetivos logrados. Se pueden mencionar la implementación de una red con arquitectura IMS [16] y el desarrollo de un laboratorio docente para actividades de IPTV sobre IMS [17].

Por otro lado, es importante mencionar que este trabajo pertenece a un proyecto de desarrollo con trabajos paralelos que convergen en diversos objetivos comunes. Estos trabajos son por un lado el diseño de metodologías para la medición de Calidad de Servicio (QoS) y Calidad de Experiencia (QoE – *Quality of Experience*) sobre plataformas IMS montadas en LTE [18] y el diseño de un curso teórico y práctico que utilizará herramientas y servicios desarrollados tanto en este trabajo como en el recién mencionado [19].

3.3 ARQUITECTURA DISPONIBLE

Según el acuerdo entre ZTE y la Universidad de Chile mencionado anteriormente, para el año 2011 se instaló y se puso en marcha el laboratorio con tecnología LTE. Dicho laboratorio estuvo disponible para la realización de pruebas y experiencias relacionadas con este trabajo de memoria.

Una descripción más detallada de los elementos disponibles en el laboratorio se puede encontrar en [19]. A continuación se presenta un resumen de los aspectos más relevantes para este trabajo. Algunas de las figuras se han tomado directamente de [19].

Los principales equipos del núcleo de la red en conjunto con algunas antenas “indoor” fueron instaladas en el segundo piso de Electrotecnologías en el departamento de Ingeniería Eléctrica.

Por otro lado, para la simulación de un entorno más semejante a un escenario real, también se instalaron estaciones base y antenas “outdoor” (eNodeB completos) en dos puntos distantes. Esto último permite generar varias celdas de cobertura y poder experimentar con el paso de una celda a otra en tiempo real. Una de las estaciones fue ubicada en la terraza superior del Departamento de Ingeniería Eléctrica (DIE) y la otra en la terraza del Edificio de Computación (DCC).

La Figura 3.1 muestra una vista satelital (Google Earth) de la ubicación de las antenas identificando las celdas de cobertura que generan. La cobertura que se muestra es sólo una aproximación que solo tiene el propósito de representar la dirección de estas celdas.



FIGURA 3.1: COBERTURA DE LTE DE LA FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS

Los equipos disponibles permiten un funcionamiento de la red LTE en modo TDD y FDD y utiliza las siguientes frecuencias de acuerdo a cada modo:

- La SUBTEL asignó a la FCFM las bandas de frecuencias; (2530 Mhz -2550 Mhz) y (2640 Mhz -2660 Mhz), para uso del Laboratorio LTE en modalidad FDD.
- El Operador VTR facilitó, para efectos de prueba la Banda 38 de frecuencia (2580 Mhz - 2600 Mhz) centrada en 2590 Mhz, para uso del Laboratorio LTE en modalidad TDD.

La Figura 3.2 muestra los equipos que se instalaron para cumplir con las prestaciones mencionadas anteriormente. Adicionalmente se muestra diagrama del Núcleo EPC y Acceso E-UTRAN.

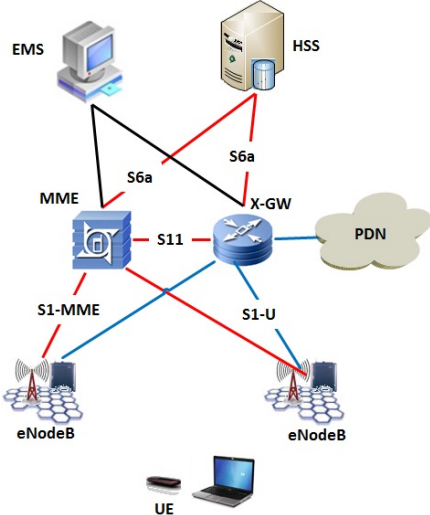
Equipamiento LTE - ZTE	Diagrama del Core EPC y Acceso E-UTRAN
<p>7 * eNodeB:</p> <ul style="list-style-type: none"> •FDD-LTE : 4 eNodeB <ul style="list-style-type: none"> 3X 1B8200&2R8880 1X 1B8200&1R8880 ·TDD-LTE : 3 eNodeB <ul style="list-style-type: none"> 3X 1B8200&2R8928D <p>1* EPC:</p> <ul style="list-style-type: none"> 1*MME (uMAC) 1*S-GW/P-GW (x-GW) 1*HSS <p>1*EMS:</p> <ul style="list-style-type: none"> 1*NetNumen M31 <p>1 * Service System:</p> <ul style="list-style-type: none"> 1*Application Server (PDN) <p>4 * UE:</p> <ul style="list-style-type: none"> 4* UE de Prueba (ZTE MF820 LTE modem) 	 <p>El diagrama ilustra la arquitectura de red LTE. En la parte superior izquierda, un EMS (Equipo de Monitoreo y Gestión) se conecta a un MME (Controlador de Movilidad) y a un X-GW (Gateway). En la parte superior derecha, un HSS (Sistema de Suscripción de Usuarios) se conecta al MME y al X-GW. El MME y el X-GW están interconectados por la interfaz S11. El X-GW está conectado a un PDN (Red de Datos). Dos eNodeB (Estaciones Base) están conectados al MME y al X-GW a través de las interfaces S1-MME y S1-U, respectivamente. En la parte inferior, un UE (Equipo de Usuario) se conecta a los eNodeB.</p>

FIGURA 3.2: EQUIPAMIENTO INSTALADO EN EL LABORATORIO

En la Figura 3.3 se muestra un diagrama de red de los equipos incluyendo conexiones, IP y lugar físico donde se instala:

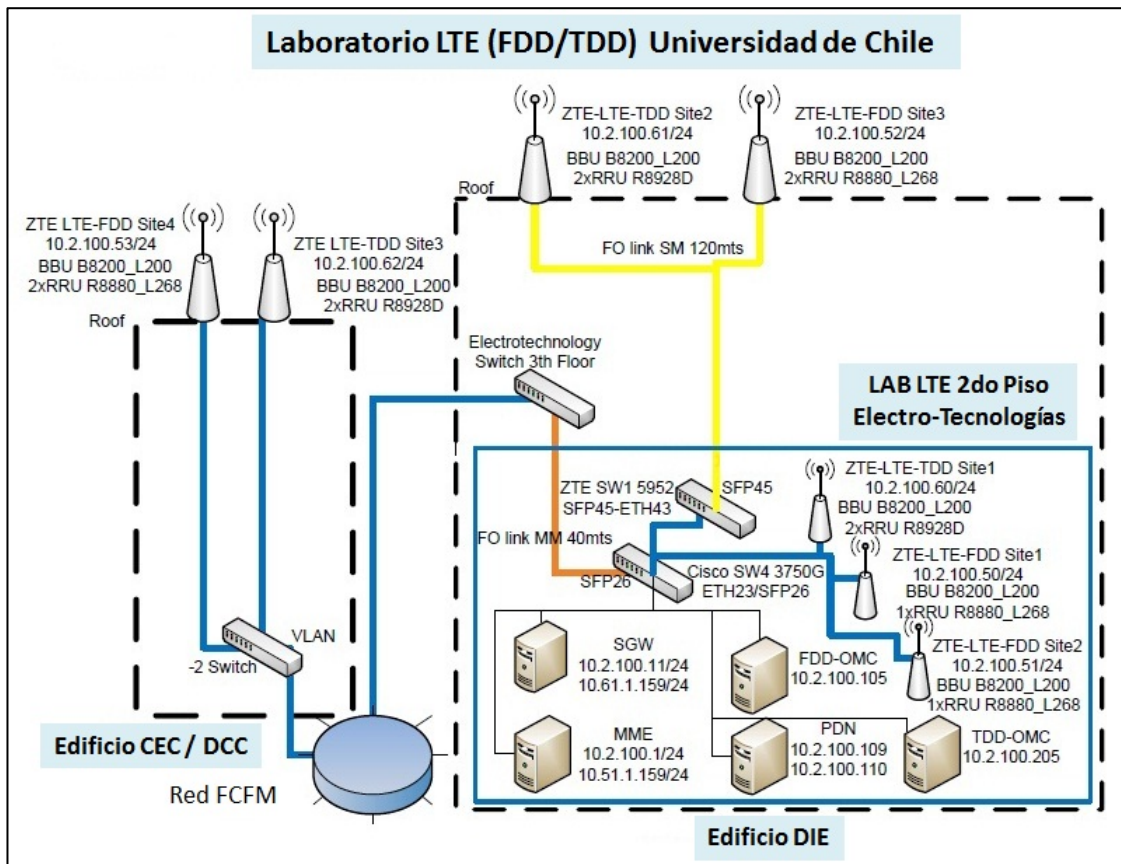


FIGURA 3.3: ARQUITECTURA DE LA RED LTE DISPONIBLE EN EL LABORATORIO

3.4 DISEÑO GENERAL

En esta sección se detalla el diseño de los servicios a implementar considerando limitaciones y características que se busca que posean presentando cada elemento que lo conformará de manera de satisfacer los objetivos planteados.

3.4.1 ALCANCES DEL DISEÑO

En este trabajo se pretende implementar servicios de televisión que operen sobre redes NGN de manera de permitir su uso de manera convergente en cualquier red de acceso, incluidas las redes para dispositivos móviles. Por otro lado se busca que el servicio permita la interacción con el usuario y entre usuarios, permitiéndole a estos la elección del contenido en horarios flexibles y en un ambiente seguro con posibilidad de cobro inteligente, ya sea por tiempo de uso o por

contenido individualizado. Se pretende desarrollar servicios que logren adaptarse de mejor manera a las necesidades que los consumidores de televisión tienen hoy en día, ofreciendo mejores interfaces gráficas, pero que a la vez logre entregar las facilidades para ofrecerlo de cara al operador de red.

Como ya se mencionó, este trabajo continúa los trabajos de título [16] [17], que establecieron el camino a seguir para la implementación de servicios sobre redes convergentes. En este sentido, se vuelve a elegir IMS como la arquitectura del núcleo de red en la cual se desarrollaran nuevos servicios aprovechando las características que tiene. IMS permite el desarrollo y despliegue de servicios de manera transparente a la tecnología de acceso, justamente una de las características que se busca en un servicio de televisión de próxima generación. Además, IMS es capaz de ofrecer prestaciones que mejoren la experiencia de los servicios que se desarrollen sobre su arquitectura gracias a que permite una fácil interacción con otros servidores de aplicación o servicios completos. Un ejemplo de esto es la integración con un servicio de presencia para la visualización de contactos que estén conectados. También es importante mencionar que las empresas de telecomunicaciones están en proceso de adopción de estas arquitecturas, lo que hace aún más relevante su utilización para el desarrollo de servicios.

Se elige aprovechar el laboratorio de tecnología LTE para la red de acceso ya que como se describió en el capítulo anterior, sus prestaciones logran solucionar diversos problemas que han tenido los servicios de televisión IP. Por un lado, con LTE se logran mejores desempeños en aplicaciones intensivas en ancho de banda, elemento característico de los servicios de televisión, en particular con la mejora en resolución del video como en películas en alta definición. Por otro lado, LTE permite obtener transmisiones con muy baja latencia, característica fundamental para mejorar la calidad de experiencia en servicios en tiempo real como lo es la televisión en vivo habitual.

Estas razones y las explicadas inicialmente en el Capítulo I, comprenden las razones por las que se eligió una red de acceso LTE e IMS como la plataforma que permitirá el despliegue de los servicios a desarrollar.

En el capítulo anterior se describió la arquitectura de las redes IMS y cómo esta permite justamente la fácil incorporación de servicios de valor agregado a través de servidores de aplicación, permitiendo así un sin número de alternativas que se pueden incorporar. En este sentido, se hace necesario delimitar algunos alcances del servicio que se pretende desarrollar considerando limitantes en tiempo y recursos disponibles para este trabajo.

En primer lugar, debido a la gran estructura que posee IMS, se decide solo utilizar las componentes estrictamente necesarias para lograr el despliegue del servicio, evitando así

complicaciones adicionales que no aportan a lo fundamental de este trabajo. En segundo lugar, se descarta que las plataformas y servicios desplegados logren poseer el desempeño en calidad de servicio o seguridad necesaria para poder utilizarse como un servicio con fines comerciales. En este sentido, solo se propone que los servicios a desarrollar sean considerados como una prueba de concepto de las capacidades propias a las redes utilizadas y una muestra de lo que es posible lograr. Sin embargo, los servicios a desarrollar sí serán aptos en un contexto docente para el desarrollo de experiencias demostrativas, o bien como punto de partida para un posible desarrollo de un servicio más robusto.

Además, se aprovecharán todas las herramientas puestas a disposición en otros trabajos, logrando hacerlas convivir adecuadamente en las plataformas y redes escogidas, destacando particularmente su puesta en marcha sobre la red de acceso LTE.

3.4.2 DESCRIPCIÓN DE PRESTACIONES

Aún considerando los alcances mencionados, para poner en marcha un servicio multimedia convergente a través de diversas redes de acceso, hacia cualquier dispositivo y que cumpla con las prestaciones que se buscan, se deben fusionar diversos mecanismos y protocolos tanto del dominio de internet como de las telecomunicaciones. Esto último evidentemente no parece ser una tarea fácil y a pesar de que las instituciones de estandarización definen el marco teórico para arquitecturas como IMS que son capaces de cumplir con estos objetivos, en la práctica éstas no se involucran activamente en el desarrollo de prototipos que puedan ayudar a verificar su robustez. Debido a esto, las pruebas de concepto que implementan dichas arquitecturas juegan un rol importante en proveer de retroalimentación en el proceso de estandarización, entregando la posibilidad a los operadores y proveedores de servicios de realizar pruebas de efectividad en escenarios reales. Sin embargo, a pesar de que existen diversas implementaciones de este tipo en el mercado, la mayoría escapan al alcance del desarrollador promedio. En este sentido, los software de código libre (*open-source*) son una alternativa viable y de bajo costo que logra ofrecer lo que se necesita para efectivamente poder involucrarse en este proceso. Como resultado, aparecen diversos desarrollos que crean ambientes de prueba (*testbeds*) que imitan el comportamiento real de arquitecturas como IMS.

A continuación se propone una serie de implementaciones de código libre para la construcción de un ambiente de prueba donde se podrá desarrollar el servicio según las prestaciones y alcances que se persiguen.

En primer lugar, para montar la arquitectura IMS se utilizará el proyecto OpenIMS Core ya bastamente utilizado en este contexto. En particular se utilizará como referencia la configuración descrita en [16]. Este proyecto será la base de los servicios a desarrollar y permitirá su uso de manera convergente en cualquier red de acceso. Además, en conjunto con clientes compatibles, también permite la interacción entre usuarios a través de mensajería instantánea, llamadas de voz o video conferencias, incluso permitiendo la incorporación de un servidor de presencia (el cual no se implementará en este trabajo). Para lograr un despliegue de contenido multimedia flexible se aprovecharán los avances propuestos en [17] que utilizan un enfoque de sesiones de transmisión de video que permiten Video-on-Demand (VoD) y Live Streaming mediante un servidor de IPTV. Este enfoque permite al usuario la selección de múltiples canales disponibles y ofrece al operador la capacidad de un sistema de tarificación en tiempo real según tiempo de uso.

Finalmente para obtener un servicio más interactivo, se utilizará una plataforma de telecomunicaciones de código libre llamada Mobicents, principalmente tomando los desarrollos propuestos en [20]. En dicho trabajo se presenta la modificación de un ejemplo contenido en el proyecto Mobicents para desarrollar una plataforma que permite que los usuarios puedan navegar en un portal web para comprar contenido de video disponible en un catálogo que incluye imágenes, descripción de la trama, puntuación y precios. Si un usuario quisiera ver una sinopsis de alguna película antes de comprarla entonces basta un click en el portal web y el sistema automáticamente genera una llamada SIP asociada a la cuenta del usuario registrado. Cuando el usuario contesta, el servidor multimedia envía un anuncio de voz que pide una confirmación de parte del usuario para que se realice un *streaming* del video a su terminal. Una vez vista la sinopsis, el usuario puede agregar el video a su carro de compras del portal web y puede continuar viendo otros videos en el catálogo online. Finalmente el usuario puede comprar los videos en su carro de compras, descontando fondos de su usuario, proceso que también puede confirmarse mediante el terminal.

Con cada uno de los servicios descritos, se pretende cumplir con los objetivos enunciados en el comienzo de este capítulo. IMS juega el rol de posibilitar la convergencia de los servicios desarrollados y habilita el control de ellos de cara al operador, permitiéndoles provisionar o bloquear los servicios a los usuarios de manera independiente. El servidor de IPTV montado sobre IMS permite la elección de contenido multimedia en horarios flexibles y en un ambiente seguro con posibilidad de cobro inteligente. Finalmente la plataforma de video convergente basada en Mobicents permite agregar valor al servicio de video entregando una interfaz gráfica que se adecúa de mejor manera a las necesidades y hábitos de los clientes de hoy en día. Por otro lado, esta plataforma entrega un beneficio clave de cara al operador, permitiéndole a este la protección al abuso asegurando que sólo los usuarios registrados accedan al servicio, un elemento clave en servicios de video alojados en servidores web. Quien sea que navegue por el portal web deberá poseer una dirección y dispositivo SIP para efectivamente poder disfrutar las prestaciones del sistema. Estas cuentas SIP pueden ser manejadas por el operador en un sistema aislado de la red, lo que lo protege de ataques de terceros.

En una primera etapa se implementará esta plataforma de video de manera independiente, mostrando que es posible lograr las prestaciones descritas anteriormente. En una segunda etapa, se aprovecha la capacidad que tiene la arquitectura de IMS de servir como plataforma convergente y que la plataforma de video opera principalmente en base al protocolo SIP para hacerlas convivir. De esta manera se dará las facilidades al operador para el suministro del servicio a través de IMS, pero se mantiene la libertad del proveedor del servicio de manejar el contenido.

Es importante remarcar que si bien los servicios recién mencionados cumplen con las prestaciones que se buscan, estos solo pretenden ser una prueba de concepto y por ende no ofrecen la calidad de un servicio real. En particular se destaca el caso de la plataforma de video que a pesar de poseer una plataforma web con interfaz gráfica, esta es bastante básica y mejorarla no es el foco de este trabajo.

3.5 FUNCIONAMIENTO Y ARQUITECTURA

3.5.1 SERVICIO DE IPTV SOBRE IMS

Como ya se mencionó en secciones anteriores, para la implementación de la arquitectura IMS se utilizará el proyecto FOKUS Open IMS Core debido a que es la única implementación robusta construida en base a proyecto de código libre. Si bien existen otros servidores de aplicación SIP con prestaciones similares, e incluso existen las herramientas para su construcción en base a proyectos como el mismo Mobicents, Open IMS Core ha logrado establecerse y es quizás el proyecto más utilizado y documentado para la construcción de ambientes de prueba de IMS.

Cabe destacar que la implementación de IMS de este proyecto solo considera las entidades del núcleo de la arquitectura; HSS, P-CSCF, I-CSCF y S-CSCF. El HSS (Home Subscriber Server) representa la base de datos que permite el registro de usuarios y la información asociada a los servicios disponibles para ellos. El P-CSCF (Proxy Call Session Control Function) es el primer punto de contacto de un usuario con la red IMS. El I-CSCF (Interrogating Call Session Control Function) decide que S-CSCF debe atender a cada usuario. Por último, el S-CSCF (Serving Call Session Control Function) es el servidor SIP que sirve los requerimientos de un usuario. En el ambiente a desarrollar, los clientes pueden provenir de distintas tecnologías de acceso, pues se pretende demostrar la capacidad convergente de IMS, sin embargo, para mantener las pruebas en un ambiente controlado, todos los clientes convivirán en el mismo dominio de área local. Con este enfoque se pueden descartar los elementos de IMS que permiten la interacción con la PSTN y los *gateways* hacia redes o elementos con funcionamiento distinto, lo que hace que los elementos de núcleo de IMS sean suficientes. Otros elementos de la arquitectura también son

excluidos ya que no son estrictamente necesarios para el funcionamiento de IMS y su desarrollo no se justifica por estar fuera del foco principal de este trabajo.

Para la implementación del servicio de IPTV básico que permita Video-on-Demand (VoD) y Live Streaming se utilizará el proyecto UCT Advanced IPTV Server. Como se describe en la sección 2.6.2, su funcionamiento básico requiere el uso de otras dos componentes: un servidor adicional con la capacidad de realizar streaming y un cliente capaz de interpretar mensajes SIP con redirección de contenido. Para el servidor multimedia se utilizará el proyecto VLC Media Player debido a que es *software* libre, es altamente configurable, está bien documentado y es capaz de transmitir video tanto en formato VoD como en formato de Streaming en vivo a través del protocolo RTSP. Para el cliente se utilizará el proyecto UCT IMS Client, pues en realidad este servidor de IPTV fue específicamente diseñado para dicho cliente, lo que permite una puesta en marcha más directa.

Para agregar la función de cobro al servicio de IPTV se utilizará el proyecto UCT IMS IPTV Charging System. Cómo se describe en la sección 2.6.3, este proyecto se compone de dos entidades (OCS y CDF), que se encargan del manejo de cobros *online* y *offline*, y además modifica el servidor de IPTV de manera de incorporar los llamados a dichas entidades según corresponda.

Finalmente, en vez de que cada componente de la plataforma posea su propio servidor de resolución de nombres de dominio (DNS - Domain Name Server), se implementará uno único para todos los componentes, lo cual facilitará la administración de direcciones IPs y permitirá analizar más fácilmente todas las consultas de DNS. Para la implementación de este servidor se utilizará el DNS BIND que está disponible por defecto en todos los sistemas operativos basados en UNIX ya que es el software de su tipo más consolidado y el de más amplia utilización en desarrollos de este tipo.

En resumen, para la plataforma de IPTV sobre IMS se utilizarán las siguientes componentes:

- Cliente IMS (UCT IMS Client)
- DNS (BIND)
- P-CSCF (FOKUS OPEN IMS Core)
- I-CSCF (FOKUS OPEN IMS Core)
- S-CSCF (FOKUS OPEN IMS Core)
- HSS (FOKUS OPEN IMS Core)
- IPTV AS (UCT Advanced IPTV Server)
- Media Server (VLC Media Player)
- CDF (UCT IMS IPTV Charging System)

- OCS (UCT IMS IPTV Charging System)

En la Figura 3.4 se muestra un diagrama de interconexiones.

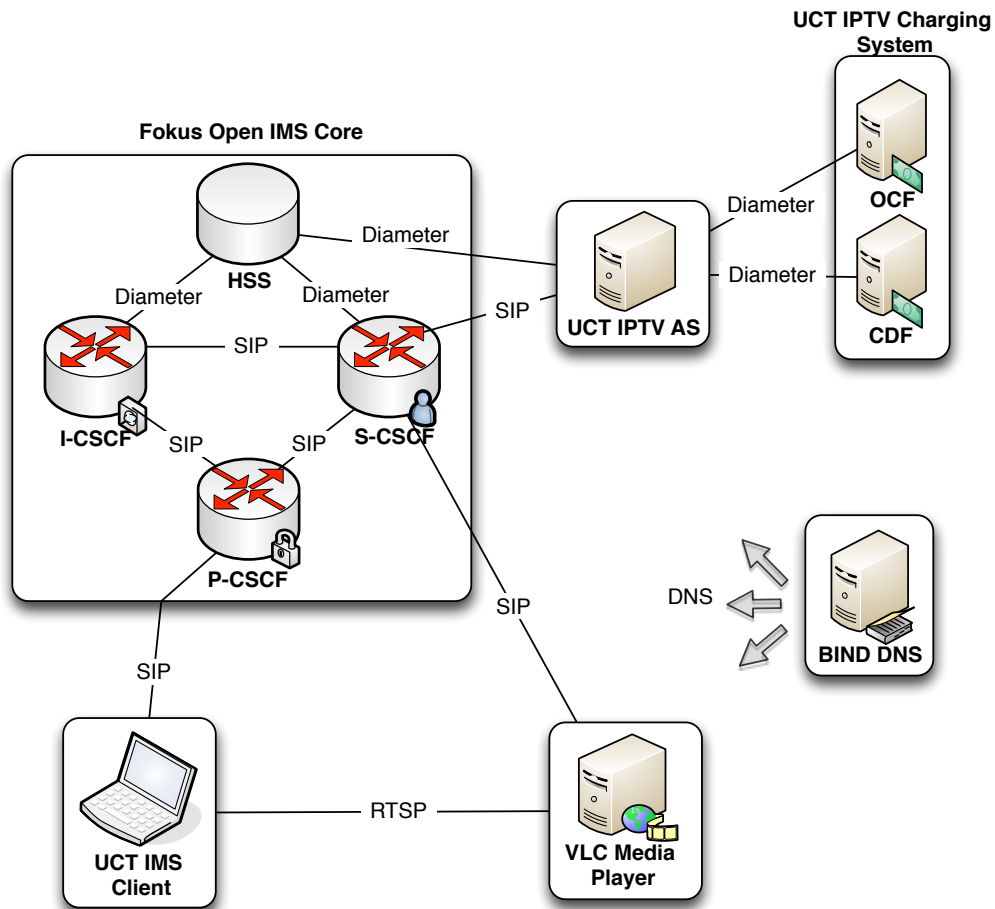


FIGURA 3.4: DIAGRAMA DE INTERCONEXIONES PARA EL SEVICIO DE IPTV SOBRE IMS

Todos los elementos están del lado del operador o proveedor de contenidos salvo el cliente UCT IMS Client.

3.5.2 SERVICIO VIDEO SHOPPING

El servicio o plataforma de *Video Shopping* pretende cumplir el objetivo de ofrecer contenido multimedia de manera interactiva, con una interfaz gráfica amigable y que se adecúe de mejor manera a las necesidades y hábitos de los usuarios de hoy en día, pero que también entregue facilidades desde el punto de vista del operador. Como ya se mencionó, se utiliza la plataforma de telecomunicaciones de código libre Mobicents. Según se describe en 2.6.5, Mobicents utiliza una serie de sub proyectos que entregan distintas prestaciones y facilidades para el desarrollo de servicios en redes convergentes. En este trabajo se utiliza como plataforma base el ambiente de

ejecución Mobicents JAIN-SLEE (descrito en 2.6.6), sobre el cuál opera el servidor multimedia Mobicents Media Server (descrito en 2.6.7). Los demás elementos de Mobicents se descartan, pues a pesar de que ofrecen características interesantes, estas no aportan significativamente a la prueba de concepto que se pretende desarrollar en este trabajo.

El servidor multimedia Mobicents Media Server contiene diversos ejemplos con el propósito de asistir en la comprensión de su funcionamiento. Uno de estos ejemplos es el *Mobicents Converged Demo* [36] el cual será modificado para el desarrollo de la plataforma de video shopping (en versiones más recientes de Mobicents, este ejemplo también está disponible como parte del proyecto Mobicents SIP Servlets). En [37] se puede encontrar un manual de usuario donde se explica su funcionamiento y código.

El ejemplo original pretende demostrar el uso de los *Resource Adaptors* (RA) de SIP y de texto a voz (TTS – *Text to Speech*) además del *framework* para aplicaciones web JBoss Seam [38]. A través de Jboss Seam se implementa un sistema que le permite a los usuarios la creación de cuentas a través de un portal web. Cada cuenta mapea a un usuario con una dirección SIP. Cuando los usuarios se autentifican en el portal, el sistema automáticamente carga en memoria su dirección SIP para su uso posterior. Una vez registrado, el usuario puede navegar a través de un catálogo de muebles que puede ir agregando a su canasta de compras. El usuario puede ver, agregar o eliminar los productos en su canasta y revisar su monto total resultante en cualquier momento.

Cuando un usuario decide terminar su compra, puede generar la orden de sus productos seleccionados, lo que automáticamente genera una llamada SIP a su terminal. Cuando el usuario contesta la llamada, el motor TTS reproduce un anuncio donde se le pide al usuario digitar con su terminal la fecha de envío para la cual quiere que se le envíen los productos. Una vez que esto se haya logrado, se da fin a la llamada y la orden se envía a la sección de e-commerce del sistema, la cual contiene la información de todas las ordenes que aún no se procesan. Este proceso es visible para el usuario, quien puede monitorear el estado de su orden en la pestaña “*My Orders*” del portal web.

Por otro lado, el ejemplo requiere de un usuario en calidad de administrador, quien debe confirmar cada orden antes de que se genere la llamada al terminal SIP, sin embargo este y otros detalles del sistema no son relevantes para el objetivo que se quiere lograr.

Este ejemplo muestra como Mobicents puede utilizarse para crear una aplicación para un servicio interactivo que logra combinar SIP, http y prestaciones multimedia. Es particularmente interesante como la interacción entre Mobicents y JBoss Seam logra generar llamadas SIP a través de la interacción con un portal web.

La idea principal, es aprovechar la funcionalidad básica de este ejemplo para permitir que los usuarios registrados puedan navegar a través de un catálogo de películas en vez de muebles y lograr las características descritas en 3.4.2.

Es importante destacar que al realizar estos cambios se mantiene la característica principal de la aplicación que es demostrar que a través de Mobicents se puede lograr la interacción entre un

terminal SIP y un portal web. Sin embargo, los videos generan un nuevo enfoque aún más interesante, pues ahora el servicio se puede entender no sólo como una plataforma de e-commerce; sino desde una perspectiva de IPTV enriquecido que ofrece la posibilidad de navegar en un portal web para la elección de contenido desde la comodidad de un computador personal pero que su visualización propiamente tal sea finalmente en otro terminal SIP (por ejemplo una Televisión con un cliente SIP implementado). Más aún, debido a que el consumo de servicios de video o televisión es algo mucho más habitual en los clientes que la compra de muebles, este nuevo enfoque puede ser el primer paso para el desarrollo de una plataforma comunitaria donde se permita la creación de listas de contactos, interacción entre usuarios, evaluación de películas, y un sinnúmero de otras alternativas, todas dentro de la misma plataforma. Esto se puede lograr justamente gracias a que los servicios en Mobicents se desarrollan en base a componentes reutilizables pensados para interactuar entre ellos. Por otro lado, debido a que el servicio opera principalmente en base al protocolo SIP, es posible hacerlo interactuar con la arquitectura IMS en la capa de control. Esto le permite al operador seleccionar a qué clientes provisionar con el servicio.

3.5.2.1 MODIFICACIONES AL EJEMPLO ORIGINAL

De una manera muy similar a como el proyecto Fokus OpenIMS Core utiliza el *framework* Hibernate [39] en la entidad FHoSS (implementación del HSS) para el mapeo de objetos y clases a tablas relacionales de la estructura SQL, este sistema utiliza HSQLDB (Hyper SQL Database) [40]. HSQLDB es un sistema para el manejo bases de datos relacionales basado en SQL escrito completamente en Java. Es utilizado por el servicio para construir la lista de películas y su información a desplegar en el portal web y los usuarios con su información respectiva. Los valores se ingresan en el código utilizando comandos SQL tradicionales contenidos en un archivo llamado *import.sql*. Cuando la plataforma del servicio se levanta, el archivo se ejecuta automáticamente y los valores se cargan en la base de datos temporal del servidor. En el ejemplo original, este archivo contenía algunos usuarios en calidad de administrador y otros catalogados como clientes. Además estaba la información de los productos, que en este caso eran muebles.

