



Universidad Carlos III de Madrid

Escuela Politécnica Superior

Grado en Ingeniería Telemática

*Servicio IPTV para dar soporte a docencia
basado en SIP y P2P*

Trabajo Fin de Grado

Autor: Sara Jiménez Carbajal

Tutor: Ivan Vidal Fernández

Junio de 2013

Trabajo Fin de Grado

Servicio IPTV para dar soporte a docencia basado en SIP y P2P

Autor

Sara Jiménez Carbajal

Tutor

Dr. Iván Vidal Fernández

Director

Dr. Iván Vidal Fernández

Realizado el acto de defensa y lectura del Trabajo Fin de Grado el día de Julio de 2013 en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, el tribunal:

PRESIDENTE:

SECRETARIO:

VOCAL:

acuerdan otorgarle la calificación de:

CALIFICACIÓN:

Leganés, a de Julio de 2013

A mi familia, por estar siempre a mi lado

Agradecimientos

Quería dar las gracias a mis padres ya que sin ellos no estaría donde estoy, por su apoyo durante estos años en los buenos y malos momentos.

No quiero olvidarme de mis abuelos, tíos, primos y resto de mi familia que me han apoyado siempre en cada una de mis decisiones.

Dar las gracias a mi tutor Iván, por su ayuda, su tiempo, sus consejos y su paciencia conmigo. Ha sido bastante fácil trabajar con él.

Mencionar a mis 3 pilares fundamentales durante estos años de carrera, Miriam, Ricardo y Javi. Gracias por todos nuestros momentos juntos, algunos mejores que otros, pero siempre mirando adelante y saliendo fortalecidos de cada uno de ellos. Todavía nos quedan muchos momentos juntos.

No quiero olvidarme tampoco del resto de compañeros que he conocido a lo largo de estos años.

Por último, pero no por ello menos importante, dar las gracias a mis amigos (los de Madrid y los de Calzada del Coto) que, aunque cuando les explicas lo que estudias te miran con cara de bicho raro, saben estar a tu lado, apoyarte y sacarte la mejor de tus sonrisas en cada uno de los momentos que pasas juntos a ellos. Gracias chic@s.

*Tu tiempo es limitado, de modo que no lo malgastes
viviendo la vida de otra persona. No quedes atrapado en el dogma
que es vivir como otros piensan que deberías vivir.*

*No dejes que el ruido de las opiniones de otros acallen tu propia
voz interior. Y, lo que es más importante, ten el coraje para hacer
lo que te dicen tu corazón y tu intuición. Ellos ya saben de algún
modo en qué quieres convertirte realmente. Todo lo demás es secundario.*

Steve Jobs (Discurso en un acto de graduación en Standford, 12 de Junio de 2005)

*En la lucha por la supervivencia, las especies que sobreviven
no son las especies más fuertes, ni las más inteligentes,
sino aquellas que se adaptan mejor a los cambios.*

Charles Darwin, El origen de las especies

Resumen

En los últimos años, la demanda de contenido multimedia por parte de los usuarios a través de Internet ha aumentado considerablemente debido a que la mayoría de las personas disponen o bien de dispositivos móviles como pueden ser un *smartphone*, una *tablet* o un ordenador portátil; o dispositivos fijos como es el caso de una televisión o un ordenador de sobremesa, en definitiva, dispositivos capaces de reproducir cualquier contenido multimedia, ya sea vídeo bajo demanda (como son el caso de Youtube, Netflix), IPTV (*IP Television*), o similares. Además otro factor decisivo en la demanda de contenido multimedia ha sido la aparición de nuevas tecnologías de acceso de banda ancha, tanto fija como móviles. Por estos motivos van apareciendo nuevas formas de transmitir dicho contenido a través de la red.

Además, debido a que tenemos la facilidad de comunicarnos con personas que están a una cierta distancia o de aprender nuevos conocimientos a través de los nuevos avances tecnológicos, principalmente a través de Internet, se abre un nuevo campo de posibilidades de comunicación dentro de las empresas o sin ir más lejos en el ámbito educativo. Es en éste último, debido a la gran demanda de usuarios que realizan sus estudios de manera semipresencial u *online* en dónde se está ampliando el uso del vídeo bajo demanda (VoD) de contenido educativo o incluso la transmisión de dichas clases a través de internet (IPTV).

El objetivo principal de este Trabajo Fin de Grado ha consistido en la implementación de un sistema IPTV para dar soporte a la actividad docente. El desarrollo, que ha sido realizado mediante herramientas open source, permite la retransmisión en directo de una sesión docente, a través de Internet, desde el equipo del profesor a un número potencialmente elevado de alumnos. El desarrollo del servicio se ha estructurado en dos planos, uno de señalización y uno de datos. En ambos planos, la implementación realizada utiliza protocolos de Internet definidos por el IETF (Internet Engineering Task Force) [IETF] que es una organización cuya misión es hacer que Internet funcione mejor mediante la elaboración de documentos técnicos de alta calidad que influyen en la forma de diseñar, utilizar y gestionar Internet.

Palabras clave: Internet Protocol Television (IPTV), Session Initiation Protocol (SIP), Session Description Protocol (SDP), Peer-to-Peer (P2P).

Abstract

In recent years, the demand of multimedia content by users through the Internet has increased considerably due to the most people have or mobile devices such as a smartphone, a tablet or a laptop, or fixed devices such as TV or a desktop computer, in short, any device able to play multimedia content whether video on demand (such as the case of YouTube, Netflix), IPTV, or similar. Also another factor in the demand for multimedia content has been the emergence of new technologies for broadband access, fixed and mobile. For these reasons new ways of transmitting the content through the network are emerging.

Also, due to we have the facility to communicate with people who are at a certain distance or to learn new knowledge through new technological advances, mainly through the Internet, opens a new field of possibilities for communication within companies or in the educational scope. It is in this last, due to the high demand of users who carry out their studies in a blended or online in where the use of video on demand (VoD) of educational content or the transmission of these classes over the Internet (IPTV) are being expanded.

The main goal of this Bachelor Thesis has been the implementation of an IPTV system to support teaching. The development, which has been performed using open source tools, allows the retransmission of a live teaching session, through the Internet, from the teacher's computer to a potentially high number of students. The development of the service is structured in two planes, one signaling and one data. In both plans, the implementation uses Internet protocols on defined by IETF (Internet Engineering Task Force) which is an organization whose mission is to make the Internet work better by producing high quality, relevant technical documents that influence the way people design, use, and manage the Internet.

Keywords: Internet Protocol Television (IPTV), Session Initiation Protocol (SIP), Session Description Protocol (SDP), Peer-to-Peer (P2P).

Índice General

Agradecimientos	VII
Resumen	XI
Abstract	XIII
Índice General	XV
Lista de Figuras	XIX
I Introducción	1
1. Introducción	3
1.1. Motivación	3
1.2. Objetivos	4
1.3. Estructura de la memoria	4
II Estado del Arte	7
2. SIP	9
2.1. Introducción	9
2.2. Identificación de usuarios	10
2.3. Elementos de SIP	10
2.3.1. SIP UA	10
2.3.2. Servidores SIP	11
2.3.2.1. Servidores Proxy	11
2.3.2.2. Servidores Redirectores	12
2.3.2.3. Servidores Registradores	13
2.3.3. B2BUA	14
2.4. Mensajes de SIP	15
2.4.1. Respuestas SIP	15
2.4.1.1. 100 TRYING	16
2.4.1.2. 180 RINGING	16
2.4.1.3. 200 OK	16

6.3.1.3. Liberación de la sesión	54
6.4. IPTV-AS	54
6.4.1. Procedimientos en el IPTV-AS	55
6.4.1.1. Procesado de las peticiones SIP	55
6.4.1.2. Procesado de las respuestas SIP	58
7. Escenario de pruebas	61
7.1. Introducción	61
7.2. Establecimiento de sesión con más de 2 estudiantes	62
7.3. Liberación de la sesión con más de 2 estudiantes	67
7.4. Tiempos medios de conexión de los usuarios	70
IV Conclusiones y futuras implementaciones	73
8. Conclusiones y futuras implementaciones	75
8.1. Conclusiones	75
8.2. Futuras implementaciones	76
V Anexos	77
A. Planificación de tareas y presupuesto	79
A.1. Descomposición en tareas	79
A.2. Planificación con el diagrama de fases de ejecución detallado	83
A.3. Recursos	85
A.4. Presupuesto de Proyecto	85
B. Manual de usuario	87
C. Solicitudes SIP	91
C.1. CANCEL	91
C.2. OPTIONS	91
C.3. SUBSCRIBE	91
C.4. NOTIFY	92
C.5. PRACK	92
C.6. UPDATE	92
C.7. REFER	93
C.8. MESSAGE	93
C.9. PUBLISH	93
C.10.INFO	93
D. Campos SDP	95
D.1. Session name	95
D.2. Information	95

D.3. Email	95
D.4. Phone number	95
D.5. URI	95
D.6. Bandwidth	96
D.7. Time	96
D.8. Repeat time	96
D.9. Time zones	96
D.10. Encryption keys	96
E. Establecimiento de sesión SIP	97
F. Configuración del VLC para el envío del primer contenido multimedia	99
Glosario	105
Bibliografía	107

Lista de Figuras

2.1. User Agent	11
2.2. Forking proxy [Joh09]	12
2.3. Servidor de redirección [Joh09]	13
2.4. Servidor de registro	14
2.5. B2BUA	14
2.6. Cabeceras SIP de un INVITE y OK	18
2.7. Sesión SIP [JR02a]	19
3.1. Carga SDP [Joh09]	22
3.2. Modelo Oferta/respuesta [JR02b][JR02a]	24
4.1. Arquitectura de un sistema IPTV típico[Gre09]	27
4.2. Arquitectura TISPAN funcional del sistema IPTV basado en IMS[ETS11a]	28
4.3. Red centralizada	30
4.4. Red Descentralizada	30
4.5. Red Semicentralizada	30
5.1. Plano de control	36
5.2. Plano de datos	36
5.3. Establecimiento de sesiones con varios usuarios	38
5.4. Liberación de sesión con varios usuarios	39
6.1. Sistema a implementar	43
6.2. Arquitectura JAIN SIP [PO03]	45
6.3. Esquema lógico	46
6.4. Vista del Applet para la conexión	47
6.5. Inicialización del Sistema	48
6.6. Generación del INVITE	48
6.7. Procesamiento del OK y generación del ACK	49
6.8. Recepción y reproducción del vídeo	50
6.9. Vista del Applet para la desconexión	50
6.10. Desconexión	51
6.11. Procesado del OK del BYE	51
6.12. Reenvío del vídeo al nuevo cliente	51

6.13. Procesado del INVITE en el servidor	53
6.14. Recepción del ACK y envío del vídeo	54
6.15. Procesado del BYE y generación del OK	54
6.16. Procesado del INVITE en el IPTV-AS	56
6.17. Generación del INVITE para el servidor	56
6.18. Generación del ACK para el servidor	57
6.19. Procesado del BYE en el IPTV-AS	58
6.20. Reconexión de clientes	58
6.21. Procesado del 200 OK en el IPTV-AS	59
6.22. Procesado del 200 OK correspondiente al BYE	59
7.1. Escenario de pruebas	61
7.2. Propiedades de los elementos del sistema	62
7.3. Establecimiento de sesión	63
7.4. Establecimiento de sesión de Estudiante 3	63
7.5. Transmisión del vídeo	66
7.6. Liberación sesión de un cliente/servidor al mismo tiempo	67
7.7. Liberación de sesión	68
7.8. Liberación de sesión de Estudiante 1	68
7.9. Tiempos de conexión de los clientes	70
A.1. Diagrama de Gantt con la planificación del proyecto resumida	83
A.2. Diagrama de Gantt con la planificación detallada del proyecto	84
B.1. Applet lanzado al estudiante	89
F.1. Interfaz de la aplicación VLC	99
F.2. Abrir archivo	100
F.3. Abrir medio	101
F.4. Medio seleccionado	101
F.5. Salida de emisión - Fuente	102
F.6. Salida de emisión - Configuración de destinos	102
F.7. Configuración de destinos - UDP	103
F.8. Emisión del contenido multimedia	103
F.9. Reproducción del contenido multimedia en el applet	104

Parte I

Introducción

Capítulo 1

Introducción

En este primer capítulo se exponen la motivación y los objetivos de este Trabajo Fin de Grado y se comenta la estructura del documento junto con una breve descripción de las diferentes partes de las que consta.

1.1. Motivación

Hoy en día, podemos definir el concepto de multimedia como una plataforma digital donde se integran diversos componentes ya sean de tipo texto, gráficos, imagen fija, imagen en movimiento (vídeo - animaciones) o audio (música, voz, sonidos, etc.). Integrándolos todos o alguno de ellos en un entorno llamativo para el usuario, le permitirá interactuar o no con ellos ya sea a través de los medios de reproducción más clásicos, como son los ordenadores o la televisión, o a través de los medios de reproducción más modernos como puedan ser tablets, smartphones o reproductores multimedia tipo *Ipod*, entre otros.

Gracias a Internet, se ha abierto una puerta muy importante para la distribución de información multimedia a través de la red. De este modo, el usuario puede acceder a un amplio rango de aplicaciones y servicios de gran interés, como lo son por ejemplo las aplicaciones de videollamada y videoconferencia, la televisión digital, los servicios de televisión a la carta o el vídeo bajo demanda.

La aparición en el mercado de nuevos terminales de usuario mejorados, así como el despliegue de nuevas tecnologías de banda ancha (fijas y móviles), posibilitan la apropiada provisión de estos servicios, los cuales, según estudios recientes¹ serán predominantes en Internet en un futuro cercano.

La tecnología IPTV (*Internet Protocol Television*) en el ámbito educativo permite a los usuarios acceder al contenido docente desde cualquier dispositivo conectado a la red y además el estudiante ya no es un mero espectador sino que se convierte en partícipe de su formación. Esto es muy importante ya que es un

¹Según un reporte elaborado por Cisco [cis13] se estima que en el año 2017 entre el 80 % y el 90 % del tráfico de Internet sea de vídeo.

medio interesante para hacer llegar la educación a personas que no pueden asistir a clases presenciales y que quieren recibir formación desde cualquier punto del planeta.

En este contexto, y como veremos a continuación, el objetivo principal de este Trabajo Fin de Grado, consiste en desarrollar un sistema IPTV para dar soporte a la actividad docente. En particular, nuestro desarrollo posibilita la retransmisión en directo de las clases desde el equipo del profesor a un número potencialmente elevado de estudiantes, los cuales disponen de conexión a internet. El diseño del sistema de este trabajo se basa en protocolos estándares de Internet, definidos por el IETF[[IET](#)]. Además en la fase de desarrollo se han utilizado herramientas y librerías open source.

1.2. Objetivos

El objetivo de este proyecto es desarrollar un sistema IPTV para retransmitir las clases impartidas por un profesor a un número potencialmente elevado de usuarios. Si profundizamos un poco más en este objetivo, podemos destacar los siguientes puntos a tener en cuenta en el desarrollo del sistema:

- Desarrollo de un servicio IPTV para dar soporte a la actividad docente.
- El servicio permitirá la retransmisión en directo de una sesión docente desde el equipo del profesor a un número potencialmente elevado de alumnos, que disponen de una conexión a Internet.
- El diseño se fundamentará en protocolos y estándares actuales de Internet para el acceso al contenido multimedia.
- En particular, el plano de control se implementará utilizando el protocolo SIP (Session Initiation Protocol) definido por el IETF en la RFC 3261[[JR02a](#)].
- Para el desarrollo del aplicativo se utilizarán herramientas open source.
- Desarrollo flexible y fácilmente extensible, de modo que nuevas funcionalidades puedan ser añadidas.
- El servicio estará disponible via web, de modo que el estudiante pueda acceder fácilmente al mismo a través de Aula Global, que es la plataforma de gestión de contenido docente de la Universidad Carlos III de Madrid (dicha plataforma esta basada en moodle[[moo](#)]).
- Estudio de los tiempos de conexión de cada usuario en el sistema.

1.3. Estructura de la memoria

Este documento se divide en varias partes, y estas a su vez se dividen en distintos capítulos. A continuación se describe brevemente el contenido de cada parte y de cada capítulo:

1. **Primera parte: Introducción.** Contiene la motivación y los objetivos del proyecto además de la estructura de la memoria. Tiene el siguiente capítulo:
 - Capítulo 1. Introducción
2. **Segunda parte: Estado del arte.** Se analiza el protocolo de señalización utilizado en el sistema, SIP; se comentan varios aspectos del protocolo SDP y por último se comentarán las distintas alternativas disponibles para la distribución multimedia en IPTV. Esta parte contiene los siguientes capítulos:
 - Capítulo 2. SIP
 - Capítulo 3. SDP
 - Capítulo 4. IPTV
3. **Tercera parte: Descripción del trabajo realizado.** En esta parte se identifican los requisitos del sistema y se describe la implementación de cada uno de los elementos. Además se presenta un escenario de la implementación propuesta. Se divide en los siguientes capítulos:
 - Capítulo 5. Identificación de los requisitos del sistema
 - Capítulo 6. Implementación de los elementos del sistema
 - Capítulo 7. Escenario de pruebas
4. **Cuarta parte: Conclusiones y futuras implementaciones.** En esta parte se explican las conclusiones obtenidas del trabajo y futuras implementaciones para ampliar el proyecto actual. Sólo contiene un capítulo:
 - Capítulo 8. Conclusiones y futuras implementaciones
5. **Quinta parte: Anexos.** En esta última parte se amplía la información de algunas partes del Trabajo Fin de Grado:
 - Anexo A: Planificación de tareas y presupuesto. En este anexo se describen las tareas del proyecto y los costes asociados.
 - Anexo B: Manual de usuario. Se describen los pasos a seguir para ejecutar la implementación.
 - Anexo C: Solicitudes SIP. Se describen las solicitudes que no han sido mencionadas en el capítulo 2.
 - Anexo D: Campos SDP. Se comentan los campos de la carga SDP para completar la información que aparece en el capítulo 3.
 - Anexo E: Establecimiento de sesión SIP. Se describen los pasos a seguir para el establecimiento de sesión.
 - Anexo F: Configuración del VLC para el envío del primer contenido multimedia. En el desarrollo realizado, utilizamos la herramienta VideoLAN player (VLC) para transmitir el vídeo desde el equipo del profesor. En este anexo se recogen los pasos a seguir dentro de la aplicación para enviar el vídeo hasta el cliente.

Parte II

Estado del Arte

Capítulo 2

SIP

2.1. Introducción

SIP [JR02a][Joh09] es un protocolo de señalización desarrollado para establecer, modificar y terminar sesiones multimedia. Como muchos de los protocolos de internet ha sido desarrollado por el IETF (Internet Engineering Task Force) [IET]. Este protocolo permite el establecimiento de sesiones multimedia entre dos usuarios finales. Las principales funciones de SIP son las siguientes:

- Localización de un usuario final.
- Contacto con un usuario final para determinar su deseo de establecer una sesión.
- Intercambio de información a través de la red para permitir el establecimiento de la sesión.
- Modificación de las sesiones existentes.
- Cancelación/terminación de las sesiones multimedia existentes.

También permite la solicitud y entrega de información de presencia (estado *online/off-line* e información de localización como si se estuviera suscrito a un grupo). Estas funciones incluyen:

- Publicar y cargar información de presencia.
- Solicitar la entrega de información de presencia.
- Presencia y otros eventos de notificación.
- Transporte de mensajes instantáneos.

Actualmente, SIP ha sido adoptado por el 3GPP (*3rd Generation Partnership Project*) y el ETSI (*European Telecommunications Standards Institute*) como protocolo de señalización para redes 3G/4G.

2.2. Identificación de usuarios

SIP identifica a los usuarios mediante URIs SIP (*Uniform Resource Identifier*) y tienen el siguiente formato:

$$sip: \langle userinfo \rangle \langle hostport \rangle \langle parameters \rangle$$

Los campos correspondientes a la URI SIP se describen a continuación:

- **userinfo**: contiene información sobre el usuario, seguida una letra "@".
- **hostport**: contiene un nombre de dominio o una dirección IP. Adicionalmente puede incluir un número de puerto.
- **parameters**: contiene parámetros adicionales. Se indican tras el carácter ";".

Un ejemplo de URI SIP es:

$$sip:Alice.Smith@domain.com$$

2.3. Elementos de SIP

SIP es un protocolo cliente-servidor y los elementos que se usan son los siguientes:

- agentes de usuario (UAC/UAS).
- servidores (*proxy, register y redirect*).
- adicionalmente está el B2BUA (*Back to Back User Agent*) que es la unión de un UAC y un UAS.

A continuación veremos en detalle alguno de ellos.

2.3.1. SIP UA

Un UA (User Agent) es un dispositivo final capaz de establecer una sesión multimedia con otro UA (figura 2.1). Ejemplos de estos dispositivos son los teléfonos móviles, ordenadores, TV, etc.

Además un UA puede actuar como cliente (UAC) generando peticiones y procesando respuestas o como servidor (UAS) generando respuestas y procesando peticiones.

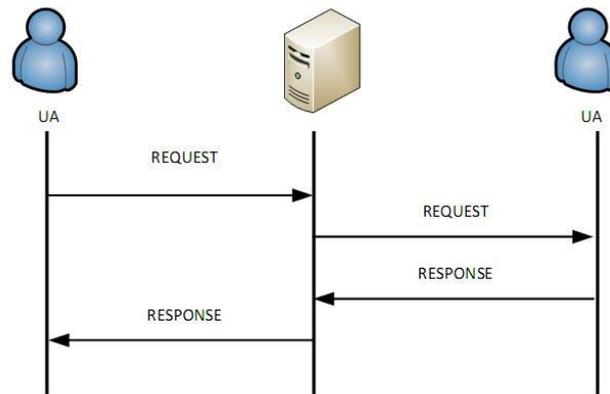


Figura 2.1: User Agent

2.3.2. Servidores SIP

Son aplicaciones que aceptan peticiones SIP y dan respuestas a ellas soportando TCP y UDP como protocolos de transporte. Además un servidor SIP no debe de ser confundido con un UA ya que es una entidad distinta.

Podemos distinguir varios tipos de servidores SIP.

2.3.2.1. Servidores Proxy

Estos servidores actúan encaminando solicitudes hacia los UASs o encaminando respuestas hacia los UACs. Durante el envío de una solicitud pueden existir varios proxies desde el UAC origen al UAS destino y la respuesta a dicha solicitud sigue exactamente el mismo camino (a la inversa) que siguió la solicitud original, es decir, que atraviesa los mismos proxies. Además, un proxy puede tomar decisiones sobre el encaminamiento de una solicitud, o incluso puede llegar a modificarla antes de reenviarla al siguiente paso. Normalmente, tienen acceso a una base de datos o a un servidor de localización para determinar el siguiente salto.

En definitiva, para entender de manera más sencilla el funcionamiento de un servidor proxy se puede interpretar como un router para mensajes SIP, pero con capacidad para interpretar, validar, redirigir, duplicar o eliminar dichos mensajes.

En el ejemplo mostrado a continuación (figura 2.2) podemos ver como un servidor proxy a partir de una solicitud, es capaz de encaminarla a más de un destino. El servidor proxy que se comporta de esta forma es denominado *forking proxy*.

Los servidores proxy se diferencian de un UA principalmente en:

- Un servidor proxy no genera peticiones SIP, solo responde a peticiones procedente de un UA.
- Un servidor proxy no tiene la capacidad de intercambiar información multimedia (ej. RTP).

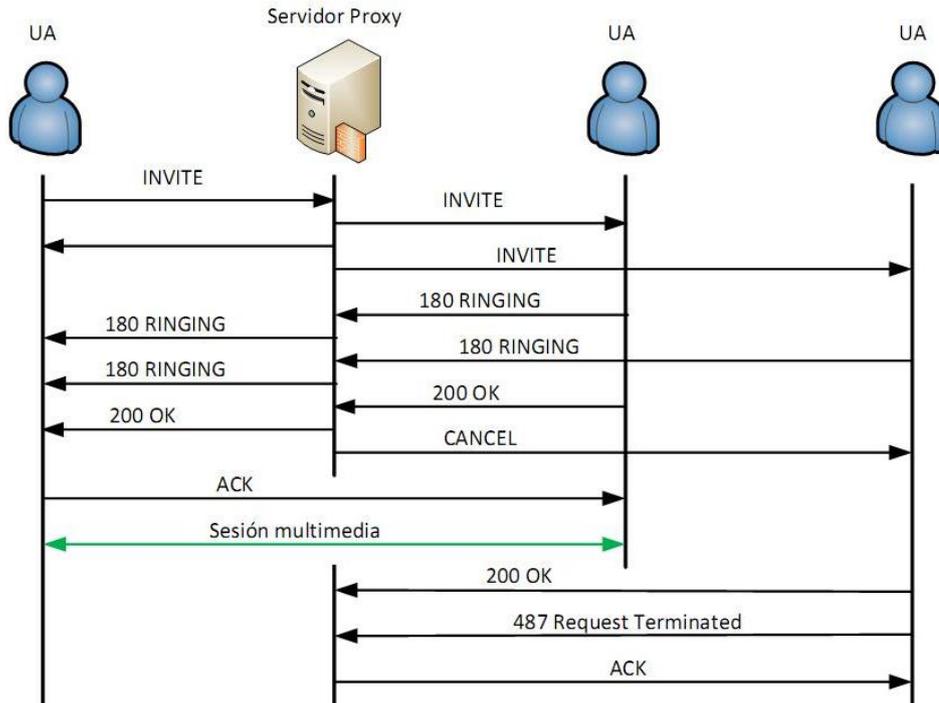


Figura 2.2: Forking proxy [Joh09]

- Un servidor proxy no procesa el contenido de los mensajes, tan solo los campos de las cabeceras SIP.

Pueden ser de dos tipos:

- **Servidor stateless** (sin memoria): es aquel que procesa cada petición o respuesta SIP basándose solamente en el contenido del mensaje. Una vez que el mensaje ha sido parseado, procesado y reenviado o respondido, el servidor no almacena ningún tipo de información sobre la petición o respuesta. Un proxy stateless nunca retransmite un mensaje y no utiliza ninguna clase de temporizador SIP.
- **Servidor stateful** (con memoria): mantiene información sobre las peticiones y respuestas recibidas y usa esa información en el procesamiento de futuras peticiones y respuestas. De este modo un proxy stateful inicia un timer cuando recibe una petición y la reenvía. Si no recibe ningún tipo de respuesta a esa petición antes de que el temporizador expire, el proxy vuelve a retransmitir la petición. También un proxy stateful puede requerir la autenticación del usuario.

2.3.2.2. Servidores Redirectores

Se trata de entidades que responden a peticiones pero no tienen capacidad de reenvío. Al igual que los proxies, los servidores de redirección utilizan una base de

datos o un servicio de localización para saber la ubicación del usuario.

En la figura 2.3 se muestra un ejemplo de señalización de llamada que incluye un servidor redirector. Como podemos observar, el servidor redirector envía un mensaje 3xx al UAC para comunicarle cual es la ubicación correcta del UAS y así poder establecer una sesión directamente con él.

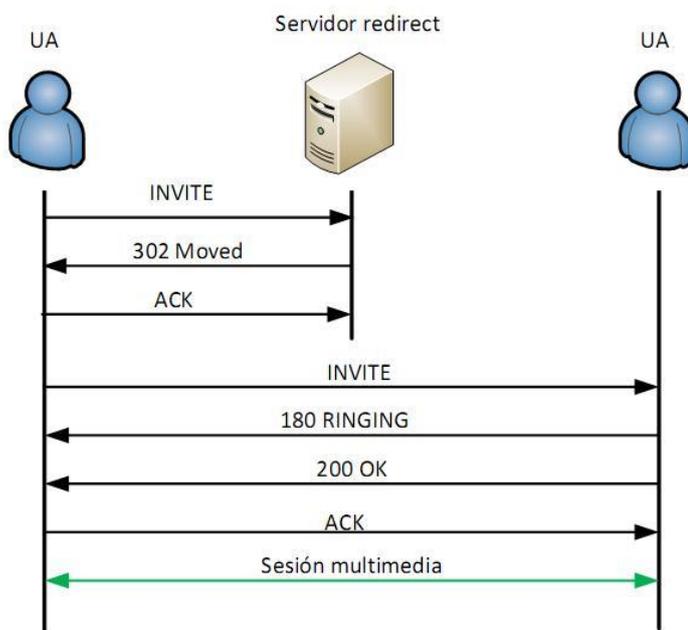


Figura 2.3: Servidor de redirección [Joh09]

2.3.2.3. Servidores Registradores

Son aquellos servidores que gestionan la información de localización de usuarios en un determinado dominio, de modo que esta información esté disponible para proxies y redirectores permitiendo establecer sesiones multimedia con un usuario independientemente de su localización.

Cada usuario tiene una dirección lógica (usuario@dominio) que no cambia respecto de la ubicación física del usuario. La dirección física (denominada "dirección de contacto") es dependiente del lugar en donde el usuario está conectado. Cuando un usuario inicializa su terminal (por ejemplo conectando su teléfono o abriendo su software de telefonía SIP), el UA SIP que está en dicho terminal, envía una petición con el método *REGISTER* a un Servidor de Registro informando a qué dirección física debe asociarse la dirección lógica del usuario y el servidor de registro realiza entonces dicha asociación (denominada binding). Podemos ver un ejemplo en la figura 2.4

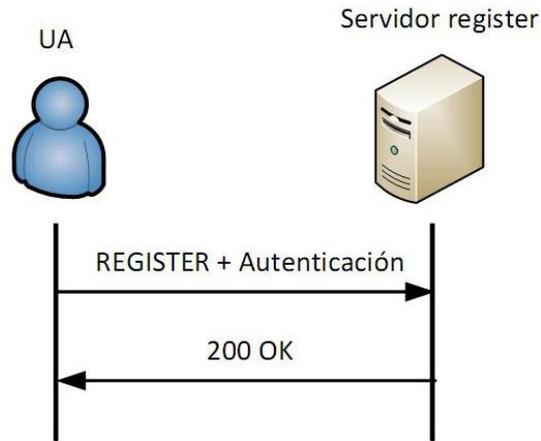


Figura 2.4: Servidor de registro

2.3.3. B2BUA

Un Back to back user agent es un tipo de SIP UA que recibe peticiones SIP y las procesa, puede generar solicitudes hacia otros servidores y además de esto, puede modificar la carga de los mensajes o contactar con varios usuarios tras recibir una petición *INVITE*, entre otras acciones. Un ejemplo es la figura 2.5

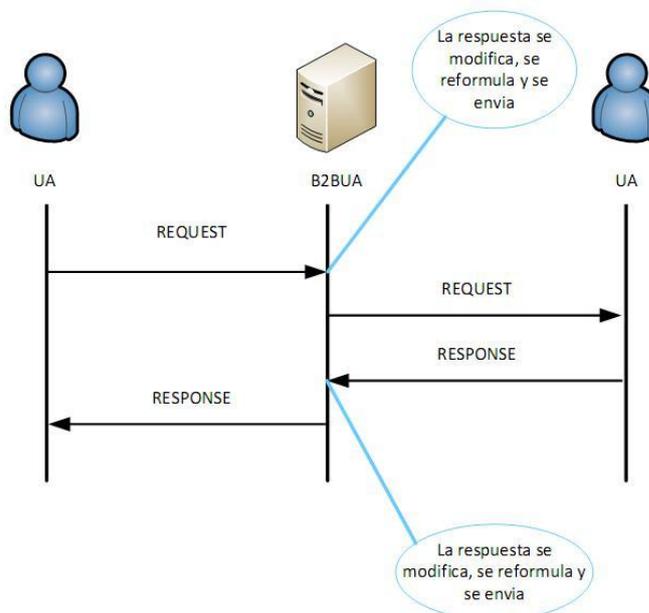


Figura 2.5: B2BUA

2.4. Mensajes de SIP

SIP es un protocolo solicitud-respuesta, estilo a HTTP (*Hypertext Transfer Protocol*) y SMTP (*Simple Mail Transfer Protocol*) donde podemos diferenciar dos tipos de mensajes.

2.4.1. Respuestas SIP

Son mensajes generados por un UAS o un Servidor SIP para responder a peticiones generadas por un UAC. Una respuesta puede contener cabeceras adicionales de información necesaria para el UAC o pueden ser simples asentimientos a peticiones anteriores. Hay 6 tipos de respuestas SIP que son mostrados en la tabla 2.1.

Clase	Descripción	Acción
1xx	Información	Indica el estado de la sesión antes de que se complete. Es la primera información que se recibe y es provisional.
2xx	Éxito	La solicitud se ha completado con éxito.
3xx	Redirección	El servidor ha cambiado de localización. El cliente ha de dirigir la solicitud a otro servidor.
4xx	Error en el cliente	La solicitud ha fracasado por fallo en el cliente. El cliente puede intentar hacer otra solicitud.
5xx	Error en el servidor	La solicitud ha fracasado por fallo en el servidor. La solicitud puede ser redirigida a otro servidor.
6xx	Fallo global	La solicitud no debe volverse a mandar a ningún servidor.

Tabla 2.1: Respuestas SIP

Si un código de respuesta SIP no es entendido por un UAC, debe ser interpretado por la clase de la respuesta. Todas las respuestas SIP se caracterizan por su *Status-Line*, que es la primera línea de cualquier respuesta, según la terminología indicada en la RFC 3261[JR02a] cuyo formato es el siguiente:

<SIP-Version> <Status-Code> <Reason-Phrase>

Los campos correspondientes a la Status-Line se describen a continuación:

- **SIP-Version:** indica la versión de SIP (2.0 en la RFC 3261).
- **Status-Code:** es un código de resultado entero de 3 dígitos como los de la tabla anterior.
- **Reason-Phrase:** descripción del Status-Code.

A continuación veremos algunas de las respuestas SIP.

2.4.1.1. 100 TRYING

Es una respuesta provisional *hop-by-hop* que indica que la solicitud ha sido recibida por el servidor del siguiente salto, nunca es reenviada y no puede contener cuerpo del mensaje. Esta respuesta puede ser generada por cualquier servidor proxy o por un agente de usuario.

2.4.1.2. 180 RINGING

Se envía para indicar que se ha recibido un *INVITE* y que el UA destinatario está intentando alertar al usuario. Con esta respuesta también se le indica al iniciador del *INVITE* que el teléfono del llamado está sonando.

2.4.1.3. 200 OK

Esta respuesta sirve para indicar que una solicitud se ha cursado con éxito. Cuando se envía para aceptar una solicitud *INVITE*, el cuerpo del mensaje incluye típicamente una carga SDP.

2.4.2. Solicitudes SIP

Se trata de un tipo de peticiones o mensajes que desde que son enviadas al usuario destino se espera que una acción se lleve a cabo. Los mensajes *INVITE*, *REGISTER*, *BYE*, *ACK*, *CANCEL* y *OPTION* son las seis solicitudes SIP originales. A posteriori y debido a la aparición de las extensiones de SIP, definidas en otras RFCs, se introducen 7 nuevos mensajes: *REFER*, *SUBSCRIBE*, *NOTIFY*, *PUBLISH*, *MESSAGE*, *UPDATE*, *INFO* y *PRACK*. Todas las solicitudes SIP se caracterizan por su *Request-Line*, que es la primera línea de cualquier solicitud, según la terminología indicada en la RFC 3261[JR02a], cuyo formato es el siguiente:

<Method> <Request-URI> <SIP-Version>

Los campos correspondientes a la Request-Line se describen a continuación:

- **Method:** indica el propósito de la solicitud.
- **Request-URI:** es una URI SIP que indica el usuario destinatario de la solicitud.
- **SIP-Version:** indica la versión de SIP (2.0 en la RFC 3261).

A continuación veremos algunas de las solicitudes mencionadas anteriormente.

2.4.2.1. INVITE

Se usa para establecer sesiones multimedia entre diferentes agentes de usuario. El cuerpo del *INVITE* típicamente contiene información que permite describir la sesión multimedia que el usuario iniciador de la petición *INVITE* desea establecer. Dicha información se describe de acuerdo al formato definido por el IETF en SDP[[MH06](#)], el cual se describe en el capítulo 3.

2.4.2.2. ACK

Se envía para asentir respuestas finales a una petición *INVITE*. Respuestas finales a otro tipo de respuestas finales nunca son asentidas. Este tipo de respuestas son definidas por los códigos 2xx, 3xx, 4xx, 5xx o 6xx.

2.4.2.3. BYE

Esta petición se usa para terminar una sesión multimedia previamente establecida. Es enviada sólo por UAs participantes en la sesión, ya que es un mensaje *end-to-end*.

2.4.2.4. REGISTER

Es utilizada por un UA para informar a un servidor registrador de la localización actual del usuario. Una solicitud *REGISTER* puede contener un cuerpo en el mensaje aunque su uso no está definido en el standard.

En el Anexo C se pueden encontrar más detalles acerca de las solicitudes SIP restantes.

2.5. Campos de las cabeceras SIP

Las solicitudes y respuestas SIP tienen, entre otras, las siguientes cabeceras:

- **To:** especifica el receptor de la solicitud. Siempre incluye una URI.

To: Alice <sip:alice@biloxi.com>;tag=26854

- **From:** indica el originario de una solicitud. Siempre incluye una URI.

From: Bob <sip:bob@atlanta.com>;tag=45s6df4

- **CSeq:** contiene un número de secuencia y un nombre de método. Con este campo podemos ordenar solicitudes, diferenciar entre nuevas solicitudes y retransmisiones y asociar respuestas con solicitudes.

Cseq: 1 INVITE

- **Call-ID:** identifica de forma unívoca una solicitud *INVITE*, así como todos los registros realizados por un cliente.

Call-ID: f81d4fae-7dec-11d0-a765-00a0c91e6bf6@biloxi.com

- **Max-Forwards:** se usa para evitar bucles de encaminamiento. Cada proxy que gestiona una solicitud decremента su valor en uno y si llega a cero la solicitud se descarta.

Max-Forward: 20

- **Via:** mantiene un registro de los proxies atravesados por una solicitud. La respuesta atraviesa los mismos proxies que la solicitud pero en orden inverso.

Via: SIP/2.0/UDP erlang.bell-telephone.com:5060;branch=z9hG4bK87asdk7

- **Contact:** contiene una URI con la información del contacto originario del mensaje.

Contact: sip:bob.biloxi@telephone.com

- **Expires:** indica el intervalo de tiempo durante el cual la solicitud o el contenido del mensaje SIP son válidos.

Expires: 30

Unos ejemplos de mensajes SIP serían los siguientes:

<pre>INVITE sip:picard@wcom.com SIP/2.0 Via: SIP/2.0/UDP host.wcom.com:5060 From: Alan Johnston <sip:alan.johnston@wcom.com> To: Jean Luc Picard <sip:picard@wcom.com> Call-ID: 314159@host.wcom.com CSeq: 1 INVITE Contact: sip:alan.johnston@wcom.com Content-Type: application/sdp Content-Length: 124</pre>	<pre>SIP/2.0 200 OK Via: SIP/2.0/UDP host.wcom.com From: Alan Johnston <sip:alan.johnston@wcom.com> To: Jean Luc Picard <sip:picard@wcom.com> Call-ID: 314159@host.wcom.com CSeq: 1 INVITE Contact: sip:picard@wcom.com Content-Type: application/sdp Content-Length: 107</pre>
---	---

Figura 2.6: Cabeceras SIP de un INVITE y OK

2.6. Establecimiento y terminación de sesión

En la figura 2.7 podemos observar un establecimiento de sesión SIP completo extraído de la RFC 3261 [JR02a].

Vemos como el usuario "Alice" pretende establecer una sesión con el usuario "Bob" y para ello genera la petición *INVITE* pertinente. Mientras la petición llega a su destino, los servidores por los que atraviesa van respondiendo con la respuesta temporal *100 TRYING*. Al llegar el *INVITE* a destino y hasta que la respuesta *200 OK* es generada, se envía un *180 RINGING* para indicar que el *INVITE* ha sido recibido. A continuación el usuario destino envía la respuesta *200 OK* al usuario iniciador de la sesión y posteriormente éste asiente la respuesta con un *ACK*.

Una vez la sesión está establecida, se empieza a transmitir el contenido multimedia hasta que uno de los dos usuarios decide abandonar la sesión. En este caso, es el usuario "Bob" es el que envía una petición *BYE* mostrando su interés por abandonar la sesión. Cuando el usuario "Alice" recibe esta solicitud, responde a ella con un *200 OK* para finalizar la sesión entre ambos usuarios.

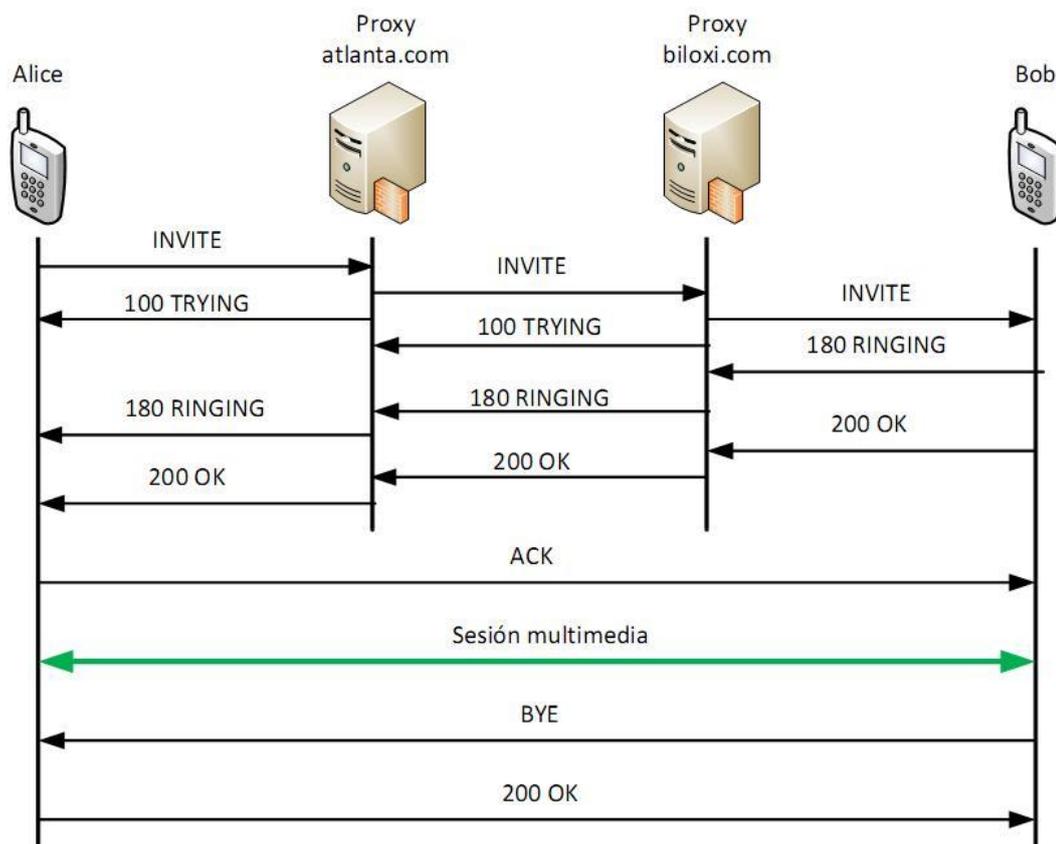


Figura 2.7: Sesión SIP [JR02a]

Capítulo 3

SDP

SDP [MH06] es un protocolo que surge para describir sesiones multimedia. Tiene una sintaxis muy clara y con el modelo de oferta/respuesta se permite la negociación de la descripción de la sesión (lo veremos en detalle en el apartado 3.1).