Nuevos usuarios se agregan a la base de datos USERS usando un comando de la siguiente manera:

```

INSERT INTO USERS
(USERID,DTYPE,FIRSTNAME,LASTNAME,ADDRESS1,ADDRESS2,CITY,STATE,ZIP,EMAIL,PHONE
,CREDITCARDTYPE,CC_NUM,CC_MONTH,CC_YEAR,USERNAME,PASSWORD)
VALUES (3,'customer','Sebastian','Rivas','4 Privet Drive','Cupboard under the
Stairs' , 'QSDPAGD' , 'SD' , 24101, 'h.potter@hogwarts.edu' ,
'sip:srivas@mobicents.uchile.cl:5070' , 1 , '1979279217775911' , 03,2012 ,
'srivas' , 'srivas')

```

Por otro lado, los videos se almacenan en una base datos llamada PRODUCTS y se pueden agregar de la siguiente manera.

```

INSERT INTO PRODUCTS (PROD_ID, ASIN, TITLE, PRICE, IMAGE_URL, TRAILER_URL,
DESCRIPTION)
VALUES ('1', '11fj-v0siZL_AA160_', 'Burn after reading', 99.99,
'/img/burn_after_reading_poster.jpg',
'rtsp://media.mobicents.uchile.cl:554/channel1', 'Osbourne Cox, a Balkan
expert, is fired at the CIA, so he begins a memoir. His wife wants a divorce
and expects her lover, Harry, a philandering State Department marshal, to
leave his wife. A diskette falls out of a gym bag at a Georgetown fitness
center. Two employees there try to turn it into cash: Linda, who wants money
for elective surgery, and Chad, an amiable goof. Information on the disc
leads them to Osbourne who rejects their sales pitch; then they visit the
Russian embassy. To sweeten the pot, they decide they need more of Osbournes
secrets. Meanwhile,Lindas boss likes her, and Harrys wife leaves for a book
tour. All roads lead to Osbournes house. ');

```

Es importante destacar que el comando contiene un campo llamado “TRAILER_URL”. Este valor indica la dirección RTSP donde se encuentra el servidor de streaming que contiene el video.

La siguiente modificación tiene que ver con el SLEE propiamente tal y la manera en que responde a eventos del portal web (recordar que SLEE es un ambiente de ejecución orientado a eventos). El contenedor SLEE está capacitado para manejar eventos que provienen de fuera del entorno SLEE a través de RAs (Resource Adapters). Por ejemplo el SIP RA otorga al SLEE la capacidad de interpretar mensajes SIP provenientes de un cliente SIP. El RA adapta estos mensajes para generar eventos que luego pueden ser ruteados apropiadamente por el enrutador de

eventos del SLEE hacia un bloque que pueda consumirlo. No obstante, el SLEE no solo puede manejar eventos que provengan de entidades externas, sino que también es capaz de manejar eventos definidos por el desarrollador. Este servicio define bloques o SBBs (Service building blocks) específicamente diseñados para ser capaces de reaccionar a estos eventos personalizados.

Según se describe en [37], en total son cinco los SBB que se definen en el ejemplo original. *CallControlSbb*, es un bloque que se encarga de la señalización. Establece la conexión entre el terminal de usuario y el Mobicents Media server y le solicita a este último la reproducción de mensajes de voz una vez conectada la llamada. *AdminSbb* genera la solicitud al administrador para que acepte o rechace las ordenes de determinado usuario. *UserSbb* es el encargado de realizar la llamada al cliente correspondiente cuando este haga click en el portal web. Evidentemente depende de *CallControlSbb*. *OrderDeliveryDateSbb* realiza la solicitud al cliente para que digite la fecha que le parezca conveniente para el envío de sus productos. También depende de *CallControlSbb*. Finalmente *OrderShipDateSbb* realiza una llamada al usuario para recordarle su fecha de envío, el cual también depende de *CallControlSbb*.

En el caso de video algunos de estos SBB no son necesarios y también se deben realizar algunas modificaciones para que actúen de manera adecuada. Por ejemplo no se quiere que exista la llamada que pregunta por la fecha de envío. Esto se logra definiendo nuevos eventos que ahora serán interpretados de distinta manera. Por ejemplo, cuando un usuario hace click en un link en el portal web para reproducir un tráiler, el Framework Seam ahora generará un nuevo evento llamado *OrderTrailer*, el cual le ordenará al *Usersbb* tomar una acción distinta.

Los eventos personalizados se definen en un archivo XML y en una clase evento (archivo Java). Cada evento tiene un nombre, un proveedor y una versión que identifican al evento únicamente. De esta manera el SBB que los interpreta sabe a qué clase llamar para ejecutar la acción correspondiente. El archivo que contiene los identificadores de los eventos se llama *event-jar.xml* y a continuación se muestra cómo se definió el evento *OrderTrailer*.

```
<event event-direction="Receive" initial-event="True">
  <event-name>OrderTrailer</event-name>
  <event-type-ref>
    <event-type-name>
      org.mobicents.slee.service.dvddemo.ORDER_TRAILER
    </event-type-name>
    <event-type-vendor> org.mobicents</event-type-vendor>
    <event-type-version>1.0</event-type-version>
  </event-type-ref>
  <initial-event-select variable="ActivityContext" />
</event>
```


Por otro lado, se debe crear una sección dentro de UserSBB para poder manejar este evento. También se utiliza la API multimedia para interactuar con el RA TTS (Text-to-Speech) para generar un anuncio que le solicite al usuario aceptar la recepción del video. La Figura 3.5 muestra un diagrama simplificado del sistema y qué rutas toman los mensajes SIP como eventos de SLEE y los eventos personalizados que genera SEAM.

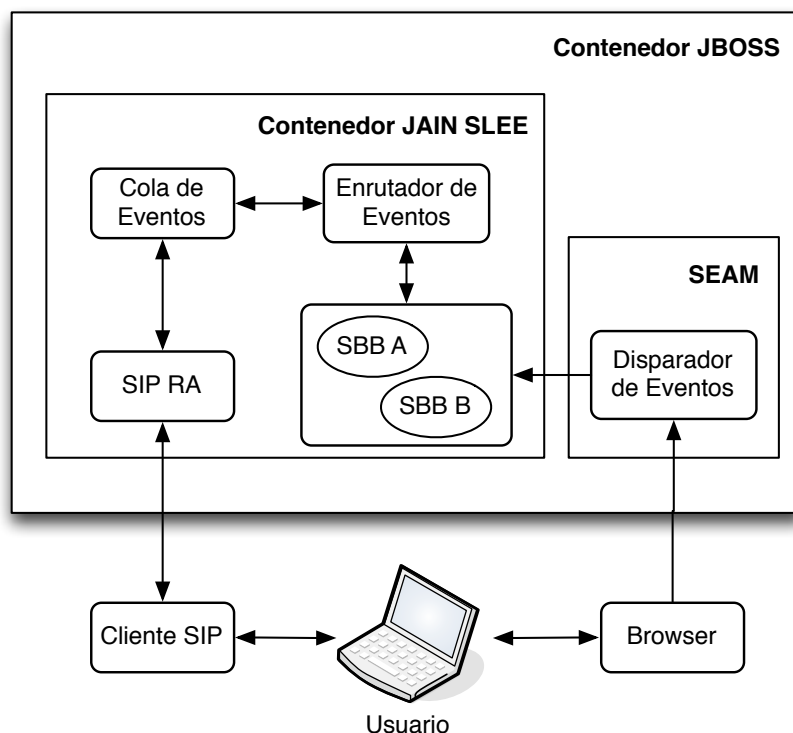


FIGURA 3.5: DIGRAMA DE FUNCIONAMIENTO DEL SERVICIO DE VIDEO SHOPPING

Una vez que el usuario recibe la llamada, debe confirmar que quiere ver el tráiler presionando el botón para llamar en su cliente SIP. Esto genera lo que se conoce como un mensaje SIP de tipo reINVITE. Básicamente es un mensaje INVITE pero dentro de una sesión donde ya existe conexión, lo que comúnmente se utiliza en el contexto de SIP para cambiar las propiedades multimedia de la sesión. No todos los clientes SIP disponibles soportan esta función, por lo que esto pasa a ser un requerimiento del cliente para su utilización con este servicio.

Al igual que el caso del UCT Advanced IPTV Server, este sistema utiliza el mecanismo de la IETF de redirección de contenido RFC 4483 (*Content Indirection*) [25]. Es decir, una vez que el mensaje reINVITE haya sido recibido en el servidor de aplicación, éste responde con un mensaje

SIP de tipo OK que contiene la dirección RTSP del servidor de streaming donde se contiene el video solicitado. El cliente IMS debe ser capaz de extraer la dirección de ese mensaje y automáticamente iniciar una sesión RTSP con el servidor de streaming.

El ejemplo original no contiene dicho mecanismo y se debe agregar que el último mensaje OK que envía el *UserSBB*, contenga la siguiente información:

```
ContentType: message/external-body; access-type="URL"; expiration="Sat,
01 January 2011 09:00:00 GMT"; URL="rtsp://media.mobicents.uchile.cl:554/
channel11"
```

Este mecanismo requiera la incorporación de servidor de streaming con compatible con RTSP. Al igual que en el servicio de IPTV, se utilizará el proyecto VLC Media Player. De esta manera incluso se podría tener un único servidor con contenido dedicado a cada servicio.

Más allá de las modificaciones en el ejemplo original, el funcionamiento descrito impone tres restricciones sobre el cliente SIP; la capacidad de realizar SIP reINVITES, tener compatibilidad con el mecanismo de redirección de contenido y tener soporte para el protocolo RTSP para efectivamente lograr que ocurra el streaming. Si bien existen diversos clientes SIP disponibles como proyectos open source o de distribución gratuita (Monster, Mercurio IMS Client, Xlite, iDoubs), son muy pocos los clientes que cumplen todos los requisitos. JAIN-SIP-PHONE [41] es uno de ellos y para el caso de conectar esta plataforma a la arquitectura IMS, entonces también el cliente UCT IMS Client pasa a estar disponible.

Por último, también se incorpora un servidor de resolución de nombres de dominio (DNS - *Domain Name Server*) de manera de permitir que cada entidad pueda tener un dominio distinto. Para la implementación de este servidor se utilizará el DNS BIND que está disponible por defecto en todos los sistemas operativos basados en UNIX.

3.5.2.2 ARQUITECTURA

Según lo descrito anteriormente, para la plataforma de video shopping sin la incorporación de IMS se utilizarán las siguientes componentes:

- Cliente SIP (JAIN SIP Phone)
- DNS (BIND)
- JBoss AS
- JBoss SEAM Web Framework (Incluido en JBoss AS)

- Mobicents JAIN SLEE
- Mobicents Media Server

La Figura 3.6 muestra un diagrama de interconexiones.

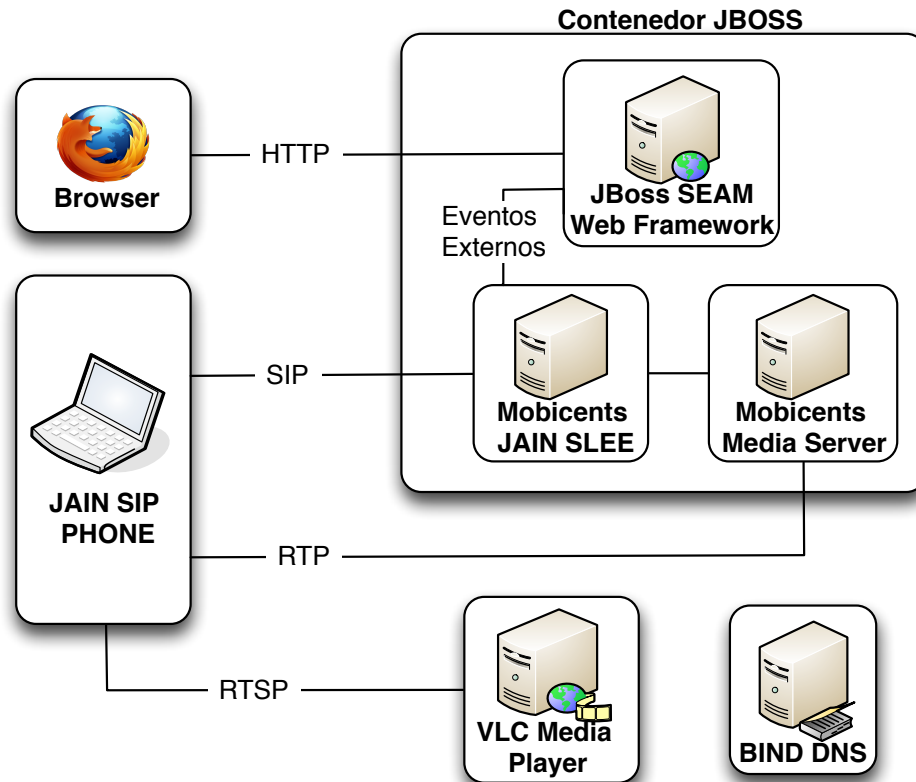


FIGURA 3.6: DIAGRAMA DE INTERCONEXIONES SERVICIO VIDEO SHOPPING

Si ahora se considera la inclusión de la arquitectura IMS en la capa de control, entonces se utilizan las siguientes componentes:

- Cliente IMS (UCT IMS Client)
- DNS (BIND)
- JBoss AS
- JBoss SEAM Web Framework (Incluido en JBoss AS)
- Mobicents JAIN SLEE
- Mobicents Media Server
- P-CSCF (FOKUS OPEN IMS Core)
- I-CSCF (FOKUS OPEN IMS Core)
- S-CSCF (FOKUS OPEN IMS Core)
- HSS (FOKUS OPEN IMS Core)

La Figura 3.7 muestra el nuevo diagrama de interconexión considerando estas componentes.

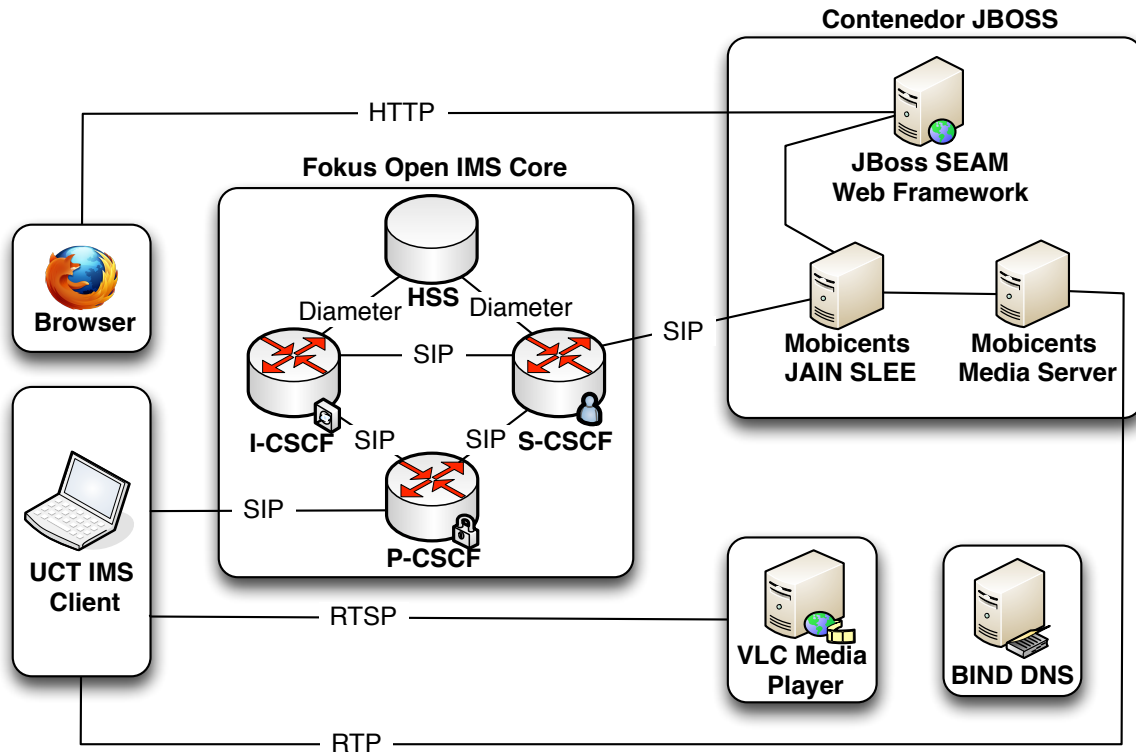


FIGURA 3.7: DIAGRAMA DE INTERCONEXIONES SERVICIO VIDEO SHOPPING

3.6 IMPLEMENTACIÓN DE LOS SERVICIOS

La implementación de los Servicios antes descritos requiere la instalación, configuración y puesta en marcha de todas las componentes descritas durante el diseño y además hacerlas compatibles con la red del laboratorio de LTE. Debido a que se pretende que este trabajo pueda ser replicado con el fin de realizar experiencias demostrativas o bien para continuar el proceso constructivo de desarrollos sobre redes convergentes, se separa la implementación de cada servicio por etapas. Para el servicio de IPTV en primer lugar se levanta el Núcleo y Cliente IMS, luego se levanta el sistema de IPTV y finalmente el sistema de tarificación. En el caso del servicio de Video Shopping en primer lugar se levanta la plataforma Mobicents y el cliente SIP y luego se incorpora el IMS ya levantado en el servicio anterior. Al finalizar todas estas etapas se habrá logrado levantar los servicios descritos en la sección anterior en una red de área local (LAN –

Local Area Network) donde cada componente podrá tener su dominio e IP arbitraria. El siguiente paso es entonces poner en marcha los servicios utilizando la conexión móvil de la red LTE.

El hecho de que cada componente pueda tener su propia IP permite además la separación de estas en distintos computadores, lo que es bastante natural debido a que justamente se busca replicar un servicio donde el usuario y los proveedores de servicio no tiene acceso directo a las componentes de red. Para facilitar este trabajo se decide realizar toda la implementación en una máquina virtual ubicada en un mismo computador para realizar las pruebas y luego basta clonarla e instalarla en más computadores cuantas veces sea necesario para separar componentes sin la necesidad de volver a poner en marcha cada componente.

La implementación se lleva a cabo en un computador con procesador Intel Core 2 Duo de 2.66 Ghz y memoria RAM de 4GB. El desarrollo se realiza sobre el sistema operativo GNU/Linux, y la distribución escogida es Ubuntu 8.04 (Hardy Heron). Se escoge la utilización de Linux debido a que la mayoría de los proyectos que se utilizan fueron diseñados para este sistema operativo. Por otro lado, se escoge dicha distribución, en desmedro de otras más recientes, con el propósito de que cada máquina virtual sea algo más liviana y así lograr ejecutar varias en simultáneo sin problemas. El software de virtualización escogido es VirtualBox debido a que es de distribución gratuita, contiene interfaces amigables y permite una fácil portabilidad de la máquina virtual hacia cualquier computador.

Todo el proceso de implementación detallado se incluye en los Anexos. Lo que se describe a continuación es un resumen de los aspectos más relevantes de la implementación

3.7 IMPLEMENTACIÓN SERVICIO DE IPTV

El primer paso en la implementación del servicio de IPTV es la puesta en marcha de los componentes de IMS. Como se describe en 3.5.1, para este propósito se utiliza el proyecto Fokus Open IMS Core, el cual provee de las componentes P-CSCF, I-CSCF, S-CSCF y HSS. Este proyecto permite la configuración de un dominio común para todas las entidades y sub dominios dentro de dicho dominio para cada componente. El proyecto contiene un script que justamente realiza la labor de configuración de dominios de manera automática. Por defecto el sistema define el dominio open-ims.test, donde cada subdominio es pcscf.open-ims.test, icscf.open-ims.test, scscf.open-ims.test, hss.open-ims.tes. Para agregar personalización al servicio, se cambio el dominio principal a ims.uchile.cl, manteniendo las estructura de los subdominios.

El proyecto Open IMS Core también requiere la instalación de un sistema de base de datos y un servidor DNS. Para la base de datos se utilizará MySQL-Server 5.0 [42] debido a que es un proyecto de código libre y es fácilmente accesible mediante el gestor de paquetes de Linux APT (Advanced Packet Tool) desde la línea de comandos. Para el servidor DNS, como se menciona en 3.5.1, se utilizará BIND9.

Por otro lado, también por defecto, el proyecto Open IMS Core provee de dos usuarios pre configurados en la base de datos HSS; Alice y Bob. A ambos clientes se les asocia direcciones sip que pertenecen al dominio común del sistema, por ende sus direcciones sip públicas son de la forma “sip:alice@ims.uchile.cl” y “sip:bob@ims.uchile.cl”. Sin embargo, este proyecto implementa nativamente una plataforma web basada en Tomcat [43] que permite modificar los datos en el HSS. De esta manera se agregan usuarios nuevos como por ejemplo “sip:srivas@ims.uchile.cl”.

Una vez instalados los componentes del núcleo de IMS, se debe levantar un cliente IMS que permita registrarse en el sistema y solicitar canales del futuro servicio de IPTV. De todas maneras es importante mencionar que para las pruebas a realizar en este momento, cualquier cliente compatible con la arquitectura IMS bastaría. Se verá en las siguientes secciones que este punto pasará a ser relevante en el caso de la interconexión con el laboratorio de LTE. El cliente IMS escogido es el proyecto UCT IMS Client.

Una vez puesto en marcha todos los elementos recién mencionados, es posible registrarse en IMS y se logra la interacción entre clientes mediante sesiones paralelas del cliente UCT IMS Client. Es posible el envío de mensajes de texto, llamadas de voz y videollamadas. Esta será la base para gran parte de los desarrollos expuestos a continuación.

La siguiente tabla contiene todos los elementos que componen el núcleo IMS que se utilizará en el resto de la implementación junto con el Cliente IMS y el servidor de resolución de nombres de dominio. Contiene además nombres de dominio y sus puertos respectivos. La asignación de las direcciones IP a cada dominio se deja pendiente, pues se pretende compatibilizar esta red con los dominios de la red del laboratorio de LTE, tema que se revisará en las próximas secciones.

Elemento	Nombre del Dominio	Puerto
P-CSCF	pcscf.ims.uchile.cl	4060
I-CSCF	icscf.ims.uchile.cl	5060
S-CSCF	sccsf.ims.uchile.cl	6060
HSS	hss.ims.uchile.cl	3868, 3869, 3870, 8080
DNS	ns.ims.uchile.cl	-
Cliente	ue.ims.uchile.cl PUI: "sip:srivas@ims.uchile.cl"	5061

TABLA 3.1: ELEMENTOS DE IMS Y SUS NOMBRES DE DOMINIO

Continuando con la implementación del diseño descrito en 3.5.1, se debe poner en marcha el servidor de aplicación de IPTV, UCT Advanced IPTV Server y el servidor de streaming multimedia, VLC Media Player. Como en la estructura de IMS los clientes no se relacionan directamente con los servidores de aplicación, es necesario agregar el servidor de IPTV como servicio disponible en los registros del HSS. De esta manera, cuando el cliente solicite contenido del servidor, IMS podrá interceptar la llamada y redirigirla correctamente al servidor de IPTV. Para lograr esto se utiliza la interfaz web del HSS para crear un *Application Server* el cual debe contener un nombre, dirección SIP y puerto. Posteriormente se crea un *Trigger Point* que realiza la función de interceptar los mensajes SIP de tipo INVITE dirigidos hacia el servidor de aplicación de IPTV. Finalmente se debe agregar un iFC (*Initial Filter Criteria*) que guarda el registro de los usuarios que tienen provisión del servicio. Siguiendo la estructura hasta el momento, se asigna el subdominio `iptv.ims.uchile.cl` a este servidor. Todos los detalles de este proceso se describen en los Anexos.

En cuanto a la configuración del IPTV AS propiamente tal, se debe incorporar la tabla que mapea nombres de canales solicitados con las direcciones RTSP del servidor de *streaming*. Se definen dos canales con el propósito de poder hacer pruebas en formato VoD y en *live Streaming*.

El siguiente componente a agregar es el servidor de *streaming* multimedia. En su configuración solo se requiere que la dirección RTSP en la cual se levantará coincida con las direcciones que entrega el IPTV AS. Se asigna el subdominio `media.ims.uchile.cl`.

Tras haber incorporado a la red los dos elementos ya mencionados, además de las acciones hasta el punto anterior, ahora los usuarios pueden solicitar canales del servicio IPTV y visualizarlos tanto en modo VoD como *live streaming*. Además el usuario puede acceder a la guía de programación previamente configurada por el proveedor del servicio.

La Tabla 3.2 muestra los nuevos elementos incorporados al sistema.

Elemento	Nombre del Dominio	Puerto
IPTV AS	iptv.ims.uchile.cl	8010
Media Server	media.ims.uchile.cl	5554

TABLA 3.2: ELEMENTOS DEL SISTEMA DE IPTV Y SUS NOMBRES DE DOMINIO

El último paso en la implementación del servicio de IPTV es la incorporación del sistema de tarificación, UCT IMS IPTV Charging System. Como se detalla en la sección 2.6.3, este proyecto considera la incorporación de dos nuevos elementos; CDF (*Charging Data Function*) y OCF (*Online Charging Function*) además de ciertas modificaciones al IPTV AS (principalmente para la incorporación del CTF - *Charging Trigger Function*, que se encarga de enviar las notificaciones de cobro a las nuevos elementos). Debido a la modificación del IPTV AS, en el proyecto se incluye el nuevo código fuente modificado, lo que en este sentido impone la necesidad de volver a levantar el nuevo servidor en desmedro del anterior. Se opta por instalar el anterior primero pues genera un punto intermedio de testeo si se quisiera replicar esta implementación.

Luego de haber incorporado estos últimos elementos, el sistema de IPTV es capaz de cobrar por el servicio tanto de manera *offline* como *online*. En particular se utiliza el cobro *online*, lo que se traduce en que cada usuario tiene asignada una cierta cantidad de fondos, los cuales se van descontando en tiempo real por cada segundo que se consuma el servicio. Si eventualmente el usuario quedara sin fondos, entonces el sistema desconecta la transmisión automáticamente. Como el proyecto que se utiliza para este desarrollo aún se encuentra en una versión beta, la cantidad de fondos debe ser configurada en el código fuente antes de ejecutar el sistema.

La Tabla 3.3 muestra los dominios elegidos para estas dos nuevas componentes (El IPTV AS mantendrá el dominio anterior).

Elemento	Nombre del Dominio	Puerto
CDF	cdf.ims.uchile.cl	3892
OCS	ocs.ims.uchile.cl	3891

TABLA 3.3: ELEMENTOS DEL SISTEMA DE IPTV Y SUS NOMBRES DE DOMINIO

3.8 IMPLEMENTACIÓN DEL SERVICIO DE VIDEO SHOPPING

Para la implementación del servicio de Video Shopping, también se procede por etapas. En primer lugar se levantará la plataforma Mobicents y el cliente SIP y luego se incorporará el IMS ya levantado en el servicio anterior. Todo el proceso expuesto a continuación se describe en detalle en los Anexos.

Según se describe en la sección 3.5.2, con respecto al proyecto Mobicents, en este trabajo sólo se utiliza el ambiente de ejecución Mobicents JAIN-SLEE y el servidor multimedia Mobicents Media Server. Ambos proyectos pueden ser descargados en conjunto en un paquete que contiene además el servidor de aplicación JBoss. Es importante destacar que no se utilizó la versión más reciente de estos proyectos, pues gran parte de los desarrollos que se exponen a continuación se basan en avances previos propuestos en [20]. Las plataformas de Mobicents han sido actualizadas desde la publicación de dicho trabajo y su utilización requeriría de aún más adaptaciones al código publicado. Si bien este esfuerzo adicional si aportaría para mantener al día el servicio según las versiones de Mobicents, el resultado no mejora significativamente las características del servicio final. En este sentido, y considerando que el servicio es solo una prueba de concepto sin pretensiones de transformarse en un servicio comercial, se opta por mantener las versiones anteriores de la plataforma de Mobicents que se utilizan en [20], las cuales corresponden a la versión 1.0.2.GA en el caso del servidor multimedia y a la versión 4.2.3.GA en el caso del servidor de aplicación Jboss.

El primer paso para la implementación de este servicio es la puesta en marcha del servidor Jboss con los servidores Mobicents JAIN-SLEE y el servidor multimedia Mobicents Media Server. Existe una alternativa de descarga que contiene los binarios precargados, por lo que este paso es relativamente directo. En este sentido, la única modificación por el momento será cambiar el dominio sobre el cual se levanta. Se elige el dominio `mobicents.uchile.cl`, el cual no pertenece al subdominio de la plataforma de IMS. Esto se hizo justamente con el propósito demostrar que los proveedores de servicios pueden tener los servicios alojados en sus propias redes y basta que exista interconexión con la red del operador de red (IMS en este caso) para que el servicio pueda ser accedido por los usuarios.

Como se describe en 3.5.2, el servidor de aplicación JBoss sobre el cual se levantan las plataformas de Mobicents incorpora un *framework* para aplicaciones web. Por defecto este mecanismo se utiliza para levantar un portal web llamado *Management-Console*, donde se puede monitorear y cambiar el estado actual del SLEE. Además, se podrá navegar a través de los ejemplos demostrativos contenidos tanto en el servidor SLEE como en el Media Server. Uno de estos ejemplos es el *converged-demo* [36].

El siguiente paso es preparar el servidor SLEE para la instalación del servicio. Como el servicio está basado en el ejemplo *converged-demo*, se debe cumplir con todos los requerimientos para su puesta en marcha. Este ejemplo necesita la instalación de los RA (*Resource Adapter*) de SIP y de

TTS (*Text-to-Speech*). Se puede verificar la correcta instalación de estos RA en el portal web de monitoreo *Management-Console*.

Una vez que se haya cumplido con los requerimientos de instalación del ejemplo, se debe reemplazar su código fuente por aquel que incorpora las modificaciones pertinentes descritas en 3.5.2.1. Este código fuente se pone a libre disposición en [44] bajo una licencia GPL y existe una discusión abierta de su funcionamiento en la comunidad de Mobicents. A pesar de que existe algo de documentación disponible, es necesario hacer una serie de cambios adicionales para lograr su correcto funcionamiento.

La primera modificación al código fuente tiene que ver con las dirección SIP de los usuarios y las direcciones RTSP donde se ubican los videos. Esto se realiza en el archivo `import.sql` según se describe en 3.5.2.1. Se define un usuario con la dirección SIP en el dominio donde se levanta la plataforma de Mobicents (por ejemplo `sip:srivas@mobicents.uchile.cl`). En cuanto a la ubicación de los videos, se les asigna direcciones RSTP en el subdominio `media.mobicents.uchile.cl`.

Luego se modifica la dirección SIP que utiliza el sistema para establecer la llamada con los clientes cuando estos solicitan la visualización de una sinopsis. Se asigna la dirección `sip:shoppingdemo@shopping.mobicents.uchile.cl`. Es importante remarcar que esta dirección no depende del dominio donde se levante el servidor pues es solo un nombre dentro del contexto SIP que se define para que los clientes puedan establecer una llamada directa con el servidor.

Una vez realizadas las modificaciones anteriores, es necesario compilar las fuentes y luego instalarlas en el servidor SLEE para que se carguen cada vez que este se levante. Para compilar las fuentes, se utiliza Maven [45], una herramienta de gestión y construcción de software típicamente escritos en lenguaje Java. Maven utiliza un archivo XML para describir el proyecto de *software* a construir, sus dependencias de otros módulos y componentes externos, y el orden de construcción de los elementos. Es necesario modificar algunas de las direcciones de las componentes externas que utiliza este ejemplo para que efectivamente pueda compilar (revisar Anexos).

Una vez realizada la compilación e instalación, debería ser posible acceder al portal web del servicio en la dirección web `http://mobicents.uchile.cl:8080/ShoppingDemo`, donde además es posible registrarse con los usuarios que se crearon en los pasos anteriores.

Continuando con la puesta en marcha del servicio, se debe levantar un cliente SIP capaz de manejar el protocolo RTSP y que sea compatible con el mecanismo de redirección de contenido. Como se ha explicado anteriormente, para este propósito se utilizará el proyecto JAIN SIP

APPLET PHONE, el cual implementa un cliente SIP que cumple con ambas características. Adicionalmente en [20], se incorporan algunas modificaciones a dicho proyecto para poder realizar algunas pruebas de diagnóstico antes de hacerlo interactuar con la plataforma Mobicents. El cliente modificado también está puesto a libre disposición en [44]. No obstante, algunas de las modificaciones que se realizaron tenían el propósito de hacerlo compatible con el servidor multimedia Darwin, y como la plataforma que se busca montar en este trabajo no utiliza dicho servidor, se deben revertir. Este proceso se describe en detalle en los Anexos. Por otro lado, se debe configurar el cliente con una dirección SIP accesible por el servidor SLEE y que además esté configurada en la base de datos de usuarios. Se utiliza “sip:srivas@mobicents.uchile.cl”. También debe ser configurado el dominio al cual se registrará, en este caso se configura con la dirección del servidor de Mobicents, mobicents.uchile.cl.

Finalmente se debe levantar el servidor de *streaming* multimedia. Al igual que en el caso del servicio de IPTV, se utiliza el proyecto VLC Media Player. En su configuración solo se requiere que la dirección RTSP en la cual se levantará coincida con las direcciones que entrega Mobicents cuando el usuario solicita en el portal web la visualización de una sinopsis. Por esta razón se asigna el subdominio media.mobicents.uchile.cl.

La creación de los nuevos dominios utilizados requiere la configuración del DNS para que estas sean mapeadas a IPs válidas. Al igual que en el servicio de IPTV, la asignación de las direcciones IP a cada dominio se deja pendiente, pues se pretende compatibilizar esta red con los dominios de la red del laboratorio de LTE, tema que se revisará en las próximas secciones.

A este punto, es posible registrarse en el portal web del servicio y navegar por el catálogo de películas. También es posible registrar el cliente SIP que se asocia al usuario del portal web. Con el cliente JAIN SIP APPLETT PHONE, a través de la herramienta de diagnóstico, se pueden solicitar los videos directamente al servidor multimedia VLC. Si esto no tiene problemas, entonces también es posible recibir los videos mediante la solicitud en el portal web.

La Tabla 3.4 contiene todos los elementos que componen el servicio de Video Shopping sin la utilización del núcleo IMS. Se pueden observar los nombres de dominio y sus puertos respectivos.

Elemento	Nombre del Dominio	Puerto
JBoss AS	mobicents.uchile.cl	5060, 8080
Media Server	media.mobicents.uchile.cl	554
DNS	ns.mobicents.uchile.cl	-
Cliente	“sip:srivas@mobicents.uchile.cl”	5070

TABLA 3.4: ELEMENTOS DEL SERVICIO DE VIDEO SHOPPING

Una vez que el servicio de Video Shopping esté operativo se procede a su incorporación en la arquitectura IMS. Como se muestra en 3.5.2.2, dentro de la arquitectura, se dispone el servidor Jboss como un servidor de aplicación accesible por IMS que representa un servicio que puede prestar. Los usuarios no interactúan directamente con el servicio, sino que hacen un requerimiento a IMS, y este redirige las llamadas correspondientes.

Este nuevo enfoque requiere de diversas modificaciones. En primer lugar se debe agregar el servicio a los registros del HSS tal como se hizo con el servidor de IPTV en el servicio anterior. De esta manera, cuando el usuario solicite la visualización de una sinopsis, el servidor SLEE envía un mensaje SIP que puede ser interceptado por IMS para poder redirigirlo correctamente al cliente. Lo mismo ocurre cuando el cliente realice el SIP reINVITE para confirmar la recepción del video y que se le envíe la dirección RTSP. Para lograr esto se utiliza la interfaz web del HSS para crear un *Application Server* el cual debe contener un nombre, dirección SIP y puerto. Posteriormente se crea un *Trigger Point* que realiza la función de interceptar los mensajes SIP de tipo INVITE entre cliente y servidor. Finalmente se debe agregar un iFC (*Initial Filter Criteria*) que guarda el registro de los usuarios que tienen provisión del servicio.

La segunda modificación a realizar es la creación de nuevos usuarios en la plataforma Mobicents que contengan una dirección SIP propia al dominio de IMS. Se crea un nuevo usuario con la dirección “sip:srivas@ims.uchile.cl” aprovechando que ya existe dicho usuario en el HSS de IMS.

Finalmente, se debe cambiar el puerto en el que se levanta el servidor web que permite realizar modificaciones en el HSS, pues por defecto utiliza exactamente el mismo valor que el portal web del servicio de Video Shopping. Se elige el puerto 9080 (Esta elección es arbitraria y cualquier otro valor podría utilizarse siempre y cuando no interfiera con algún otro componente).

Con estas modificaciones es posible acceder al servicio de Video Shopping con las mismas prestaciones descritas anteriormente, pero sin necesidad de registrarse en el dominio de

Mobicents. Ahora, mediante el uso usuarios propios a IMS es posible recibir las llamadas y videos del servicio de Video Shopping. Esto impone la necesidad de volver a utilizar un cliente capaz de registrarse en IMS, por lo que se utiliza el proyecto UCT IMS Client.

La Tabla 3.5 contiene todos los elementos que componen el servicio de Video Shopping funcionando en conjunto con la arquitectura IMS. Se pueden observar los nombres de dominio y sus puertos respectivos.

Elemento	Nombre del Dominio	Puerto
JBoss AS	mobicents.uchile.cl	5060, 8080
Media Server	media.mobicents.uchile.cl	554
P-CSCF	pcscf.ims.uchile.cl	4060
I-CSCF	icscf.ims.uchile.cl	5060
S-CSCF	scscf.ims.uchile.cl	6060
HSS	hss.ims.uchile.cl	3868, 3869, 3870, 9080
DNS	ns.ims.uchile.cl	-
Cliente	ue.ims.uchile.cl PUI: "sip:srivas@ims.uchile.cl"	5061

TABLA 3.5: ELEMENTOS DEL SERVICIO DE VIDEO SHOPPING A TRAVÉS DE IMS

3.9 INTERCONEXIÓN CON LABORATORIO LTE

En la sección 3.3 se describe la arquitectura y elementos disponibles en el laboratorio de LTE ubicado en la FCFM de la Universidad de Chile. Estos elementos permiten la conexión móvil a la red LTE mediante terminales USB (ZTE MF820) que se conectan a computadores personales. Por otro lado también es posible conectarse mediante cable de red a través del switch principal al cual están conectados los elementos del núcleo de red EPC. Esta red es operada por personal de la empresa ZTE y mayores detalles de su puesta en marcha y configuración no son el tema central de este trabajo. Sin embargo, es importante destacar que como se muestra en la Figura 3.3, los elementos de la red LTE están interconectados según el dominio de red 10.2.100.0 y que además los computadores conectados con terminales USB reciben una IP automáticamente en el dominio 70.70.2.0 de acuerdo al protocolo DHCP (*Dynamic Host Configuration Protocol*).

Considerando lo anterior, se decide que tanto el núcleo IMS como los servicios desarrollados se conectarán directamente a la red a través de un cable de Ethernet hacia el switch principal. El cliente será el único elemento que se conectará de forma móvil mediante los terminales USB. De

cierta manera esto no escapa tanto de las implementaciones de la vida real, pues es de esperar que los operadores tengan en su núcleo de red los elementos de IMS y algunos servicios.

A cada servicio se le asocia una dirección IP distinta, considerando que no necesariamente pertenecen al mismo proveedor. De esta manera los dominios y direcciones IP de todos los componentes se definen según se presenta en la Tabla 3.6.

Elemento	Nombre del Dominio	Puerto	Dirección IP
P-CSCF	pcscf.ims.uchile.cl	4060	10.2.100.80
I-CSCF	icscf.ims.uchile.cl	5060	10.2.100.80
S-CSCF	scscf.ims.uchile.cl	6060	10.2.100.80
HSS	hss.ims.uchile.cl	3868, 3869, 3870, 9080	10.2.100.80
DNS	ns.ims.uchile.cl	-	10.2.100.80
IPTV AS	iptv.ims.uchile.cl	8010	10.2.100.81
Media Server (IPTV)	media.ims.uchile.cl	5554	10.2.100.81
CDF	cdf.ims.uchile.cl	3892	10.2.100.81
OCS	ocs.ims.uchile.cl	3891	10.2.100.81
JBoss AS	mobicents.uchile.cl	5060, 8080	10.2.100.82
Media Server (Video Shopping)	media.mobicents.uchile.cl	554	10.2.100.82
Cliente SIP	“sip:srivas@mobicents.uchile.cl”	5070	70.70.2.5
Cliente	ue.ims.uchile.cl PUI: “sip:srivas@ims.uchile.cl”	5061	70.70.2.5

TABLA 3.6: DOMINIOS, PUERTOS, Y DIRECCIONES IP DE TODOS LOS ELEMENTOS QUE CONFORMAN LOS SERVICIOS

Las Figuras 3.8, 3.9 y 3.10 muestran los diagramas de interconexión de componentes de acuerdo a las direcciones IP que se definieron. Es importante destacar que no se muestran todos los elementos de la red LTE, pues el detalle completo se muestra en la Figura 3.3. Cada conjunto enmarcado corresponde a un único computador donde se levantan las componentes que se muestran en su interior. Por esta razón, solo se muestra un cable de red desde cada conjunto de elementos que compone algún determinado servicio.

El laboratorio de LTE tiene antenas instaladas tanto en el Departamento de Ingeniería Eléctrica como en el edificio CEC para asegurar cobertura en casi la totalidad de las dependencias de la FCFM. Sin embargo, para realizar las pruebas y de manera de tener fácil acceso a las

componentes que se conectan físicamente al núcleo de red, se prefiere utilizar el eNodo B instalado en el mismo Laboratorio que contiene una antena indoor de baja potencia. Esto se refleja en los diagramas que se muestran a continuación.

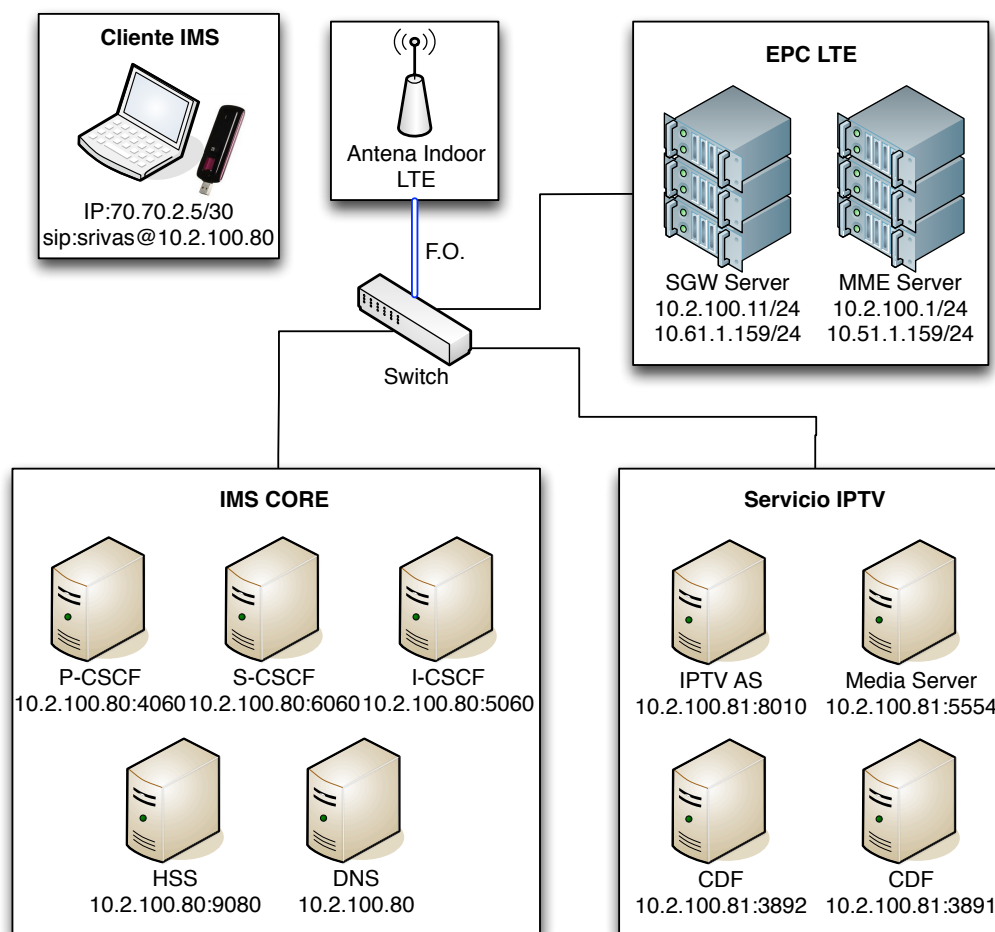


FIGURA 3.8: DIAGRAMA DE INTERCONEXIÓN RED LTE, NÚCLEO IMS Y COMPONENTES SERVICIO DE IPTV

La Figura 3.8 muestra la interconexión de las componentes del servicio de IPTV sobre IMS en la red LTE. En este servicio se debe levantar el núcleo IMS y los cuatro elementos que componen el servicio de IPTV propiamente tal. Como se explicó anteriormente, se asigna distintas IPs a estos dos bloques de elementos con la intención de representar a un “operador de red” (propietario del núcleo IMS) que tiene instalada su arquitectura en un entorno que no depende de las instalaciones del “proveedor de servicios” (propietario de los elementos del servicio IPTV).

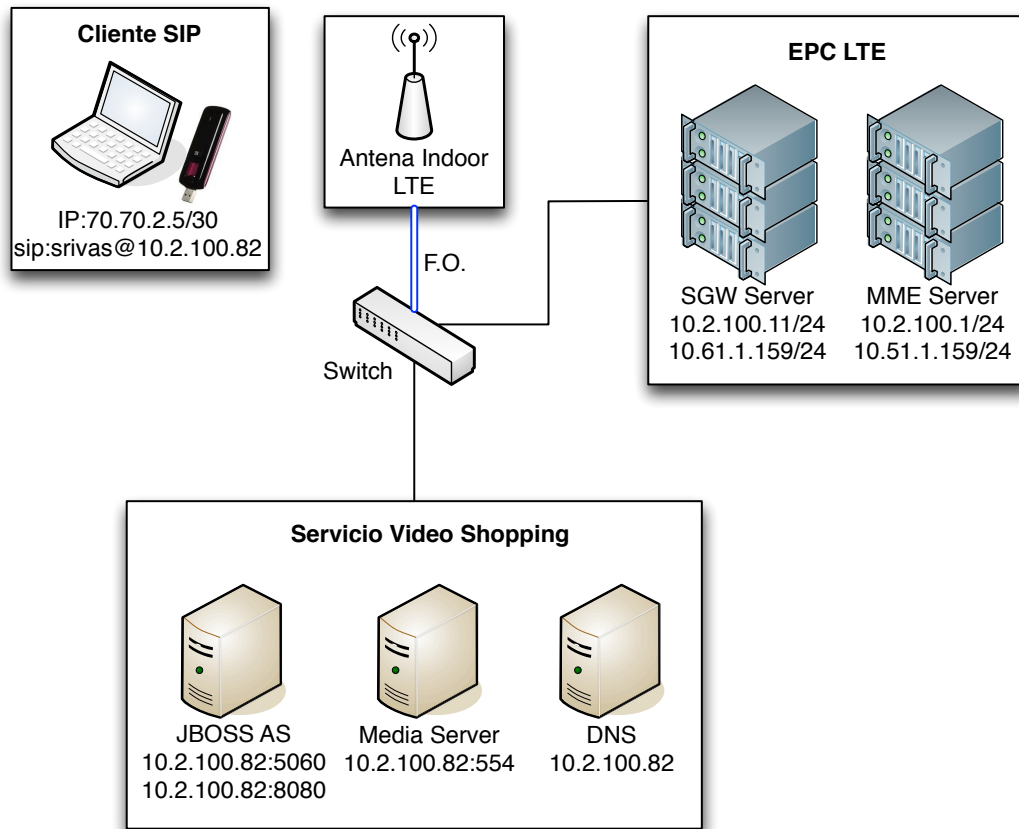


FIGURA 3.9: DIAGRAMA DE INTERCONEXIÓN RED LTE Y COMPONENTES SERVICIO DE VIDEO SHOPPING

La Figura 3.9 muestra la interconexión de las componentes del servicio de Video Shopping en la red LTE. Como en este caso no se considera la interacción de los elementos de IMS, solo se utiliza un computador para la implementación de todas las componentes de este servicio. Se puede observar que en esta implementación se utiliza un cliente SIP en desmedro de uno asociado a IMS.

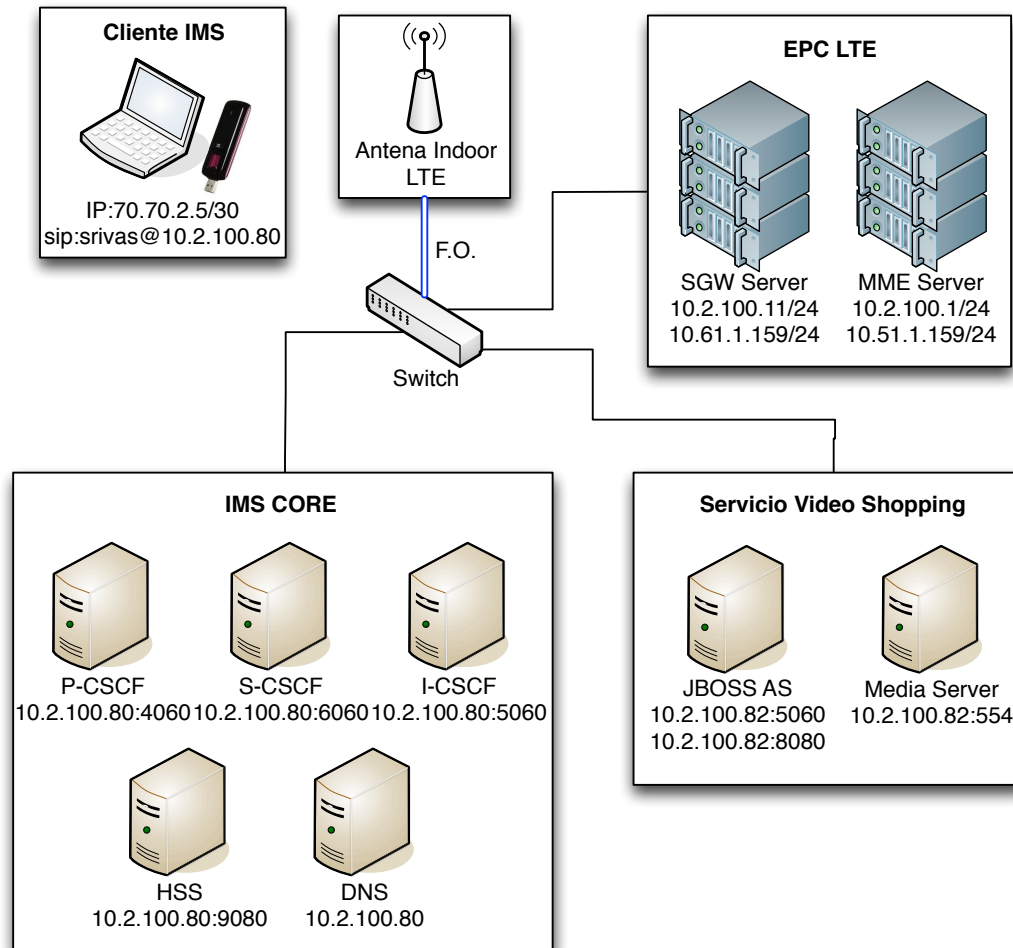


FIGURA 3.10: DIAGRAMA DE INTERCONEXIÓN RED LTE Y COMPONENTES SERVICIO DE VIDEO SHOPPING

La Figura 3.10 muestra la interconexión de las componentes del servicio de Video Shopping en la red LTE pero esta vez utilizando los elementos del núcleo de IMS. Se puede observar que nuevamente se han separado estos elementos en dos IPs distintas con el propósito de que no sean instalados en las mismas dependencias. Por otro lado, también se puede observar que nuevamente se utiliza el cliente IMS para poder realizar el registro en el HSS.

Es importante mencionar que es necesario que la red LTE tenga habilitado el modo P2P (*Peer-to-Peer*), es decir que se permita que los clientes interactúen entre sí sin necesidad de redirigir sus mensajes a través de un servidor intermedio. Esto es particularmente relevante en el caso de los servicios de llamados de voz y video entre clientes, pues en este caso IMS solo aporta con la señalización inicial, pero la conversación propiamente tal ocurre directamente entre clientes mediante el protocolo RTP. Los operadores de red en ciertos casos prefieren bloquear dicha modalidad ya que prefieren tener control sobre el tráfico y servicios que existan entre clientes.

Finalmente debido a algunos problemas con los terminales disponibles y su incompatibilidad con el sistema operativo Linux, algunas de las pruebas se debieron realizar con clientes fijos

conectados directamente al switch del núcleo EPC. Se propone como alternativa de solución la utilización del paquete de Linux USB_ModeSwitch [46], el cual da soporte para que el sistema operativo Linux reconozca diversos dongles de internet móvil, dentro de los cuales específicamente se encuentra el modelo utilizado.

En la siguiente sección se presentan algunos resultados considerando al cliente conectado directamente en el core EPC utilizando la IP 10.2.100.83.

4 CAPÍTULO IV: RESULTADOS

En este capítulo se exhiben los resultados del diseño e implementación que se muestran en el capítulo anterior. Se muestra el funcionamiento de los servicios mediante diagramas de flujo de protocolos y se describen sus características mostrando imágenes del servicio en acción.

Considerando que durante el diseño de la arquitectura de los servicios se muestran dos distintos enfoques (IPTV y Video Shopping), en este capítulo se mantiene dicha estructura, dividiendo los resultados en dos secciones.

En general se puede observar que tanto el servicio de IPTV como el servicio de Video Shopping se comportan tal como se espera. A través de las componentes de IMS y su cliente compatible, se logra realizar llamadas de voz y video y se permite mensajería instantánea entre clientes. El servidor de IPTV efectivamente es capaz de entregar contenido tanto en modalidad VoD como *Live Streaming*. Se logra desplegar el portal Web del servicio de Video Shopping y consigue realizar los cambios pertinentes para agregar clientes o películas. El servicio además realiza las llamadas adecuadamente tanto al dominio de IMS como a clientes SIP independientes de esta arquitectura.

Por otro lado también se comprueba que ambos servicios no son capaces de otorgar la seguridad necesaria para evitar que usuarios no provisionados accedan al servicio.

Finalmente se hacen algunos comentarios en relación al manual detallado de instalación y puesta en marcha de los elementos, el cual es un resultado fundamental de este trabajo considerando que el laboratorio de LTE disponible en la FCFM genera un espacio de desarrollo constante en esta línea de investigación.

4.1 RESULTADOS DE LA IMPLEMENTACIÓN DEL SERVICIO DE IPTV

4.1.1 CARACTERÍSTICAS GENERALES

Como se describe en secciones anteriores, la implementación del servicio de IPTV contempla la puesta en marcha de IMS y de un cliente compatible, además de los servidores específicos a IPTV. IMS por si solo implementa diversas prestaciones que cumplen algunos de los alcances

propuestos. Con el cliente compatible es posible realizar llamadas de voz, video y mensajería instantánea. La Figura 4.1 muestra la interfaz gráfica del cliente que se utiliza.

Se puede ver que el cliente incorpora una ventana donde se puede monitorear el flujo de mensajes SIP entre IMS y el cliente. Si bien esto no es lo que se busca en un servicio de cara al usuario final, en el contexto de desarrollo, esta característica aporta significativamente en la depuración de errores y en la comprensión de ciertos mecanismos.

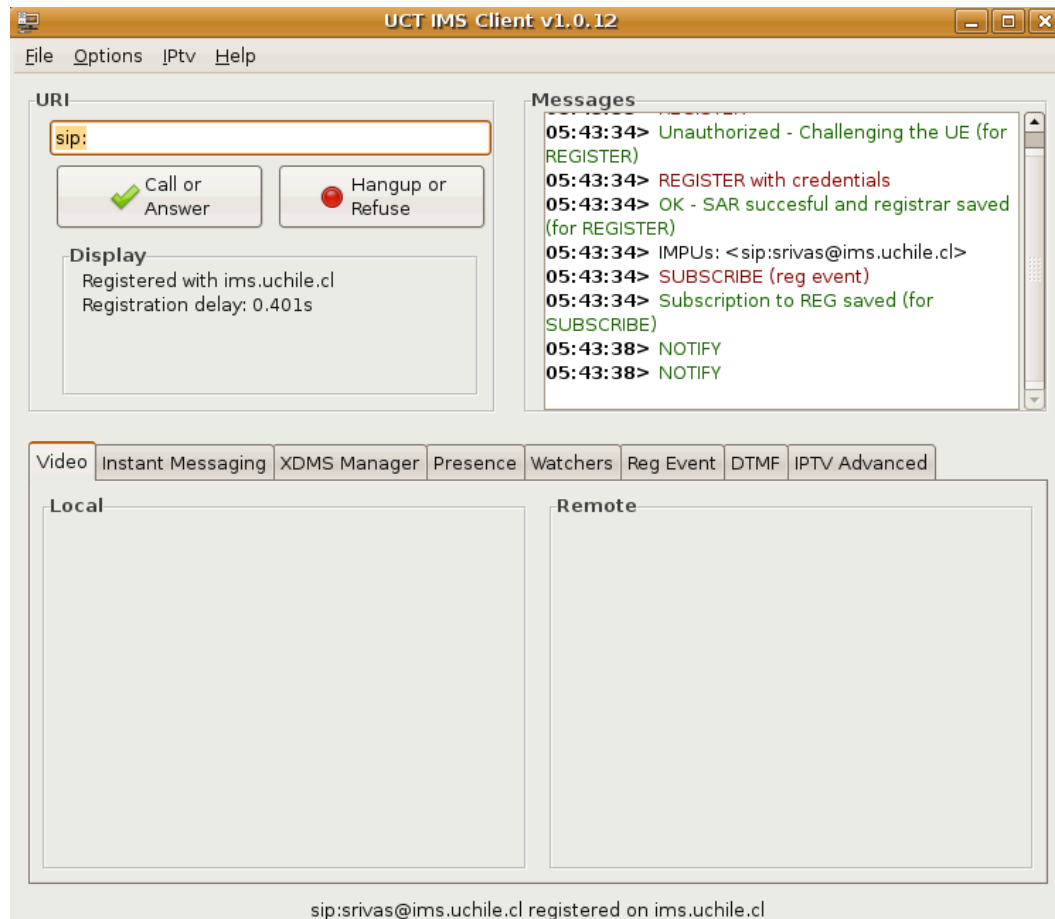


FIGURA 4.1: INTERFAZ GRÁFICA CLIENTE IMS UTILIZADO

En cuanto a la puesta en marcha de los elementos específicamente diseñados para el servicio de IPTV, efectivamente se logra su adecuada configuración, logrando que el cliente pueda acceder a los servicios de VoD y *Live Streaming*. Para solicitar contenido multimedia se debe hacer una llamada SIP de tipo INVITE con el siguiente formato: “sip:channel@iptv.ims.uchile.cl”, donde “channel” corresponde al nombre del canal al cual se quiere acceder. IMS se encarga de redirigir este mensaje al servidor de aplicación de IPTV, y este retorna la dirección RTSP a la cual el cliente debe contactar para recibir el contenido multimedia del canal solicitado. Todo este proceso viene implementado por defecto en el cliente utilizado y basta escribir el número del canal que se desea ver en la viñeta “IPTV Advanced” que se observa en la Figura 4.1.

La Figura 4.2 muestra un ejemplo de cómo se ve el servicio de VoD a través de la interfaz de video del cliente utilizado.