La carga SDP puede transportarse en el cuerpo del mensaje SIP. Dicha carga se describe con una serie de atributos, cada uno en una línea, llamados **campos** cuyos nombres dentro del mensaje están abreviados a una letra minúscula indicando el tipo de campo que representan seguido de un = y a continuación su valor.

La información almacenada en la carga SDP se puede estructurar de la siguiente forma:

- **Descripción de la sesión:** contiene información referente a la sesión y al creador de la misma.
- **Descripción de tiempos:** contiene información sobre cuanto tiempo la sesión estará activa y las repeticiones.
- **Descripción de componentes multimedia:** contiene información sobre el protocolo de transporte a utilizar y los formatos multimedia soportados.

Un ejemplo de carga SDP podría ser el mostrado en la figura 3.1.

Los campos SDP definidos en la RFC 4566 [MH06] son los mostrados en la tabla 3.1. Prestaremos especial atención a los que se han utilizado en el desarrollo de este proyecto. El resto de campos aparecen detallados en el Anexo D.

- **version:** contiene el número de versión de SDP. Dado que la versión actual del protocolo es 0, el valor ha de ser $v=0$.
- **origin:** almacena la información a cerca de quién originó la sesión y sobre los identificadores de ella. Se utiliza para identificar de manera unívoca la sesión. El formato de este campo es el siguiente:

```
o=<username> <session-id> <version> <network-type> <address-type>  
<address>
```

```

v=0
o=johnston 2890844526 2890844526 IN IP4 43.32.1.5
s=IETF Update
i=This broadcast will cover the latest from the IETF
u=http://www.sipstation.com
e=Alan Johnston alan@avaya.com
p=+1-314-555-3333 (Daytime Only)
c=IN IP4 225.45.3.56/236
b=CT:144
t=2877631875 2879633673
m=audio 49172 RTP/AVP 0
a=rtpmap:0 PCMU/8000
m=video 23422 RTP/AVP 31
a=rtpmap:31 H261/90000

```

Figura 3.1: Carga SDP [Joh09]

Campo	Nombre	Obligatorio/Opcional
v=	número de versión del protocolo	obligatorio
o=	origen e identificador de sesión	obligatorio
s=	nombre de la sesión	obligatorio
i=	información de la sesión	opcional
u=	URI de descripción	opcional
e=	email	opcional
p=	número telefónico	opcional
c=	información de conexión	obligatorio
b=	información de ancho de banda	opcional
t=	tiempo durante el cual la sesión estará activa	obligatorio
r=	número de repeticiones	opcional
z=	ajustes de zona horaria	opcional
k=	clave de cifrado	opcional
a=	atributos de sesión	opcional
m=	información multimedia	obligatorio

Tabla 3.1: Campos SDP

- username: nombre del usuario que genera la sesión.
 - session-id: cadena numérica que constituye un identificador único.
 - version: número que identifica la versión de la sesión.
 - network-type: identifica, en formato texto, el tipo de red que se está utilizando. Su valor es siempre *IN* para Internet.
 - address-type: define, en formato texto, el tipo de dirección. Los valores que toma son *IP4* para direcciones IPv4 e *IP6* para direcciones IPv6.
 - address: dirección desde la cual se creó la sesión.
- **connection:** contiene información sobre la conexión. El formato de este campo es el siguiente:

$$c = \langle network\text{-}type \rangle \langle address\text{-}type \rangle \langle connection\text{-}address \rangle$$

- network-type: es definido como *IN* para Internet.

- **address-type:** toma los valores *IP4* para direcciones IPv4 e *IP6* para direcciones IPv6.
 - **connection-address:** para sesiones unicast es la dirección IP y el puerto al que se envía el contenido multimedia. Para sesiones multicast es la dirección IP del grupo multicast y el puerto de destino del flujo de información multicast.
- **media:** almacena información sobre el tipo de sesión multimedia que se va a transmitir. El formato de este campo es el siguiente:

$$m=\langle media \rangle \langle port \rangle \langle transport \rangle \langle format-list \rangle$$

- **media:** puede tener varios valores en función del flujo de datos a transmitir: *audio*, *video*, *text*, *application*, *message*, *image* o *control*.
- **port:** contiene el número de puerto por el que se envía el flujo multimedia.
- **transport:** indica el protocolo de transporte sobre el que se va a enviar el flujo multimedia.
- **format-list:** muestra todos los formatos en los que el flujo multimedia puede ser codificado. Es una cadena de caracteres numéricos separados por un espacio en el que cada número representa un formato de codificación.

Un posible ejemplo puede ser:

$$m=audio\ 49430\ RTP/AVP\ 0\ 6\ 8\ 99$$

- **attributes:** son el principal medio para extender SDP. Los atributos pueden ser definidos para ser utilizados tanto a nivel de sesión como a nivel multimedia o ambos.
 - Nivel de sesión: el atributo aparece antes de la primera línea de *media* en la carga SDP y este atributo se aplica a todas las líneas de *media* por debajo de ella.
 - Nivel multimedia: el atributo aparece a continuación de una línea *media* y sólo aplica a este flujo de *media* en particular.

Si el atributo es definido de los dos tipos, el atributo de nivel multimedia prevalece sobre el atributo de nivel de sesión.

El campo de atributos puede presentarse en dos formatos:

- Atributos de propiedad: que se muestran de la forma $a=\langle attribute \rangle$ para indicar si se soporta una determinada propiedad.
- Atributos de valor: que extienden a los anteriores de la forma $a=\langle attribute \rangle:\langle valor \rangle$. No sólo especifican que se encuentra una determinada propiedad sino que además se incluye el valor correspondiente.

3.1. Modelo oferta/respuesta

Este modelo fue definido en la RFC 3264 [JR02b]. Lo que se pretende es acordar la descripción de una sesión entre dos usuarios. Consiste en lo siguiente: un usuario (ofertante) genera una carga SDP (oferta SDP) con sus preferencias multimedia y se la envía al otro usuario. El otro usuario puede generar una respuesta o puede rechazar la oferta. Si decide generarla, con la carga SDP recibida, genera la suya propia (respuesta SDP) y se la envía al usuario ofertante. La respuesta contendrá el mismo número de líneas $m=$ que se recibieron en la oferta.

Un ejemplo de este modelo podría ser el mostrado en la figura 3.2. Vemos que en la oferta SDP se propone establecer una sesión multimedia con un componente de audio y dos componentes de vídeo. Para cada componente, el ofertante indica el listado de formatos o codecs soportados. A modo de ejemplo, para el primer componente de audio, el generador de la oferta SDP soporta el codec PCMU. En la respuesta SDP podemos observar como dos de ellos se aceptan (uno de vídeo y uno de audio).

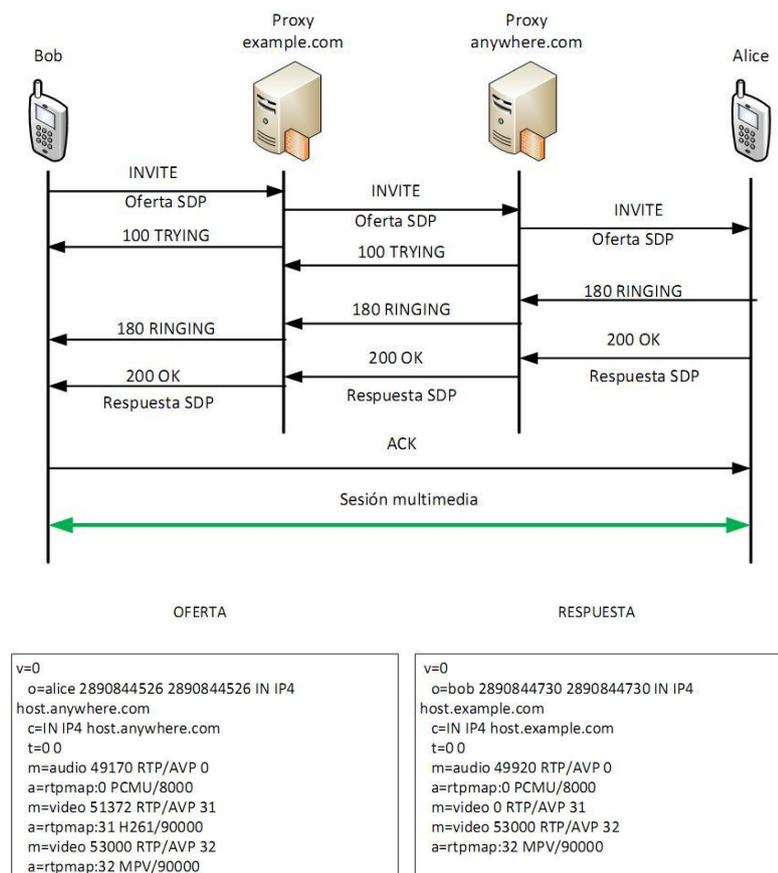


Figura 3.2: Modelo Oferta/respuesta [JR02b][JR02a]

Capítulo 4

IPTV

4.1. Multicast IP

Por multicast o multidifusión entendemos que es el envío de la misma información a múltiples destinos interesados en ella.

Antes del envío de la información, deben establecerse una serie de parámetros:

- Los equipos que quieran recibir la información deberán constituir un grupo multicast con una dirección IP multicast asociada. Para llevar a cabo esta suscripción los participantes hacen uso del protocolo IGMP (*Internet Group Management Protocol*) [W. 97][B. 02]. Dicho protocolo establece un mecanismo para el intercambio y actualización de información sobre la pertenencia de los equipos a un grupo multicast. El diálogo se desarrolla entre los equipos miembros del grupo y los routers de acceso con capacidad multicast, que les proporcionan a dichos equipos conectividad IP posibilitando la conexión y desconexión de los equipos a los grupos multicast. Los equipos informan acerca de su pertenencia a un grupo, y los routers multicast sondean periódicamente el estado de los equipos suscritos.
- El que va a enviar la información indicará como dirección destino la del grupo multicast.
- Los routers de la red se encargarán de hacer llegar a cada miembro del grupo multicast una copia de la información. Para ello, los routers hablan un protocolo de encaminamiento multicast, como pueden ser PIM-SM (*Protocol Independent Multicast-Sparse Mode*)[BF06], MOSPF (*Multicast OSPF*)[Moy94], CBT (*Core-Based Trees*)[Bal97], PIM-DM (*Protocol Independent Multicast-Dense Mode*)[AA05] y DVMRP (*Distance Vector Multicast Routing Protocol*)[DW88]. PIM-SM es la alternativa más usada en la actualidad[Sav08].

Para la identificación de grupos multicast IP se utilizan direcciones de clase D (224.0.0.0 a 239.255.255.255) [Z. 01]. Para cada dirección multicast puede existir

cero o más destinatarios activos, que recibirán una copia del datagrama enviado a dicha dirección.

Independientemente de su naturaleza dinámica, los grupos multicast pueden ser permanentes o transitorios.

- Grupo permanente: tiene asociada una dirección IP multicast fija, independientemente del número de miembros que tenga el grupo. Ejemplos:
 - 224.0.0.1 - Todos los sistemas de la subred
 - 224.0.0.2 - Todos los routers de la subred
 - 224.0.0.4 - Todos los routers DVMRP de la subred
 - 224.0.0.5 - Todos los routers OSPF (*Open Shortest Path First*) del dominio

Se puede hacer uso del servicio DNS para localizar la dirección asociada a un grupo multicast permanente (dominio mcast.net) y lo mismo para sus resoluciones inversas (224.in-addr.arpa.)

- Grupo transitorio: se crea dinámicamente (en el momento que se lanza una aplicación multicast), y dejará de existir cuando deje de tener miembros activos.

El grupo multicast se considera siempre de receptores, no de emisores. Esto implica que el originante de un datagrama multicast no tiene por qué ser miembro del grupo al que envía dicho datagrama, solo ha de conocer la dirección del grupo multicast.

La mayoría de los protocolos de aplicaciones existentes que usan multicast lo hacen sobre UDP (*User Datagram Protocol*) [Pos80]. Otras aplicaciones, sobre todo aquellas que tienen que transmitir contenidos multimedia, lo hacen usando UDP o el protocolo RTP (*Real Time Protocol*) [H. 03]; además del protocolo RTCP (*RTP Control Protocol*) [H. 03] para proporcionar información sobre la calidad de la distribución de los datos.

4.2. Sistemas IPTV

En este apartado comentaremos diferentes soluciones de sistemas IPTV.

4.2.1. Sistemas Walled-Garden

Para grandes sistemas de IPTV, a menudo hay una jerarquía en las instalaciones construidas para entregar las señales de vídeo a través de la extensión del territorio. Están formados por un *super head end* (SHE) (o dos, para evitar pérdidas) que puede servir a millones de usuarios mediante el procesamiento de los canales de vídeo que son comunes a todos los abonados del servicio. Un *video serving office* (VSO) que se

encuentra en cada región como requisito para manejar la programación y los canales específicos para una sola ciudad o área geográfica local. Un *central office* (CO) o *remote terminal* (RT) puede servir como un *local end office* (LEO) que contiene el equipo necesario para entregar la programación a los clientes en un área local [Gre09]. Un ejemplo de este sistema podemos verlo en la figura 4.1.

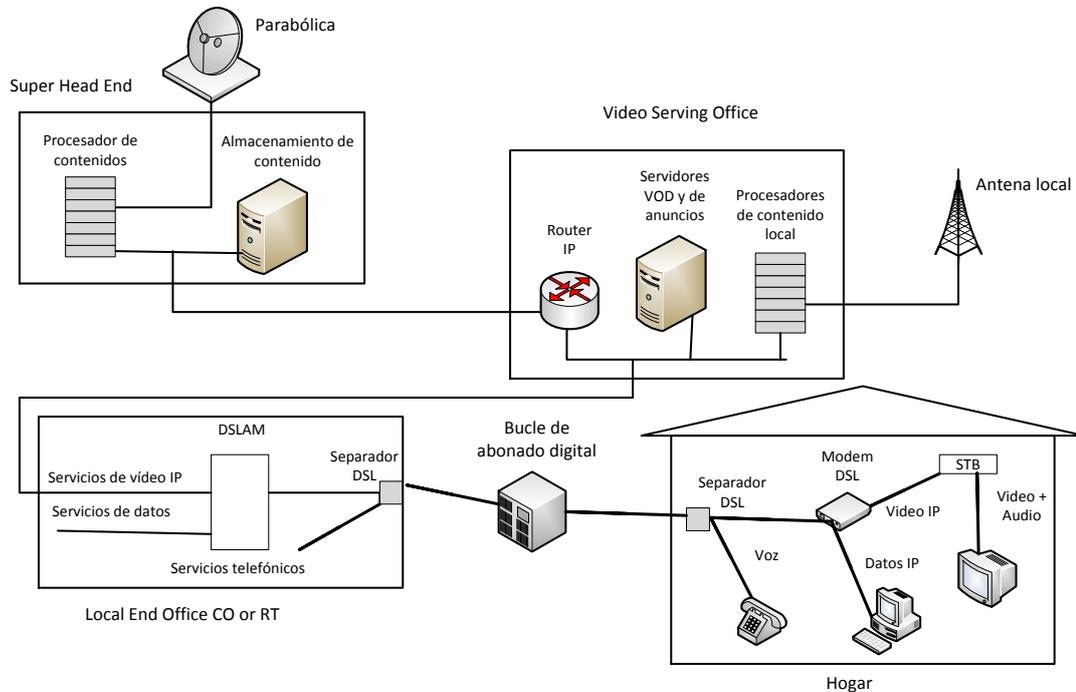


Figura 4.1: Arquitectura de un sistema IPTV típico [Gre09]

4.2.2. Arquitectura de TISPAN

Hay varios organismos de estandarización internacionales trabajando en IPTV. Estas actividades de estandarización han resultado en distintas especificaciones técnicas sobre la arquitectura y funciones de un sistema IPTV. Una de las actividades más relevante en este campo es la realizada por el grupo TISPAN (*Telecommunications and Internet converged Services and Protocols for Advanced Networking*) de ETSI (*the European Telecommunications Standards Institute*) cuyo principal objetivo es la especificación de un nuevo modelo de red de próxima generación o NGN (*Next Generation Networks*).

En la versión 2 de especificaciones de la red NGN, TISPAN añade IPTV como un nuevo servicio en la arquitectura de la red. Para aprovechar los esfuerzos de estandarización realizados en IMS (*IP Multimedia Subsystem*) en esta versión se hace referencia a la arquitectura de servicios IPTV basada en IMS. Esta arquitectura implementa un conjunto de requisitos a nivel de servicio (figura 4.2). Por otro

lado, también se define un sistema IPTV integrado en las NGN, con el objetivo de fomentar los requisitos antes mencionados y permitir la integración de soluciones IPTV existentes definidas por otras organizaciones, tales como DVB (*Digital Video Broadcasting*), ATIS IIF (*ATIS IPTV Interoperability Forum*), or ITU (*International Telecommunication Union*). [EM11]

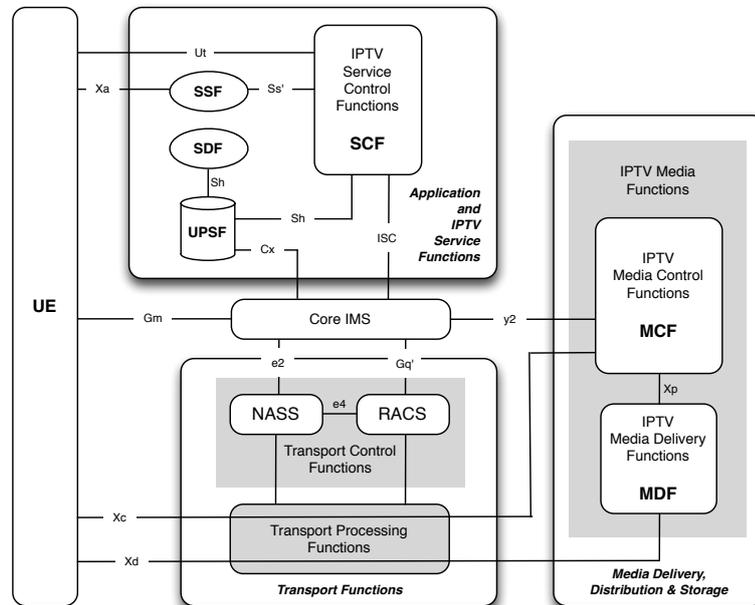


Figura 4.2: Arquitectura TISPAN funcional del sistema IPTV basado en IMS[ETS11a]

En la versión 2, TISPAN define 3 servicios principales para el sistema IPTV basado en IMS: *Broadcast TV*, *Content on Demand (CoD)* y *Network-Personal Video Recorder (N-PVR)*. Posteriormente, en la versión 3 se añaden un amplio conjunto de servicios de valor añadido. Algunos ejemplos son la televisión interactiva, la interacción entre los usuarios y los canales personalizados. La especificación ETSI TISPAN para los sistemas IPTV basados en IMS y para los sistemas IPTV integrados en las NGN se pueden ver en [ETS11a] y [ETS11b] respectivamente.

4.2.3. P2PTV

P2PTV es una técnica de transmisión y difusión de contenidos audiovisuales a través de la red de Internet usando la arquitectura de los sistemas P2P, donde unos equipos se conectan a otros para recibir los flujos de información de vídeo y audio.

Una red P2P se define como una red de computadoras en la que todos se comportan como iguales, es decir, actúan al mismo tiempo como clientes y servidores respecto a los demás nodos de la red. Esto permite el intercambio directo de información entre los equipos conectados.

Las redes *peer-to-peer* aprovechan, administran y optimizan el uso del ancho de

banda de los demás usuarios de la red por medio de la conectividad entre los mismos y obtienen así más rendimiento en las conexiones y en las transferencias que con algunos métodos centralizados convencionales, donde una cantidad relativamente pequeña de servidores provee el total del ancho de banda y recursos compartidos para un servicio o aplicación.

Dichas redes son útiles para diversos fines y a menudo son usadas para compartir archivos de cualquier tipo (audio, video, ...) pero también suelen usarse en telefonía VoIP para hacer más eficiente la transmisión de datos en tiempo real.

Las redes P2P deberían de tener las siguientes características [RS05]:

- **Escalabilidad:** las redes P2P tienen una cobertura mundial con cientos de millones de usuarios potenciales. Lo deseable es que cuantos más nodos estén conectados a una red P2P, mejor será su funcionamiento.
- **Robustez:** en caso de haber fallos en algunos nodos de la red, no tiene repercusiones en el resto de nodos.
- **Descentralización:** por definición todos los nodos son iguales, por lo que no existen nodos con funciones especiales y por consiguiente no hay ningún nodo que sea imprescindible para el correcto funcionamiento de la red.
- **Distribución de costes entre los usuarios :** todos los usuarios comparten recursos a cambio de recursos.
- **Anonimato:** en estas redes, salvo un pequeño número de usuarios, nadie da a conocer su verdadera identidad.
- **Seguridad:** es una de las características más deseables de las redes P2P pero no está muy desarrollada ya que existen nodos maliciosos, contenido infectado, espionaje de las comunicaciones entre nodos,...