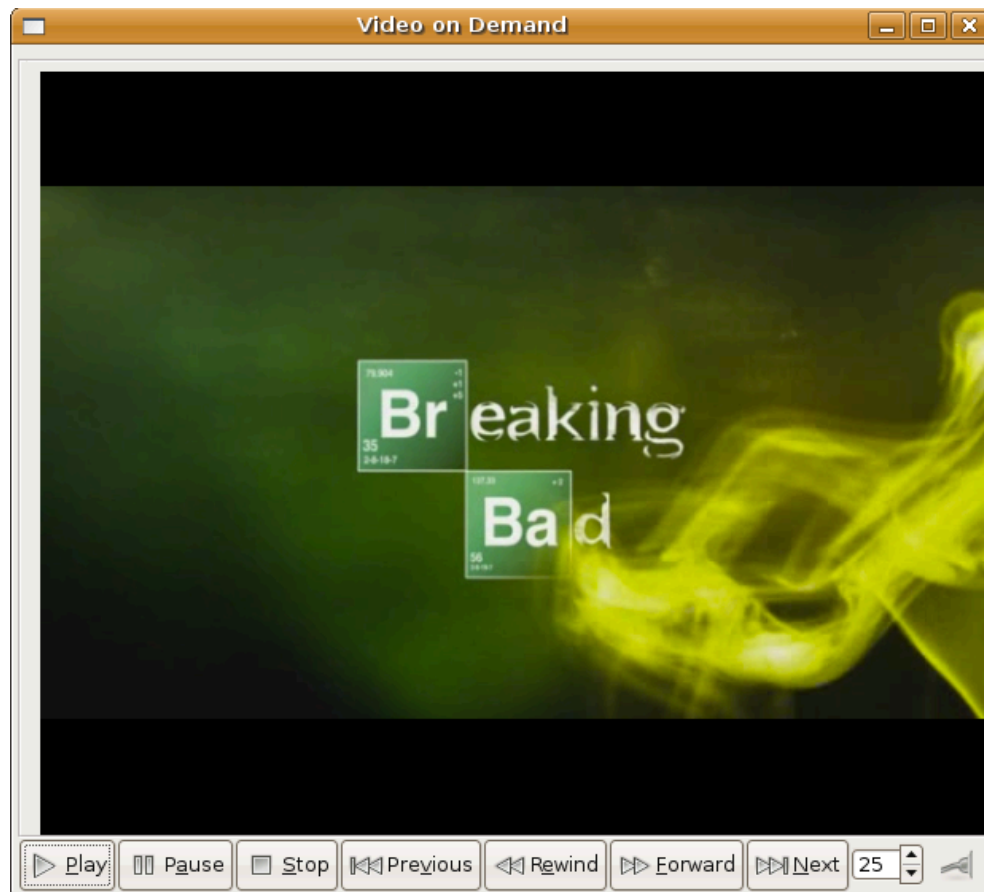


FIGURA 4.2: EJEMPLO DE VIDEO-ON-DEMAND EN EL SERVICIO DE IPTV

Como se puede ver en la imagen anterior, el cliente incorpora botones para detener, pausar, adelantar y retroceder la reproducción. Estos botones gatillan el envío de mensajes RTSP hacia el servidor multimedia para cambiar las características de la sesión de transmisión de datos. Sin embargo, según los elementos que se levantan solo es posible detener la reproducción, pues los demás requieren de un servidor de almacenamiento de datos.

En las pruebas en modalidad *Live Streaming*, se logra observar que el video no comienza su reproducción al momento de su solicitud (como en el caso de VoD), sino que esta comienza al momento de levantar el servidor multimedia. En este sentido, cuando se solicita un canal en modalidad *Live Streaming*, se verá el video desde un punto intermedio. Esto es justamente lo que ocurre en los servicios de televisión por cable tradicionales.

En esta modalidad también se puede observar que en los primeros segundos de reproducción, la imagen está bastante distorsionada, mostrando manchas de color verde que lentamente van desapareciendo a medida que avanza la reproducción. Esto es un problema no menor en el caso

de querer implementar un servicio de IPTV con pretensiones comerciales, pues esto ocurre cada vez que el usuario cambie de canal. Esto empeora notablemente la calidad de experiencia del servicio pues los clientes deben esperar varios segundos para poder ver una imagen aceptable.

La Figura 4.3 muestra el fenómeno que se describe.



FIGURA 4.3: INICIO DE TRANSMISIÓN EN MODALIDAD LIVE STREAMING EN SERVICIO DE IPTV

También se puede destacar que el cliente no hace distinción en la interfaz entre la modalidad de VoD y la de *Live Streaming*. Por esta razón se titula VoD la ventana de video a pesar de que no lo sea.

El estudio de la calidad de servicio y experiencia en servicios desarrollados sobre IMS se aborda con mayor detalle en [18], trabajo que se realizó en paralelo a este.

El cliente además incorpora una guía de programación electrónica que muestra un detalle del contenido por canales de ambas modalidades de transmisión. Estas listas se cargan en base a archivos XML contenidos dentro de la carpeta del cliente. De todas maneras es importante

destacar que para lograr que estas listas efectivamente se actualicen automáticamente, sería necesario desarrollar un sistema que envíe la información pertinente y reemplace los valores en estos archivos XML.

La Figura 4.4 muestra la interfaz gráfica de la guía de programación electrónica

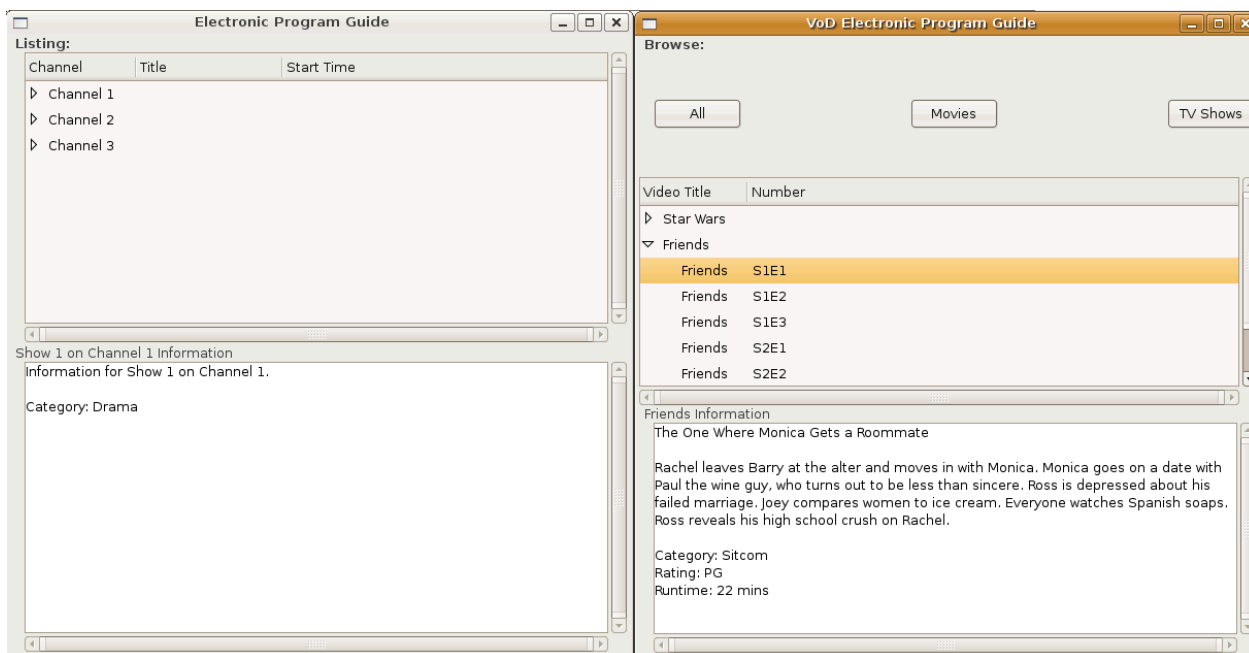


FIGURA 4.4: ELECTRONIC PROGRAM GUIDE EN EL SERVICIO DE IPTV

Respecto al sistema de tarificación, se tiene que es un sistema exclusivo de cobro del servicio de IPTV y no de los otros servicios IMS que son ofrecidos dentro del Laboratorio de IPTV, por lo tanto no es posible tarificar los servicios de llamadas de videoconferencias o de VoIP, esto debido a que solo el IPTV AS contiene un Charging Trigger Function para estos fines. Como ya se ha comentado, la plataforma posee un OCS encargado de la tarificación online que opera bajo la lógica de sistema de prepago y un CDF encargado de la tarificación offline que permite realizar cobros de postpago del servicio. El proyecto de software libre desde el cual se basa la implementación de estos componentes está en una fase Beta, por lo cual es una solución muy precaria en cuanto a la posibilidad de configurar y sacar provecho sin necesidad de modificar sus códigos fuentes. Considerando todo esto, ambos módulos fueron ocupados en la medida de lo posible. Respecto a los resultados obtenidos con el sistema de tarificación Online se pudo distinguir que para la tarificación se utiliza un modelo de cobro conocido como *Unit Reservation* [12], esto significa que el IPTV AS solicita de forma periódica al OCS la asignación de créditos durante el uso del servicio. Si el usuario detiene manualmente la transmisión del canal de IPTV, el sistema de tarificación se detiene. En caso de que se le acaben los créditos al usuario, el sistema de tarificación pide el corte inmediato del servicio.

No existen archivos XML para la configuración tarifaria, toda esta información se encuentra contenida en los códigos fuentes del programa, por lo que cualquier cambio implica la nueva compilación del OCS. Por defecto se tiene que el número de créditos inicial es de 200, la tarificación es por tiempo de uso del servicio y el precio fijado es de un crédito por segundo. Al estudiar el código fuente de los elementos involucrados en la tarificación, se encontró que están las funciones asociadas a las otras modalidades de tarificación, que son el caso de Direct Debiting [4] que corresponde a un cobro único por uso del servicio y el caso de Unit Reservation según flujo de datos transmitido, con lo que se podría tarificar el MB consumido en vez del tiempo de uso de los servicios, sin embargo la utilización de estos dos tipos de métodos significaría un esfuerzo mayor de coordinación con el IPTV AS o el PCEF en el caso del cobro por flujo de datos. Al hacer pruebas con el sistema de tarificación offline se corroboró el envío del tiempo de uso del servicio con una periodicidad de 30 segundos desde el IPTV AS al CDF, sin embargo el CDF no almacena ni crea ningún Charging Data Record [12] con el cual se pueda facturar a los usuarios posteriormente, por lo que la información de uso del servicio se pierde.

Es importante mencionar que este servicio carece de algunos aspectos de seguridad. Si bien IMS permite restringir el uso del servicio mediante la configuración de la provisión en el HSS. El servidor multimedia no realiza ningún tipo de verificación con IMS en relación a qué cliente es quien solicita el video. Esto provoca que un determinado cliente no provisionado puede eventualmente solicitar el envío del video directamente al servidor multimedia si consigue la dirección RTSP correspondiente. Esto se puede ver en los diagramas de funcionamiento que se muestran en la siguiente sección.

4.1.2 FUNCIONAMIENTO

En esta sección se muestran capturas de tráfico de partes esenciales del servicio. Para esto se utiliza el software de código libre Wireshark [47] , ya ampliamente utilizado en este contexto.

Se utiliza la arquitectura de conexión que se muestra en Figura 3.8 la cual considera la selección de direcciones IPs según se muestra en la Tabla 3.6.

En esta arquitectura se considera una única dirección IP para toda la estructura de IMS y otra para toda la estructura de IPTV. En este sentido, los siguientes diagramas no muestran mensajes entre elementos de dichas estructuras, como los que ocurren por ejemplo entre las CSCFs.

El primer proceso relevante en este servicio, es el registro del cliente en IMS. La interacción entre IMS y cliente se muestra en la Figura 4.5.

Time	10.2.100.83	10.2.100.80	Comment
5.417	Request: REGISTER s (5060)	(4060)	SIP: Request: REGISTER sip:ims.uchile.cl
5.436	Status: 401 Unautho (5060)	(4060)	SIP: Status: 401 Unauthorized - Challenging the UE (0 bindings)
5.541	Request: REGISTER s (5060)	(4060)	SIP: Request: REGISTER sip:ims.uchile.cl
5.578	Status: 200 OK - SA (5060)	(4060)	SIP: Status: 200 OK - SAR successful and registrar saved (1 bindings)
5.741	Request: SUBSCRIBE (5060)	(4060)	SIP: Request: SUBSCRIBE sip:rivas@ims.uchile.cl
5.743	Status: 200 Subscri (5060)	(4060)	SIP: Status: 200 Subscription to REG saved
7.057	Request: NOTIFY sip (5060)	(4060)	SIP: Request: NOTIFY sip:rivas@10.2.100.83:5060
7.057	Status: 200 OK (5060)	(4060)	SIP: Status: 200 OK

FIGURA 4.5: MENSAJES SIP EN REGISTRO DE CLIENTE EN IMS

En primer lugar se puede observar que el cliente IMS falla en el primer intento de registro. Esto ocurre debido a que IMS exige el uso de credenciales. El cliente por defecto siempre envía un primer mensaje de registro sin el uso de estas credenciales.

Por otro lado se puede observar que el registro propiamente tal ocurre en solo una fracción de segundo, lo que es muy positivo considerando que estas plataformas en la práctica suelen atender a una cantidad de clientes del orden de los cientos de miles.

El segundo proceso relevante en el servicio de IPTV es la solicitud de transmisión de contenido multimedia. La Figura 4.6 muestra un diagrama de flujo de mensajes SIP de este proceso.

Se puede observar que cuando el cliente envía un mensaje SIP de tipo INVITE según el formato “sip:channel@iptv.ims.uchile.cl”, esta es interceptada por IMS y luego reenviada adecuadamente al servidor de aplicación de IPTV. Después de algunas notificaciones de que efectivamente se está reenviando el mensaje, el Servidor de IPTV responde con un mensaje OK que contiene la dirección RTSP a la cual el cliente debe solicitar el contenido. Este mensaje nuevamente es interceptado por IMS y redirigido correspondientemente. El cliente a través del mecanismo de redirección de contenido extrae la dirección RTSP y solicita el canal directamente al servidor de streaming (el protocolo RTSP no figura en este flujo). Esto último genera la transmisión de video mediante el protocolo UDP (desde 10.2.100.81 hacia 10.2.100.83), el cual también ha sido extraído del análisis pues al ser enviado en pequeños paquetes, entorpece el estudio de los demás mensajes. Finalmente cuando el cliente decide terminar la transmisión envía un mensaje SIP de tipo BYE, el cual es redirigido al servidor de IPTV según el mismo método anterior.

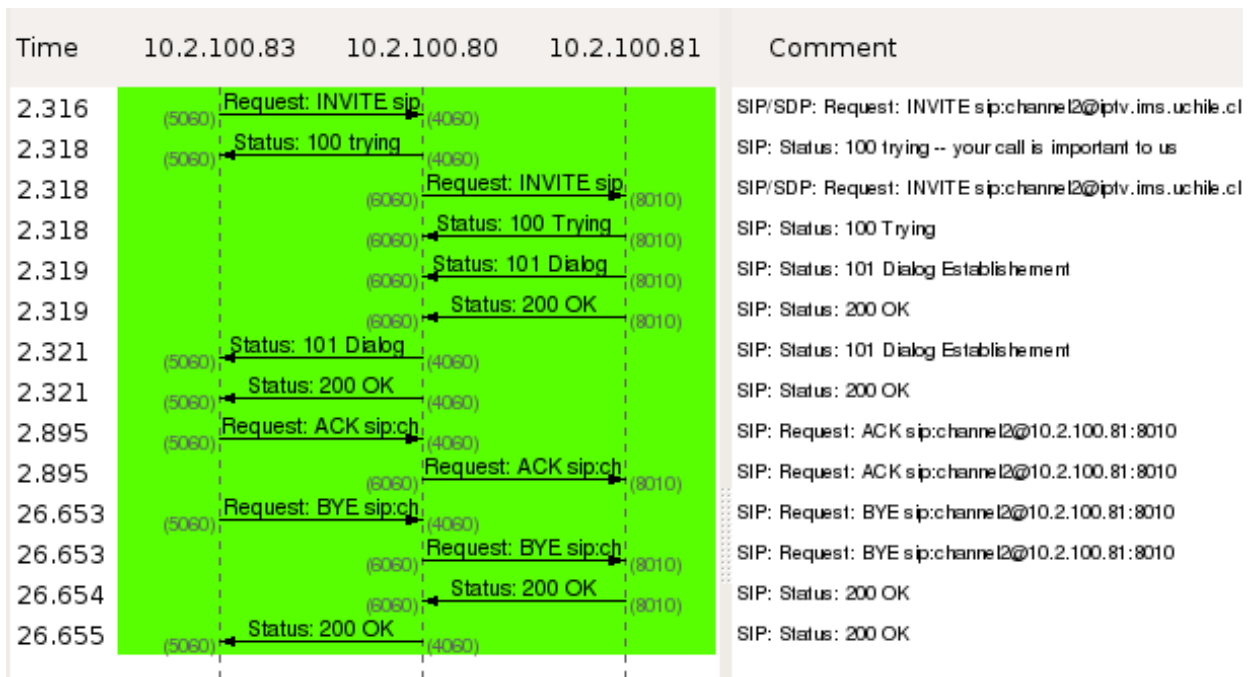


FIGURA 4.6: MENSAJES SIP EN SOLICITUD DE

Estos flujos de mensajes SIP evidencian que la plataforma se comporta tal como se estipulaba en su diseño.

4.1 RESULTADOS DEL VIDEO SHOPPING SOBRE LTE

4.1.1 CARACTERÍSTICAS GENERALES

Como se describe en la sección 3.8, la implementación del servicio de Video Shopping considera dos etapas. En primer lugar se levanta la plataforma de Mobicents y se accede al servicio directamente mediante un cliente SIP. Cuando se cumple esta etapa, se incorpora el núcleo de IMS al sistema para permitir un mejor control del servicio. Se logra implementar ambas etapas exitosamente.

La plataforma de Mobicents que se levanta incluye un servidor que ejecuta la lógica del servicio (JSLEE), un servidor multimedia y un servidor web construido en base a herramientas de desarrollo de servicios.

Gran parte de las virtudes de este servicio se encuentran en el portal web que se desarrolla en base a estas herramientas. Uno de los objetivos principales que se persigue, es la mejora de las interfaces del usuario de manera que estas permitan una interacción más amistosa y más cercana a los servicios propios al dominio de internet. En este sentido, el servicio cumple satisfactoriamente con el requisito. La Figura 4.7 muestra el diseño del aspecto del portal web del servicio.



FIGURA 4.7: PORTAL WEB DEL SERVICIO DE VIDEO SHOPPING

Si bien el aspecto del portal es bastante simplista, de todas maneras se pone en evidencia el gran potencial que posee y debe considerarse como una prueba de concepto o como punto de partida para un desarrollo posterior. Se destaca el hecho de que la plataforma de Mobicents ofrezca herramientas del dominio de internet (http, AJAX) para el desarrollo de interfaces en servicios propios al dominio de las telco.

Más allá del aspecto, el portal web ofrece una serie de características que logran enriquecer el servicio de video. La Figura 4.8 muestra la interfaz para la creación de usuarios que implementa el portal web.

FIGURA 4.8: CREACIÓN DE USUARIOS ONLINE EN EL PORTAL WEB DEL SERVICIO DE VIDEO SHOPPING

Es importante destacar que la creación de usuarios mediante este medio solo guarda los datos en la memoria temporal del servidor. Por lo tanto, si este se reinicia, estos datos se pierden.

Remove?	Title	Actor	Price	Quantity
<input type="checkbox"/>	Burn after reading	Audrey Hepburn	\$99.99	1
<input type="checkbox"/>	Evan Almighty	Audrey Hepburn	\$69.99	1

Subtotal: \$169.98
 Tax (8.25%): \$14.02
 Total: \$184.00

FIGURA 4.9: CARRO DE COMPRAS EN EL PORTAL WEB DEL SERVICIO DE VIDEO SHOPPING

La Figura 4.9 muestra la implementación de un carro de compras dentro del portal web. El usuario puede agregar, quitar o modificar la cantidad de videos en cualquier momento antes de realizar la compra.

Otro aspecto principal del servicio de Video Shopping en su capacidad de traducir la interacción en el portal web en llamadas SIP que además luego desencadenan la transmisión de video desde un servidor multimedia. Para que esto ocurra, se debe poner en marcha un cliente SIP capaz de comunicarse mediante el protocolo RTSP y que sea compatible con el mecanismo de redirección de contenido. El cliente se debe registrar en la plataforma y luego se debe hacer click en el botón “*watch trailer*” en el portal web. Esto desencadena una llamada SIP desde el servidor JSLEE al cliente SIP y mediante su confirmación es posible recibir la dirección RTSP en la que se aloja el video que se solicitó. Mediante el mecanismo de redirección de contenido, el cliente extrae la dirección y automáticamente se contacta con el servidor multimedia. El cliente elegido logra cumplir estos requisitos y se desempeña de buena manera.

La siguiente imagen muestra el cliente reproduciendo un tráiler de una película solicitado desde el portal web.

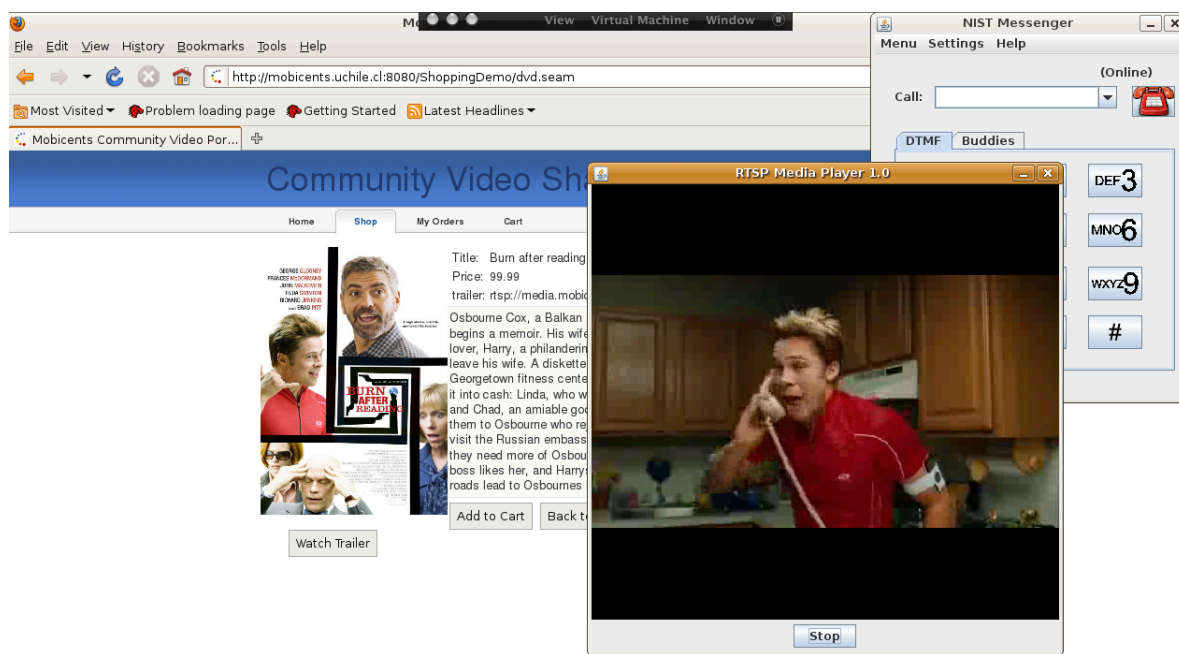


FIGURA 4.10: RECEPCIÓN DE VIDEO MEDIANTE CLIENTE SIP EN SERVICIO DE VIDEO SHOPPING

Cuando el usuario decide aceptar su orden en el carro de compras, este recibe una llamada a su terminal SIP en la que se le pide confirmar la compra. Este mecanismo es muy similar al que ocurre cuando el cliente solicita ver el tráiler de video y naturalmente también funciona sin problemas.

Finalmente, cuando se incorpora IMS al sistema que compone este servicio, es posible observar que ya no es necesario registrar el usuario en la plataforma de Mobicents. Ahora basta con registrar al cliente directamente en IMS. Las llamadas entre la plataforma de Mobicents y el cliente ahora ocurren a través de IMS y se logra obtener los mismos resultados del caso anterior.

La Figura 4.11 muestra el cliente IMS reproduciendo un video solicitado desde el portal web.

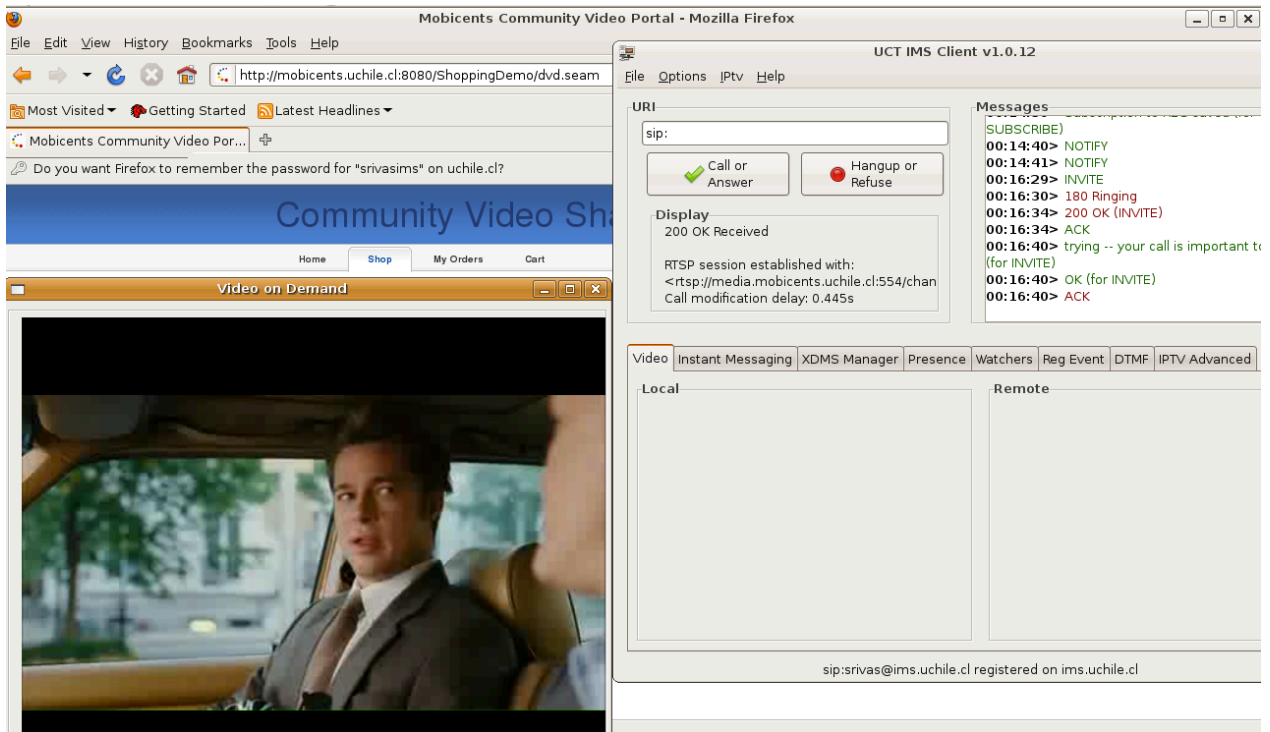


FIGURA 4.11: UTILIZACIÓN DE CLIENTE IMS PARA ACCEDER AL SERVICIO DE VIDEO SHOPPING

Es importante mencionar que al igual que el caso del servicio de IPTV, el servicio de Video Shopping tiene exactamente el mismo problema de seguridad. Si bien la plataforma web permite restringir el uso del servicio mediante la configuración previa de usuarios que pueden realizar el Login. El servidor multimedia no realiza ningún tipo de verificación con el servidor de Mobicents en relación a qué cliente es quien solicita el video. Esto provoca que un determinado cliente no provisionado podría eventualmente solicitar el envío del video directamente al servidor multimedia consigue la dirección RTSP correspondiente. Esto se puede ver en los diagramas de funcionamiento que se muestran en la siguiente sección.

4.1.1 FUNCIONAMIENTO

Al igual que en el caso del servicio de IPTV, en esta sección se muestran capturas de tráfico de partes esenciales del servicio utilizando el software WireShark.

Se utiliza la arquitectura de conexión que se muestra en las Figuras 3.13 y 3.14, las cuales consideran la selección de direcciones IPs según se muestra en la Tabla 3.6.

En esta arquitectura se considera una única dirección IP para toda la estructura de IMS y otra para toda la estructura de IPTV. En este sentido, los siguientes diagramas no muestran mensajes entre elementos de dichas estructuras, como los que ocurren por ejemplo entre las CSCFs.

El proceso más relevante de este servicio es la interacción que ocurre entre componentes desde el momento en que se hace click en el portal web en el botón para solicitar el video hasta que este finalmente llega al cliente. En primer lugar se muestra este proceso en el enfoque sin IMS y luego con IMS.

Time	10.2.100.83	10.2.100.82	Comment
24.177	(5070)	(5060)	SIP: Request: REGISTER sip:mobicents.uchile.cl:5060
24.184	(5070)	(5060)	SIP: Status: 200 OK (1 bindings)
34.274	(5070)	(5060)	SIP: Request: INVITE sip:srivas@10.2.100.83:5070
34.277	(5070)	(5060)	SIP: Status: 180 Ringing
35.955	(5070)	(5060)	SIP/SDP: Status: 200 OK, with session description
36.027	(5070)	(5060)	SIP/SDP: Request: ACK sip:srivas@10.2.100.83:5070;transport=udp, with session description
40.154	(5070)	(5060)	SIP: Request: INVITE sip:shoppingdemo@10.2.100.82:5060;transport=udp
40.175	(5070)	(5060)	SIP: Status: 200 OK
40.179	(5070)	(5060)	SIP: Request: ACK sip:shoppingdemo@10.2.100.82:5060;transport=udp
40.375	(5070)	(5060)	SIP: Status: 100 Trying

FIGURA 4.12: MENSAJES SIP EN SERVICIO DE VIDEO SHOPPING SIN IMS

La Figura 4.12 muestra todos los mensajes desde que el cliente se registra hasta que se recibe el mensaje SIP de tipo OK que contiene la dirección RTSP a ser extraída mediante el mecanismo de redirección de contenido.

Como se puede observar, en primer lugar el cliente envía un mensaje SIP de registro al servidor JSLEE ubicado en 10.2.100.82, el cual responde con un mensaje de tipo OK, si este proceso no presenta problemas. El paso siguiente (no detallado en este flujo) es solicitar el video a través del portal web. Esta acción genera un que el motor de procesamiento web (Mobicents SEAM Web Framework) envíe un evento dentro de la plataforma Mobicents. Este evento provoca que el servidor de ejecución JSLEE envíe un mensaje SIP de tipo INVITE al cliente. El cliente automáticamente responde al cliente con un mensaje SIP que indica el status de la llamada, pero

el usuario debe contestar la llamada para que el flujo pueda continuar. Una vez contestada la llamada se genera el mensaje SIP de tipo OK desde el cliente al servidor, el cual es respondido por un mensaje de tipo ACK. En este momento se genera una transmisión mediante el protocolo RTP (tampoco detallada en este flujo) que contiene un mensaje de voz que solicita al usuario confirmar la recepción del video. El cliente posteriormente puede realizar esta confirmación realizando lo que se conoce como un SIP reINVITE, lo que en este contexto se traduce como un mensaje INVITE tradicional pero enviado dentro de una llamada en curso. El servidor responde con un mensaje OK. Este es el mensaje que contiene la dirección RTSP del video que se ha solicitado. El cliente automáticamente extrae esta dirección, responde al servidor, y luego contacta al servidor de contenido multimedia, lo que se muestra en la Figura 4.13.