Si nos fijamos en su grado de centralización, podemos realizar la siguiente clasificación de las redes P2P [RS05]:

- **Redes centralizadas:** todos los intercambios de información se realizan a través de un único servidor que sirve de punto de enlace entre el resto de nodos distribuyendo y reenviando la información y las peticiones de los clientes. Estas redes tienen limitada la privacidad de los usuarios y carecen de escalabilidad. Ejemplo en la figura 4.3.
- **Redes descentralizadas o puras:** son las más comunes ya que todas las comunicaciones se realizan directamente de usuario a usuario con ayuda de un nodo (que es otro usuario) que permite enlazar esas comunicaciones. No requieren de un gestionamiento central de ningún tipo. Ejemplo en la figura 4.4.
- **Redes semicentralizadas o híbridas:** en este tipo de red se puede observar un servidor central que sirve como *hub* y administra los recursos de banda

ancha, enrutamientos y comunicación entre los nodos pero sin saber la identidad de cada uno y sin almacenar información alguna, sólo actúa de coordinador. Los nodos son los responsables de almacenar la información. Ejemplo en la figura 4.5.

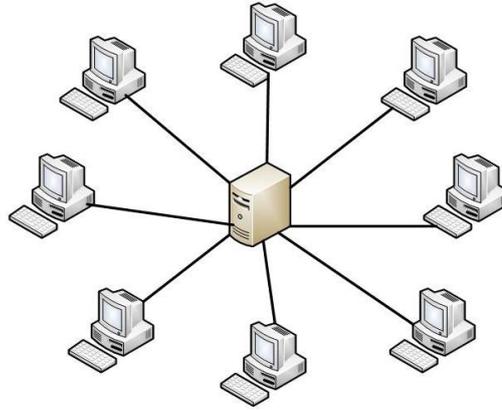


Figura 4.3: Red centralizada

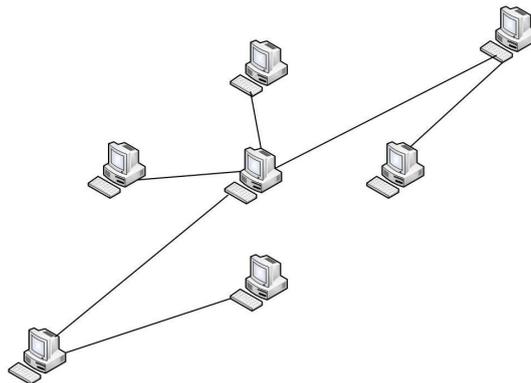


Figura 4.4: Red Descentralizada

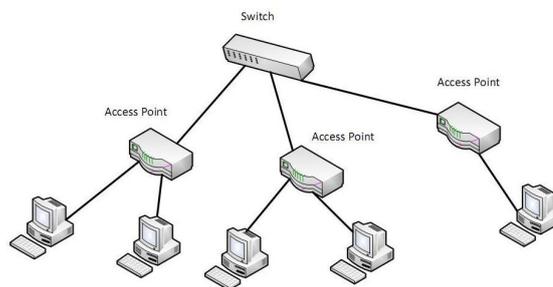


Figura 4.5: Red Semicentralizada

Entre las distintas alternativas de P2PTV, en este Trabajo Fin de Grado nos centraremos en la arquitectura propuesta en el artículo "*A peer-to-peer IPTV service architecture for the IP multimedia subsystem*" [AF09]. En este artículo se describe

una arquitectura de servicio IPTV para IMS, que en lugar de utilizar multicast utiliza un enfoque P2P. Su diseño se divide en dos planos, uno de señalización basado en SIP y uno de datos.

En el capítulo 5 se proporcionarán más datos sobre esta solución y sobre su particularización al entorno de este Trabajo Fin de Grado, el soporte a la labor docente.

Parte III

Descripción del trabajo realizado

Capítulo 5

Identificación de los requisitos del sistema

5.1. Introducción

Este Trabajo Fin de Grado está basado en el artículo "*A peer-to-peer IPTV service architecture for the IP multimedia subsystem*"[AF09] en el que se propone una arquitectura de servicio IPTV para el IMS (*IP Multimedia Subsystem*) que combina un plano de control (en el que se utiliza el protocolo de señalización SIP) y un plano de datos distribuido (siguiendo un enfoque P2P), que depende de los dispositivos del IMS o de los *peers* ubicados en los clientes locales para actuar como nodos de reenvío de la transmisión.

Nuestra implementación será una simplificación de lo expuesto en este artículo, y nos centraremos en el desarrollo de un plano de control utilizando el protocolo SIP y de un plano de datos distribuido siguiendo un enfoque P2P. Además esta implementación estará formada de tres componentes: cliente, IPTV-AS y profesor.

En la figura 5.1 se observa un ejemplo con 3 estudiantes en la que se muestran las interconexiones en el plano de control y en la figura 5.2 se observa el mismo ejemplo para el plano de datos.

El plano de control (5.1) para el establecimiento y liberación de la sesión estará basado en el protocolo SIP. La idea fundamental es que el Estudiante 1 y Estudiante 2 establezcan una sesión de streaming de video udp con el profesor a través del IPTV-AS, que es el elemento central de la señalización, y que a su vez ellos proporcionen el contenido recibido del profesor a los estudiantes que se les conecten. En primer lugar se producirá el establecimiento de sesión a través de SIP y una vez que la sesión ya esté establecida, se procederá a la difusión de contenidos a los diferentes usuarios del sistema. Posteriormente cualquier estudiante podrá abandonar la sesión sin que la difusión de contenidos afecte al resto de usuarios.

El plano de datos (figura 5.2) comprende el procesamiento y el envío del contenido multimedia siguiendo de un enfoque P2P a los usuarios del sistema. Para ello, una vez

establecida la sesión SIP entre el usuario y su servidor de contenidos, la difusión de éste será de manera directa entre ambos utilizando Sockets de JAVA para gestionar el envío y la recepción del mismo.

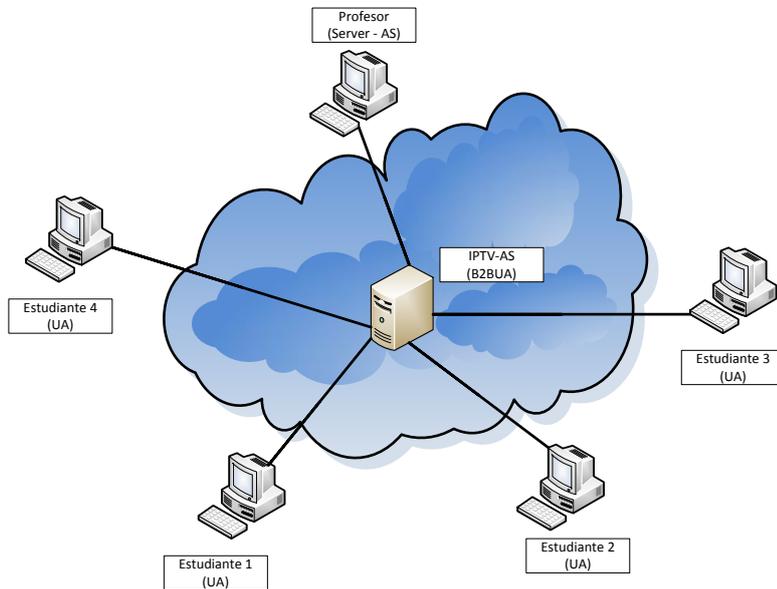


Figura 5.1: Plano de control

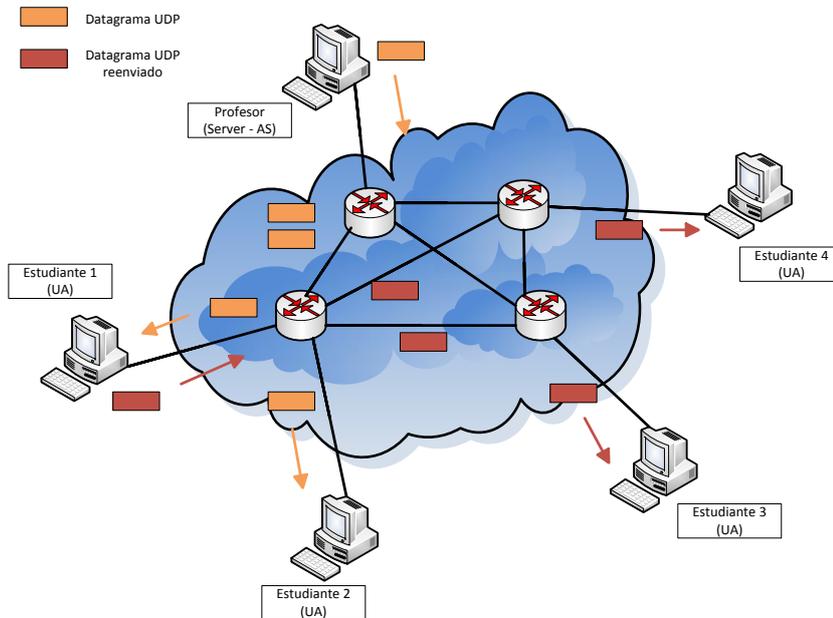


Figura 5.2: Plano de datos

5.2. Modelo de uso

Este es un servicio que la Universidad deberá ofrecer al profesor para poder impartir la actividad docente si así lo desea y para ello, tendrá que disponer del IPTV-AS que proporcione este servicio dentro de sus infraestructuras.

Para su utilización, el profesor contactará con el personal responsable del servicio IPTV y solicitará una retransmisión en directo. El equipo responsable del servicio le dará al profesor un enlace para que el alumno pueda conectarse al servicio durante la sesión. El profesor, a su vez, publicará el enlace que le han proporcionado en la plataforma docente de la Universidad (por ejemplo, Aula Global en la UC3M) para que los alumnos tenga acceso al servicio al inicio de la sesión.

5.3. Plano de control o señalización

Para esta parte tenemos identificadas varias tareas a realizar:

- Desarrollar un módulo cliente que se encargue de iniciar y terminar la sesión SIP. Este cliente debe ser accesible via web, de modo que pueda ser fácilmente integrable en Aula Global. Este módulo se ejecutará en el equipo del estudiante.
- Implementar un módulo servidor que sea el que envíe el contenido multimedia una vez que se tiene una sesión establecida con el cliente. Este módulo será utilizado en el equipo del profesor y de los estudiantes.
- Desarrollar un IPTV-AS que haga de intermediario para establecer y liberar la sesión SIP entre el cliente y el servidor.

Antes de continuar con la definición de los requisitos, tenemos que entender el marco en el que se encuentran los mismos. La idea principal es que tanto los clientes como el servidor interactúen con el IPTV-AS y por ello veremos un poco más a fondo su funcionamiento.

El IPTV-AS es un B2BUA de SIP, que está formado por un agente de usuario cliente (UAC) y un agente de usuario servidor (UAS) que se encuentran conectados mediante cierta lógica específica del servicio proporcionado por el B2BUA. El IPTV-AS sólo participa en la fase de establecimiento y liberación de la sesión. Una vez está establecida la sesión, el IPTV-AS no interviene en el envío de los datos multimedia, de eso se encargan ya el cliente y el servidor directamente.

En la figura 5.3 se muestra un ejemplo con los mensajes de señalización SIP intercambiados cuando tres estudiantes (Estudiante 1, Estudiante 2 y Estudiante 3 en la figura) acceden al servicio.

Los clientes (Estudiante 1, Estudiante 2 y Estudiante 3) y los servidores (Profesor y Estudiante 1) serán los encargados de generar las solicitudes y respuestas SIP.

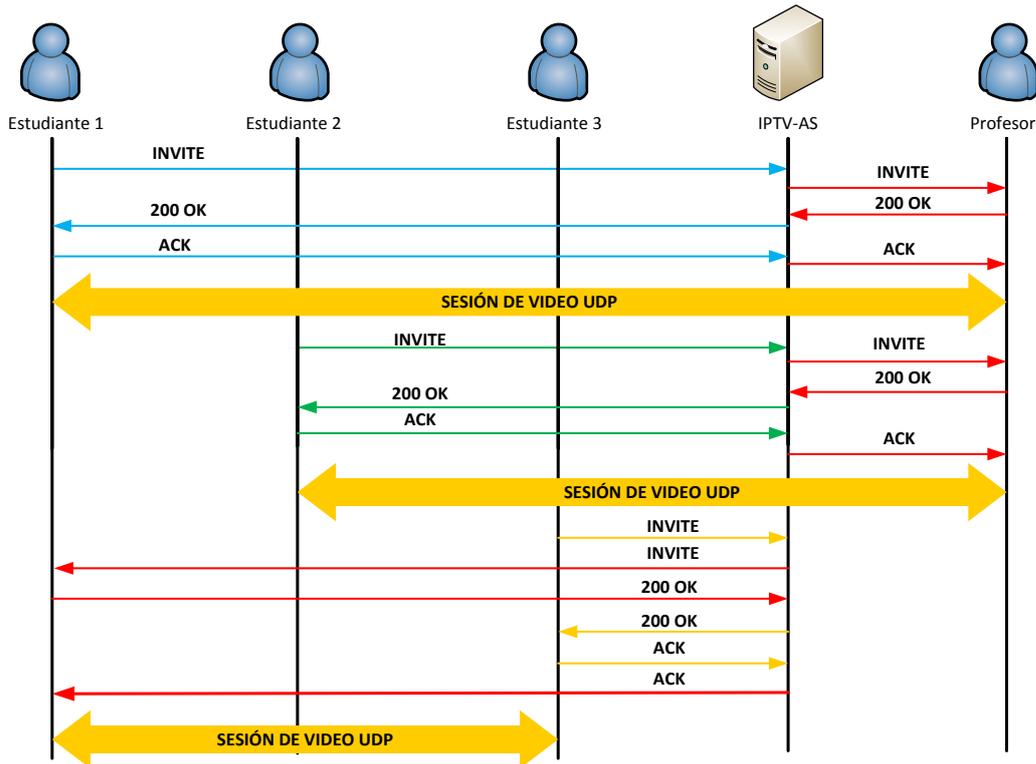


Figura 5.3: Establecimiento de sesiones con varios usuarios

Inicialmente, Estudiante 1 accede al servicio. De este modo, su módulo cliente genera una solicitud *INVITE* de SIP que es dirigida al IPTV-AS y tras comprobar que se trata del primer usuario conectado al sistema, redirige la solicitud al módulo servidor del profesor. Como resultado de esto, se establece una sesión multimedia entre los equipos Estudiante 1 y el profesor, de modo que el Estudiante 1 comienza a recibir el contenido multimedia correspondiente a la clase.

A continuación, un nuevo usuario (Estudiante 2) accede al sistema y al igual que el Estudiante 1 su módulo cliente genera una solicitud *INVITE* de SIP que es dirigida al IPTV-AS y tras comprobar que es el segundo usuario conectado al sistema, redirige la solicitud al módulo servidor del profesor. Como resultado de esto, se establece una sesión multimedia entre los equipos Estudiante 2 y el profesor, de modo que el Estudiante 2 comienza a recibir el contenido multimedia correspondiente a la clase.

Por último, accede al sistema otro usuario (Estudiante 3) que nuevamente genera una solicitud *INVITE* de SIP que es dirigida al IPTV-AS y tras comprobar que el profesor ya está proporcionando el contenido multimedia a dos usuarios, el IPTV-AS elegirá al usuario Estudiante 1 como servidor del contenido multimedia del usuario Estudiante 3 y por tanto redirige la solicitud al módulo servidor del Estudiante 1. Como resultado de esto, se establece una sesión multimedia entre los equipos Estudiante 3 y Estudiante 1, de modo que el Estudiante 3 comienza a recibir el contenido multimedia correspondiente a la clase.

En el Anexo E podemos encontrar información más detallada de las peticiones y respuestas SIP del establecimiento de sesión.

Otra parte importante del establecimiento de sesión es la negociación de la descripción de sesión. La sesión establecida consistirá en un componente de vídeo y los estudiantes indicarán en una oferta SDP la información de direccionamiento en la que desean recibir este vídeo. Asimismo, indican el listado de codecs de vídeo soportados. Esta negociación es la que se ha comentado en el capítulo 3, el modelo de oferta/respuesta de SDP.

En la figura 5.3 tendríamos 3 ofertas SDP, 1 por cada estudiante, y 3 respuestas SDP con las que responden los servidores, 2 del profesor y 1 del Estudiante 1.

Del mismo modo también se propone una liberación de la sesión en caso de que uno de los usuarios decida abandonar la sesión como podemos observar en la figura 5.4. Cuando un usuario decida abandonar la sesión establecida, habrá que tener en cuenta los siguientes problemas:

- Algunos usuarios pueden quedar huérfanos y habrá que reconectarlos.
- Tendremos servidores para los cuales habrá que modificar o terminar sus sesiones.

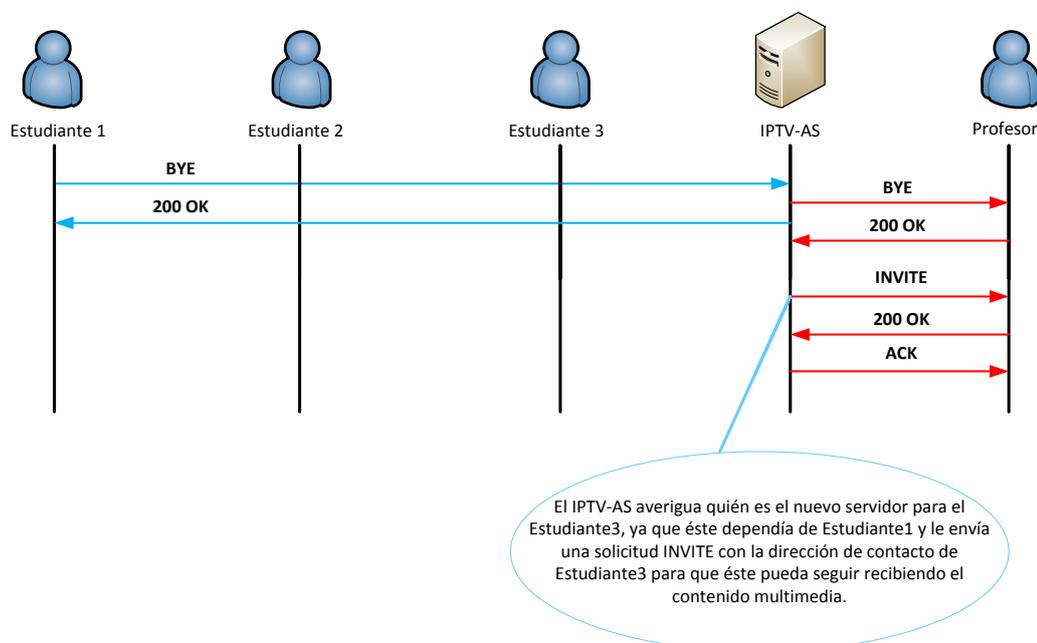


Figura 5.4: Liberación de sesión con varios usuarios

5.3.1. Estudiante/Cliente

El estudiante será quien se encargue de generar el inicio del establecimiento de la sesión y su posterior liberación. Esta implementación estará disponible para el estudiante a través de la web y se le mostrará un applet donde podrá visualizar el contenido multimedia recibido.

Para que comience el establecimiento de sesión entre el estudiante y el resto de componentes del sistema, se tendrá que pulsar sobre el botón *Conexión* en la parte inferior del applet. Una vez que el profesor transmita el contenido a dos estudiantes, dichos estudiantes pasarán a ser servidores del contenido que están recibiendo. Para la posterior liberación de la sesión se tendrá que pulsar sobre el botón *Desconexión* en la parte superior del applet.

Los requisitos que deberá cumplir el applet son los siguientes:

1. Para que el applet pueda usar `vlcj[vlcb]` para la reproducción del contenido multimedia, el estudiante deberá configurar la ruta de las librerías de la aplicación VLC[VLCa].
2. El estudiante deberá de definir un nombre de usuario con el que se le identificará en toda la sesión. Dicho nombre puede ser cualquier cadena alfanumérica.
3. El estudiante también deberá de especificar en qué puerto y en qué IP escuchará las solicitudes y respuestas SIP. Estos dos parámetros irán dentro de la cabecera *Contact* del mensaje *INVITE* inicial. En nuestro caso, la dirección IP será la propia de la máquina en la que se lance el applet y el puerto será el que por defecto utiliza SIP, el 5060.
4. Además, el cliente también deberá de especificar el puerto por dónde se va a recibir el contenido multimedia procedente del servidor. Este puerto ha de ser diferente al que se utiliza para las peticiones y respuestas de SIP.
5. El estudiante tiene que configurar la URI SIP correspondiente al IPTV-AS. Dicha URI SIP contiene un nombre de usuario, así como el nombre de dominio correspondiente al servicio IPTV. Se utilizará el puerto por defecto, el 5060.
6. La petición *INVITE* inicial irá dirigida a la URI SIP que identifica al IPTV-AS. Dicha URI SIP se obtendrá de los parámetros comentados en el punto 4.
7. El cliente originario deberá especificar qué tipo de contenido multimedia se va a transmitir. En nuestro caso será vídeo.
8. Dado que puede haber varias clases en directo a disposición del estudiante, éste debe indicar en la carga SDP de la petición *INVITE* cual es la clase a la que desea acceder. Para ello se utiliza el atributo SDP `bc_service` definido por TISPAN [tis], que permite identificar un canal en un servicio IPTV.

5.3.2. Profesor/Servidor

En la implementación realizada, por simplicidad, el profesor (así como cualquier estudiante) puede retransmitir el vídeo correspondiente a la clase a 2 estudiantes. Esta limitación se ha impuesto por simplicidad, sin embargo el código implementado es fácilmente extensible para acomodar cualquier otro algoritmo para la generación del árbol de distribución.

Los requisitos que deberá cumplir el profesor son los siguientes:

1. El profesor deberá de definir un nombre de usuario con el que se le identificará en toda la sesión. Dicho nombre puede ser cualquier cadena alfanumérica.
2. El profesor tendrá que especificar en qué puerto y en qué IP escuchará las solicitudes y respuestas SIP. Estos dos parámetros irán dentro de la cabecera *Contact* del mensaje *200 OK* que el servidor envía como respuesta al *INVITE* inicial. En nuestro caso, la dirección IP será la propia de la máquina en la que se lance el servidor y el puerto será el que por defecto utiliza SIP, el 5060. En este trabajo se asume que la URI SIP del profesor es conocida en el IPTV-AS.
3. El servidor deberá de tener almacenado a que usuarios les ha proporcionado el contenido multimedia.
4. Deberá de configurar la ruta de la aplicación VLC[VLCa] en su equipo.

5.3.3. IPTV-AS

El IPTV-AS será el encargado de recibir las peticiones procedentes de los clientes y de reenviarlas al servidor correspondiente. A su vez también se encargará de hacer el proceso contrario, es decir, recibir las respuestas procedentes del servidor y reenviarlas al cliente correspondiente. Como ya se comentara al principio de este capítulo, sólo participa en la fase de establecimiento y liberación de la sesión y una vez está establecida la sesión, el IPTV-AS no interviene en el envío de los datos multimedia.

Los requisitos que deberá cumplir el IPTV-AS son los siguientes:

1. El IPTV-AS deberá de definir un nombre de usuario con el que se le identificará en toda la sesión. Dicho nombre puede ser cualquier cadena alfanumérica.
2. El IPTV-AS también deberá de especificar en qué puerto y en qué IP escuchará las solicitudes y respuestas SIP. Estos dos parámetros irán dentro de la cabecera *Contact* del mensaje *INVITE* que el IPTV-AS genere después de recibir el *INVITE* inicial. En nuestro caso, la dirección IP será la propia de la máquina en la que se lance el IPTV-AS y el puerto será el que por defecto utiliza SIP, el 5060.

3. El IPTV-AS tiene que configurar la URI SIP correspondiente al profesor. Dicha URI SIP contiene un nombre de usuario, así como el nombre de dominio correspondiente al servicio IPTV. Se utilizará el puerto por defecto, el 5060.
4. Tendrá que almacenar el estado de los diálogos del plano de señalización para saber en todo momento cuál es el estado de la sesión entre cada cliente y servidor.
5. Deberá de saber de quien depende cada elemento del sistema, es decir, tendrá que formar una estructura de árbol binario para saber la ubicación de cada elemento a nivel interno del desarrollo del sistema.
6. Tendrá que ser capaz de reconectar a un usuario cuyo servidor ha decidido abandonar la sesión.

5.4. Plano de datos

Para el plano de datos identificamos los siguientes requisitos:

1. El profesor, una vez finalizado el establecimiento de la sesión SIP, conocerá las direcciones IP de los estudiantes a los que tiene que proporcionar el contenido multimedia.
2. Cuando los estudiantes hagan de servidores, tendrán la misma situación descrita en el requisito anterior.
3. Cuando el estudiante reciba el vídeo, deberá de reproducirlo en el applet que se lanzó al inicio de la sesión. Además permitirá retransmitir el vídeo a otros estudiantes, en caso de ser requerido por el IPTV-AS.

Capítulo 6

Implementación de los elementos del sistema

6.1. Introducción

Una simplificación del sistema que vamos a implementar se muestra en la siguiente figura.

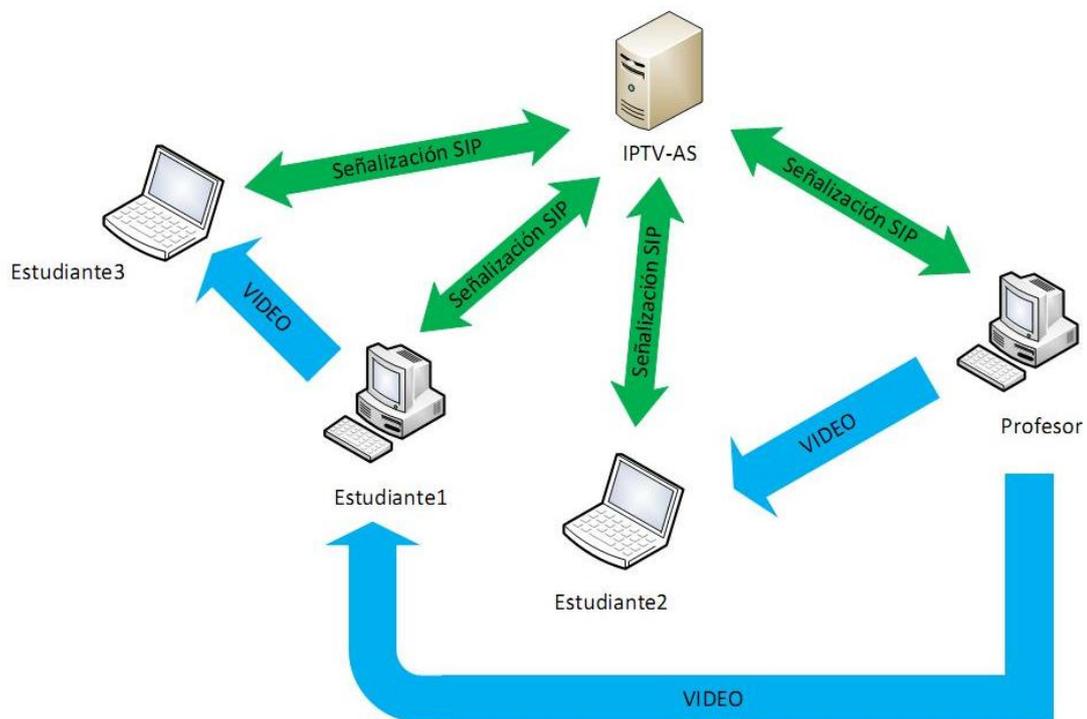


Figura 6.1: Sistema a implementar

Como se puede observar en la figura, podemos diferenciar dos partes en la implementación del sistema:

- Un plano de señalización que se realizará mediante el protocolo de señalización SIP. Se utilizará este protocolo para gestionar la señalización necesaria antes de empezar la transmisión del contenido multimedia entre el servidor y el cliente. En esta parte, el IPTV-AS participará activamente ya que todas las solicitudes SIP que generen los clientes serán enviadas al él, éste las procesará y reenviará la petición recibida al correspondiente servidor de contenidos multimedia. Los servidores, a su vez, generarán las respuestas a las peticiones que reciban y serán enviadas al IPTV-AS y éste las enviará al iniciador de la sesión. En pocas palabras, toda la señalización se realiza a través del IPTV-AS. Además también se gestionará la liberación de la sesión por parte de un cliente en cualquier momento.
- La transmisión de paquetes con contenido multimedia a través de la técnica P2P. Una vez finalizada la señalización y sabiendo que los elementos del sistema han aceptado la sesión, los paquetes con contenido multimedia serán enviados directamente desde el servidor al puerto del cliente que se estableció en la fase de señalización. En esta parte el IPTV-AS no intervendrá.

Vamos a describir el entorno sobre el que se ha desarrollado esta aplicación. La solución está formada por varias clases desarrolladas en JAVA.

Para la señalización de SIP en JAVA existe una interfaz llamada *JAIN SIP* [Pro07][PO03]. Con esta interfaz gestionamos la señalización a nivel de transacción sin preocuparnos por la complejidad de la sintaxis de los mensajes SIP y del transporte de los mismos, ya que esto se realiza a través de una interfaz con una pila de protocolos SIP. Esta interfaz se basa en eventos/mensajes y la implementación de SIP Listener y SIP Provider que interactúan con la Pila SIP (figura 6.2). El SIP Provider convierte mensajes SIP que recibe de la red en eventos para dárselos a la aplicación, mientras que el SIP Listener escucha estos eventos y los encapsula dentro de mensajes para que la aplicación los gestione.

De la interfaz *JAIN SIP* utilizaremos las siguientes librerías para la gestión de la señalización de SIP:

- **jain-sip-api-1.2.jar**: contiene las interfaces de SIP y las clases principales. Se trata de un API de JAVA para el acceso a bajo nivel de la pila de protocolo SIP.
- **jain-sip-ri-1.2.90.jar**: contiene la implementación de referencia de SIP.
- **log4j-1.2.8.jar**: es una librería que permite gestionar los mensajes de debugging y los avisos de la aplicación.
- **concurrent.jar**: es una librería que gestiona la ejecución de procesos concurrentes.

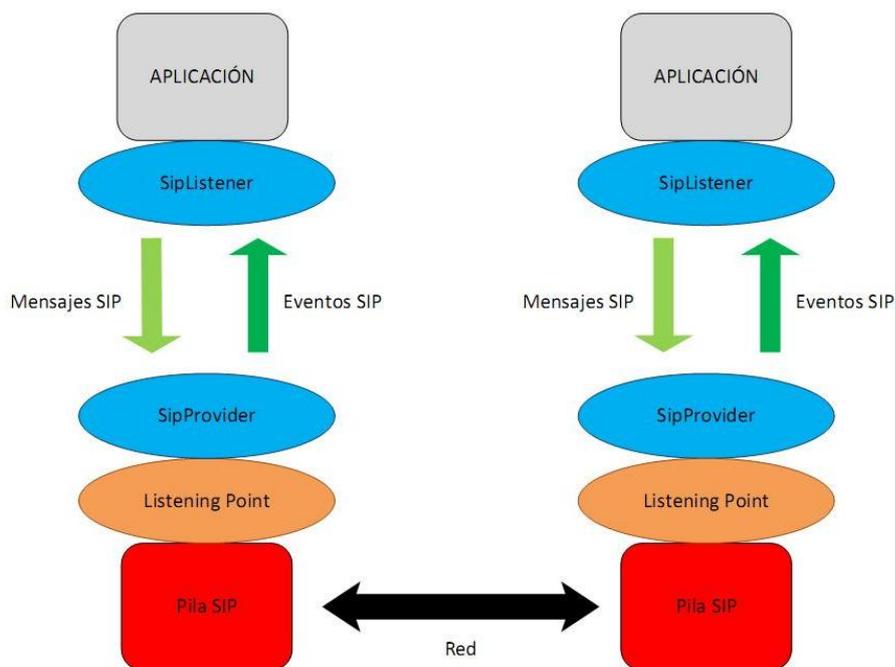


Figura 6.2: Arquitectura JAIN SIP [PO03]

Para el plano de datos de la aplicación se ha utilizado la siguiente librería del *framework* vlcj[vlcb] de JAVA:

- **vlcj-1.2.2.jar**: contiene la implementación de las clases de vlcj.

Y también hemos necesitado usar otras dos librerías más:

- **jna.jar** y **platform.jar**: nos permiten acceder a las librerías nativas del sistema operativo de forma sencilla.

En la figura 6.3 podemos ver un esquema de la aplicación completa.

En el caso del cliente (Estudiante) disponemos de un applet cuando lanzamos el aplicativo. Dicho applet es el que se encarga de comunicarse con el IPTV-AS generando las solicitudes y procesando las respuestas SIP.

En el caso del servidor (Profesor/Estudiante) se comunicará con el IPTV-AS para gestionar las solicitudes y respuestas SIP y una vez se tenga establecida la señalización se comunicará con la aplicación de emisión de vídeo para que ésta, a su vez, retransmita dicho contenido al applet del estudiante.

A continuación vamos a describir de manera más detallada la implementación de cada uno de los elementos que forman parte de esta solución.

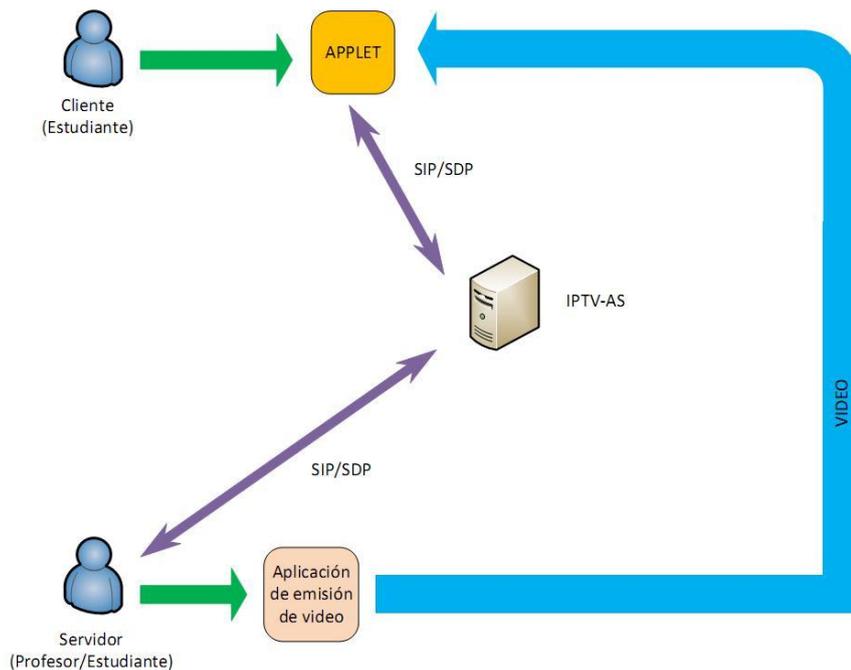


Figura 6.3: Esquema lógico

6.2. Estudiante

6.2.1. Interfaz gráfica

La implementación del estudiante/cliente está basada en una interfaz gráfica, más concretamente en un applet desarrollado en Java. Cuando el cliente lance el applet se le presentará una ventana como la mostrada en la figura 6.4.

El applet está compuesto por dos partes bien diferenciadas:

- En la parte inferior tenemos el botón *Conexión*. Si pulsamos sobre él comienza el intercambio de mensajes SIP para establecer la sesión multimedia con el servidor.
- En el resto del applet tenemos la parte donde se va a visualizar el contenido multimedia, en nuestro caso, como ya hemos comentado anteriormente será vídeo.

Los parámetros de configuración básicos que se mencionan en los requisitos definidos en el capítulo 5.3.1 serán establecidos de la siguiente forma:

- Como un parámetro del applet, la ruta¹ del sistema operativo dónde se tengan las librerías de la aplicación VLC (requisito 1).

¹En la parte *xxxxxx* del siguiente fragmento código se incluirá la ruta de la aplicación:
`NativeLibrary.addSearchPath(RuntimeUtil.getLibVlcLibraryName(), "xxxxxx");`

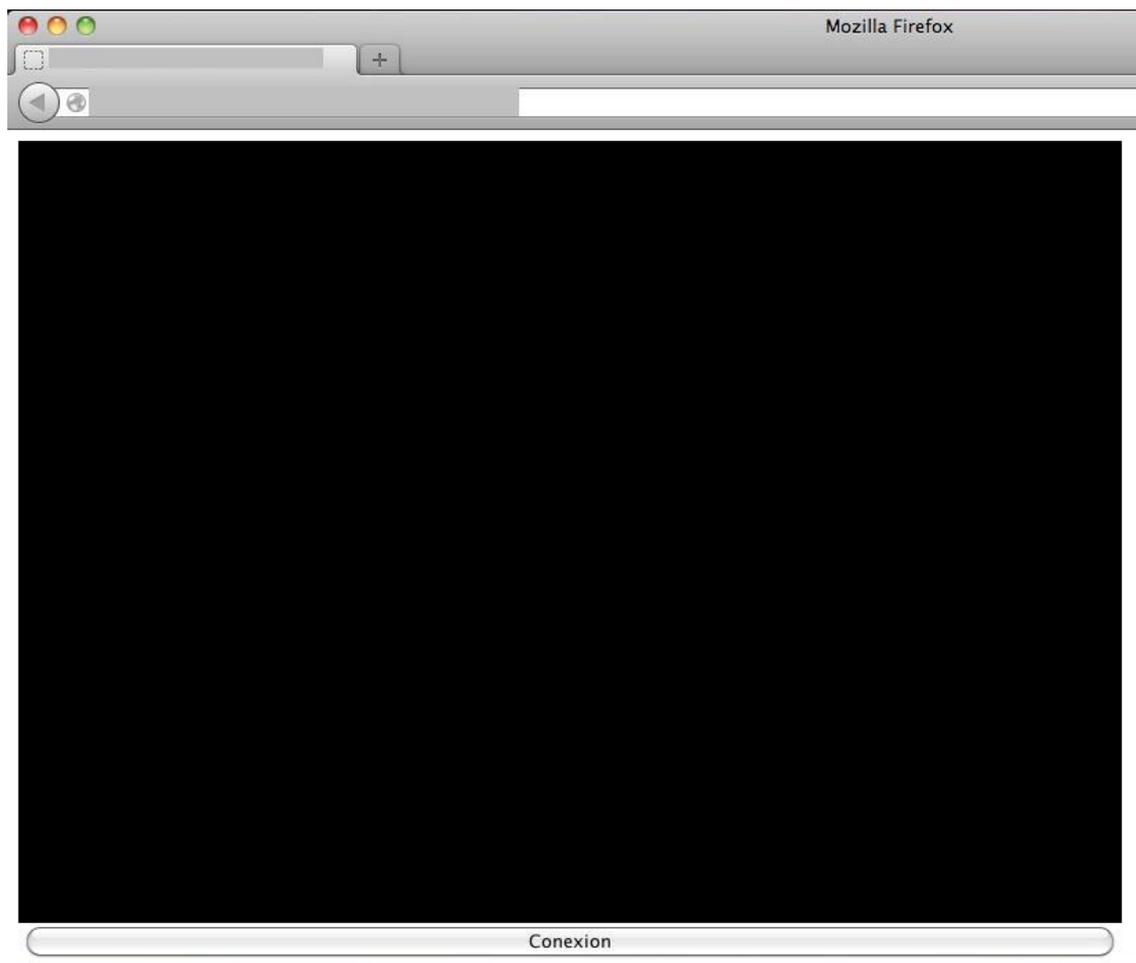


Figura 6.4: Vista del Applet para la conexión

- Como otro parámetro del applet, la dirección IP o nombre de dominio del IPTV-AS (parte del requisito 5).
- Como variables globales de la implementación del applet, los requisitos del 2 al 5.
- Como contenido de la carga SDP del mensaje *INVITE* generado, el requisito 7.
- Como otro parámetro del applet, el atributo SDP para identificar el canal en el servicio IPTV (requisito 8).

6.2.2. Procedimientos en el cliente

En este apartado se explicarán todos los procesos que se suceden desde que el cliente inicia la sesión hasta que la libera.

Un ejemplo de rutas:

Para Windows -> C:\\Archivos de programa\\VideoLAN\\VLC\
Para Mac OS X -> /Applications/VLC.app/Contents/MacOS/lib/

6.2.2.1. Establecimiento de sesión

Antes de empezar con la señalización, se tendrán que tener establecidos correctamente los parámetros de configuración comentados en el apartado anterior 6.2.1 y a continuación se procederá a ejecutar la aplicación para inicializar las variables del sistema. En la figura 6.5 se describe este proceso.

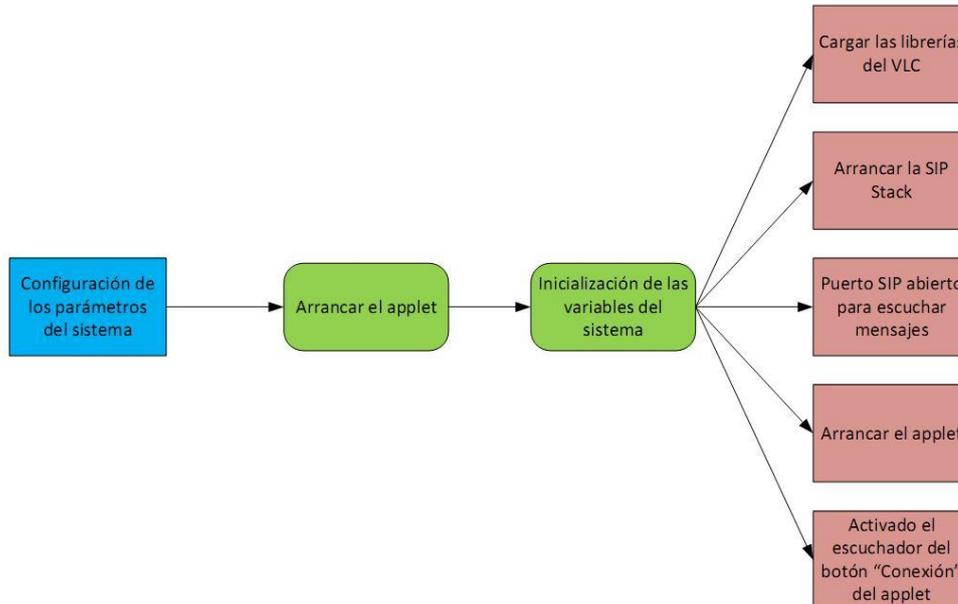


Figura 6.5: Inicialización del Sistema

Una vez que se han inicializado las variables, se procederá a realizar la señalización SIP. Para ello, el estudiante deberá de pulsar el botón *Conexión* del applet y a partir de ese momento se generará una petición *INVITE* dirigida al IPTV-AS que es el conocedor del servidor de contenido multimedia. Este proceso podemos verlo en la figura 6.6.