Time	10.2.100.83	10.2.100.82	Comment
40.231	OPTIONS rtsp://medi		RTSP: OPTIONS rtsp://media.mobicents.uchile.cl:554/channel1 RTSP/1.0
40.359	Reply: RTSP/1.0 200		RTSP: Reply: RTSP/1.0 200 OK
40.360	DESCRIBE rtsp://med		RTSP: DESCRIBE rtsp://media.mobicents.uchile.cl:554/channel1 RTSP/1.0
40.612	Reply: RTSP/1.0 200		RTSP/SDP: Reply: RTSP/1.0 200 OK, with session description
40.645	SETUP rtsp://10.2.1		RTSP: SETUP rtsp://10.2.100.82:554/channel1/trackID=0 RTSP/1.0
40.775	Reply: RTSP/1.0 200		RTSP: Reply: RTSP/1.0 200 OK
40.811	SETUP rtsp://10.2.1		RTSP: SETUP rtsp://10.2.100.82:554/channel1/trackID=1 RTSP/1.0
40.975	Reply: RTSP/1.0 200		RTSP: Reply: RTSP/1.0 200 OK
40.991	PLAY rtsp://media.m		RTSP: PLAY rtsp://media.mobicents.uchile.cl:554/channel1 RTSP/1.0
41.187	Reply: RTSP/1.0 200		RTSP: Reply: RTSP/1.0 200 OK
74.826	PAUSE rtsp://media.		RTSP: PAUSE rtsp://media.mobicents.uchile.cl:554/channel1 RTSP/1.0
74.923	Reply: RTSP/1.0 200		RTSP: Reply: RTSP/1.0 200 OK
74.976	TEARDOWN rtsp://med		RTSP: TEARDOWN rtsp://media.mobicents.uchile.cl:554/channel1 RTSP/1.0
75.151	Reply: RTSP/1.0 200		RTSP: Reply: RTSP/1.0 200 OK

FIGURA 4.13: MENSAJES RTSP EN SERVICIO DE VIDEO SHOPPING SIN IMS

En este flujo de mensajes se puede ver que el cliente contacta el servidor de *streaming* con un mensaje RTSP de tipo DESCRIBE el cual le entrega al servidor una descripción de un objeto multimedia que el servidor es capaz de manejar. En respuesta, el servidor envía un mensaje RTSP de tipo OK que contiene la información de inicialización para la transmisión. El cliente utiliza esta información para construir un mensajes de tipo SETUP que indican los parámetros a ser usados en la trasmisión. Todos estos mensajes son respondidos con mensajes RTSP de tipo OK. Una vez que se hayan establecido todos los parámetros para la transmisión, el cliente envía un mensaje de tipo PLAY, el cual efectivamente hace que la transmisión comience. En este paso es donde el cliente usa la totalidad de la información que se extrajo del mensaje RTSP que envió el servidor JSLEE, llamando directamente al canal que señala la dirección RTSP. Finalmente, el usuario puede terminar la transmisión en cualquier momento mediante el envío de una mensaje de tipo TEARDOWN.

La siguiente imagen muestra el flujo de mensajes SIP al incorporar IMS a la plataforma del servicio.



FIGURA 4.14: MENSAJES SIP EN SERVICIO DE VIDEO SHOPPING CON IMS

Es importante mencionar que la Figura 4.14 no muestra los mensajes SIP que se transmiten en el momento del registro del usuario a IMS, pues ocurre exactamente lo mismo que en el registro que se muestra en la sección anterior.

De lo anterior, también es relevante que en este caso no existe registro directo en el servidor de Mobicents, pues ahora el IMS se encarga de enrutar los mensajes correctamente.

Por otro lado, se puede observar que los demás mensajes son exactamente los mismos que en el caso de la implementación sin IMS, con la salvedad que estos deben ser repetidos. Por cada mensaje que el IMS reciba de alguna de las partes, debe volver a generarlo hacia el otro sentido, en cada caso anuncia que está realizando dicha acción.

En definitiva, IMS logra actuar correctamente en la capa de control permitiendo en flujo de mensajes entre cliente y servidor cuando no existe registro entre ellos.

5 CAPÍTULO V: DISCUSIÓN

En este capítulo se establece una discusión acerca de diversos aspectos relacionados con el trabajo de memoria. Se comparan las características buscadas para los servicios desarrollados en contraste con los obtenidos. Se mencionan algunos beneficios que ofrecen los servicios puestos en marcha en comparación con los que se pueden observar hoy en el mercado. Finalmente se discute acerca del impacto del desarrollo de servicios mediante la utilización de ambientes de creación servicios.

En este trabajo se pretende la implementación y puesta en marcha de servicios de televisión con un carácter convergente, que permita interacción con usuario y entre usuarios, que permita la elección de contenido de manera flexible y con cobro flexible, que ofrezcan mejores interfaces gráficas que hagan del servicio algo más cómodo para el usuario y que finalmente se ofrezcan ciertas facilidades para el operador de red.

Considerando estas metas, es posible argumentar que el desarrollo de servicios fue exitoso. Cada uno de los proyectos utilizados cumple con alguno de los criterios recién mencionados. Se logra el carácter convergente a través de la implementación sobre la arquitectura de IMS, se permite interacción entre usuarios mediante los servicios que IMS implementa por si solo (mensajería, llamadas de voz y videoconferencias). Se logra la elección de contenido en horarios flexibles mediante la plataforma de IPTV que implementa VoD. Se logra el cobro inteligente a través de la plataforma que se adjunta a los servicios de IPTV que realiza cobros por segundo consumido y no mediante un cargo fijo. Se permite la interacción entre usuarios y la mejora de las interfaces de usuario a través del servicio de Video Shopping que implementa un portal web interactivo que realiza llamadas SIP. Por último se logra dar facilidades al operador de red mediante la incorporación de IMS en la plataforma de Video Shopping pues permite el control de las llamadas a través de un elemento típicamente ubicado en el núcleo de red de las Telco. Además se logra interconectar todos estos proyectos de manera satisfactoria, por lo que los servicios finales cumplen con los objetivos planteados.

Sin embargo, también es importante remarcar algunos alcances de los servicios desarrollados. Si bien cumplen con los objetivos de diseño, como se ha mencionado, estos por ningún motivo pretenden transformarse en un servicio con características de un servicio comercial. Ambos servicios deben ser considerados como una prueba de concepto de lo que es posible desarrollar. En este sentido, y además considerando que todo lo desarrollado se construye en base a proyectos de código libre, este trabajo es un excelente aporte docente para la realización de experiencias demostrativas o bien para el desarrollo de nuevos servicios tomando ideas de lo que se expone en este documento.

En cuanto a las limitaciones, en particular se puede mencionar algunos aspectos de seguridad del cual carecen los servicios desarrollados. En ambos casos, el servidor multimedia de *streaming*, no realiza ningún tipo de verificación de qué usuario está solicitando el video, ya sea con la plataforma Mobicents o con algún componente de IMS. Si bien, IMS a través de la provisión de servicios en el HSS y Mobicents a través de las puertas de Login con clave, restringen el uso del servicio sólo a usuarios previamente registrados, cualquier usuario que consiga las direcciones RTSP de dónde están alojados los videos podría eventualmente solicitarlos directamente al servidor multimedia. En el caso del servicio de Video Shopping se da solución parcial a este problema, pues el *streaming* de videos sólo ocurre cuando se solicitan las sinopsis, pero la compra propiamente tal se realiza en el portal web. En un caso hipotético en que las películas se distribuyan en algún formato físico (por ejemplo en DVD), entonces el servicio en realidad agrega algo de seguridad pues cada usuario además de realizar el Login, debe también confirmar la compra con un dispositivo SIP que esté provisionado en IMS. Sin embargo, en este trabajo se pretende el desarrollo de servicios de video de próxima generación y no la implementación de elementos de seguridad en una tienda que distribuya elementos físicos, por lo que se descarta dicho caso y se mantiene pendiente la incorporación de mayor seguridad en la plataforma.

Otras limitaciones tienen que ver con las dependencias de los proyectos utilizados. En diversos casos se dificulta la instalación de componentes en distintos sistemas operativos o bien utilizan librerías que han dejado de actualizarse. Se puede mencionar el caso del cliente JAIN-SIP-Applet-Phone. Cómo se puede ver en la sección 8.4.4, dicho proyecto depende de las librerías JMF y gstreamer, ambas han dejado de actualizarse desde hace bastante tiempo y su puesta en marcha sobre sistemas operativos más recientes es algo inestable. Lo mismo ocurre con el código fuente del servicio de Video Shopping el cual sólo puede levantarse en una versión obsoleta del servidor JSLEE de Mobicents.

En cuanto a los servicios de manera individual, se puede mencionar que el servicio de IPTV es bastante útil como ejemplo particular del rol de los servidores de aplicación en la arquitectura IMS. Aporta lo necesario para comprender el funcionamiento de la ejecución de servicios en IMS para luego pasar al desarrollo de un servicio algo más elaborado como lo es el servicio de Video Shopping. También es relevante mencionar que el cliente que se utiliza inicialmente en esta plataforma termina siendo lo suficientemente robusto como para hacerlo interactuar con el segundo servicio desarrollado. Más aún, su diseño contempla la incorporación de aún más elementos para enriquecer servicios, como podría ser por ejemplo, un servidor de presencia.

En cuanto al servicio de Video Shopping, se pueden comentar diversos factores positivos. En un mercado inundado de alternativas, las interfaces gráficas son cada vez más determinantes en las preferencias de los usuarios a la hora de la elección de un servicio por sobre otro. Más aún, sistemáticamente los usuarios comienzan a reemplazar los servicios tradicionales por otros en el dominio de internet que responden mejor a sus necesidades, en cuanto ofrecen mayor flexibilidad

en la entrega de contenido a través de plataformas más cómodas y amigables, aún cuando presenten una calidad de servicio significativamente más pobre. En este sentido, las operadoras de red son las llamadas a modernizar las plataformas de manera de permitir el desarrollo de servicios convergentes con prestaciones que se adecuen de mejor manera a la tendencia actual. El servicio de Video Shopping es justamente un ejemplo de lo que se puede lograr mediante la incorporación de una arquitectura convergente y a través de la utilización de ambientes de desarrollo de servicios. Lo que se logra en este servicio es solo una muestra de lo que se puede implementar mediante la utilización de herramientas que facilitan procesos más complejos como lo es la interacción del dominio de internet con el dominio de las Telco.

Esta plataforma puede seguir desarrollándose, por ejemplo mediante la incorporación de listas de contactos en el portal web con los cuales se pueda compartir críticas de películas y recomendaciones. Se podría desarrollar un cliente SIP compatible con televisores actuales para que efectivamente los videos puedan verse en la comodidad de una pantalla grande. El ambiente de creación de servicios de Mobicents incorpora diversas herramientas que buscan justamente facilitar labores de este estilo.

Lo anterior sitúa al servicio de Video Shopping como un excelente punto de partida para continuar con el enriquecimiento de los servicios convergentes, más aún considerando que se dispone de una red LTE compatible sobre la cual se pueden montar servicios exigentes en consumo de recursos de red.

6 CAPÍTULO VI: CONCLUSIONES

En términos generales se puede afirmar que se cumple con el objetivo principal de este trabajo. Se lleva a cabo una metodología de trabajo adecuada y se logran desarrollar servicios de televisión de próxima generación para núcleos IMS/EPC sobre la red LTE de la empresa ZTE, que se encuentra instalada en el Laboratorio de Telecomunicaciones del Departamento de Ingeniería Eléctrica de la Universidad de Chile.

Para lograr el desarrollo de estos servicios, se hace un revisión previa de las arquitecturas relacionadas y su funcionamiento. Debido a este estudio, se logra identificar los elementos clave que hacen posible la interconexión de arquitecturas y plataformas. Luego, se hace una amplia investigación de alternativas de solución, lo que lleva a la selección de diversos proyectos de código libre que terminan por conformar por completo la arquitectura de los servicios. Posteriormente, el estudio de la arquitectura de red disponible en el laboratorio permite identificar la manera de conectar las plataformas correspondientes a los proyectos seleccionados a la red de LTE, generando así un diseño de los servicios a implementar. La puesta en marcha de dicho diseño resulta exitosa y es corroborada con diversas pruebas de funcionamiento y capturas de tráfico.

Es importante mencionar que se logra contribuir al proyecto de desarrollo y estudio de redes convergentes que se lleva a cabo a la Facultad de Ciencias Físicas y Matemáticas en diversos sentidos. Por un lado, se aprovechan implementaciones de trabajos anteriores, enriqueciendo los avances que se postulaban. En este sentido, se continúa con el proceso constructivo de enriquecimiento de herramientas disponibles para la construcción de ambientes docentes para el estudio de redes. Más aún, este trabajo fue desarrollado en simultáneo con otros dos trabajos de título y en conjunto se busca dar un enfoque más global que no solo incluya el desarrollo de servicios, sino que además el estudio de su calidad de servicio y experiencia y el posterior desarrollo de cursos teóricos y prácticos.

Por otro lado, junto a los servicios desarrollados, en este trabajo se entrega una guía de instalación que detalla paso a paso la puesta en marcha de las plataformas y proyectos utilizados en un proceso separado por etapas. Este proceso contiene diversos puntos de control donde se exponen los resultados parciales que se debiera estar logrando, lo que permite la fácil detección de eventuales problemas en las futuras puestas en marcha.

También se destaca que el diseño de los servicios finales, al estar basada sólo en proyectos de código libre, logra una implementación de muy bajo costo, dando mayores facilidades a su posterior estudio.

En cuanto al diseño y puesta en marcha del servicio propiamente tal, como se discutió en el capítulo anterior, se puede mencionar que fue un proceso exitoso en cuanto se logra cumplir con todas las pretensiones y características que se buscaba desarrollar.

Como líneas de trabajo futuro en base a los desarrollos que se presentan en esta memoria se proponen dos alternativas.

Por un lado, es posible realizar un trabajo aplicado donde se puede implementar aspectos que mejoren los servicios desarrollados. Como se describe en el capítulo anterior, existen diversas limitaciones que tienen los proyectos utilizados y es posible dedicar otro trabajo al mejoramiento de la plataforma. Se puede actualizar la versión de Mobicents, adaptar el servicio al nuevo entorno y aprovechar las nuevas herramientas de creación de servicios que ofrece la nueva versión. En particular, se podría incorporar un servidor de presencia para la creación de listas de contactos y el monitoreo de su conexión. Con esto se podría transformar el portal web en una red comunitaria de videos agregándole la funcionalidad de mensajería y video llamadas dentro del entorno web.

Por otro lado, es posible la utilización de algunas herramientas que se muestran en este trabajo y en los trabajos de memoria paralelos, para realizar estudios más orientados a la línea de investigación. Por ejemplo, un estudio de esquemas óptimos para la asignación de recursos de red podría beneficiarse de la existencia de un prototipo de servicio de televisión con características convergentes que opera sobre LTE, pues son justamente estos servicios los que producen mayores consumo de recursos. Esto complementado con que en paralelo a este trabajo se presentan metodologías de medición de calidad de servicio y calidad de experiencia, podría conformar un ambiente propicio para este estudio.

7 CAPÍTULO VII: BIBLIOGRAFÍA

- [1] GSM Association. GSM World Statistics, Disponible Online, Visitada en Agosto 2012.
URL: http://www.gsmworld.com/ewsroom/market-data/market_data_summary.htm
- [2] ITU. About ITU. <http://www.itu.int/en/about/Pages/default.aspx> , 2011.
- [3] ITU. Guidelines for Evaluation of Radio Transmission Technologies for IMT-2000. Recommendation M.1225. International Telecommunication Union, 1997.
- [4] 3GPP, A Global Initiative. About 3GPP. Disponible Online, Visitada en Agosto 2012. URL: www.3gpp.org.
- [5] 3GPP2. About 3GPP2. Disponible Online, Visitada en Agosto 2012. URL: www.3gpp.org
- [6] International Telecommunication Union, ITU-T Recommendation Y.2001, General Overview of NGN, 2004.
- [7] Farooq Khan, *LTE for 4G Mobile Broadband: Air Interface Technologies and Performance*, Cambridge University Press, 2009.
- [8] Tara Ali-Yahiya, *Understanding LTE and its Performance*, Springer, 2011.
- [9] Overview of 3GPP Release 8, V0.2.2. Abril, 2011.
- [10] LTE, the UMTS Long Term Evolution, from theory to practice. Editorial 'Wiley'. 2009.
- [11] System Engineering for IMS Networks, Arun Handa. Editorial Elsevier Inc, 2009.
- [12] POIKSELKÄ, Miikka; MAYER, Georg: *The IMS, IP Multimedia Concepts and Services*. 3rd Edition, Wiley, 2009.
- [13] ITU, Focus Group on IPTV. *Mandate and Terms of Reference of FG IPTV Working Groups. Telecommunications Standardization Sector*. Study Period 2005-2008.
- [14] Gerard O'Driscoll, *Next Generation IPTV Services and Technologies*. First Edition, Wiley, 2008.
- [15] Universidad de Chile. Noticias. Disponible Online, Visitada en Agosto 2012.
<http://www.uchile.cl/noticias/70339/primer-laboratorio-de-tecnologias-lte-en-latinoamerica>
- [16] Andrés Peñaloza, *Diseño e Implementación de un Proveedor de Servicios Genérico con Arquitectura IMS*. Memoria de Título para optar a la carrera de Ingeniero Civil Electricista. Santiago, Chile. Universidad de Chile, Facultad de Ciencias Físicas y Matemáticas, Abril 2008.
- [17] Kurt Rottman, *Diseño e Implementación de un Laboratorio de IPTV*. Memoria de Título para optar a la carrera de Ingeniero Civil Electricista. Santiago, Chile. Universidad de Chile, Facultad de Ciencias Físicas y Matemáticas, Enero 2010.
- [18] Cristian Segura, *Diseño e Implementación de una Metodología para la Medición de QoS/QoE en Servicios OTT Montados sobre una Plataforma LTE/IMS*. Memoria de Título

- para optar a la carrera de Ingeniero Civil Electricista. Santiago, Chile. Universidad de Chile, Facultad de Ciencias Físicas y Matemáticas, Agosto 2012.
- [19] John Oliva, *Diseño e implementación de un curso y laboratorio de servicios sobre Acceso LTE y PacketCore IMS*. Memoria de Título para optar a la carrera de Ingeniero Civil Electricista. Santiago, Chile. Universidad de Chile, Facultad de Ciencias Físicas y Matemáticas, Abril 2012.
 - [20] Mosiuoa Tsietsi, *A Structural and Functional Specification of a SCIM for Service Interaction Management and Personalisation in the IMS*. Submitted in fulfilment of the requirements of the degree Doctor of Philosophy at Rhodes University. August 2011.
 - [21] Fraunhofer FOKUS NGNI. The Open Source IMS Core Project. Disponible Online, Visitada en Agosto 2012. URL: <http://www.openimscore.org/>.
 - [22] R. Spiers, R. Marston, R. Good, and N. Ventura. *The UCT IMS IPTV Initiative*. InNGMAST '09: Third International Conference on Next Generation Mobile Applications, Services and Technologies, pages 503–508. IEEE Computer Society, Septiembre 2009.
 - [23] D. Waiting, R. Good, R. Spiers, and N. Ventura. *The UCT IMS Client*. In TRIDENT '08: Third International Conference on Testbeds and Research Infrastructure for the Development of Networks and Communities, Abril 2009.
 - [24] Vitalis G. Ozianyi, Neco Ventura, *Charging in IP Multimedia Networks*, Proceedings of South Africa Telecommunication Networks and Applications Conference (SATNAC'10), Spier Estate, South Africa, 5-8 Septiembre 2010, ISBN: 978-0-620-47934-9
 - [25] E. Burger, *A Mechanism for Content Indirection in Session Initiation Protocol (SIP) Messages*, RFC 4483 (Proposed Standard), Internet Engineering Task Force, Mayo 2006. Disponible Online, Visitada en Agosto 2012. URL: <http://www.ietf.org/rfc/rfc4483.txt>
 - [26] 3GPP, “Policy and Charging Control Architecture,” *TS 23.203, 3rd Generation Partnership Project (3GPP)*, March 2009.
 - [27] University of Cape Town. How to Setup UCT Advanced IPTv. Disponible Online, Visitada en Agosto 2012. URL: http://uctimsclient.berlios.de/uctviptv_advanced_howto.html
 - [28] University of Cape Town. How to Setup UCT IMS IPTV Charging System. Disponible Online, Visitada en Agosto 2012. URL: <http://uctimsclient.berlios.de/uctimscharging.html>
 - [29] University of Cape Town. UCT IMS Client. Disponible Online, Visitada en Agosto 2012. URL: <http://uctimsclient.berlios.de>
 - [30] Red Hat. Mobicents – The Open Source Cloud Communications Platform. Disponible Online, Visitada en Agosto 2012. URL: <http://www.openimscore.org/>
 - [31] Java Community Process – Community Development of Java Technology Specifications. *JSR 240: JAIN SLEE (JSLEE) v1.1*. Disponible Online, Visitada en Agosto 2012. URL: <http://jcp.org/en/jsr/detail?id=240>
 - [32] Red Hat. Mobicents – The Open Source JAIN SLEE. Disponible Online, Visitada en Agosto 2012. URL: <http://www.mobicents.org/slee/intro.html>

- [33] Red Hat. Mobicents –Mobicents Media Server. Disponible Online, Visitada en Agosto 2012. URL: <http://www.mobicents.org/mms/mms-main.html>
- [34] VideoLAN Organization. Disponible Online, Visitada en Agosto 2012. URL: <http://wiki.videolan.org/>
- [35] VideoLAN Organization. VLC Media Player Disponible Online, Visitada en Agosto 2012. URL: <http://wiki.videolan.org/vlc/>
- [36] Red Hat. Mobicents –Mobicents Shopping Demo. Disponible Online, Visitada en Agosto 2012. URL: <http://www.mobicents.org/shopping-demo.html>
- [37] Red Hat. Mobicents – Mobicents JAIN SLEE Shopping Demo Example User Guide. Disponible Online, Visitada en Agosto 2012. URL: http://docs.jboss.org/mobicents/jain-slee/2.5.0.FINAL/examples/converged-demo/user-guide/en-US/pdf/Mobicents_SLEE_Example_Converged_Demo_User_Guide.pdf
- [38] Red Hat. JBoss –The Seam Framework, Next generation enterprise Java development. Disponible Online, Visitada en Agosto 2012. URL: <http://seamframework.org/>
- [39] Red Hat. JBoss Community –Hibernate. Disponible Online, Visitada en Agosto 2012. URL: <http://www.hibernate.org/>
- [40] The HSQL Development Group. HSQLDB– The 100% Java Database. Disponible Online, Visitada en Agosto 2012. URL: <http://hsqldb.org/>
- [41] Oracle Corporation. Java.net. Disponible Online, Visitada en Agosto 2012. URL: <http://java.net/projects/jain-sip-applet-phone>
- [42] Oracle Corporation. MySQL – The world’s most popular open source database. Disponible Online, Visitada en Agosto 2012. URL: <http://www.mysql.com/>
- [43] The Apache Software Foundation. Apache Tomcat. Disponible Online, Visitada en Agosto 2012. URL: <http://tomcat.apache.org/>
- [44] Mosiuoa's Research Portal. Converged Movie-Demo. Disponible Online, Visitada en Agosto 2012. URL: <http://hezekiah.rucus.net/converged-movie-demo/>
- [45] The Apache Software Foundation. Apache Maven Proyect. Disponible Online, Visitada en Agosto 2012. URL: <http://maven.apache.org/>
- [46] Draisberghof. Software. USB_ModeSwitch. Disponible Online, Visitada en Agosto 2012. URL: http://www.draisberghof.de/usb_modeswitch/
- [47] Wireshark Foundation. Wireshark. Disponible Online, Visitada en Agosto 2012. URL: <http://www.wireshark.org/>
- [48] Mark G. Sobbel. *A Practical Guide to Linux: Commands, Editors, and Shell Programming*. 2nd Edition, Prentice Hall, November 2009.
- [49] Linux.ie. The Beginners Linux Guide. Disponible Online, Visitada en Agosto 2012. URL: <http://www.linux.ie/newusers/beginners-linux-guide/>
- [50] LXDE Wiki. Ubuntu. Disponible Online, Visitada en Agosto 2012. URL: <http://wiki.lxde.org/en/Ubuntu>

- [51] Official Ubuntu Documentation. Ubuntu Installation Guide. Disponible Online, Visitada en Agosto 2012. URL: <https://help.ubuntu.com/8.04/installation-guide/>
- [52] Fraunhofer FOKUS NGNI. OpenIMScore Installation Guide. Disponible Online, Visitada en Agosto 2012. URL: http://www.openimscore.org/installation_guide
- [53] University of Cape Town. How to Install the Open IMS Core on Ubuntu. Disponible Online, Visitada en Agosto 2012. URL: http://uctimsclient.berlios.de/openimscore_on_ubuntu_howto.html
- [54] University of Cape Town. Register and Make Calls with the UCT IMS Client. Disponible Online, Visitada en Agosto 2012. URL: http://uctimsclient.berlios.de/uctimsclient_on_ubuntu_howto.html
- [55] JBoss Community. Maven Getting Started – Developers. Disponible Online, Visitada en Agosto 2012. URL: <https://community.jboss.org/wiki/MavenGettingStarted-Developers>
- [56] Mobicents JAIN SLEE Shopping Demo Example User Guide. Disponible Online, Visitada en Agosto 2012. URL: http://docs.jboss.org/mobicents/jain-slee/2.5.0.FINAL/examples/converged-demo/user-guide/en-US/pdf/Mobicents_SLEE_Example_Converged_Demo_User_Guide.pdf

8 ANEXOS

8.1 ANEXO A: ACRÓNIMOS

3GPP:	Third Generation Partnership Project
3GPP2:	Third Generation Partnership Project 2
APT:	Advanced Packet Tool
AS:	Application Server
CDF:	Charging Data Function
CDMA:	Code Division Multiple Access
CSCFs:	Conjunto de entidades Call Session Control Function
CTF:	Charging Trigger Function
DHCP:	Dynamic Host Configuration Protocol
DNS:	Domain Name Server
EPC:	Evolved Packet Core
EPG:	Electronic Programming Guide
FCFM:	Facultad de Ciencias Físicas y Matemáticas
FDD:	Frequency Division Duplex
GPRS:	General Packet Radio Service
GSM:	Global System for Mobile Communications
HSQldb:	Hyper Structured Query Language Database
HSPA:	High Speed Packet Access
HSS:	Home Subscriber Server
I-CSCF:	Interrogating Call Session Control Function
IETF:	Internet Engineering Task Force
IMS:	IP Multimedia Subsystem
iFC:	Initial Filter Criteria
IP:	Internet Protocol
IPTV:	Internet Protocol Television
ITU:	International Telecommunication Union
JAIN:	Java APIs for Intelligent Networks
JCP:	Java Community Process
JMF:	Java Media Framework
JSR:	Java Specification Request
JTA:	Java Transaction API
LAN:	Local Area Network
LTE:	Long Term Evolution
MGCP:	Media Gateway Control Protocol
MSCML:	Media Server Control Markup Language
NGN:	Next Generation Networks
OCF:	Online Charging Function
OSI:	Open System Interconnection
P2P:	Peer-to-Peer
P-CSCF:	Proxy Call Session Control Function
PSTN:	Public Switched Telephone Network
QoE:	Quality of Experience
QoS:	Quality of Service

RA:	Resource Adapter
S-CSCF:	Serving Call Session Control Function
SBB:	Service Building Block
SCE:	Service Creation Environment
SDP:	Service Delivery Platform
SIP:	Session Initiation Protocol
SLEE:	Java Logic and Execution Environment
TDD:	Time Division Duplex
UE:	User Equipment
UMTS:	Universal Mobile Telecommunications System
VoD:	Video on Demand
VoIP:	Voice over IP
XML:	Extensible Markup Language

8.2 ANEXO B: CREACIÓN DE UNA MÁQUINA VIRTUAL CONECTADA A UNA LAN

Esta guía muestra como crear una máquina virtual con una dirección IP visible en la LAN a la que pertenece su host y se basa principalmente en los avances que se proponen en [17].

En esta guía se asume que se tienen los conocimientos básicos del uso del sistema operativo Linux, tales como saber sobre uso de terminales, instalar paquetes binarios, compilar programas y configurar el acceso a la red. Si este no es el caso, se recomienda la lectura de algún manual introductorio al sistema operativo. Existen manuales genéricos de Linux para el estudio del uso del terminal [48] y manuales online [49].

El software de virtualización utilizado es VirtualBox OSE 3.0.8, este se puede obtener directamente desde su página oficial para diversas arquitecturas de procesador y sistemas operativos accediendo a la zona de descarga a través del siguiente url: <http://www.virtualbox.org/wiki/Downloads>. En particular se encuentran paquetes binarios para numerosas distribuciones de GNU/Linux en la siguiente dirección: http://www.virtualbox.org/wiki/Linux_Downloads. Sin embargo si se está utilizando una distribución moderna lo más sencillo es instalarlo directamente desde los repositorios, para el caso de Ubuntu 9.10 (Karmic Koala) basta con utilizar apt-get, con esto se realizará la instalación incluido la carga de los módulos específicos del kernel que necesita Virtualbox para ejecutarse:

```
$ sudo apt-get update
$ sudo apt-get install virtualbox
```

Con VirtualBox abierto, se selecciona el botón que dice “Nueva”, esto dará inicio a la configuración de la nueva máquina virtual. Lo primero que se debe hacer es asignar un nombre para identificar posteriormente a la máquina virtual, también se debe elegir el sistema operativo y distribución que corresponda, en este caso se elige Linux y Ubuntu.



FIGURA 8.1: SELECCIÓN DE NOMBRE Y SISTEMA OPERATIVO PARA MÁQUINA VIRTUAL.

El siguiente paso será elegir el tamaño de la memoria RAM que tendrá la máquina virtual. Para hacer una instalación mínima de Ubuntu se debe asignar al menos 128 MB.



FIGURA 8.2: SELECCIÓN DE MEMORIA RAM PARA MÁQUINA VIRTUAL.

Posteriormente es necesario crear un nuevo disco duro virtual, para que sea posteriormente utilizado por la máquina virtual.



FIGURA 8.3: CREACIÓN DE DISCO DURO PARA MÁQUINA VIRTUAL.

Se deberá escoger la manera como se comportará el archivo que representará al disco duro virtual. Una opción es que sea un archivo que crezca de forma dinámica mientras se va llenando de datos, o que inmediatamente posea el tamaño máximo del disco duro virtual. Teóricamente se obtiene un mejor rendimiento al elegir la opción de tamaño fijo, al no tener que redimensionarse los discos mientras está corriendo la máquina virtual.

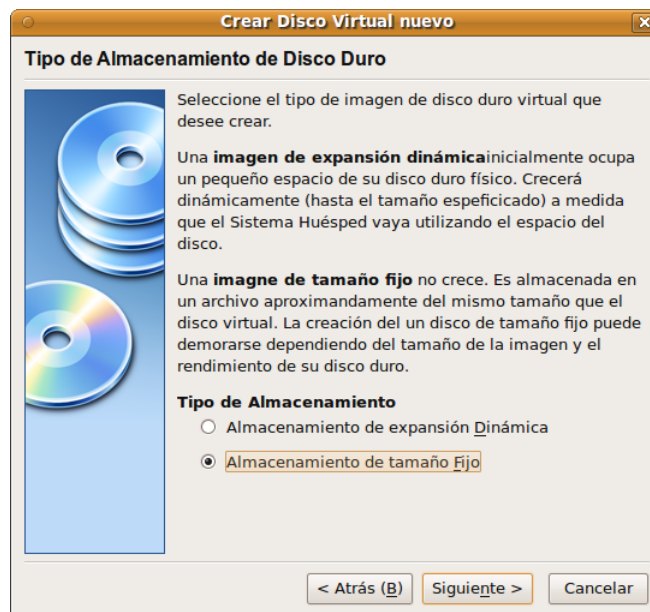


FIGURA 8.4: SELECCIÓN DE TIPO DE ALMACENAMIENTO DE LA MÁQUINA VIRTUAL.

Se debe escoger el tamaño y nombre del disco duro. EL tamaño recomendado para una instalación mínima de Ubuntu es de al menos 2 GB.



FIGURA 8.5: SELECCIÓN DE CAPACIDAD DE DISCO DURO PARA MÁQUINA VIRTUAL.

Con esto se da por terminada la creación de la máquina virtual, sin embargo es necesario configurar la tarjeta de red. Para eso se debe apretar el botón “Configuración” y seleccionar en el menú de la izquierda la opción de Red para tener acceso a las configuraciones específicas de la tarjeta de red de la máquina virtual. Aquí se debe escoger la opción de “Adaptador Punte (Bridge)” y la tarjeta de red del Host que se desea utilizar. Se recomienda usar el adaptador de red Intel PRO/1000 MT Desktop (82540EM) que fue con el que se obtuvo los mejores resultados.

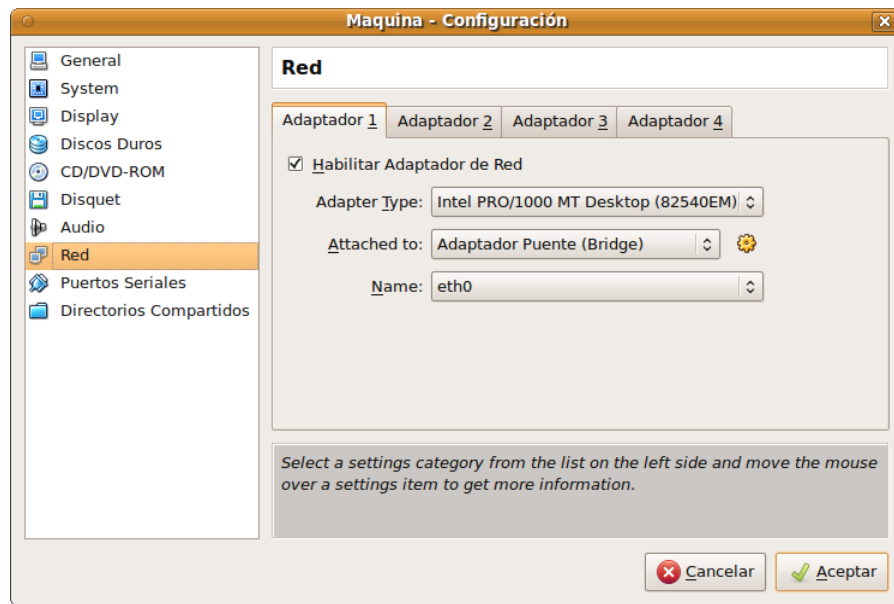


FIGURA 8.6: CONFIGURACIÓN DE TARJETA DE RED DE LA MÁQUINA VIRTUAL.

Antes de terminar con la configuración se puede obtener un mejor rendimiento si se habilita la opción de utilización del set de instrucciones de virtualización presentes en procesadores de alta gama tanto de Intel como AMD, por lo tanto si se dispone de uno conviene hacerlo. Para esto se debe seleccionar la opción System y luego la lengüeta Acceleration.

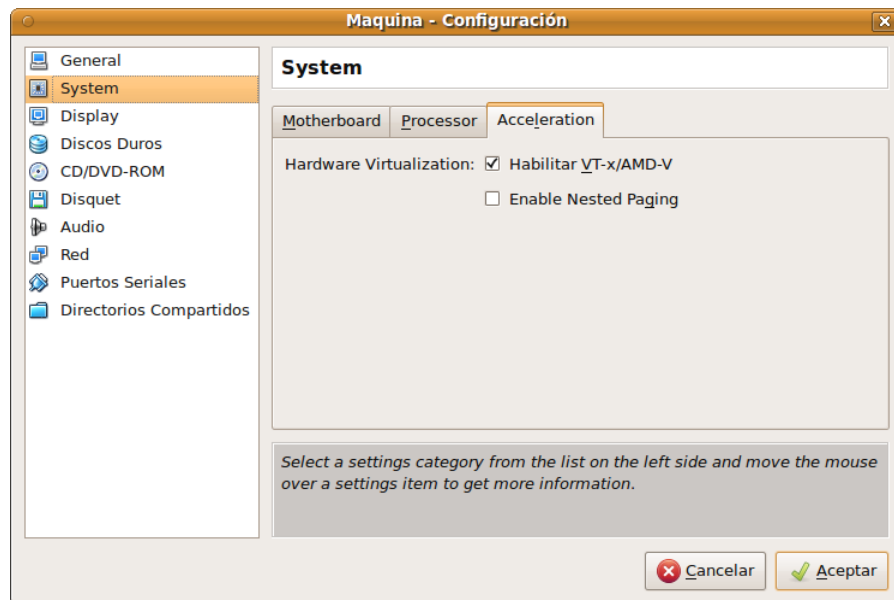


FIGURA 8.7: HABILITACIÓN DE SET DE INSTRUCCIONES DE VIRTUALIZACIÓN.

Con esto se da por terminada la creación de una máquina virtual con acceso a la red local y apta para ser utilizada en el laboratorio de IPTV.

8.3 ANEXO C: INSTALACIÓN Y CONFIGURACIÓN DEL SERVICIO DE IPTV

A continuación se presenta una guía para la instalación y puesta en marcha del servicio de IPTV que utiliza la arquitectura IMS como base del sistema. Gran parte de lo que viene a continuación se basa en los avances que se proponen en [17].

Se incluyen los pasos a seguir para obtener un adecuado entorno de desarrollo, instalar los elementos que componen el sistema y realizar las configuraciones pertinentes de modo de disponer de una plataforma adecuada para el análisis y estudio que este trabajo de título busca realizar.

Todos los componentes que conforman el servicio de IPTV se ejecutan sobre el sistema operativo GNU/Linux. Las instrucciones que en esta guía se enuncian fueron hechas para la distribución Ubuntu 8.04 (Hardy Heron), sin embargo son extensibles y aplicables a otras distribuciones de GNU/Linux si quien las ejecuta posee los conocimientos adecuados. La razón de elegir Ubuntu para la implementación por sobre otras distribuciones de GNU/Linux, como por ejemplo Debian GNU/Linux, es que el software que compone el sistema de IPTV, a excepción del núcleo IMS y el Media Server, ha sido exclusivamente probado y corroborado su funcionamiento en esta distribución por parte de sus desarrolladores. Es por lo tanto requisito para seguir las instrucciones tener conocimientos previos en el sistema operativo GNU/Linux, tales como saber sobre uso de terminales, instalar paquetes binarios, compilar programas y configurar el acceso a la red.

La instalación se hará por etapas, las primeras consistirán en el levantamiento gradual de todos los procesos corriendo en una misma máquina. Finalizado esta labor, se espera obtener un completo y correcto funcionamiento del sistema dadas las configuraciones por defecto que trae cada componente. De esta manera se evitan dificultades generadas por la personalización del sistema o por la configuración de la red. Posteriormente se explicita como se debe replicar cada parte del sistema en una red de área local de modo de tener cada componente corriendo en computadores distintos. Se terminará con las configuraciones específicas de cada elemento que personalizarán la plataforma.

8.3.1 PREPARACIÓN DEL COMPUTADOR Y SISTEMA OPERATIVO

Antes de comenzar la instalación de la plataforma de IPTV, es preciso disponer de un computador apto para su uso. Existen dos opciones: utilizar un computador real o utilizar una máquina virtualizada. Ambas opciones son igualmente validas si se desea utilizar el sistema con fines de estudio. En cuanto a requisitos mínimos para las características de estos equipos estos están directamente relacionados con la distribución de GNU/Linux que se desee utilizar. Para una instalación de Ubuntu 8.04 por defecto los requisitos mínimos deberán ser:

- Procesador: 1GHz (Intel x86 compatible).

- Memoria RAM: 256 MB.
- Disco Duro: 4 GB.
- Tarjeta de red, audio y video.

Sin embargo, si se hace una instalación mínima de Ubuntu, lo cual es lo recomendado si se desea virtualizar numerosas máquinas, los requisitos son los siguientes:

- Procesador: 1 Ghz (x86 compatible).
- Memoria RAM: 128 MB.
- Disco Duro: 2 GB.
- Tarjeta de red, audio y video.

Una vez que se dispone de un computador o máquina virtualizada con las características apropiadas y contándose además acceso a Internet, se deberá proceder con la instalación de Ubuntu 8.04. Una vez terminado esta tarea, se debe actualizar el sistema para dejar los paquetes de software al día. Las instrucciones enmarcadas que se describen en esta guía corresponden a los comandos que deben ser ingresados en un terminal.

```
$ sudo apt-get update
$ sudo apt-get upgrade
```

En caso de que el computador disponible posea características de hardware muy limitadas o se quiere utilizar una máquina virtual se recomienda hacer una instalación mínima de Ubuntu de modo de tener un sistema más ligero, esto se logra escogiendo la opción de “Install a command Line System” en el momento de iniciar con la instalación. Una vez terminado este proceso, será necesario agregar algún escritorio ligero como XFCE o LXDE. Si escogemos LXDE que es más liviano se deberá agregar el repositorio de LXDE antes de instalarlo, ya que este no existe en Ubuntu 8.04.

```
$ sudo sh -c 'echo "deb http://ppa.launchpad.net/lxde/ubuntu hardy main" >>
/etc/apt/sources.list'
$ sudo apt-get update
$ sudo apt-get upgrade
$ sudo apt-get install xorg lxde
```

Se recomienda no instalar un Display manager, como podría ser GDM, XDM o Slim, de modo de ahorrar memoria en el sistema. Esto dado que no es necesario disponer de una interfaz grafica si se desea ejecutar en el futuro sólo uno de los componentes de la plataforma de IPTV. Luego si se necesitase una interfaz grafica se puede ejecutar directamente el comando “startx” con que se da

inicio al servidor grafico X.org. Para esto es necesario primero ejecutar el siguiente comando una única vez:

```
$ sudo update-alternatives --config x-session-manager
```

Y se escoge la opción “startlxde”. Con esto el sistema quedará configurado para iniciar el escritorio LXDE cada vez que se ejecute el comando:

```
$ startx
```

En esta sección se utilizó [50] y [51] como fuentes de consulta para la construcción de los pasos que se mostraron.

8.3.2 INSTALACIÓN Y CONFIGURACIÓN DE COMPONENTES DE IMS Y CLIENTE COMPATIBLE

Partiendo de la base que se tiene un computador apto con GNU/Linux instalado, como fue descrito en el punto anterior, se continuará con la instalación del núcleo IMS y del cliente IMS. Partiendo por el Core IMS, lo primero será obtener su código fuente, para esto se necesita contar con el gestor de versiones Subversion instalado en nuestro equipo:

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

Luego, creamos las carpetas donde se va a descargar el código fuente y le cambiamos el dueño a la carpeta OpenIMSCore a nuestro nombre de usuario (reemplazar con el nombre correspondiente donde dice nombre_de_usuario) con el fin de tener todos los privilegios necesarios.

```
$ sudo mkdir /opt/OpenIMSCore
$ sudo chown -R nombre_de_usuario /opt/OpenIMSCore/
$ cd /opt/OpenIMSCore
$ mkdir ser_ims
$ mkdir FHoSS
```

Ahora se descargan los códigos fuentes desde las ramas principales de los repositorios existentes para los CSCFs y para el HSS a través del gestor de versiones. Tras cada descarga Subversion informa de la revisión de software obtenida, que corresponde a la última disponible en el repositorio.

```
$ svn checkout http://svn.berlios.de/svnroot/repos/openimscore/ser_ims/trunk
ser_ims
$ svn checkout http://svn.berlios.de/svnroot/repos/openimscore/FHoSS/trunk
FHoSS
```

Este trabajo de memoria fue realizado utilizándose la revisión 1175.

Ahora hay que compilar las fuentes que se han descargado, para esto es necesario primero instalar en la máquina las herramientas necesarias para la compilación y posterior ejecución de los programas.

```
$ sudo apt-get install sun-java6-jdk mysql-server libmysqlclient15-dev
libxml2-dev bind9 ant flex bison build-essential ipsec-tools lib6-dev
libcurl4-openssl-dev
```

Ahora ya es posible realizar las compilaciones de los CSCFs:

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

Y se procede a compilar el HSS. Para esto es requisito que este antes correctamente configurada la variable de entorno JAVA_HOME.

```
$ cd /opt/OpenIMSCore/FHoSS/
$ export JAVA_HOME=/usr/lib/jvm/java-6-sun/
$ ant compile deploy
```

Tras estos pasos ya se encuentran compilados los elementos que componen el núcleo IMS, ahora hace falta poblar las bases de datos MYSQL con la información correspondiente al I-CSCF y al HSS.

```
$ 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
```

Continuando con la implementación del sistema se vuelve necesario configurar el servidor de resolución de nombres de dominio. Para esto es necesario realizar una serie de modificaciones a ciertos archivos que operará BIND para establecer las consultas de DNS. Se comienza copiando el archivo que contiene los nombres de dominio a resolver “open-ims.dnszone” que viene con el código de OpenIMS Core a la carpeta donde BIND guarda sus archivos de configuración.

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

También es necesario modificar el archivo “named.conf.local” para eso ejecutamos:

```
$ sudo gedit /etc/bind/named.conf.local
```

Y pegamos el siguiente texto al final del archivo:

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

También hay que preocuparse de que el archivo “/etc/resolv.conf” este apuntando a la dirección del DNS, en este caso a la dirección “127.0.0.1” para esto simplemente editamos el archivo:

```
$ sudo gedit /etc/resolv.conf
```

Y lo dejamos de la siguiente manera:

```
nameserver 127.0.0.1
```

En este punto es importante recordar que si esta activado un cliente de DHCP en la máquina, esta dirección que recién fue modificada, será sobrescrita la próxima vez que se reinicie la unidad. Tras los cambios se debe reiniciar BIND para que tomen efecto las modificaciones realizadas.

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

Se pueden hacer pruebas para comprobar que el DNS funcione correctamente, para esto solo bastará con ver si hay respuesta cuando se hace “ping” a los nombres de dominio que ocuparan los elementos del núcleo IMS. Ejemplo, al HSS sería:

```
$ ping hss.open-ims.test
```

Ahora solo falta copiar los archivos de configuración y los scripts para lanzar los componentes del core IMS a la carpeta OpenIMSCore antes de hacer funcionar el núcleo IMS.

```
$ cp /opt/OpenIMSCore/ser_ims/cfg/*.cfg /opt/OpenIMSCore/  
$ cp /opt/OpenIMSCore/ser_ims/cfg/*.xml /opt/OpenIMSCore/  
$ cp /opt/OpenIMSCore/ser_ims/cfg/*.sh /opt/OpenIMSCore/
```

Para levantar el core IMS abrimos cuatro terminales, o tabs en un mismo terminal si el que se está utilizando tiene la opción (Ctrl+Shift+T en el terminal de Gnome) y ejecutamos en cada una las siguientes entradas.

Para el P-CSCF:

```
$ cd /opt/OpenIMSCore  
$ ./pcscf.sh
```

Para el I-CSCF:

```
$ cd /opt/OpenIMSCore  
$ ./icscf.sh
```

Para el S-CSCF:

```
$ cd /opt/OpenIMSCore  
$ ./scscf.sh
```

Para el HSS:

```
$ cd /opt/OpenIMSCore  
$ export JAVA_HOME=/usr/lib/jvm/java-6-sun/  
$ ./fhoss.sh
```

Si todo ha salido bien, se debiera tener funcionando correctamente el core IMS. Para terminar, sólo faltaría ejecutar un cliente IMS para poder registrarse y utilizar los servicios que entrega el sistema hasta este punto. Para esto descargaremos el paquete binario del Cliente IMS y lo instalaremos después de haber hecho lo mismo con sus dependencias.

```
$ cd  
$ wget http://download.berlios.de/uctimsclient/uctimsclient1.0.12.deb  
$ sudo apt-get install libosip2-2 libexosip2-4 libgstreamer0.10-0  
gstreamer0.10-plugins-base gstreamer0.10-plugins-good gstreamer0.10-plugins-  
bad gstreamer0.10-plugins-ugly gstreamer0.10-ffmpeg vlc libvlc0 libcurl3  
$ sudo dpkg -i uctimsclient1.0.12.deb
```

Para ejecutar el Cliente IMS solo bastará con ejecutarlo desde un terminal:

```
$ uctimsclient
```

En esta sección se utilizó [52], [53] y [54] como fuentes de consulta para la construcción de los pasos que se mostraron.

8.3.3 INSTALACIÓN Y CONFIGURACIÓN DE SISTEMA DE IPTV BÁSICO

Antes de comenzar con la implementación de los componentes esenciales que darán vida a la plataforma de IPTV es necesario que el núcleo IMS en el cual se basa el sistema esté previamente instalado y configurado como es descrito en el punto anterior y exista certeza de su correcto funcionamiento. Se comenzará por el IPTV Application Server, para esto se debe descargar el código fuente, instalar las librerías necesarias y realizar su compilación:

```
$ cd
$ wget http://download.berlios.de/uctimsclient/uctiptv_advanced1.0.0.tar.gz
$ tar -xvf uctiptv_advanced1.0.0.tar.gz
$ sudo apt-get install libosip2-dev libexosip2-dev
$ cd uctiptv_advanced
$ make
```

Se necesita hacer que la dirección `iptv.open-ims.test` este presente en el DNS, para eso se debe agregar una línea al archivo “`open-ims.dnszone`”

```
$ sudo gedit /etc/bind/open-ims.dnszone
```

Y se debe agregar al final del archivo la siguiente línea:

```
iptv          1D IN A      127.0.0.1
media        1D IN A      127.0.0.1
```

Luego se necesita reiniciar el DNS para que se incorporen estos datos:

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

Posteriormente, habrá que configurar manualmente a través de la interfaz web del HSS la información relevante para que el IPTV AS quede a disposición del núcleo IMS de modo que le reenvíe las peticiones que llegan desde el Cliente IMS. Para acceder a la interfaz web del HSS,

una vez que se esta ejecutando el HSS como fue descrito en el punto anterior, se debe abrir un navegador web e ingresar a la pagina <http://hss.open-ims.test:8080> que corre en el servidor web que se levanta el HSS cuando se ejecuta. Los datos para ingresar al sistema son:

- Username: hssAdmin
- Password: hss

Una vez adentro hay que ir al apartado SERVICES, y lo primero es crear un Application Server utilizando el menú ubicado en el costado izquierdo de la pagina, luego ingresamos la siguiente información y guardamos la configuración. Notar que el IPTV Application Server escucha en el puerto 8010.

Application Server -AS-

ID	2
Name*	iptv
Server Name*	sip:iptv.open-ims.test:8010
Diameter FQDN*	iptv.open-ims.test
Default Handling*	Session - Continued
Service Info	
Rep-Data Limit	1024

Sh Interface - Permissions			
Permission for	UDR	PUR	SNR
Allowed Request	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Repository-Data	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
IMPU	<input type="checkbox"/>		<input type="checkbox"/>
IMS User State	<input type="checkbox"/>		<input type="checkbox"/>
S-CSCF Name	<input type="checkbox"/>		<input type="checkbox"/>
iFC	<input type="checkbox"/>		<input type="checkbox"/>
Location	<input type="checkbox"/>		
User-State	<input type="checkbox"/>		
Charging-Info	<input type="checkbox"/>		
MS-ISDN	<input type="checkbox"/>		
PSI Activation	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
DSAI	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Aliases Rep Data	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Mandatory fields were marked with "*"

FIGURA 8.8: DATOS PARA CREACIÓN DEL APPLICATION SERVER.

Lo siguiente será añadir un Trigger Point que coincida con todas las peticiones que contengan *.iptv.open-ims.test.* en su cabecera SIP, como podría ser la solicitud del Canal 1 del servicio de IPTV a través de la dirección sip:channel1@iptv.open-ims.test . Para esto seleccionamos Create Trigger Point en el menú y se deben ingresar los siguientes datos para luego guardar la configuración.

Trigger Point -TP-

ID	2	Attach IFC Select IFC... <input type="button" value="Attach"/>
Name*	iptv_trigger	
Condition Type CNF*	Conjunctive Normal Format	

Mandatory fields were marked with "**"

ID	IFC Name	Detach

Add SPTs to Trigger Point

Not	<input type="checkbox"/>	SIP Method	INVITE	<input type="button" value="Delete"/>
OR				
			Request-URI	<input type="button" value="+"/>
AND				
Not	<input type="checkbox"/>	SIP Header	To	<input type="button" value="Delete"/>
		SIP Header Content	.*iptv.open-ims.test.*	
OR				
			Request-URI	<input type="button" value="+"/>
AND				
			Request-URI	<input type="button" value="+"/>

FIGURA 8.9: DATOS PARA CREACIÓN DEL TRIGGER POINT.

Continuando, se debe enlazar el Application Server y el Trigger Point con el Initial Filter Criteria, para eso primero creamos el Initial Filter Criteria seleccionando Create en el menú izquierdo e ingresando la siguiente información para luego presionar Guardar.

Initial Filter Criteria -IFC-

ID	2
Name*	iptv_filter
Trigger Point	iptv_trigger
Application Server*	iptv
Profile Part Indicator	Any

Mandatory fields were marked with "**"

FIGURA 8.10: DATOS PARA CREACIÓN DEL INITIAL FILTER CRITERIA.

Ahora es necesario agregar el Initial Filter Criteria recién creado al Shared IFC que ya posee el sistema. Para esto se escoge la opción Search que esta junto a "Shared IFC Sets" en el menú de la

izquierda, luego se debe presionar en el botón Search nuevo para finalmente escoger default_shared_set. Después de esto se adjunta el Initial Filter recién creado con prioridad 2 como se aprecia en la siguiente imagen.

Shared iFC Sets -Sh-iFC-

ID-Set	1
Name*	default_shared_set

Mandatory fields were marked with "*"

Save Refresh Delete

Attach iFC

Select iFC...	Priority	2	Attach
---------------	----------	---	--------

Warning: Priority values defined here can overwrite priority values defined in SP-iFC setup!

List of attached iFCs

ID	Name	Priority	Detach
1	default_ifc	0	Detach
2	iptv_filter	2	Detach

FIGURA 8.11: DATOS DE CONFIGURACIÓN EN SHARED IFC SETS.

Para terminar con la configuración del HSS, se pone Search en Service Profiles y luego se hace presiona el nuevo botón Search que aparece para poder elegir al Service Profile por defecto que tiene registrado el HSS, luego se debe adjuntar el Initial Filter Criteria que se creo para el sistema de IPTV con prioridad 2 y el default_shared_set que se modifiko recientemente, tal como se aprecia en la siguiente imagen:

Service Profile -SP-

ID	1
Name*	default_sp
Core Network Service Auth	0

Mandatory fields were marked with "*"

Attach IFC

0

Attach Shared-IFC-Set

List of attached IFCs

ID	IFC Name	Priority	Detach
1	default_ifc	0	<input type="button" value="Detach"/>
2	iptv_filter	2	<input type="button" value="Detach"/>

List of attached Shared-IFC-Sets

ID-Set	Name	Detach
1	default_shared_set	<input type="button" value="Detach"/>

FIGURA 8.12: DATOS DE CONFIGURACIÓN EN SERVICE PROFILE.

Antes de ejecutar el IPTV AS, es necesario configurar el archivo XML que mapeará las direcciones SIP con las direcciones RTSP correspondientes al Media Server. Para esto se editará un archivo a partir del archivo `key_value_file` que se encuentra dentro de la carpeta en que venia el código fuente del IPTV AS:

```

$ cd
$ cd uctiptv_advanced
$ cp key_value_file key_value_file.xml
$ gedit key_value_file.xml
    
```

Y editamos las direcciones de los canales 1, 2 y 3 dejando el archivo de la siguiente manera:

```

<?xml version="1.0" encoding="UTF-8"?>
<key-value_pairs>
<key-value_pair>
  <key>channel1</key>
  <value>rtsp://media.open-ims.test:5554/channel1</value>
</key-value_pair>
<key-value_pair>
  <key>channel2</key>
  <value>rtsp://media.open-ims.test:5554/channel2</value>
</key-value_pair>
<key-value_pair>
  <key>channel3</key>
  <value>rtsp://media.open-ims.test:5554/channel2</value>
</key-value_pair>
</key-value_pairs>
    
```

Ahora ya es posible correr el IPTV AS agregando como parámetro el archivo recién configurado en un nuevo terminal:

```
$ cd
$ cd uctiptv_advanced
$ ./uctiptv_as key_value_file.xml
```

Luego es necesario levantar el Media Server. Se utilizará el programa Videolan - VLC Media Player para estos fines. Lo primero que se debe hacer será instalarlo:

```
$ sudo apt-get install vlc
```

Posteriormente será necesario crear un archivo de configuración de VLC que incluya la información relativa a los canales. En este ejemplo se creará uno de tipo live streaming y dos de VoD. El segundo de VoD servirá para que el cliente pueda ver las grabaciones agendadas remotamente.

```
$ gedit vlm.conf
```

Y copiamos lo siguiente reemplazando las rutas de los videos por las correspondientes a 2 videos que estén contenidos en el computador:

```
new channel1 broadcast enabled
setup channel1 input /carpeta/video_live.avi
setup channel1 output #rtp{sdp=rtsp://media.open-ims.test:5554/channel1}
control channel1 play

new channel2 vod enabled
setup channel2 input /carpeta/video_vod.avi

new channel3 vod enabled
setup channel3 input /carpeta/video_grabado.avi
```

Finalmente solo queda levantar el Media Server a través del siguiente comando, en donde la dirección del parámetro rtsp-host debe ser la dirección IP del Media Server, en este ejemplo 192.168.0.108:

```
$ vlc -I telnet --vlm-conf vlm.conf --rtsp-host media.open-ims.test:5554
```

Si siguiendo con la configuración del sistema de IPTV, es necesario hacer una configuración sencilla en el cliente IMS antes de poder ver exitosamente los canales de IPTV, esto consiste en activar la opción de Video Llamadas que se encuentra en Options → Preferences → Profile.

Ejecutamos el cliente y luego dejamos configurado lo mencionado como se aprecia en la siguiente imagen.

```
$ uctimsclient
```

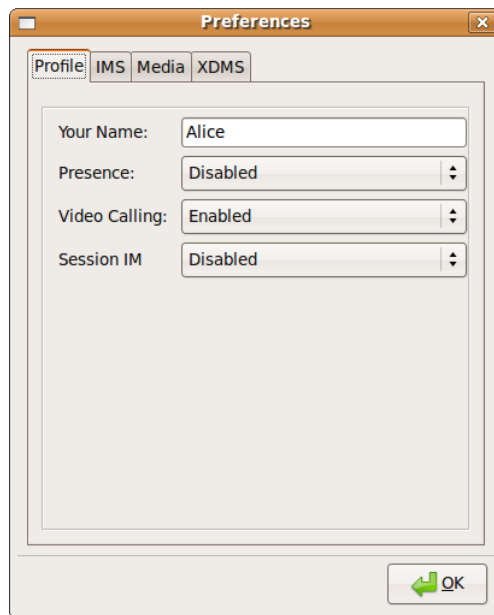


FIGURA 8.13: CONFIGURACIÓN CLIENTE IMS.

Como el cliente UCT IMS Client aún está en desarrollo, si en el contexto del servicio de IPTV, recibe una dirección RTSP errónea, simplemente se caerá sin mostrar en la consola ningún mensaje que haga referencia a la raíz del problema. Si esto ocurre, revisar que tanto el servidor de IPTV como el servidor de *Streaming* de VLC estén configurados en direcciones RTSP correctas.

En esta sección se utilizó [27] [34] como fuentes de consulta para la construcción de los pasos que se mostraron.