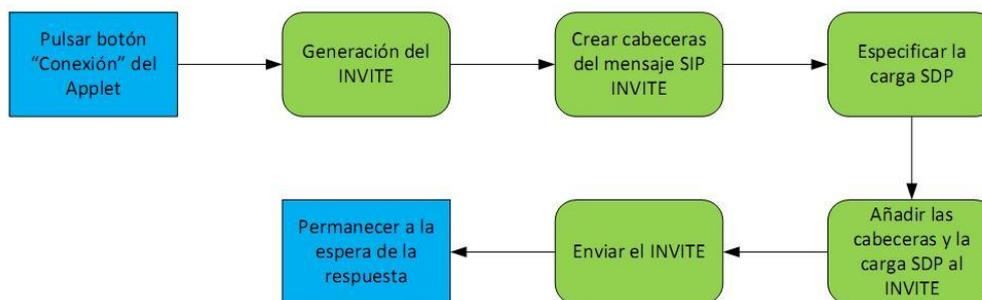


Figura 6.6: Generación del INVITE

Una vez realizado el envío de la petición anterior, la aplicación se encuentra a la espera de recibir la respuesta del IPTV-AS. Cuando llega la respuesta *200 OK*

procedente del IPTV-AS, el cliente la procesa y genera el *ACK* correspondiente para asentar la respuesta recibida como podemos ver en la figura 6.7.

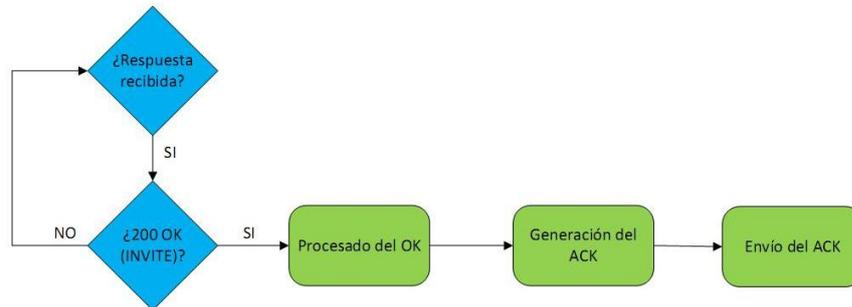


Figura 6.7: Procesamiento del OK y generación del ACK

Llegados a este punto, el estudiante ya tiene establecida la sesión multimedia y además se queda escuchando los posibles mensajes SIP que pudiera recibir de otros usuarios para establecer una sesión multimedia, en este caso, siendo él el servidor del vídeo.

6.2.2.2. Reproducción del contenido multimedia

Una vez tenemos establecida la sesión multimedia empezaremos a recibir el contenido multimedia procedente del servidor correspondiente. Dicho contenido nos llegará a través del puerto que se haya establecido en la carga SDP para la recepción del vídeo.

Para reproducir el vídeo en el applet, lo reenviaremos a otro puerto del equipo que hemos establecido en la implementación. Este reenvío interno se ha realizado para facilitar la reproducción y la posterior difusión de dicho contenido multimedia cuando este cliente sea también servidor, ya que de esta forma no dejamos bloqueada nuestra implementación con la reproducción del contenido multimedia en el cliente, permitiendo la recepción de nuevos mensajes con el fin de establecer nuevas sesiones multimedia con otros usuarios. Los pasos seguidos para la reproducción del contenido multimedia se pueden ver en la figura 6.8.

6.2.2.3. Liberación de la sesión

Cuando el cliente quiera abandonar la sesión tendrá que pulsar el botón *Desconexión* que se encuentra en la parte superior del applet (figura 6.9).

Una vez hecho esto, el cliente generará un mensaje *BYE* con destino el IPTV-AS para indicarle que quiere abandonar la sesión (figura 6.10).

Cuando reciba la respuesta *200 OK* al *BYE* enviado anteriormente, se cerrará el applet poniendo fin a la reproducción del contenido multimedia (figura 6.11).

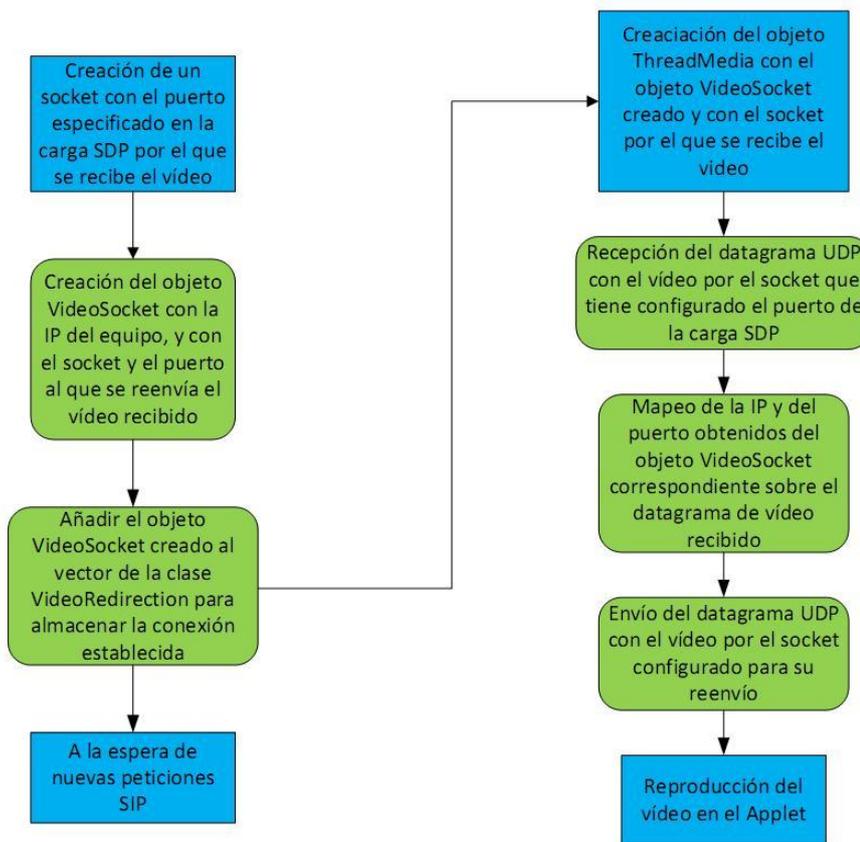


Figura 6.8: Recepción y reproducción del vídeo

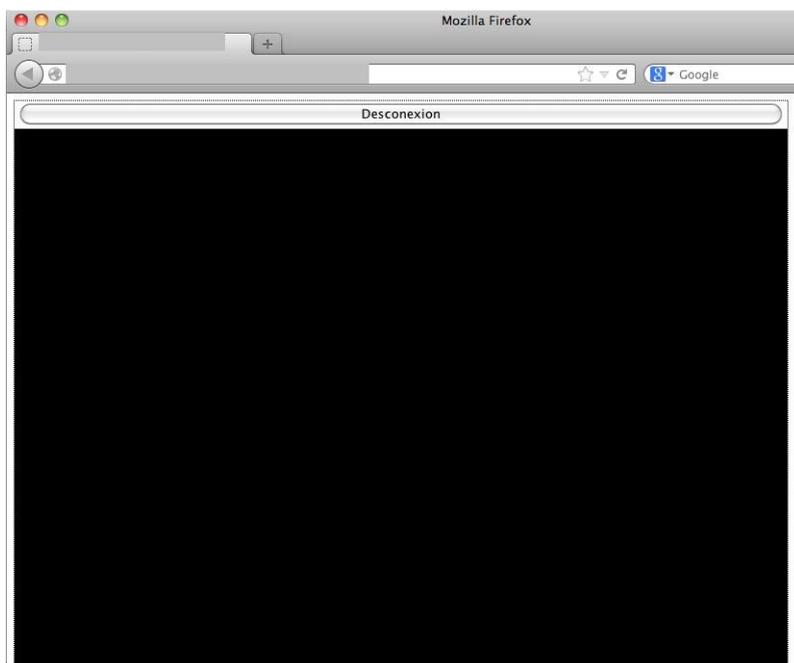


Figura 6.9: Vista del Applet para la desconexión

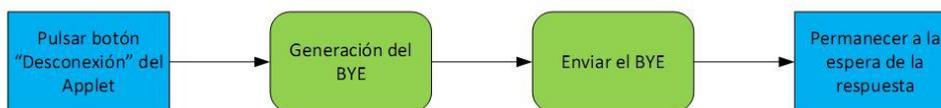


Figura 6.10: Desconexión

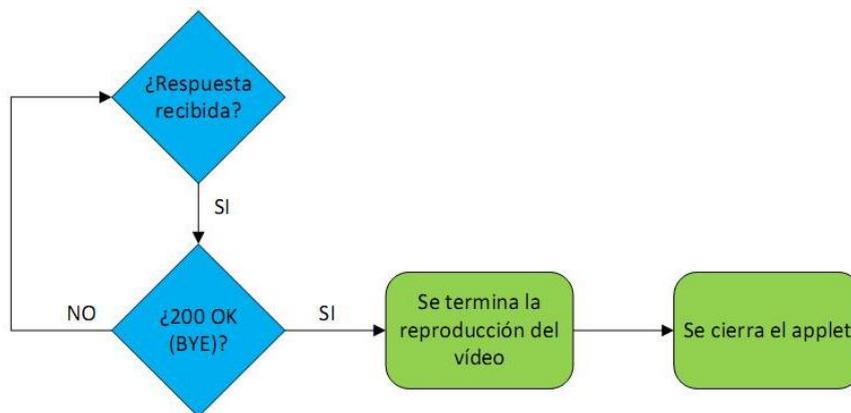


Figura 6.11: Procesado del OK del BYE

6.2.2.4. Envío del contenido multimedia cuando el cliente también actúa como servidor

Llegados a este punto ya tenemos establecida la sesión multimedia entre el cliente que inició la sesión y en este caso, el cliente que hace a su vez de servidor de contenidos, por lo que sólo nos quedará ver como se va a enviar el contenido multimedia. Para ello, como en este caso el servidor del nuevo cliente ya está recibiendo el contenido multimedia de su propio servidor, sólo hará falta crear un socket JAVA con el puerto por donde se va a enviar el vídeo al cliente que se acaba de conectar y a continuación, reenviarle los datagramas que él recibe de su servidor (figura 6.12).

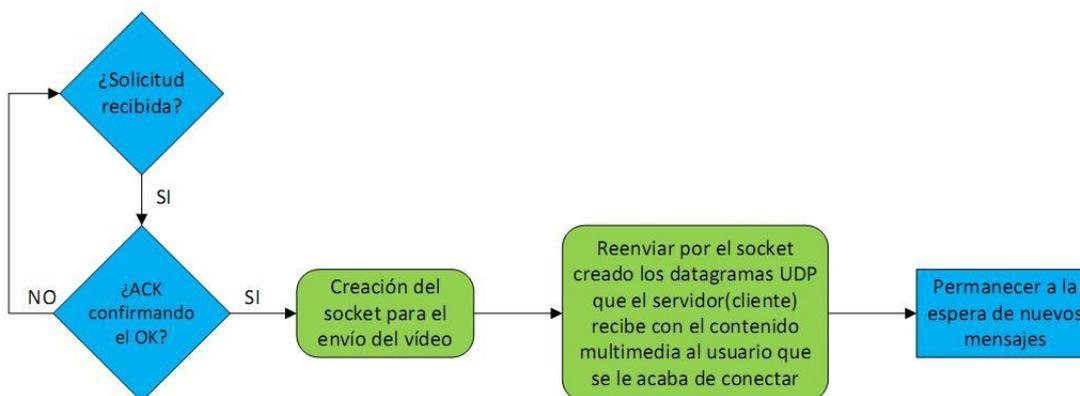


Figura 6.12: Reenvío del vídeo al nuevo cliente

6.3. Profesor

A diferencia del cliente, en este caso, para el profesor/servidor se ha realizado una aplicación JAVA y no disponemos de una interfaz gráfica, ya que no es un objetivo de este proyecto.

Para su ejecución se utilizará un terminal de comandos (o en su defecto Eclipse) dónde se lanzará el servidor y se le pasará como parámetro el nombre de usuario que queremos utilizar para identificar al servidor durante la sesión. Este parámetro es el requisito 1 de los definidos en el capítulo 5.3.2.

El resto de los parámetros de configuración básicos acorde con los requisitos definidos en el capítulo 5.3.2 serán establecidos de la siguiente forma:

- Como variables locales del método principal de la implementación del servidor, el requisito 3.
- Como variable global de la implementación, el requisito 5.
- Como ruta ² del sistema dónde se encuentra el ejecutable de la aplicación VLC en el servidor, el requisito 4.

6.3.1. Procedimientos en el servidor

En este apartado se explicarán todos los procesos que se suceden desde que el servidor recibe la petición *INVITE* para el establecimiento de la sesión hasta que recibe un *BYE* para la liberación de la sesión correspondiente.

6.3.1.1. Establecimiento de sesión

Una vez que las variables están inicializadas, el servidor permanece en estado de espera hasta que recibe la primera solicitud que será un *INVITE* procedente del IPTV-AS. Cuando esta solicitud llega, el servidor actúa tal y como se muestra en la figura 6.13.

Básicamente, lo que hace el servidor es mirar en la carga SDP que recibe en el *INVITE* y a partir de ella conoce la dirección IP y el puerto hacia donde irán dirigidos los paquetes con el contenido multimedia.

Conocidos estos valores, el servidor generará una respuesta *200 OK* con su respuesta SDP correspondiente, en la que el codec de vídeo será el mismo que viniera

²En la parte *xxxxx* del siguiente fragmento código se incluirá la ruta de la aplicación:

```
Runtime.getRuntime().exec("xxxxx");
```

Un ejemplo de rutas:

Para Windows -> C:\\Archivos de programa\\VideoLAN\\VLC\\vlc.exe

Para Mac OS X -> /Applications/VLC.app/Contents/MacOS/VLC

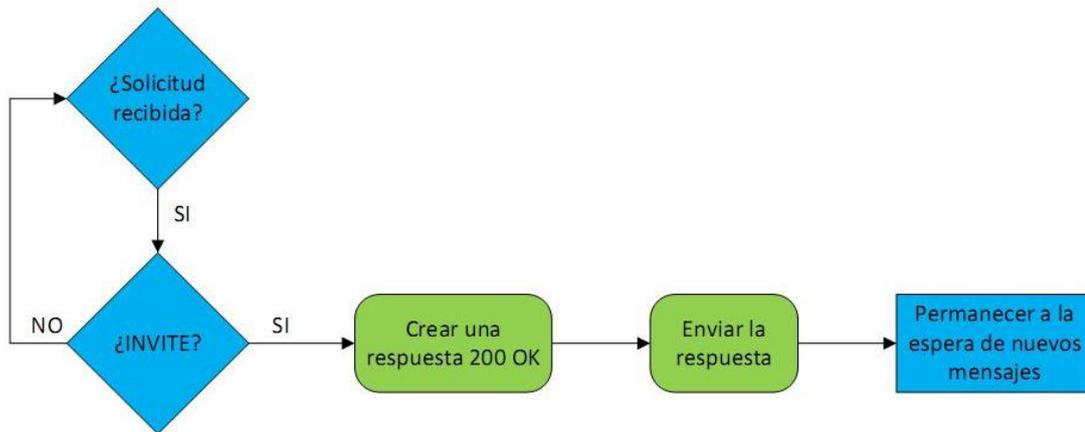


Figura 6.13: Procesado del INVITE en el servidor

en la oferta SDP en el *INVITE*. Una vez creada esta respuesta, se enviará al IPTV-AS para que éste se la haga llegar al iniciador de la sesión.

Finalizado el proceso mostrado en la figura 6.13, el servidor pasa a un estado de escucha a las siguientes solicitudes SIP.

El siguiente mensaje que debe de recibir el servidor es un *ACK* que le envía el IPTV-AS. Este *ACK* es para confirmar que el destinatario ha aceptado el establecimiento de la sesión.

6.3.1.2. Envío del contenido multimedia

Llegados a este punto ya tenemos establecida la sesión multimedia entre el cliente y el servidor, por lo que sólo nos quedará ver como se va a enviar el contenido multimedia. Para ello, el servidor creará un socket JAVA con el puerto de su equipo por donde va a enviar el contenido multimedia al cliente. A continuación lanzará la aplicación **VLC** para empezar la transmisión del contenido multimedia al iniciador de la sesión tal y como podemos ver en la figura 6.14 junto con la recepción del mensaje *ACK*.

La configuración a seguir en el **VLC** para proceder al envío podemos verlo en detalle en el Anexo F.

Una vez llegados a este punto el servidor ya está transmitiendo el vídeo y permanece a la espera de recibir nuevos mensajes SIP de clientes para establecer una nueva sesión multimedia con él.

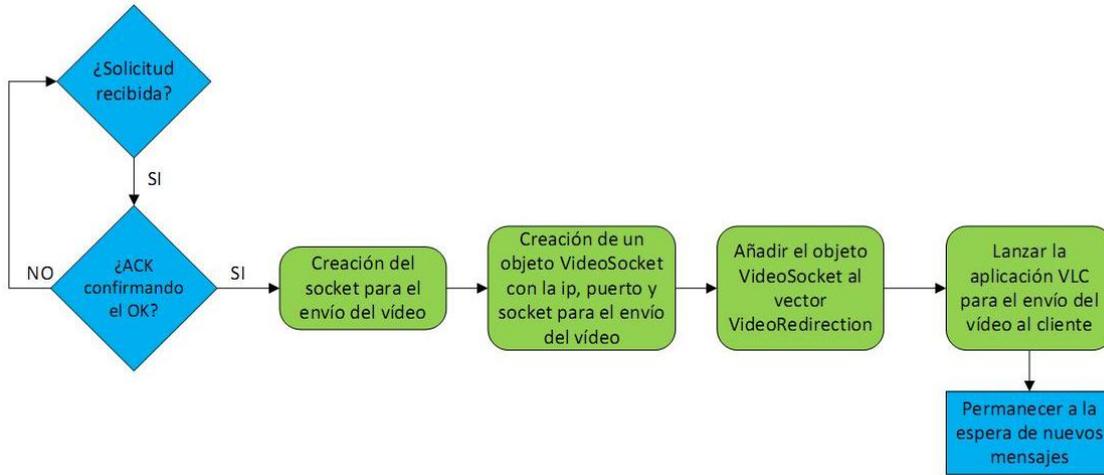


Figura 6.14: Recepción del ACK y envío del vídeo

6.3.1.3. Liberación de la sesión

Cuando el servidor recibe un mensaje *BYE* procedente del IPTV-AS lo que hace es dejar de transmitir el contenido multimedia al cliente que decide abandonar la sesión cerrando el socket que tenía establecido para esa transmisión.

A continuación se genera la respuesta *200 OK* a la petición recibida anteriormente como podemos ver en la figura 6.15.

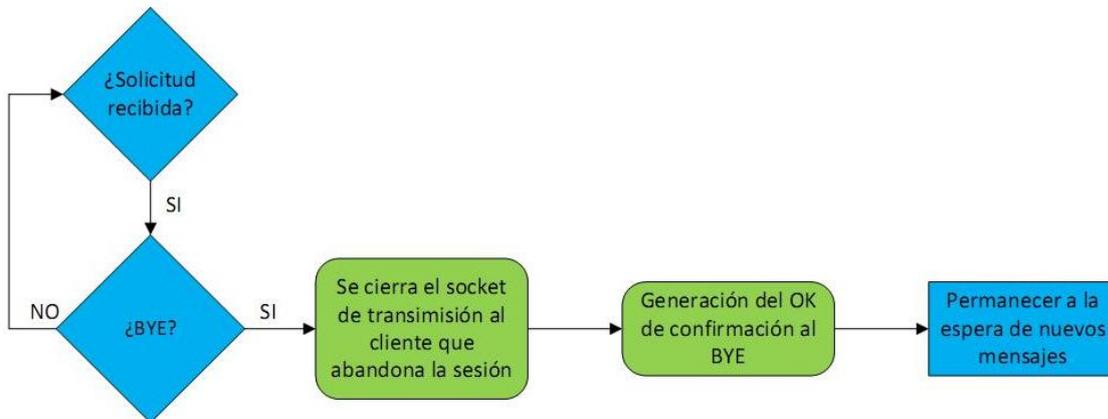


Figura 6.15: Procesado del BYE y generación del OK

6.4. IPTV-AS

Al igual que para el servidor, en este caso, para el IPTV-AS se ha realizado una aplicación JAVA y no disponemos de una interfaz gráfica, ya que no es necesario debido a que sólo se encarga de la parte de los mensajes SIP.

Para su ejecución se utilizará un terminal de comandos (o en su defecto Eclipse) dónde se lanzará el IPTV-AS y se pasarán como parámetros:

- El nombre de usuario que queremos utilizar para identificar al IPTV-AS durante la sesión (requisito 1 de los definidos en el capítulo 5.3.3)
- La URI SIP del profesor que va a recibir las solicitudes SIP (requisito 3 de los definidos en el capítulo 5.3.3).

El resto de los parámetros de configuración básicos acorde con los requisitos definidos en el capítulo 5.3.3 serán establecidos de la siguiente forma:

- Como variables locales del método principal de la implementación del IPTV-AS, el requisito 2.
- Como variables globales de la implementación, los requisitos 4 y 5.

6.4.1. Procedimientos en el IPTV-AS

En este apartado se explicarán todos los procesos que se suceden desde que el IPTV-AS recibe la petición *INVITE* procedente del cliente hasta que se recibe el mensaje *BYE* indicando que un usuario abandona la sesión.