8.3.4 INSTALACIÓN Y CONFIGURACIÓN DE SISTEMA DE TARIFICACIÓN

El sistema de tarificación consiste en un IPTV Charging Server con capacidad de tarificación, un Charging Data Function (CDF) y un Online Charging System (OCS). Este IPTV Charging Server es un IPTV Application Server que reemplazará al que fue implementado y utilizado en el punto anterior, dado que posee las mismas funcionalidades de IPTV sumándole la capacidad de operar con el CDF y OCS. Para instalar estos tres elementos, será necesario descargar sus códigos fuentes y compilar cada componente.

```
$ cd
```

```
$ wget http://download.berlios.de/uctimsclient/uctimscharging1.0.1.tar.gz
$ tar -xvzf uctimscharging1.0.1.tar.gz
$ sudo apt-get install libosip2-dev libexosip2-dev
$ cd UCTIMSCHARGING-IPTV
$ make
$ cd ../UCTCDF
$ make
$ cd ../UCTOCS
$ make
```

Una vez terminada esta etapa, será necesario modificar los archivos de configuración de cada elemento antes de levantarlos. Comenzando por el IPTV AS, primero es necesario corregir el nombre del archivo de configuración de uctimscharging-iptv.xml a uctimscharging.xml que viene incorrectamente nombrado, luego hay que modificar la dirección en donde escuchará el servidor cambiándola de 127.0.0.1 a iptv.open-ims.test lo cual solo influirá más adelante cuando se realice la implementación distribuida de las componentes. Para eso se hace:

```
$ cd
$ cd UCTIMSCHARGING-IPTV
$ mv uctimscharging-iptv.xml uctimscharging.xml
$ gedit uctimscharging.xml
```

Dentro del editor Gedit buscamos la siguiente línea:

```
<Acceptor port="3871" bind="127.0.0.1" />
```

Y la reemplazamos por esto:

```
<Acceptor port="3871" bind="iptv.open-ims.test" />
```

Similar a lo que se hizo con el anterior IPTV AS, se debe editar el archivo de configuración que posee las direcciones RTSP del Media Server.

```
$ gedit key_value_file.xml
```

Y se editan las direcciones de los canales 1, 2 y 3 dejando el archivo de la siguiente manera:

```
<?xml version="1.0" encoding="UTF-8"?>
<key-value_pairs>
<key-value_pair>
    <key>channell</key>
    <value>rtsp://media.open-ims.test:5554/channell</value>
</key-value_pair>
<key-value_pair>
```

```
<key>channel2</key>
  <value>rtsp://media.open-ims.test:5554/channel2</value>
</key-value_pair>
<key-value_pair>
  <key>channel3</key>
  <value>rtsp://media.open-ims.test:5554/channel2</value>
</key-value_pair>
</key-value_pairs>
```

La decisión de si el sistema de tarificación ocupará el OCS o el CDF queda determinado por la variable `ch_type` que está en el archivo `server.c` al momento de la compilación del UCTIMSCHARGING-IPTV. Por defecto viene configurado para que el sistema de tarificación utilice el OCS. Si se cambia el valor de `ch_type` de 1 a 2 y se vuelve a compilar, el sistema de tarificación utilizará el CDF en vez del OCS. La línea en cuestión es la siguiente:

```
ch_type = 1; /* This is where the default charging type is defined */
```

Ahora es el turno del CDF, se abre el archivo `cdf.xml` con el editor de texto.

```
$ cd
$ cd UCTCDF
$ gedit cdf.xml
```

Y se reemplaza la siguiente línea:

```
<Acceptor port="3892" bind="127.0.0.1" />
```

Por esto:

```
<Acceptor port="3892" bind="cdf.open-ims.test" />
```

Análogamente para el OCS:

```
$ cd
$ cd UCTOCS
$ gedit ocf.xml
```

Se cambia la línea:

```
<Acceptor port="3891" bind="127.0.0.1" />
```

Por esta:


```
<Acceptor port="3891" bind="ocf.open-ims.test" />
```

Se necesita hacer que las direcciones cdf.open-ims.test y ocf.open-ims.test estén presentes en el DNS, para eso se debe agregar un par de líneas al archivo “open-ims.dnszone”

```
$ sudo gedit /etc/bind/open-ims.dnszone
```

Y se debe agregar al final del archivo la siguiente línea:

```
cdf          1D IN A          127.0.0.1
ocf          1D IN A          127.0.0.1
```

Luego se necesita reiniciar el DNS para que se incorporen estos datos:

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

Una vez terminada la configuración de los elementos involucrados en el sistema de tarificación procedemos a levantar todos estos componentes. Antes de ejecutar el nuevo IPTV AS, se debe ir al terminal en que se está corriendo el IPTV AS que no posee las prestaciones de tarificación y cerramos su proceso con la combinación de teclas Ctrl + C. Posteriormente se procede a ejecutar el nuevo IPTV AS:

```
$ cd
$ cd UCTIMSCHARGING-IPTV
$ ./uctimscharging-iptv
```

En otro terminal (o tab), se levanta el CDF:

```
$ cd
$ cd UCTCDF
$ ./uctimscdf
```

Y también en otro terminal se corre el OCS:

```
$ cd
$ cd UCTOCS
$ ./uctimsocs
```

En esta sección se utilizó [28] como fuente de consulta para la construcción de los pasos que se mostraron.

8.3.5 IMPLEMENTACIÓN DISTRIBUIDA Y PERSONALIZACIÓN DEL SERVICIO DE IPTV

Al seguir la guía de instalación hasta este punto, se tiene como resultado que la plataforma completa esta funcionando sobre una misma máquina, es decir doce entidades levantadas sobre un mismo computador. Además se tiene que el sistema está configurado con sus principales parámetros por defecto en donde destacan el dominio del core IMS: open-ims.test y los usuarios registrados en el sistema Alice y Bob, con sus respectivas direcciones SIP: alice@open-ims.test y bob@open-ims.test. A continuación se presentan los pasos a seguir para levantar cada componente en una máquina distintas de modo de obtener un una plataforema que corrobore su que la implementación puede ser distribuida y que se puedan realizar pruebas tales como capturas de paquetes de tráfico en la red. Además se cambiara el dominio del sistema a uno personalizado, que será ims.toip.uchile.cl, por ultimo se creará un nuevo usuario personalizado que sea capaz de utilizar los servicios que ofrece la plataforma de IPTV.

Antes de realizar la distribución de componentes es necesario contar con suficientes computadores, de lo contrario se tendrán que instalar más de un componente en una misma máquina y en consecuencia tendrán estos elementos una misma dirección IP. Es muy probable que no se cuente con doce computadores disponibles en el laboratorio, es por esto que se vuelve interesante la opción de virtualizar suficientes máquinas de modo de que cada elemento quede separado con su propia dirección IP. En el caso de los computadores reales, lo que se debe hacer es instalar el componente deseado en siguiendo los pasos mencionados previamente en los puntos anteriores. En el caso de que se vayan a utilizar máquinas virtuales para este fin, es posible realizar una clonación de discos virtuales de modo de ahorrar tiempo. Basta con haber instalado todos los componentes en una misma máquina virtual, para solo tener que duplicar el disco duro y cargarlo en una nueva máquina virtual, sin tener que instalar el sistema operativo ni las entidades nuevamente. Esto se puede hacer de la siguiente manera, si se utilizó el software Virtualbox para la virtualización, como fue el caso de la implementación de este trabajo de título. Se debe ingresar a la carpeta que contiene los discos virtuales en el Host y utilizar el comando VboxManage clonevdi.

```
$ cd .VirtualBox/  
$ VBoxManage clonevdi Máquina1.vdi Máquina2.vdi
```

Este comando realizará una clonación del disco, ahora solo será necesario crear una máquina virtual como se detalla en el punto 9.3, pero en vez de crear un nuevo disco virtual, se debe utilizar el que ha sido creado a través del proceso de clonación de discos virtuales. Este procedimiento puede ser repetido el numero de veces que sea necesario hasta disponer del numero de máquinas suficientes para que todos los componentes puedan correr aisladamente. Un detalle importante es que la primera vez que se inicien las máquinas virtuales nuevas y cargue el sistema operativo, será necesario borrar el archivo 70-persistent-net.rules que es creado por udev

para el caso de Ubuntu/Debian. Este archivo contiene la MAC de la tarjeta del computador al cual su disco duro fue clonado, por lo que causará problemas si se mantiene dado que no coincidirá su información con la de la MAC de la tarjeta de red que poseen las nuevas máquinas virtuales.

```
$ sudo rm /etc/udev/rules.d/70-persistent-net.rules
```

Una vez borrado este archivo, al reiniciarse el sistema operativo se creará uno nuevo con la información correcta de la actual MAC que posee la máquina virtual. Una vez que está presente cada componente en un computador, ya sea real o virtual, será necesario hacer algunas modificaciones con respecto a las direcciones IP que están definidos en los archivos de configuraciones de cada elemento. Por defecto todas estas direcciones corresponden a localhost (127.0.0.1) dado que todos los componentes corrían en una misma máquina, pero ahora que los elementos están separados es necesario que cada servidor escuche en la IP que corresponda su dirección en la LAN y que utilice las direcciones del nuevo dominio para comunicarse con sus pares. Al utilizarse un nuevo dominio, la lista de direcciones IP contenida en el DNS debe ser actualizada también. Justamente para estos fines, viene un script junto con los archivos de configuración de OpenIMS Core llamado configurator.sh. Al cual se le da como parámetros los archivos a modificar y luego el dominio que reemplazará todas las apariciones de open-ims.test en aquellos archivos, para luego solicitar la dirección IP que reemplazará a 127.0.0.1 cada vez que se presente en los archivos. Se deberá correr el script para los archivos de configuración de cada componente del sistema:

Cada vez que se ejecuta el script configurator.sh se debe ingresar el nuevo dominio que es ims.uchile.cl, seguido de la dirección IP del computador dentro de la red.

P-CSCF:

```
$ cd /opt/OpenIMSCore/  
$ ./configurator.sh pcscf.cfg pcscf.qos.cfg pcscf.qos.xml
```

I-CSCF:

```
$ cd /opt/OpenIMSCore/  
$ ./configurator.sh icscf.cfg icscf.thig.cfg icscf.xml ser_ims/cfg/icscf.sql  
$ mysql -uroot -p < /opt/OpenIMSCore/ser_ims/cfg/icscf.sql
```

S-CSCF:

```
$ cd /opt/OpenIMSCore/
```

```
$ ./configurator.sh scscf.cfg scscf.xml
```

HSS:

```
$ cd /opt/OpenIMSCore/FHoSS/  
$ /opt/OpenIMSCore/configurator.sh deploy/DiameterPeerHSS.xml  
deploy/hss.properties scripts/hss_db.sql scripts/userdata.sql  
$ mysql -uroot -p < /opt/OpenIMSCore/FHoSS/scripts/hss_db.sql  
$ mysql -uroot -p < /opt/OpenIMSCore/FHoSS/scripts/userdata.sql
```

Esto borrará los datos existentes en el HSS, por lo cual hay que repetir los pasos del punto 9.4.3 para agregar el IPTV AS, pero reemplazando en esta ocasión todas las referencias de open-ims.test a ims.uchile.cl.

IPTV AS:

```
$ cd ~/uctiptv_advanced/  
$ /opt/OpenIMSCore/configurator.sh key_value_file.xml  
$ cd ~/UCTIMSCHARGING-IPTV/  
$ ./configurator.sh key_value_file.xml uctimscharging.xml server.c charging.c  
$ make
```

CDF:

```
$ cd ~/UCTCDF/  
$ /opt/OpenIMSCore/configurator.sh cdf.xml
```

OCF:

```
$ cd ~/UCTOCS/  
$ /opt/OpenIMSCore/configurator.sh ocf.xml
```

PCRF:

```
$ cd /opt/OpenIMSCore/uctpcrf/  
$ /opt/OpenIMSCore/configurator.sh config/DiameterPeerPCRF.xml  
config/pcrf.properties policies/domain_policies.xml  
policies/default_domain_policies.xml
```

PCEF:

```
$ cd /opt/OpenIMSCore/uctpcrf/  
$ /opt/OpenIMSCore/configurator.sh config/DiameterPeerPCEF.xml  
config/pcef.properties
```

DNS:

Para el caso del DNS hay que hacer modificaciones que permitan actualizar la lista de direcciones al dominio nuevo. Para eso se creará una nueva lista capaz de resolver las direcciones pertenecientes al dominio `ims.toip.uchile.cl`

```
$ sudo cp /etc/bind/open-ims.dnszone /etc/bind/ims.uchile.dnszone
```

Luego es necesario modificar este archivo.

```
$ sudo gedit /etc/bind/ims.uchile.dnszone
```

Hay que cambiar todos los dominios `open-ims.test` a `ims.toip.uchile.cl`, para lo cual se puede utilizar el script `configurator.sh`, sin embargo este no servirá para cambiar las direcciones IP de los componentes que aparecen listadas en este archivo las cuales deben ser cambiadas manualmente a las de cada componente en la LAN. También es necesario modificar el archivo “`named.conf.local`” para eso ejecutamos:

```
$ sudo gedit /etc/bind/named.conf.local
```

Y pegamos el siguiente texto al final del archivo:

```
zone "ims.uchile.cl" {  
    type master;  
    file "/etc/bind/ims.uchile.dnszone";  
};
```

Una vez hecho todo esto se debe reiniciar BIND en el DNS.

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

Para que los componentes puedan hacer consultas al DNS es necesario que los archivos “`/etc/resolv.conf`” de cada uno este apuntando a la dirección del DNS, para esto simplemente editamos los archivos en cada máquina:

```
$ sudo gedit /etc/resolv.conf
```

Y los dejamos de la siguiente manera, si por ejemplo la dirección del servidor de DNS fuera `10.2.100.80`:

```
nameserver 10.2.100.80
```

8.4 ANEXO D: INSTALACIÓN DEL SERVICIO DE VIDEO SHOPPING

A continuación se presenta una guía para la instalación y puesta en marcha del servicio de Video Shopping que utiliza la plataforma Mobicents como base. Este proceso contempla dos etapas, primero se muestra la instalación de la plataforma de manera individual y luego se muestra la configuración necesaria para montar el servicio sobre la arquitectura IMS. En la configuración de la segunda etapa será necesario que IMS y un cliente compatible estén funcionando previamente.

En ambas etapas, al igual que en la sección anterior, se incluyen los pasos a seguir para poder replicar exactamente el entorno de desarrollo que se utilizó en este trabajo de memoria, considerando la configuración individual de cada componente.

Para la instalación de cada componente, se utiliza como base el sistema operativo Linux. En particular, se elige la última distribución Ubuntu 8,04 (Hardy Heron), pues logra ser un sistema operativo suficientemente liviano como para poder abrir diversas máquinas virtuales en paralelo en un mismo computador. Cabe destacar que con los conocimientos adecuados, es posible instalar cada componente tanto en otras distribuciones de Linux o bien en otros sistemas operativos. Sin embargo, algunos de los comandos que se muestran a continuación no necesariamente serán válidos. Mobicents en particular puede instalarse y ejecutarse sin dificultades en el sistema operativo Windows, pero se debe tener en cuenta que la instalación de las componentes del núcleo de IMS debe ser sobre Linux.

En esta guía se asume que se tienen los conocimientos básicos del uso del sistema operativo Linux, tales como saber sobre uso de terminales, instalar paquetes binarios, compilar programas y configurar el acceso a la red. Si este no es el caso, se recomienda la lectura de algún manual introductorio al sistema operativo. Existen manuales genéricos de Linux para el estudio del uso del terminal [48] y manuales online [49].

Como se mencionó, el proceso de instalación se divide en dos etapas principales, la instalación de la plataforma individual y la incorporación de la arquitectura IMS. Además de estas etapas se subdivide la instalación en otras etapas intermedias de manera de generar algunos puntos de control que aseguren la correcta instalación final. En primer lugar, se instalan los servidores de Mobicents (JSLEE y Media Server). Recordando que el servicio de Video Shopping se basa principalmente en la modificación de un ejemplo de funcionamiento del Media Server, se procederá instalando la versión original de dicho ejemplo. Esto asegura que se tiene todo lo necesario para continuar. Posteriormente se instalan las fuentes modificadas y se configuran adecuadamente según los usuarios y dominios que se quieran utilizar. A continuación, se instala y configura un cliente SIP (JAIN-SIP-APPLET-PHONE) para poder hacer uso de la plataforma

levantada. Finalmente, una vez que el servicio esta operando correctamente se configuran los últimos detalles para poder hacerlo interactuar con la arquitectura de IMS.

Todo este proceso supone que ya se dispone de computador o máquina virtual con un sistema operativo apto para la ejecución del software requerido. De lo contrario revisar la sección 8.3.1.

8.4.1 INSTALACIÓN DE MOBICENTS

La plataforma Mobicents contiene diversos proyectos que cumplen distintos propósitos. En este caso, se requiere particularmente del servidor JAIN SLEE y el Media Server. Se descargará la versión del Media Server 1.0.2.GA. en un paquete que incluye ambos servidores además del servidor de aplicación JBOSS sobre el cual ambos se ejecutan. La elección de dicha versión se discute en la sección 3.8.

La descarga se realiza mediante los siguientes comandos:

```
$ cd
$ mkdir ms-1.0.2.GA
$ cd ms-1.0.2.GA
$ wget
http://sourceforge.net/projects/mobicents/files/Mobicents%20Media%20Server/1.0.2.GA/mobicents-media-server-all-1.0.2.GA.zip
$ unzip mobicents-media-server-all-1.0.2.GA.zip
```

Este paquete incluye los binarios ya compilados de los servidores que se mencionaron, luego basta descomprimirlo para finalizar su instalación.

```
$ unzip mobicents-media-server-all-1.0.2.GA.zip
```

Para poder verificar que se haya descargado y descomprimido adecuadamente, se levantaran los servidores. Para esto es necesario configurar algunas variables de entorno.

```
$ export JBOSS_HOME=/home/srivas/ms-1.0.2.GA/jboss-4.2.3.GA
$ export JAVA_HOME=/usr/lib/jvm/java-6-sun/
```

Notar que se utiliza “srivas” como nombre de usuario del sistema operativo. Este nombre debe cambiarse según corresponda.

Por último se levanta el servidor con el siguiente comando.

```
$ ./jboss-4.2.3.GA/bin/run.sh
```

El servidor ha terminado de levantarse cuando en el terminal entregue algo de la siguiente forma.

```
Started in 50s:742ms
```

Una vez que está levantado, es posible acceder al servidor web que está incorporado en JBOSS. A modo de ejemplo es posible revisar que la consola de gestión se haya levantado correctamente. Para acceder, basta entrar a la dirección web “<http://localhost:8080/management-console>” en cualquier navegador. Más adelante se verá como cambiar el dominio de esta página web.

En el navegador web, debería poder verse algo como lo que se muestra en la

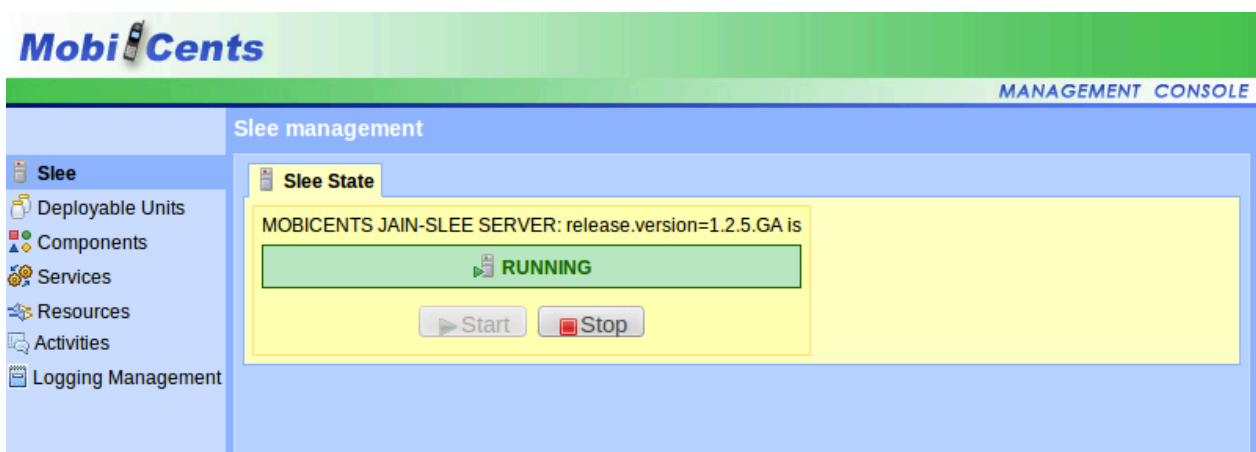


FIGURA 8.14: MANAGEMENT-CONSOLE DEL SERVIDOR DE MOBICENTS

Esta consola sirve además para realizar algunas operaciones en el servidor JSLEE mediante una interfaz gráfica o bien para corroborar que los comandos hechos en el terminal hayan tomado efecto.

Para cerrar el servidor basta cancelar el proceso en el terminal correspondiente (*ctrl+c*) o bien en otro terminal ejecutar el script para detener el servidor:

```
$ cd ~/ms-1.0.2.GA
$ ./jboss-4.2.3.GA/bin/shutdown.sh -S
```

El servidor se ha detenido por completo cuando el terminal arroje el siguiente mensaje:


```
INFO [Server] Shutdown complete
Shutdown complete
Halting VM
```

Todo este proceso se puede estudiar con más detalle en el manual correspondiente a esta versión disponible en la página web del proyecto Mobicents [30]. Dicho manual se incluye en el paquete descargado en el directorio “*Documentation/jain-slee*”.

8.4.2 INSTALACIÓN CONVERGED-DEMO ORIGINAL

En esta sección se instalará el ejemplo original llamado *Converged-Demo* en el cual se basa el servicio de Video Shopping que se utiliza en este trabajo. Este paso se realiza por dos razones, en primer lugar su instalación provoca la descarga automática de diversas librerías necesarias para su ejecución que también se necesitarán para el servicio de Video Shopping. En segundo lugar, este paso genera un punto de control intermedio que permite verificar que los pasos hasta el momento se estén ejecutando correctamente.

El primer paso es descargar la herramienta de gestión y construcción de software Maven que se utiliza para compilar los proyectos utilizando librerías instaladas localmente.

```
$ sudo apt-get install maven2
```

Una vez descargado Maven, es necesario cambiar su archivo de configuración para permitir la descarga de librerías asociadas a JBOSS y los proyectos de Mobicents. Esto se realiza en el archivo *setting.xml* de la siguiente manera:

```
$ sudo gedit /etc/maven2/settings.xml
```

Este archivo contiene diversos *tags* que se abren y se cierran de acuerdo a comandos de la forma `<nombreTag>` y `</nombreTag>`. Todo lo que esté dentro de esos comandos es contenido correspondiente a dicho *tag*.

En este caso, se debe buscar el tag “*<pluginGroups>*” e insertar el siguiente texto:

```
<pluginGroup>org.jboss.maven.plugins</pluginGroup>
```

Notar que se está agregando un “*pluginGroup*” dentro del tag “*pluginGroups*”

Luego se debe buscar el tag “<profiles>” e insertar el siguiente texto:

```
<profile>
<id>jboss</id>
<repositories>
  <repository>
    <id>jboss-public-repository-group</id>
    <name>JBoss Public Repository Group</name>
    <url>http://repository.jboss.org/nexus/content/groups/public/</url>
    <layout>default</layout>
    <releases>
      <enabled>>true</enabled>
      <updatePolicy>never</updatePolicy>
    </releases>
    <snapshots>
      <enabled>true</enabled>
      <updatePolicy>never</updatePolicy>
    </snapshots>
  </repository>
</repositories>
<pluginRepositories>
  <pluginRepository>
    <id>jboss-public-repository-group</id>
    <name>JBoss Public Repository Group</name>
    <url>http://repository.jboss.org/nexus/content/groups/public/</url>
    <releases>
      <enabled>true</enabled>
    </releases>
    <snapshots>
      <enabled>true</enabled>
    </snapshots>
  </pluginRepository>
</pluginRepositories>
</profile>
```

Por último, al final del documento, justo antes de cerrar el tag de *settings* agregar el siguiente texto:

```
<activeProfiles>
  <activeProfile>jboss</activeProfile>
</activeProfiles>
```

Notar que el archivo contiene diversos comentarios donde se explica el formato de cómo se debe agregar nuevos comandos. Es necesario verificar que el texto que se ha agregado no haya quedado dentro de los *tags* de esos comentarios.

Además, es necesario descargar una API de Java (JTA - *Java Transaction API*) que permite la participación de recursos (RA's) en transacciones.

```
$ cd ~/Downloads
```

```
$ wget https://edelivery.oracle.com/otn-pub/jcp/7089-jta-1.0.1B-mr-class-oth-
JSpec/jta-1_0_1B-classes.zip
```

Si la dirección de descarga anterior deja de funcionar, se recomienda la descarga manual de los Class Files del "Java Transaction API Specification 1.0.1B" en <http://java.sun.com/products/jta/>.

La instalación de esta API en el repositorio de Maven se realiza mediante el siguiente comando:

```
$ mvn install:install-file -DgroupId=javax.transaction -DartifactId=jta -
Dversion=1.0.1B -Dpackaging=jar -Dfile=/home/srivas/Downloads/jta-1_0_1B-
classes.zip
```

Notar que se hace referencia a la carpeta de usuario "srivas", dicha carpeta se debe reemplazar por el nombre de usuario local según corresponda.

Una vez que Maven esté adecuadamente configurado, se procede instalando algunos *Resource Adapters* que requiere el ejemplo para poder operar. Para esto, en primer lugar, es necesario que el servidor JSLEE esté levantado. Esto se realiza de acuerdo a los pasos que se detallan en la sección anterior. Además, se aprovechará de agrandar la memoria disponible en el servidor para prepararlo para la posterior despliegue del servicio. Esto se hace configurando la variable de entorno JAVA_OPTS. Los comandos son los siguientes:

```
$ cd ~/ms-1.0.2.GA
$ export JBOSS_HOME=/home/srivas/ms-1.0.2.GA/jboss-4.2.3.GA
$ export JAVA_HOME=/usr/lib/jvm/java-6-sun/
$ export JAVA_OPTS="-Xmn256m -XX:CMSInitiatingOccupancyFraction=80 -
Dorg.jboss.resolver.warning=true -Dsun.rmi.dgc.client.gcInterval=3600000 -
XX:MaxPermSize=256M -Xmx2000M -Xms512M -XX:+UseConcMarkSweepGC"
$ ./jboss-4.2.3.GA/bin/run.sh
```

Una vez levantado el servidor, se debe ejecutar los siguientes comandos en un nuevo terminal:

```
$ export JBOSS_HOME=/home/srivas/ms-1.0.2.GA/jboss-4.2.3.GA
$ export JAVA_HOME=/usr/lib/jvm/java-6-sun/
$ cd ~/ms-1.0.2.GA/examples/sip-services
$ ant deploy-all-proxy-without-initial-invite
```

Con este comando se estará instalando el RA de SIP. Notar que en el terminal del servidor se debería poder observar el proceso de instalación.

A continuación se instala el RA de TTS mediante el siguiente comando:

```
$ cd ~/ms-1.0.2.GA/resources/tts
$ ant deploy
```

Finalmente, se instala el RA de media.

```
$ cd ~/ms-1.0.2.GA/resources/media
$ ant deploy
```

Con esto se ha instalado en el servidor JSLEE todos los recursos necesarios para poder levantar el ejemplo *Converged-Demo*. Esto se puede verificar en el *Management-Console* del servidor. Para entrar a la consola, al igual que en la sección anterior, basta entrar a la dirección web “<http://localhost:8080/management-console>” en cualquier navegador. En la viñeta *Deployable Units* del menú de la izquierda, debería poder verse algo como se muestra en la Figura 8.15.

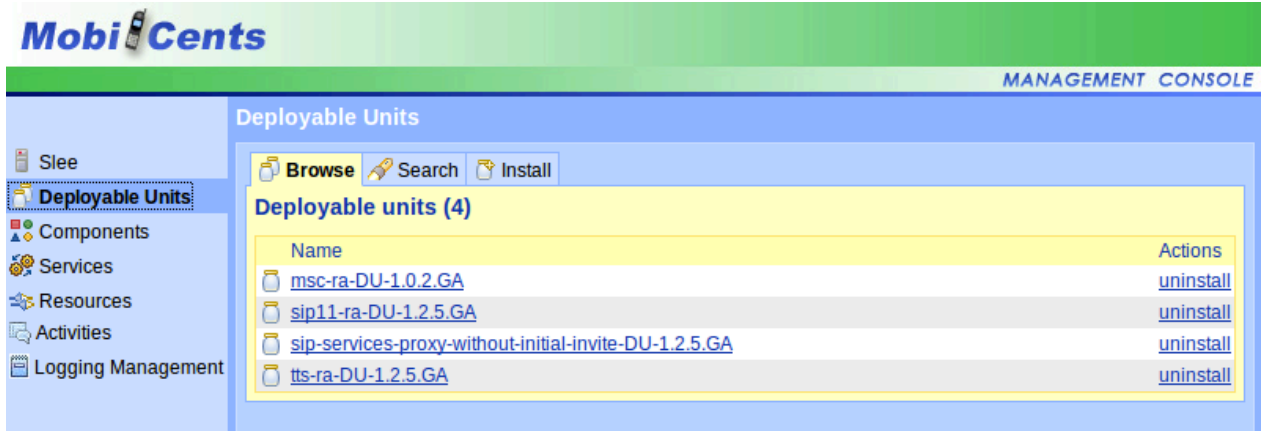


FIGURA 8.15: RESOURCE ADAPTERS NECESARIOS PARA LA INSTALACIÓN DEL SERVICIO

Una vez que se haya corroborado que los RAs estén instalados adecuadamente, es posible compilar e instalar las fuentes del ejemplo *Converged-Demo*. Como se explicó anteriormente, esto se realiza para permitir que se descarguen algunas librerías al repositorio local de Maven.

El paquete del servidor de Mobicents que se descargó solo contiene archivos binarios, por lo que se necesita descargar las fuentes de los ejemplos para poder realizar este paso.

```
$ cd ~/ms-1.0.2.GA
$ wget
http://sourceforge.net/projects/mobicents/files/Mobicents%20Media%20Server/1.0.2.GA/mobicents-media-server-1.0.2.GA-src.zip
```

Luego, se necesita extraer y mover estas fuentes en conjunto con la carpeta de recursos a la ubicación indicada para poder compilarlo. Tener en cuenta que con la carpeta de recursos en este directorio, no se puede instalar otros recursos en el servidor.

```
$ unzip mobicents-media-server-1.0.2.GA-src.zip
$ cd mobicents
$ mv servers/ ~/ms-1.0.2.GA/
$ cd ..
$ rmdir mobicents
$ mv resources/ servers/media/
```

Ahora está todo preparado para la instalación del ejemplo, la cual se realiza con el siguiente comando:

```
$ export JBOSS_HOME=/home/srivas/ms-1.0.2.GA/jboss-4.2.3.GA
$ export JAVA_HOME=/usr/lib/jvm/java-6-sun/
$ mvn clean install
```

La declaración de las variables de entorno no es necesaria si ya se ha hecho en este terminal. Notar que esta instalación provoca la descarga de diversos paquetes y plugins relacionados con los proyectos de mobicents y no es un proceso rápido.

Una vez finalizado este procedimiento, Maven ya dispone de las librerías necesarias para la instalación del ejemplo modificado que compone el servicio de Video Shopping. Sin embargo, para verificar que todo los pasos se hayan ejecutado correctamente, se levantará este ejemplo en el servidor. Este paso no es necesario para la instalación final, pero se recomienda realizarlo.

El comando para levantar el ejemplo en el servidor es el siguiente:

```
$ ant deploy-all
```

Cuando el proceso haya terminado, se puede acceder al servidor web que levanta este ejemplo en la dirección <http://localhost:8080/ShoppingDemo>. Se debería poder ver algo como lo que muestra la Figura 8.16.