6.4.1.1. Procesado de las peticiones SIP

Una vez que las variables están inicializadas, el IPTV-AS permanece en estado de espera hasta que recibe alguna solicitud SIP pudiendo ser una de las siguientes.

1. INVITE

La primera solicitud que recibirá será un *INVITE* procedente del cliente. Cuando esta solicitud llega, el IPTV-AS se comporta tal y como se muestra en la figura 6.16.

Lo que hace el IPTV-AS es almacenar ciertos datos de la sesión para tener en todo momento localizados los elementos que forman parte de ella. Para ello, realiza varias operaciones:

- Almacena en un vector de operaciones pendientes (perteneciente a la clase *PendingOperations.java*) objetos del tipo *ActualStateDialog* (pertenecientes a la clase *ActualStateDialog.java*) para poder identificar este diálogo iniciado por el cliente en futuros mensajes recibidos. Este objeto contiene diferentes datos del cliente y del servidor que van a establecer una sesión.
- Mira en la carga SDP que recibe en el *INVITE* y a partir de ella conoce el codec de vídeo que se utilizará para la transmisión.

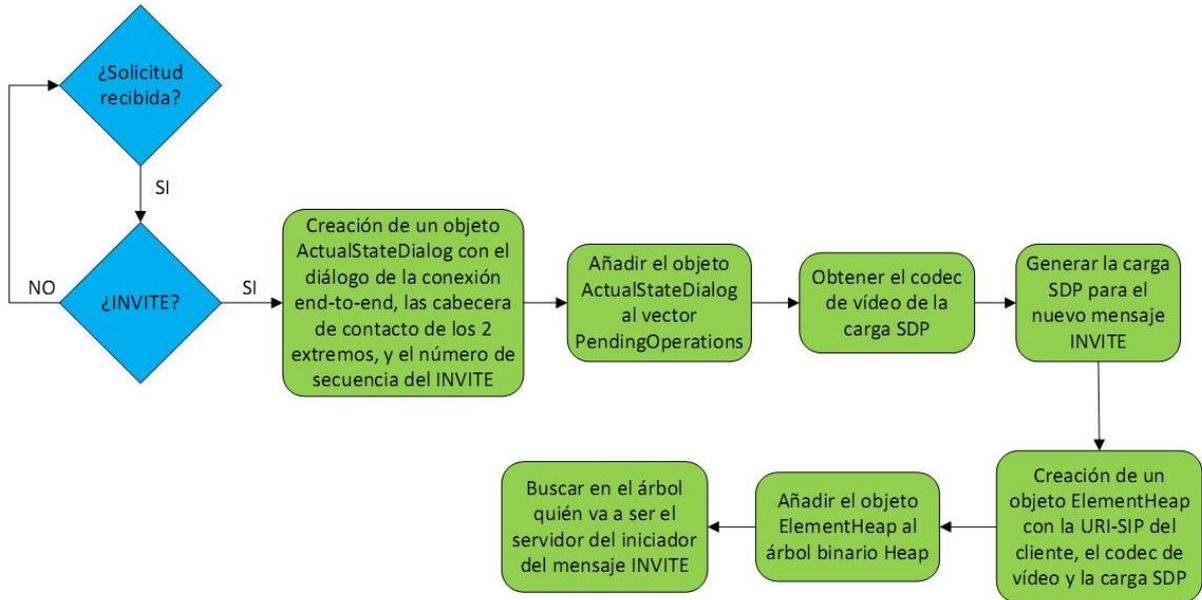


Figura 6.16: Procesado del INVITE en el IPTV-AS

- El IPTV-AS genera la carga SDP del nuevo mensaje *INVITE* que se va a enviar al servidor.
- Después añadirá un elemento nuevo al árbol binario (implementado con las clases *ElementHeap.java* y *Heap.java*) que ha sido inicializado anteriormente (requisito 5 del capítulo 5.3.3) donde se almacenarán la dirección de contacto, el codec y la nueva carga SDP del mensaje *INVITE* que va a generar el IPTV-AS a continuación.
- Dentro del árbol, se realizará un búsqueda para conocer quién será el servidor para el cliente del mensaje *INVITE* que acaba de recibir.

Una vez realizadas las acciones anteriores, el IPTV-AS procederá a la generación del nuevo mensaje *INVITE* con algunos de los datos obtenidos en el proceso anterior y a su posterior envío al servidor correspondiente tal y como podemos ver en la figura 6.17.



Figura 6.17: Generación del INVITE para el servidor

Después del envío del mensaje *INVITE*, el IPTV-AS se quedará nuevamente a la espera de recibir nuevos mensajes SIP.

2. ACK

Otro de los mensajes que puede recibir el IPTV-AS es un *ACK* como asentimiento de la respuesta *200 OK* del cliente.

En este caso lo único que hace el IPTV-AS es procesar el *ACK* recibido para poder generar el suyo y enviarlo al servidor correspondiente como podemos ver en la figura 6.18.

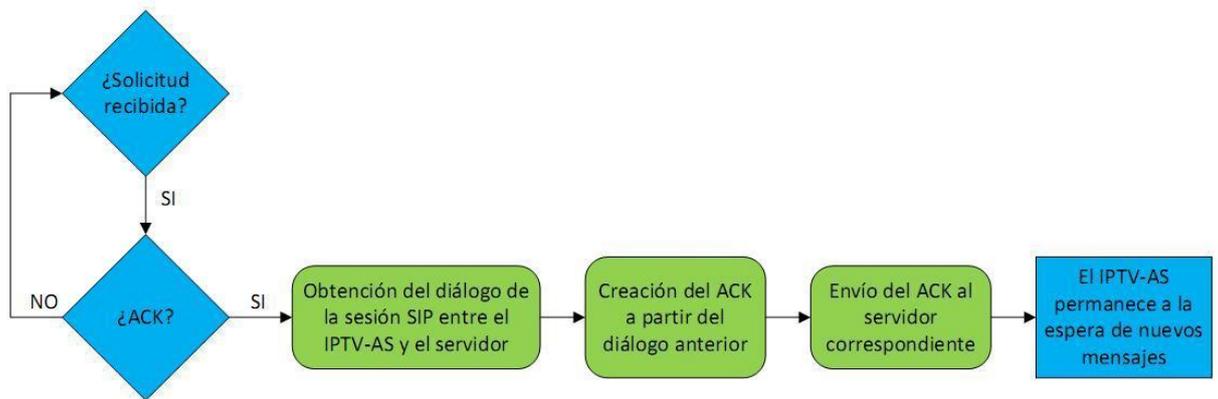


Figura 6.18: Generación del ACK para el servidor

Y a continuación, nuevamente se quedará a la espera de recibir nuevos mensajes SIP.

3. BYE

Cuando el IPTV-AS recibe una solicitud *BYE* procedente de un cliente, lo procesa y genera el correspondiente *200 OK* para que el cliente pueda abandonar la sesión (figura 6.19).

Además se genera una solicitud *BYE* para el servidor que estaba proporcionando el contenido multimedia al cliente que quiere abandonar la sesión.

Por otra parte, el IPTV-AS tendrá que gestionar la reconexión de otros clientes que dependieran del usuario que abandona la sesión, si los hubiera. Para ello, buscará un nuevo servidor para cada cliente y le mandará un mensaje *INVITE* indicándole la dirección de contacto de su nuevo cliente (figura 6.20).

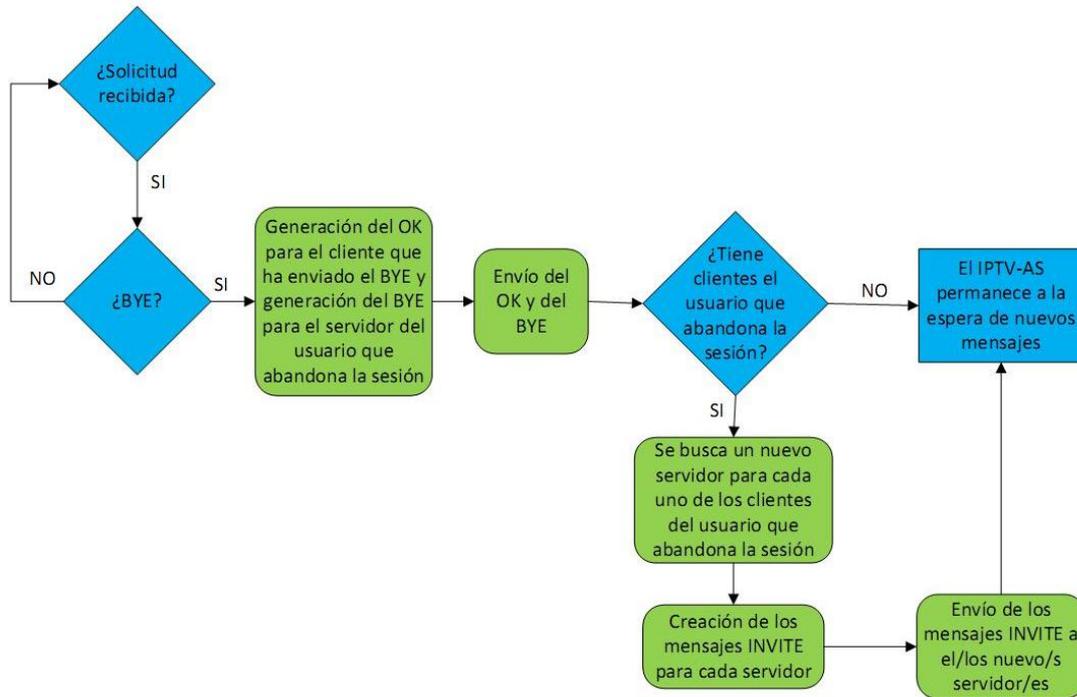


Figura 6.19: Procesado del BYE en el IPTV-AS

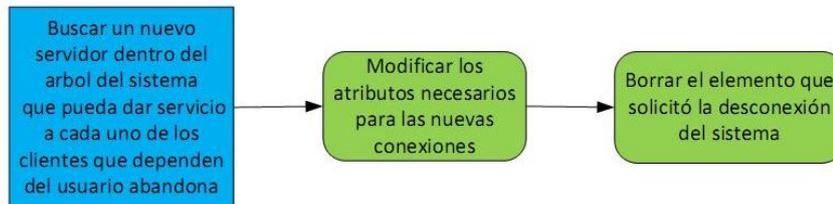


Figura 6.20: Reconexión de clientes

6.4.1.2. Procesado de las respuestas SIP

Una vez que el IPTV-AS haya procesado alguna petición SIP, comenzará a recibir respuestas a dichas peticiones para que también sean procesadas.

1. Respuestas a solicitudes *INVITE*

Podrá recibir dos tipos de *200 OK* como respuesta a peticiones *INVITE*:

- Si un cliente solicitó un inicio de sesión, cuando el IPTV-AS reciba un *200 OK* procedente del servidor deberá generar el *200 OK* correspondiente al *INVITE* de ese cliente.
- Si un IPTV-AS generó un *INVITE* después de una desconexión de un cliente deberá generar el *ACK* correspondiente al *200 OK* procedente del servidor como asentimiento al establecimiento de sesión.

Ambos tipos de respuesta son representados en la figura 6.21.

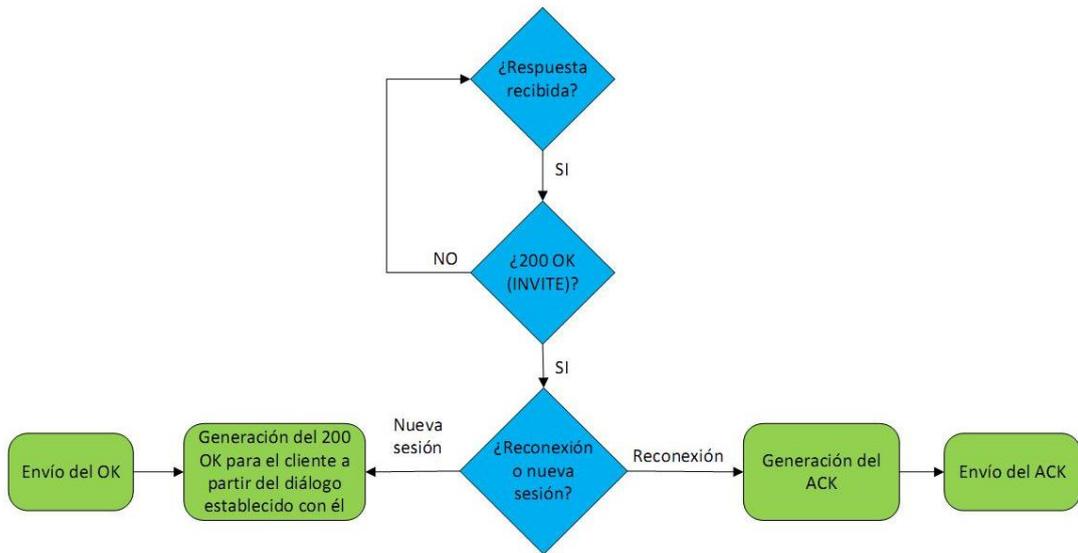


Figura 6.21: Procesado del 200 OK en el IPTV-AS

2. Respuesta a solicitud *BYE*

Recibirá un *200 OK* como respuesta a la petición *BYE* enviada al servidor del cliente que abandona la sesión. Al recibir esta respuesta borrará del vector de operaciones pendientes las dos entradas correspondientes a los diálogos establecidos para la sesión del cliente que se va. (figura 6.22).



Figura 6.22: Procesado del 200 OK correspondiente al BYE

Capítulo 7

Escenario de pruebas

7.1. Introducción

El escenario de pruebas se basa en un entorno de red de área local Ethernet en el que todos los elementos del sistema se encuentran conectados por cable Ethernet a un switch y este switch a un router que soporta el envío de datagramas IP. El entorno de pruebas utilizado en un aula de laboratorio del Departamento de Telemática se muestra a continuación.

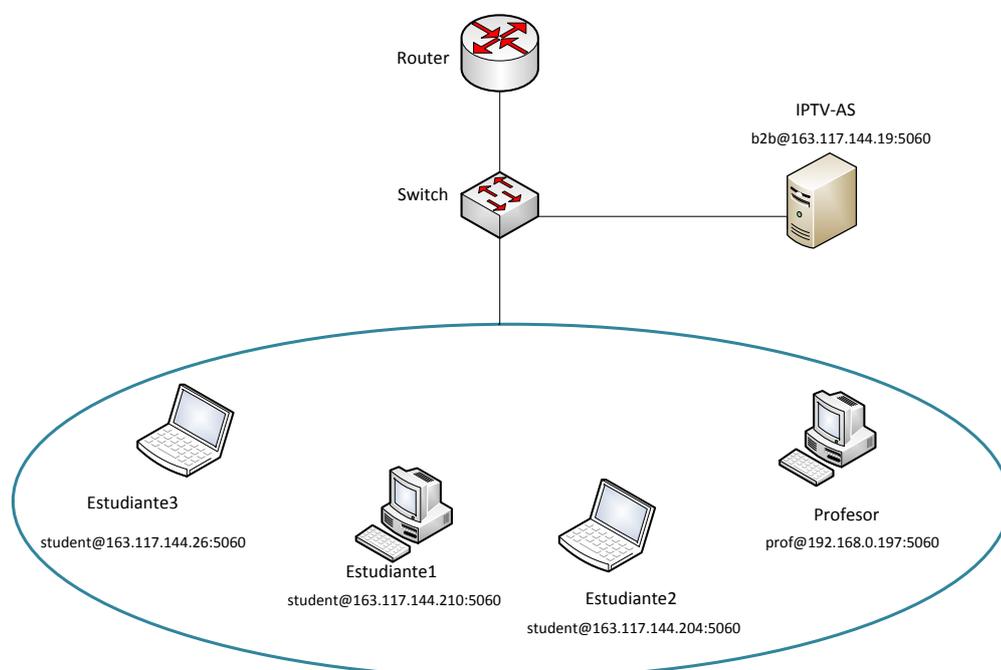


Figura 7.1: Escenario de pruebas

Los host empleados en las pruebas tienen las propiedades mostradas en la figura 7.2.

Host	Dirección IP	Puerto SIP	Puerto vídeo
prof	192.168.0.197	5060	65070
b2b	163.117.144.19	5060	-
student	163.117.144.210	5060	65070
student	163.117.144.204	5060	65070
student	163.117.144.26	5060	65070

Figura 7.2: Propiedades de los elementos del sistema

Para validar la implementación del sistema desarrollado y comprobar que cumple con los requisitos que se definieron en el capítulo 5 se han realizado una serie de pruebas que se detallan en sucesivos apartados.

7.2. Establecimiento de sesión con más de 2 estudiantes

En este caso el originador de la primera sesión será *Estudiante 1*, a continuación se conectará *Estudiante 2* y por último *Estudiante 3*. Todos ellos iniciarán la sesión con el IPTV-AS y posteriormente éste será quien decida cuál es el servidor de cada cliente.

Si tenemos en cuenta que nuestra implementación está conformada como un árbol binario con dos clientes por cada elemento del sistema, vemos que tenemos dos tipos de conexión:

- Los dos primeros estudiantes que se conecten, en este caso *Estudiante 1* y *Estudiante 2*, tendrán una conexión directa con el equipo del profesor a través del IPTV-AS.
- Sin embargo, *Estudiante 3* tendrá una conexión a través del IPTV-AS a un cliente previamente conectado al sistema (*Estudiante 1*) y por tanto será éste quien haga de cliente y de servidor al mismo tiempo.

Una vez que cada estudiante tenga establecida su sesión con su servidor, comenzará a visualizar el contenido multimedia en su equipo.

Validación y análisis de los resultados obtenidos

El diagrama de flujo de los mensajes del establecimiento de la sesión se muestra a continuación.

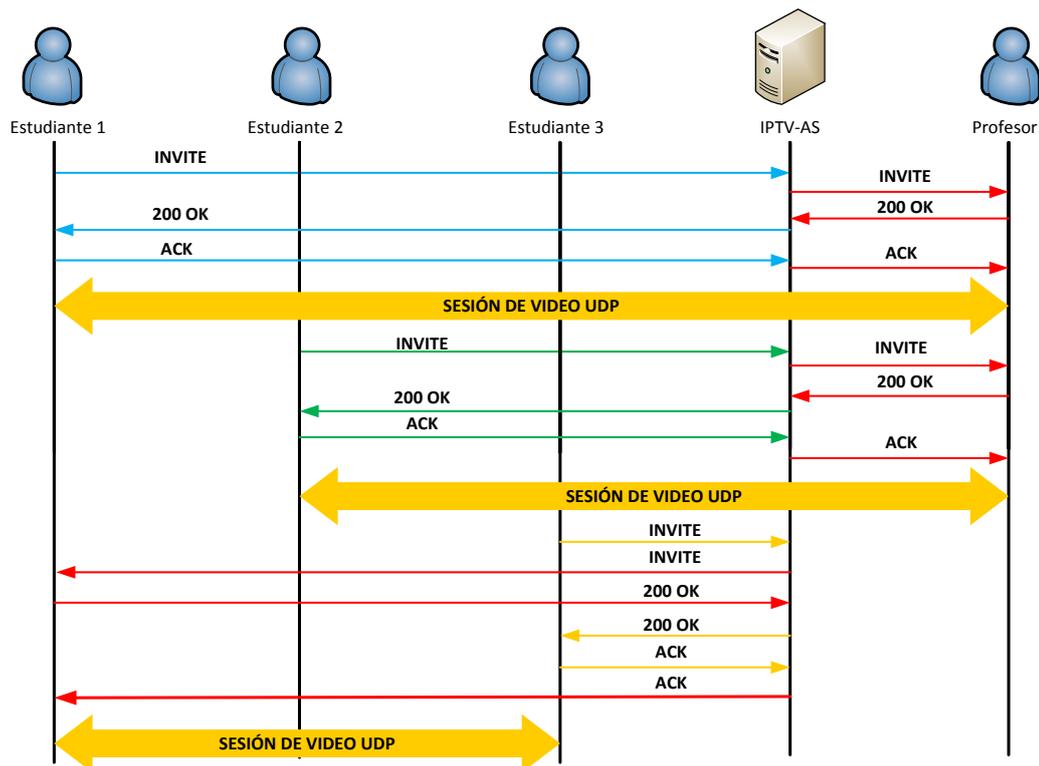


Figura 7.3: Establecimiento de sesión

El contenido de los mensajes que se cursaron para el establecimiento de sesión de *Estudiante 3* fueron los siguientes¹ (figura 7.4)².

114	29.170254	163.117.144.26	163.117.144.19	SIP/SDP	600 Request: INVITE sip:b2b@163.117.144.19:5060;transport=udp
115	29.233831	163.117.144.19	163.117.144.26	SIP/SDP	537 Status: 200 OK
116	29.303434	163.117.144.26	163.117.144.19	SIP	396 Request: ACK sip:b2b@163.117.144.19:5060

Figura 7.4: Establecimiento de sesión de Estudiante 3

Mensaje INVITE desde Estudiante 3 a IPTV-AS

```
INVITE sip:b2b@163.117.144.19:5060;transport=udp SIP/2.0
Call-ID: b2f99e76a14efb8963f8af4e49e14ccb@163.117.144.26
CSeq: 1 INVITE
From: "student" <sip:student@163.117.144.26:5060>;tag=Student
To: "b2b" <sip:b2b@163.117.144.19:5060>
Via: SIP/2.0/UDP
163.117.144.26:5060;branch=z9hG4bKe88971d121078db7ddeaae27489b3918
Max-Forwards: 70
Contact: "student1" <sip:student@163.117.144.26:5060>
Content-Type: application/sdp
Content-Length: 107
```

¹Para *Estudiante 1* y *Estudiante 2* los mensajes son similares

²Capturas realizadas con el analizador de tráfico Wireshark

```
v=0
o=student 13 13 IN IP4 163.117.144.26
c=IN IP4 163.117.144.26
m=video 65070 udp mpeg
a=bc_service:5
```

A la vista de este mensaje, podemos observar:

- La solicitud va dirigida al IPTV-AS, cuya dirección IP es la 163.117.144.19, tal y como podemos ver en la Request-Line (la primera línea del mensaje SIP).
- El mensaje contiene una carga SDP de acuerdo a la cabecera *Content-Type*.
- De la carga SDP podemos extraer que:
 - En la línea "c" se especifica la dirección IP donde se va a producir la conexión de vídeo.
 - En la línea "m" vemos especificado el puerto por dónde se va a recibir el vídeo y el protocolo de transporte utilizado en la transmisión.