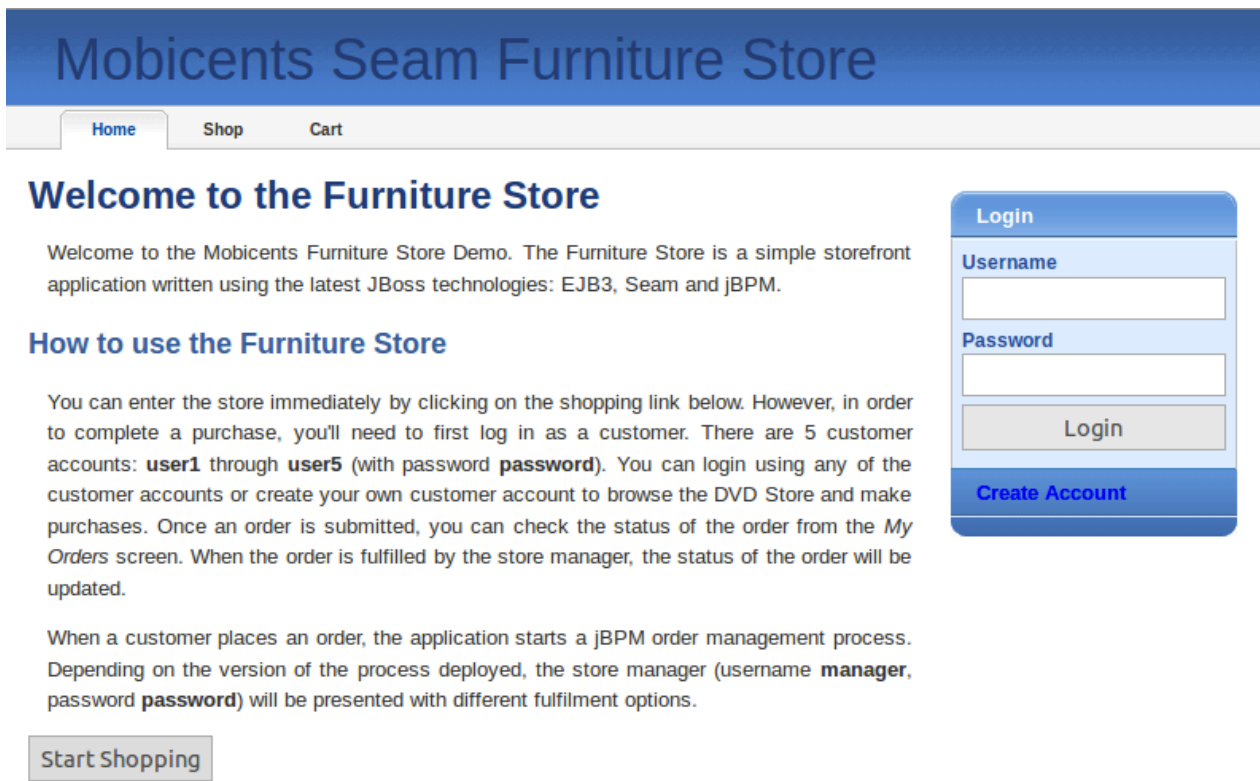


FIGURA 8.16: INTERFAZ WEB DEL EJEMPLO ORIGINAL CONVERGED-DEMO

A continuación se desinstala el ejemplo del servidor. Si bien este paso tampoco es necesario, intentar levantar el servicio de Video Shopping sin desinstalar estas fuentes podría en ciertos casos provocar problemas.

Esto se hace mediante el siguiente comando en la mismo terminal donde se acaba de levantar el ejemplo.

```
$ ant undeploy-all
```

En esta sección se utilizó [55] y [56] como fuentes de consulta para la construcción de los pasos que se mostraron.

8.4.3 INSTALACIÓN Y CONFIGURACIÓN DEL SERVICIO DE VIDEO SHOPPING

Esta sección continúa con la instalación y configuración del servicio de Video Shopping propiamente tal. Para poder ejecutar correctamente los siguientes comandos es importante haber realizado los pasos que se muestran en las secciones anteriores.

Como se explica en la sección 3.4.2, este servicio se basa principalmente en los desarrollos propuestos en [20]. Por esta razón, el primer paso, es la descarga de las fuentes que se ponen a disposición en dicho trabajo. Como las fuentes tienen el mismo nombre que las originales, además se debe eliminar o cambiar el nombre de la carpeta de carpeta antigua.

```
$ cd ~/ms-1.0.2.GA/servers/media/examples
$ mv converged-demo converged-demo-original
$ wget http://hezekiah.rucus.net/converged-movie-demo/converged-demo.zip
$ unzip converged-demo.zip
```

Estas fuentes además se incluyen en un disco anexo a este trabajo para poder tenerlas a disposición en caso de que ya no estén disponibles en el dominio <http://hezekiah.rucus.net/>.

El paso siguiente es compilar las fuentes de la misma manera que se hizo con las del ejemplo original según se detalla en la sección anterior. Recordar que para poder instalar las fuentes es necesario que previamente se haya levantado el servidor JSLEE.

```
$ export JBOSS_HOME=/home/srivas/ms-1.0.2.GA/jboss-4.2.3.GA
$ export JAVA_HOME=/usr/lib/jvm/java-6-sun/
$ mvn clean install
$ ant deploy-all
```

Una vez levantadas las fuentes, es posible acceder al servidor web del servicio entrando a la dirección <http://localhost:8080/ShoppingDemo>. Se debería poder ver algo como lo que muestra la Figura 8.17.



FIGURA 8.17: INTERFAZ WEB DEL SERVICIO DE VIDEO SHOPPING

Este paso eventualmente podría arrojar un error de la siguiente forma:

```
ERROR [compiler] Compiler Initialization Error
java.util.zip.ZipException: error in opening zip file
```

Esto ocurre en ciertas ocasiones cuando se ha compilado un proyecto muchas veces en poco tiempo. Para solucionarlo se debe desinstalar las fuentes y verificar en el *management-console* que no haya quedado ningún *plugin* relacionado al proyecto del *converged-demo* instalado en el servidor JSLEE. La consola debería verse como se muestra en la Figura 8.15. Luego, repetir el paso de instalación de las fuentes.

Una vez que se haya verificado que se puede compilar y levantar el servicio, se procede a hacer algunas modificaciones para que opere correctamente. También se creará un nuevo nombre de dominio en el DNS para que el servidor pueda ejecutarse de manera aislada. El siguiente paso requiere la instalación previa de BIND (esto debería estar instalado si se ha seguido la guía paso a paso).

```
$ sudo gedit /etc/bind/named.conf.local
```


Y se pega el siguiente texto al final del archivo:

```
zone "mobicents.uchile.cl" {  
    type master;  
    file "/etc/bind/mobicents.uchile.dnszone";  
};
```

También hay que crear el archivo que permite la resolución de IPs en esta nueva zona:

```
$ sudo gedit /etc/bind/mobicents.uchile.dnszone
```

Pegar el siguiente texto:

```
$ORIGIN mobicents.uchile.cl.  
$TTL 1W  
@           1D IN SOA      localhost. root.localhost. (  
                2006101001      ; serial  
                3H              ; refresh  
                15M             ; retry  
                1W              ; expiry  
                1D )           ; minimum  
  
           1D IN NS       ns  
ns         1D IN A        127.0.0.1  
  
mobicents.uchile.cl.  1D IN A          127.0.0.1  
  
media      1D IN A        127.0.0.1
```

Notar que se utiliza la IP 10.2.100.80 pues como se explica en la sección 3.9, esta es la IP necesaria para poder realizar la interconexión con el laboratorio de LTE. Si no se desea realizar esta conexión, se puede elegir cualquier IP arbitrario.

En este punto además se hace necesario cambiar la IP local de la máquina en la que se está trabajando (esto hará que se pierda la conexión a internet). Esto se realiza en el menú de edición de la conexión de red en la barra superior del sistema operativo. Se debe configurar una conexión con IP manual con los parámetros que se muestran en la Figura 8.18.

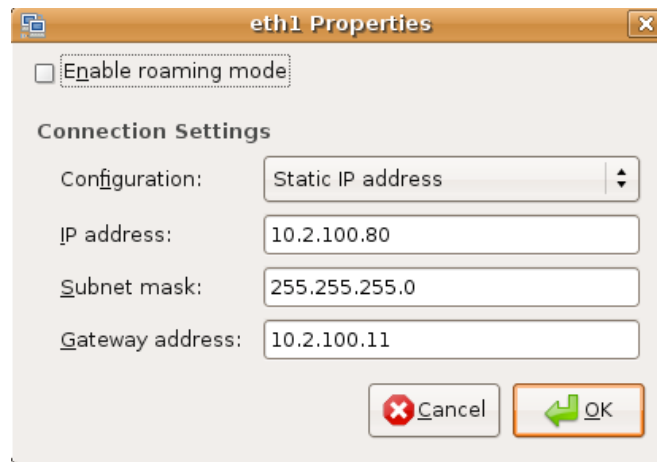


FIGURA 8.18: CONFIGURACIÓN DE PARAMETROS DE CONEXIÓN

Como ya se explicó, estos parámetros se eligen solo para poder conectarse al laboratorio de LTE.

Además hay que configurar el archivo “/etc/resolv.conf” para que apunte a la dirección del DNS, en este caso 10.2.100.80. Para esto se edita el archivo:

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

Dejándolo de la siguiente manera:

```
Nameserver 10.2.100.80
```

Una vez finalizado este paso se debe reiniciar BIND en el DNS.

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

Se puede corroborar que el procedimiento haya sido realizado como responde haciendo un ping al dominio que se acaba de crear. Por ejemplo un ping a la dirección que se le asociará al servidor multimedia que se levantará:

```
$ ping media.mobicents.uchile.cl
```

Dicho comando debería entregar tiempos de respuesta del ping.

Ahora se está en condiciones de realizar las modificaciones al código fuente del ejemplo para configurarlo de manera más personalizada.

En primer lugar, se modifica la dirección SIP del servidor que hará las llamadas a los terminales SIP de los clientes cuando estos solicitan una sinopsis de alguna película.

```
$ cd ~/ms-1.0.2.GA/servers/media/examples/converged-demo
$ gedit example.propeties
```

Reemplazar el valor del parámetro “*caller.sip*” por el siguiente valor:

```
caller.sip=sip:shoppingdemo@mobicents.uchile.cl
```

Luego se crea un usuario personalizado y se cambian los dominios de las direcciones RTSP donde están alojadas las películas. Como se explica en la sección 3.5.2.1, esto se realiza en el archivo *import.sql*.

```
$ gedit web/jar/src/main/resources/import.sql
```

Se puede modificar cualquiera de los usuarios existentes con los valores que se prefieran. Sin embargo, no es posible agregar usuarios nuevos sin modificar el código fuente. A modo de ejemplo se muestra la configuración de un nuevo usuario:

```
INSERT INTO USERS
(USERID, DTYPE, FIRSTNAME, LASTNAME, ADDRESS1, ADDRESS2, CITY, STATE, ZIP, EMAIL, PHONE
, CREDITCARDTYPE, CC_NUM, CC_MONTH, CC_YEAR, USERNAME, PASSWORD) VALUES
(3, 'customer', 'Sebastian', 'Rivas', '4 Privet Drive', 'Cupboard under the
Stairs', 'QSDPAGD', 'SD', 24101, 'serivas@ing.uchile.cl', 'sip:srivas@mobicents.uc
hile.cl:5070', 1, '1979279217775911', 03, 2012, 'srivas', 'srivas')
```

En este ejemplo se ha creado el usuario “*srivas*” con la contraseña “*srivas*” y la dirección SIP “*sip:srivas@mobicents.uchile.cl:5070*”

Además del valor de la dirección SIP, los otros valores a usar no son muy relevantes para las pruebas que se harán. Es importante que dicha dirección tenga el puerto 5070, pues es el que utiliza el cliente SIP.

A continuación se deben modificar los dominios de las películas que se insertan como Productos en la plataforma web. Se utilizará el dominio la siguiente dirección RTSP

```
"rtsp://media.mobicents.uchile.cl:5555"
```

En cada película hay que agregar el canal correspondiente tal como estaban las direcciones anteriores. A modo de ejemplo se muestra como debe quedar el comando para agregar la primera película.

```
insert into PRODUCTS (PROD_ID, ASIN, TITLE, PRICE, IMAGE_URL, TRAILER_URL,
DESCRIPTION) values ('1', '11fj-v0sizL_AA160_', 'Burn after reading', 99.99,
'/img/burn_after_reading_poster.jpg',
'rtsp://media.mobicents.uchile.cl:5555/channel1', 'Osbourne Cox, a Balkan
expert, is fired at the CIA, so he begins a memoir. His wife wants a divorce
and expects her lover, Harry, a philandering State Department marshal, to
leave his wife. A diskette falls out of a gym bag at a Georgetown fitness
center. Two employees there try to turn it into cash: Linda, who wants money
for elective surgery, and Chad, an amiable goof. Information on the disc
leads them to Osbourne who rejects their sales pitch; then they visit the
Russian embassy. To sweeten the pot, they decide they need more of Osbournes
secrets. Meanwhile, Lindas boss likes her, and Harrys wife leaves for a book
tour. All roads lead to Osbournes house. ');
```

Una vez concluidas estas modificaciones, se debe volver a compilar el servicio para incorporar los cambios. Para esto se debe levantar el servidor JSLEE. Aprovechando que se creó el nuevo nombre de dominio, ahora se está en condiciones de levantar el servidor en dicho dominio. Esto hace que el servidor web sea accesible desde fuera de la maquina virtual siempre y cuando la máquina desde donde se quiera acceder esté en la misma red y apunte al DNS que resuelve el dominio.

Para levantar el servidor en el dominio que se creó, basta ejecutar el siguiente comando en un nuevo terminal:

```
$ cd ms-1.0.2.GA
$ ./jboss-4.2.3.GA/bin/run.sh -b mobicents.uchile.cl
```

Ahora es posible compilar las fuentes:

```
$ cd ~/ms-1.0.2.GA/servers/media/examples/converged-demo
$ export JBOSS_HOME=/home/srivas/ms-1.0.2.GA/jboss-4.2.3.GA
$ export JAVA_HOME=/usr/lib/jvm/java-6-sun/
$ mvn clean install
$ ant deploy-all
```

Para verificar que los cambios hayan tomado efecto se puede intentar hacer el *login* en la plataforma con el nuevo usuario. Debido a que se cambio el dominio del servidor, ahora el portal web es accesible en la dirección <http://mobicents.uchile.cl:8080/ShoppingDemo>.

8.4.4 INSTALACIÓN DEL SERVIDOR MULTIMEDIA Y DEL CLIENTE SIP

En esta sección se detalla la instalación y configuración del servidor de *streaming* donde se alojan los archivos de video y luego se continua con la puesta en marcha del cliente SIP. Tal como se ha hecho en el resto del proceso, se incluyen diversas pruebas para verificar que los pasos se hayan ejecutado correctamente.

Para el servidor de *streaming* se utiliza el proyecto VLC, luego el primer paso es descargarlo (Si se ha seguido la guía desde el comienzo, VLC ya debería estar instalado).

```
$ sudo apt-get install vlc
```

Luego, es necesario descargar los archivos de video correspondientes a las sinopsis de las películas. A modo de ejemplo, en <http://hezekiah.rucus.net/converged-movie-demo> se ponen a disposición algunos videos que pueden ser reproducidos por el cliente SIP que se utiliza y corresponden justamente a las películas que se agregaron a la plataforma. Evidentemente, al igual que todo el contenido de la plataforma web, estos videos se pueden cambiar por otros. Si se desea utilizar estos videos, se pueden descargar con el siguiente comando:

```
$ cd ms-1.0.2.GA
$ wget http://hezekiah.rucus.net/converged-movie-demo/movies.zip
$ unzip movies.zip
```

Si se ha seguido la guía paso a paso, entonces en este momento no se tiene conexión a internet. Para poder recuperar la conexión basta configurar el método de asignación de IP a DHCP en el menú que se muestra en la Figura 8.18.

Una vez que se tengan los archivos de video, es necesario crear el archivo de configuración de VLC que incluya la dirección en disco de los videos y los nombres de los canales. Se creará dicho archivo también dentro de la carpeta del servidor de mobicents.

```
$ gedit vlm.conf
```

Luego, se pega el siguiente texto:

```
new channel1 vod enabled
setup channel1 input /home/srivas/ms-1.0.2.GA/movies/channel1.mp4

new channel2 vod enabled
```

```
setup channel2 input /home/srivas/ms-1.0.2.GA/movies/channel2.mp4

new channel3 vod enabled
setup channel3 input /home/srivas/ms-1.0.2.GA/movies/channel3.mp4

new channel4 vod enabled
setup channel4 input /home/srivas/ms-1.0.2.GA/movies/channel4.mp4

new channel5 vod enabled
setup channel5 input /home/srivas/ms-1.0.2.GA/movies/channel5.mp4

new channel6 vod enabled
setup channel6 input /home/srivas/ms-1.0.2.GA/movies/channel6.mp4

new channel7 vod enabled
setup channel7 input /home/srivas/ms-1.0.2.GA/movies/channel7.mp4

new channel8 vod enabled
setup channel8 input /home/srivas/ms-1.0.2.GA/movies/channel8.mp4
```

Es importante destacar que en este ejemplo se está utilizando la carpeta de usuario “*srivas*”, este parámetro se debe reemplazar por la carpeta donde se alojen los videos localmente.

Para levantar el servidor en la dirección RTSP *media.mobicens.uchile* (la cual ya ha sido configurada en el DNS), se ejecuta el siguiente comando:

```
$ vlc -I telnet --vlm-conf vlm.conf --rtsp-host media.mobicens.uchile.cl --rtsp-port 5555
```

En versiones anteriores de vlc, para agregar el puerto basta agregarlo junto a la dirección rtsp. Por ejemplo, el comando sería de la forma:

```
$ vlc -I telnet --vlm-conf vlm.conf --rtsp-host media.mobicens.uchile.cl:5555
```

Para probar que el servidor se haya ejecutado correctamente, es posible intentar acceder al video en otro terminal mediante el siguiente comando:

```
$ vlc rtsp://media.mobicens.uchile.cl:5555/channel1
```

Una vez que se haya corroborado que el servidor de *streaming* se haya levantado correctamente se procede con la descarga del cliente SIP. El cliente a utilizar es el JAIN-SIP-APPLET-PHONE, un cliente de código libre escrito en JAVA que utiliza la API JAIN. En los desarrollos que se muestran en [20], se realizan algunas modificaciones a dicho cliente para permitir su

compatibilidad con RTSP a través la librería gstreamer-java. El código fuente del cliente modificado también se pone a disposición en <http://hezekiah.rucus.net/converged-movie-demo>. La descarga se realiza mediante el siguiente comando:

```
$ cd
$ wget http://hezekiah.rucus.net/converged-movie-demo/jain-sip-applet-
phone.zip
$ unzip jain-sip-applet-phone.zip
```

Estas fuentes además se incluyen en un disco anexo a este trabajo para poder tenerlas a disposición en caso de que ya no estén disponibles en el dominio <http://hezekiah.rucus.net/>.

Para poder compilar las fuentes además se requiere la librería de Java JMF (*Java Media Framework*) para el manejo multimedia. Para descargar y configurar la librería en Ubuntu, se deben realizar los siguientes pasos:

```
$ cd
$ wget https://edelivery.oracle.com/otn-pub/java/jmf/2.1.1e/jmf-2_1_1e-linux-
i586.bin
$ cat jmf-2_1_1e-linux-i586.bin | sed 's/tail +309 $0 > $outname/tail -n +309
$0 > $outname/' > jmf-2_1_1e-linux-i586.bin.fixed
$ chmod u+x jmf-2_1_1e-linux-i586.bin.fixed
$ ./jmf-2_1_1e-linux-i586.bin.fixed
```

No confundir el símbolo “\$” dentro del tercer comando, como si fuese otro comando (En total hay solo 5 comandos, los cuales están marcados en negrita).

Una vez descargada la librería de JMF se procede realizando los cambios correspondientes en el cliente SIP.

El primer paso es modificar el archivo de compilación donde se definen las variables de entorno y se dan los parámetros de IP y puertos a utilizar.

```
$ cd jain-sip-applet-phone/
$ gedit build.linux.properties
```

En este archivo, reemplazar el valor de *java.home* y *jmf.home* por lo siguiente:

```
java.home=/usr/lib/jvm/java-6-sun
jmf.home=/home/srivas/JMF-2.1.1e
```

Donde el nombre de usuario “*srivas*” se debe reemplazar según corresponda.

Luego reemplazar los valores de *gateway.address* y *responder.address* por la dirección IP donde se aloja el servidor de *mobicents*. En este caso, se probará toda la plataforma en la misma máquina, por lo que nuevamente se le da el valor 10.2.100.80

```
gateway.address=10.2.100.80
responder.address=10.2.100.80
```

A continuación se configura la dirección IP del cliente. A este valor se le debe asignar la IP local que se le haya dado a la máquina que ejecutará el cliente SIP. En este caso, nuevamente se utiliza el valor 10.2.100.80. También se configura el dominio al cual se registra el cliente. En este caso, dicho valor será *mobicents.uchile.cl* y corresponde al dominio donde se ha levantado el servidor de *mobicents*. Este procedimiento se realiza editando el siguiente archivo:

```
$ gedit src/gov/nist/applet/phone/ua/Configuration.java
```

Se debe buscar y reemplazar las siguientes variables con los valores que se indican.

```
stackIPAddress="10.2.100.80";
outboundProxy="mobicents.uchile.cl";
userURI="srivas@"+outboundProxy;
```

Notar que el usuario tendrá la dirección SIP: *srivas@mobicents.uchile.cl*, la cual corresponde a la dirección que se le asignó al cliente que se ha agregado en la plataforma en la sección anterior.

En este archivo también es posible cambiar otros parámetros, como por ejemplo el puerto, que en este caso se mantiene con el valor 5070.

Por último, antes de compilar el cliente, es necesario hacer algunas modificaciones en el código fuente. En los desarrollos en [20] se trabaja con el servidor de *streaming* Darwin, en cual se requiere que se agregué la extensión del archivo de video en la dirección RTSP. Debido a esto, se modificó el código fuente original para realizar esta función y ahora es necesario remover esas secciones de código.

Abrir el siguiente archivo:

```
$ gedit src/gov/nist/applet/phone/ua/gui/GstreamerGUITest.java
```


En el editor de texto ir a *Edit/Preferences*, y marcar la casilla que dice “*Show Line Numbers*”.

Luego reemplazar las líneas 56 y 57 por lo siguiente:

```
String response = "rtsp://media.mobicens.uchile.cl:5555/";  
response = response + (String) combo.getItemAt (combo.getSelectedIndex());
```

Abrir el siguiente archivo:

```
$ gedit src/gov/nist/applet/phone/ua/gui/GstreamerGUI.java
```

Reemplazar la línea 50 por lo siguiente:

```
this.response = response;
```

Por otro lado, el cliente se ha programado para poder realizar un SIP reINVITE a una dirección determinada si se presiona el botón para llamar durante una llamada en curso. Dicho valor debe concordar con la dirección SIP que se le ha asignado a la función de plataforma Mobicents que hace las llamadas. Para cambiarla se debe editar el siguiente archivo:

```
$ gedit src/gov/nist/applet/phone/ua/gui/NISTMessenger.java
```

Reemplazar la línea 554 por lo siguiente:

```
AudioCall call = (AudioCall)  
sipMeetingManager.getCallManager().findAudioCall("sip:shoppingdemo@mobicents.  
uchile.cl");
```

También se debe editar el siguiente archivo

```
$ gedit src/gov/nist/applet/phone/ua/MessengerManager.java
```

Donde se debe reemplazar la línea 1237 por lo siguiente:

```
AudioCall call = (AudioCall)
```

```
getCallManager().findAudioCall("sip:shoppingdemo@mobicents.uchile.cl");
```

En este momento se está en condiciones de compilar el cliente. Para esto se debe volver a configurar la IP local al valor 10.2.100.80 (o bien el que se haya elegido). Utilizar el menú de configuración de red que se muestra en la Figura 8.18.

Para compilar y poner en marcha se ejecuta el siguiente comando:

```
$ ant messenger
```

Se debería ver el cliente como se muestra en la Figura 8.19.

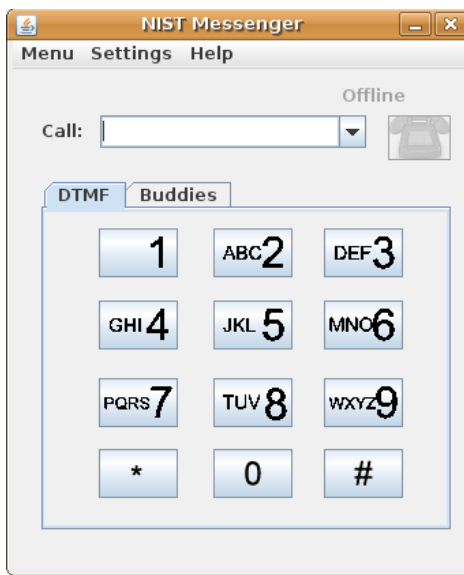


FIGURA 8.19: CLIENTE JAIN-SIP-APPLET-PHONE

Al finalizar este paso se han instalado todas las componentes necesarias para poder acceder al servicio de Video Shopping sin la utilización de la IMS. A continuación se muestran algunas pruebas que verifican su correcto funcionamiento.

Para realizar los siguientes pasos se debe levantar el servidor Mobicents y el servidor de *streaming* VLC según se detalla en pasos anteriores.

La primera prueba es verificar que el cliente SIP logra acceder y reproducir las películas. Para esto se ha incorporado en el cliente la función “*Diagnostics*”, la cual solicita los videos directamente del servidor de *streaming*. Para utilizarla, ir la viñeta *Settings* en el menú del cliente

y seleccionar *Diagnostics*. En la nueva ventana que se abre, seleccionar un canal y presionar PLAY. Se debería ver algo como lo q se muestra en la Figura 8.20.

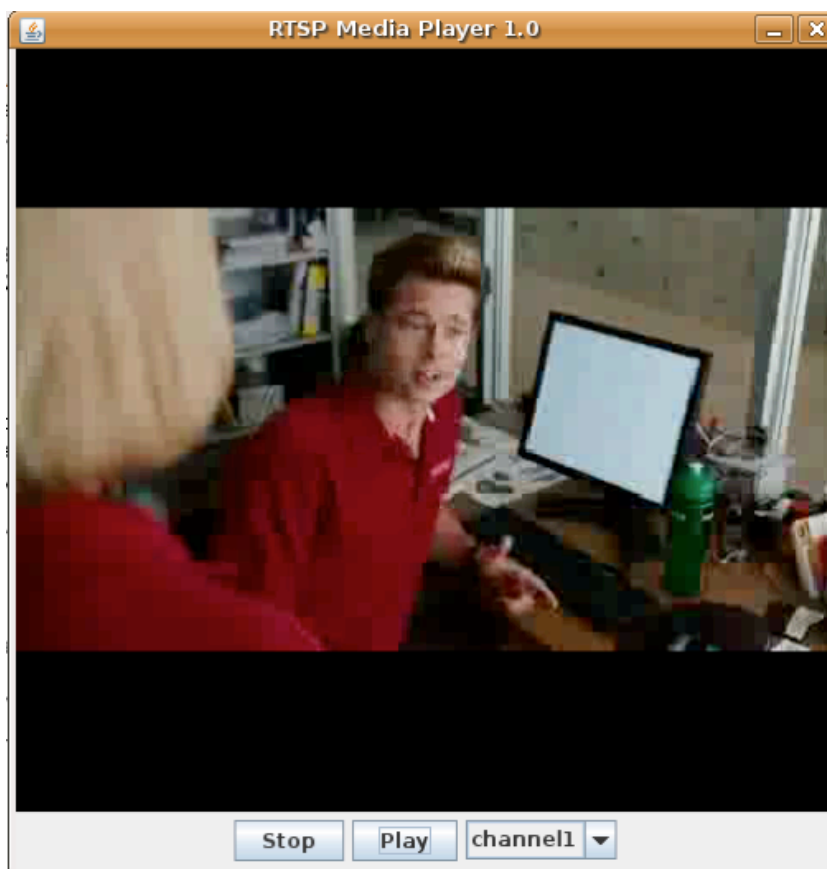


FIGURA 8.20: FUNCIÓN DE DIAGNÓSTICO EN EL CLIENTE SIP

Lo siguiente es acceder al servicio de Video Shopping propiamente tal. Para esto se debe abrir el portal web y registrarse con el usuario que se creó (en este caso *srivas*). Luego, en el cliente SIP abrir la viñeta *Menu* y seleccionar la opción *Register*. Esto registra al cliente en la plataforma de Mobicents y es posible corroborarlo en el terminal del servidor JSLEE.

Luego, navegar en el portal web y presionar en cualquier película el botón "*Watch Trailer*". Esto genera una llamada al cliente SIP, la cual se debe aceptar. Con la llamada en curso, presionar el botón en el cliente para realizar una nueva llamada (Teléfono Rojo). Esto automáticamente abre una ventana similar a la de la herramienta de diagnostico y se comienza a reproducir el video solicitado.

8.4.5 INTERCONEXIÓN CON IMS

En esta sección se muestran las modificaciones necesarias para lograr que los clientes IMS puedan acceder al servicio de Video Shopping. Esta sección supone que se haya realizado correctamente todos los pasos en este capítulo. En particular se requiere que esté instalado IMS, un cliente IMS, los servidores JSLEE y MediaServer de Mobicents y que el servicio de Video Shopping este apropiadamente configurado.

Una vez corroborado que se dispone de los elementos necesarios, el primer paso es agregar un usuario en el portal web. Como se detalla en secciones anteriores, para esto se debe modificar el archivo *import.sql*.

```
$ cd ms-1.0.2.GA/servers/media/examples/converged-demo
$ gedit web/jar/src/main/resources/import.sql
```

En este archivo editar otro de los usuarios de manera de incorporar una dirección SIP del dominio de IMS (*ims.uchile.cl*). A modo de ejemplo, se muestra el siguiente usuario:

```
INSERT INTO USERS
(USERID, DTYPE, FIRSTNAME, LASTNAME, ADDRESS1, ADDRESS2, CITY, STATE, ZIP, EMAIL, PHONE
, CREDITCARDTYPE, CC_NUM, CC_MONTH, CC_YEAR, USERNAME, PASSWORD) VALUES
(3, 'customer', 'Sebastian', 'Rivas', '4 Privet Drive', 'Cupboard under the
Stairs', 'QSDPAGD', 'SD', 24101, 'serivas@ing.uchile.cl', 'sip:srivas@ims.uchile.c
l', 1, '1979279217775911', 03, 2012, 'srivasims', 'srivas')
```

Luego, es necesario volver a compilar y levantar el servicio:

```
$ export JBOSS_HOME=/home/srivas/ms-1.0.2.GA/jboss-4.2.3.GA
$ export JAVA_HOME=/usr/lib/jvm/java-6-sun/
$ mvn clean install
$ ant deploy-all
```

A continuación se deben levantar las componentes de IMS y abrir el cliente UCT IMS Client según se describe en secciones anteriores.

El registro del cliente es ahora sobre IMS y no en la plataforma de Mobicents, luego cuando el portal web genera la llamada SIP, IMS debe la capturarla y la re-enviarla al cliente.

Por otro lado, al ingresar al portal web, esta vez se debe hacer el *Login* con el nuevo usuario con dirección SIP en el dominio de IMS, en este caso el cliente es *srivasims*.

Al igual que la prueba sobre el servicio instalado de manera individual, se debe navegar en las películas y presionar el botón “*Watch Trailer*”. Si todo se instaló correctamente, entonces la llamada debería llegar al cliente IMS y al contestarla se debería escuchar un anuncio TTS (Text to Speech) que solicita aceptar el envío del video. Basta presionar el botón para generar una llamada y automáticamente se abre la interfaz de visualización de videos.