Mensaje INVITE desde IPTV-AS a Estudiante 1

```
INVITE sip:student@163.117.144.210:5060;transport=udp SIP/2.0
Call-ID: 294f9efe20588bace85df3f51c2d05bb@163.117.144.19
CSeq: 1 INVITE
From: "b2b" <sip:b2b@163.117.144.19:5060>;tag=IPTV_AS
To: <sip:student1@163.117.144.210:5060>
Via: SIP/2.0/UDP
163.117.144.19:5060;branch=z9hG4bK62bc0ff6c6e61721e000c03d27d4031d
Max-Forwards: 70
Contact: "b2b" <sip:b2b@163.117.144.19:5060>
Content-Type: application/sdp
Content-Length: 106
```

```
v=0
o=student 13 13 IN IP4 163.117.144.26
c=IN IP4 163.117.144.26
m=video 65070 udp mpeg
a=bc_service:5
```

De este mensaje podemos extraer:

- El IPTV-AS obtiene quien va a ser el servidor de *Estudiante 3* y es a quien va dirigido este mensaje como se puede ver en la Request-Line

Mensaje 200 OK desde Estudiante 1 a IPTV-AS

```
SIP/2.0 200 OK
Call-ID: 294f9efe20588bace85df3f51c2d05bb@163.117.144.19
CSeq: 1 INVITE
From: "b2b" <sip:b2b@163.117.144.19:5060>;tag=IPTV_AS
To: <sip:student1@163.117.144.210:5060>;tag=Server
Via: SIP/2.0/UDP
163.117.144.19:5060;branch=z9hG4bK62bc0ff6c6e61721e000c03d27d4031d
Contact: <sip:student@163.117.144.210:5060>
Content-Type: application/sdp
Content-Length: 108
```

```
v=0
o=student 10 10 IN IP4 163.117.144.210
c=IN IP4 163.117.144.210
m=video 65070 udp mpeg
a=bc_service:5
```

En este mensaje podemos destacar que en la carga SDP se establecen los nuevos parámetros sobre los cuales se va a establecer la sesión.

Mensaje 200 OK desde IPTV-AS a Estudiante 3

```
SIP/2.0 200 OK
Call-ID: b2f99e76a14efb8963f8af4e49e14ccb@163.117.144.26
CSeq: 1 INVITE
From: "student1" <sip:student1@163.117.144.26:5060>;tag=Student
To: "b2b" <sip:b2b@163.117.144.19:5060>;tag=IPTV_AS
Via: SIP/2.0/UDP
163.117.144.26:5060;branch=z9hG4bKe88971d121078db7ddeaae27489b3918
Contact: <sip:b2b@163.117.144.19:5060>
Content-Type: application/sdp
Content-Length: 108
```

```
v=0
o=student 10 10 IN IP4 163.117.144.210
c=IN IP4 163.117.144.210
m=video 65070 udp mpeg
a=bc_service:5
```

Mensaje ACK desde Estudiante 3 a IPTV-AS

```
ACK sip:b2b@163.117.144.19:5060 SIP/2.0
Call-ID: b2f99e76a14efb8963f8af4e49e14ccb@163.117.144.26
CSeq: 1 ACK
Via: SIP/2.0/UDP
163.117.144.26:5060;branch=z9hG4bK005e1e876de0e26e02accd1cd5483962
From: "student" <sip:student@163.117.144.26:5060>;tag=Student
To: "b2b" <sip:b2b@163.117.144.19:5060>;tag=IPTV_AS
Max-Forwards: 70
Content-Length: 0
```

Mensaje ACK desde IPTV-AS a Estudiante 1

```
ACK sip:student1@163.117.144.210:5060 SIP/2.0
Call-ID: 294f9efe20588bace85df3f51c2d05bb@163.117.144.19
CSeq: 1 ACK
Via: SIP/2.0/UDP
163.117.144.19:5060;branch=z9hG4bK8a1c670cf9e13b1160adcc22499afad7
From: "b2b" <sip:b2b@163.117.144.19:5060>;tag=IPTV_AS
To: <sip:student@163.117.144.210:5060>;tag=Server
Max-Forwards: 70
Content-Length: 0
```

Llegados a este punto podemos concluir que el establecimiento de sesión ha concluido de acuerdo a los requisitos especificados y comienza la transmisión del vídeo como podemos ver en la siguiente captura.

Time	Source	Destination	Protocol	Length	Source Port	Destination Port
269	29.692347	163.117.144.210	163.117.144.26	MPEG TS	1358	source port: 65072 destination port: 65070
270	29.697449	163.117.144.210	163.117.144.26	MPEG TS	1358	source port: 65072 destination port: 65070
271	29.702603	163.117.144.210	163.117.144.26	MPEG TS	1358	source port: 65072 destination port: 65070

<ul style="list-style-type: none"> ⊞ Frame 269: 1358 bytes on wire (10864 bits), 1358 bytes captured (10864 bits) ⊞ Ethernet II, Src: Asiarock_32:08:a9 (00:19:66:32:08:a9), Dst: Intelcor_19:63:0f (00:19:d1:19:63:0f) ⊞ Internet Protocol Version 4, Src: 163.117.144.210 (163.117.144.210), Dst: 163.117.144.26 (163.117.144.26) ⊞ User Datagram Protocol, Src Port: 65072 (65072), Dst Port: 65070 (65070) ⊞ ISO/IEC 13818-1 PID=0x44 CC=5 Reassembled in: 275 ⊞ ISO/IEC 13818-1 PID=0x44 CC=6 Reassembled in: 275 ⊞ ISO/IEC 13818-1 PID=0x44 CC=7 Reassembled in: 275 ⊞ ISO/IEC 13818-1 PID=0x44 CC=8 Reassembled in: 275 ⊞ ISO/IEC 13818-1 PID=0x45 CC=11 Reassembled in: 273 ⊞ ISO/IEC 13818-1 PID=0x44 CC=9 Reassembled in: 275 ⊞ ISO/IEC 13818-1 PID=0x44 CC=10 Reassembled in: 275

Figura 7.5: Transmisión del vídeo

7.3. Liberación de la sesión con más de 2 estudiantes

Partiendo de la conexión anteriormente establecida y pasados unos segundos después de la conexión de *Estudiante 3*, uno de los usuarios decide abandonar la sesión enviando un BYE al IPTV-AS.

Dependiendo del estudiante que quiera abandonar la sesión podemos tener dos situaciones de liberación de la sesión:

- Si *Estudiante 1* decide abandonar la sesión, como *Estudiante 3* depende del vídeo recibido desde *Estudiante 1*, el IPTV-AS tendrá que buscarle un nuevo servidor para que pueda seguir recibiendo el contenido multimedia y no se vea afectado por el abandono de su antiguo servidor de contenidos (figura 7.6).

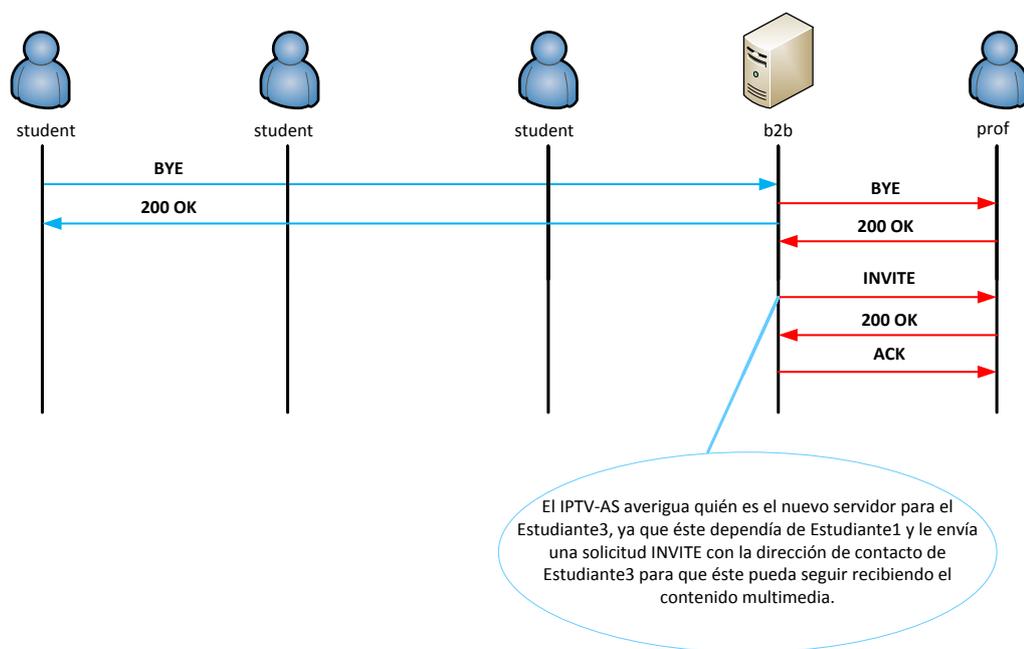


Figura 7.6: Liberación sesión de un cliente/servidor al mismo tiempo

- Si *Estudiante 2* o *Estudiante 3* deciden abandonar la sesión, como no están sirviendo su contenido, su abandono no afecta a terceros (figura 7.7).

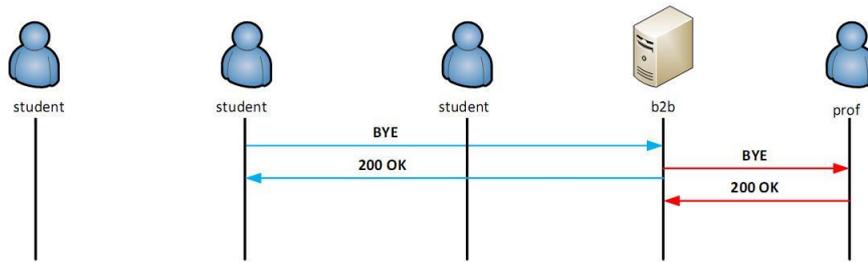


Figura 7.7: Liberación de sesión

Validación y análisis de los resultados obtenidos

1. Liberación de la sesión de un cliente/servidor al mismo tiempo

El contenido de los mensajes que se cursaron para la liberación de sesión de *Estudiante 1* fueron los siguientes (figura 7.8).

7028	56.960677	163.117.144.210	163.117.144.19	SIP	399 Request: BYE sip:b2b@163.117.144.19:5060
7030	56.967984	163.117.144.19	163.117.144.210	SIP	356 Status: 200 OK

Figura 7.8: Liberación de sesión de Estudiante 1

Mensaje BYE desde Estudiante 1 a IPTV-AS

```
BYE sip:b2b@163.117.144.19:5060 SIP/2.0
Via: SIP/2.0/UDP
163.117.144.210:5060;branch=z9hG4bK22c255e43d829b75f12b7a674804c777
CSeq: 2 BYE
Call-ID: e0d4c6d494f36cd265f7e3767c17e2a0@163.117.144.210
From: "student" <sip:student@163.117.144.210:5060>;tag=Student
To: "b2b" <sip:b2b@163.117.144.19:5060>;tag=IPTV_AS
Max-Forwards: 70
Content-Length: 0
```

Cuando este mensaje le llega al IPTV-AS busca un nuevo servidor para los clientes que dependieran de éste, en este caso como se va un usuario conectado directamente al profesor, Estudiante 3 pasaría a ser un nuevo cliente directo del profesor.

Además el IPTV-AS también mandará un mensaje *BYE* al servidor del cliente que abandona la sesión para que deje de transmitirle el contenido multimedia.

Mensaje BYE desde IPTV-AS al Servidor(Profesor)

```
BYE sip:prof@192.168.0.197:5060 SIP/2.0
Via: SIP/2.0/UDP
163.117.144.197:5060;branch=z9hG4bK32b2ed0651a10b827264b5c3633d9601
CSeq: 2 BYE
Call-ID: 65d84f8eb49f9ae299cbae025ece2922@192.168.0.197
From: "b2b" <sip:b2b@163.117.144.19:5060>;tag=IPTV_AS
To: "prof" <sip:prof@192.168.0.197:5060>;tag=Server
Max-Forwards: 70
Content-Length: 0
```

Mensaje 200 OK desde el Servidor a IPTV-AS

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP
163.117.144.197:5060;branch=z9hG4bK32b2ed0651a10b827264b5c3633d9601
CSeq: 2 BYE
Call-ID: e0d4c6d494f36cd265f7e3767c17e2a0@163.117.144.210
From: "b2b" <sip:b2b@163.117.144.19:5060>;tag=IPTV_AS
To: "prof" <sip:prof@192.168.0.197:5060>;tag=Server
Content-Length: 0
```

A continuación se enviará la confirmación (*200 OK*) al *BYE* del cliente que abandona la sesión.

Mensaje 200 OK desde IPTV-AS a Estudiante 1

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP
163.117.144.210:5060;branch=z9hG4bK22c255e43d829b75f12b7a674804c777
CSeq: 2 BYE
Call-ID: e0d4c6d494f36cd265f7e3767c17e2a0@163.117.144.210
From: "student" <sip:student@163.117.144.210:5060>;tag=Student
To: "b2b" <sip:b2b@163.117.144.19:5060>;tag=IPTV_AS
Content-Length: 0
```

Y para terminar el IPTV-AS genera un nuevo establecimiento de sesión con el nuevo servidor (los mensajes de establecimiento son similares a los comentados en el apartado [7.2](#)).

2. Liberación de sesión cliente-servidor

Para este caso, la liberación de sesión sería como la descrita el punto anterior a excepción del nuevo establecimiento de sesión con el nuevo servidor, ya que en esta situación no hay ningún usuario dependiendo del cliente que abandona la sesión.

Llegados a este punto podemos concluir que la liberación de la sesión ha concluido de acuerdo a los requisitos especificados.

7.4. Tiempos medios de conexión de los usuarios

En este apartado comentaremos la media de los tiempos de conexión de los clientes en el sistema. Para ello hemos realizado un estudio tomando 25 muestras de ejecuciones con el sistema definido para los casos de prueba.

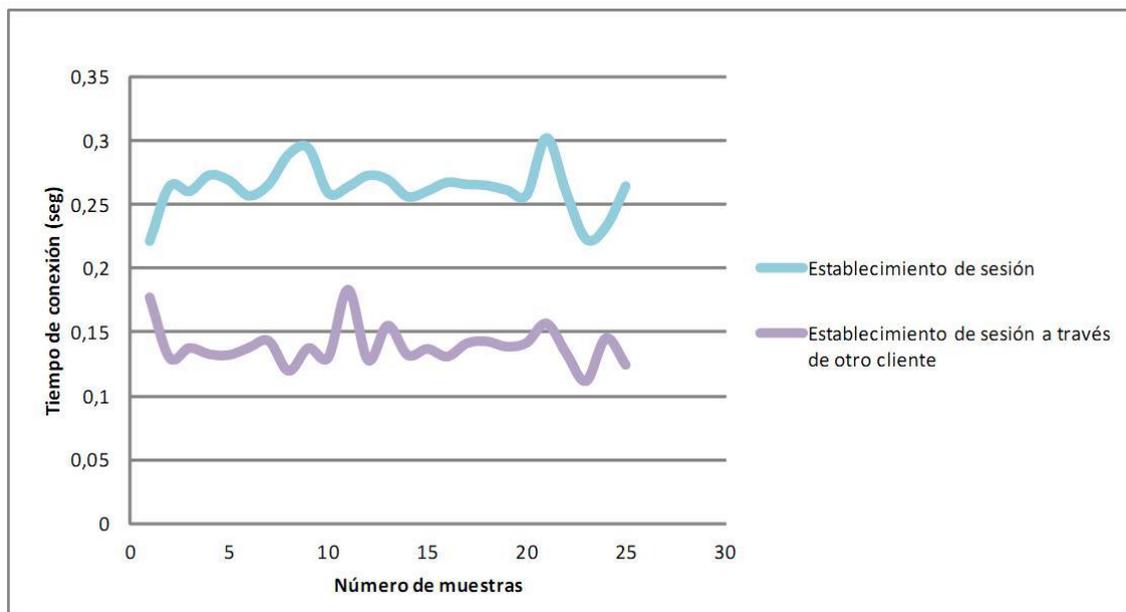


Figura 7.9: Tiempos de conexión de los clientes

Como podemos observar en la figura 7.9 el tiempo de conexión teniendo como servidor a un cliente que ha establecido previamente su sesión en el sistema, es menor que en el caso de que un cliente este conectado directamente al profesor.

En media, los tiempos de establecimiento de sesión son los siguientes:

- Estudiante-Profesor: 0,262623 segundos
- Estudiante-Estudiante: 0,138207 segundos

Esto es debido a que la primera vez que se inicia la SIP Stack tenemos un pequeño retardo de configuraciones/carga, que es lo que hace que los tiempos entre una conexión y otra sean tan dispares. En el segundo caso (Estudiante-Estudiante) el establecimiento de sesión se realiza contra un estudiante que ya ha inicializado la SIP Stack, de modo que el tiempo se reduce sensiblemente (la aportación a las 138 ms la hace prácticamente el estudiante iniciador que se conecta por primera vez).

Parte IV

Conclusiones y futuras implementaciones

Capítulo 8

Conclusiones y futuras implementaciones

8.1. Conclusiones

El objetivo de este Trabajo Fin de Grado ha sido la implementación de un sistema IPTV para retransmitir las clases docentes a un número potencialmente elevado de usuarios. Para ello se han diferenciado dos partes en el sistema: un plano de control o señalización en donde se establece la sesión y su posterior liberación mediante el protocolo de señalización SIP, y un plano de datos basado en el intercambio de datos P2P entre usuarios que nos ha permitido distribuir el contenido de unos a otros.

Se ha dispuesto de un servidor de contenidos o UAS (*User Agent Server*) al que le llegan peticiones procedentes de un IPTV-AS para proporcionar dicho contenido a distintos usuarios. El IPTV-AS, que es un B2BUA (*Back to Back User Agent*) de SIP, es capaz de procesar las peticiones que recibe de los usuarios, de responderlas y de generar nuevas peticiones para los destinatarios correspondientes.

Para el plano de control, se han implementado los usuarios o clientes SIP, el profesor, que aprovechando la modularidad de la implementación se ha implementado usando parte del código del estudiante, y el IPTV-AS que fueron desarrollados en JAVA. Por lo que el sistema consta de 1 IPTV-AS, de 1 servidor de contenidos que recibe las primeras peticiones para proporcionar los contenidos y de 1 o más usuarios del sistema.

Además la parte del cliente es accesible via web, ya que la implementación realizada es integrable en la plataforma de gestión de contenido docente de la UC3M, Aula Global.

Para el plano de datos, el envío del contenido se realiza mediante datagramas UDP sobre IP que son distribuidos usando un enfoque P2P. Para ello, una vez establecida la sesión SIP entre el usuario y su servidor de contenidos, pudiendo ser este el profesor u otro estudiante, la difusión de éste es de manera directa entre ambos utilizando Sockets de JAVA para gestionar el envío y la recepción del mismo,

además del framework *vlcj* de JAVA que es el que nos ha permitido la visualización del contenido multimedia en el cliente.

Por lo tanto podemos afirmar que se han logrado los objetivos principales que se plantearon al inicio de este Trabajo Fin de Grado.

8.2. Futuras implementaciones

Como posibles mejoras del desarrollo implementado que no se han podido llevar a cabo dado el alcance de este Trabajo Fin de Grado, se proponen los siguientes puntos:

- Gestionar el abandono desordenado del sistema por parte de los usuarios.
- Reestructuración dinámica del árbol ante la limitación del ancho de banda, es decir, si en un determinado momento algún usuario tiene problemas de red y su ancho de banda se ve decrementado, que fuera posible reubicar a sus clientes en otros nodos del sistema.
- Diferentes algoritmos para la generación del árbol de distribución de modo que sea más óptimo, ya que en esta implementación se ha optado por un árbol binario con dos usuarios por cada nodo.
- Desarrollo de una interfaz gráfica para el profesor en la que pueda retransmitir directamente el contenido multimedia sin necesidad de programas auxiliares como es el caso del VLC.
- Implementación de una interfaz para dispositivos móviles en iOS o Android para poder seguir las clases desde cualquier lugar.

Parte V

Anexos

Anexo A

Planificación de tareas y presupuesto

A.1. Descomposición en tareas

En esta sección se va a presentar la descomposición de tareas que se han llevado a cabo para realizar este Trabajo Fin de Grado.

- **Tarea A: Estudio de los framework a utilizar en el desarrollo**
 - **Subtarea A.1: Interfaz JAIN SIP**
 - **Relación con otras tareas:** esta tarea da comienzo al proyecto.
 - **Duración:** 1 semana.
 - **Recursos:** Ingeniero 0.25 ingenieros/mes.
 - **Subtarea A.2: Framework VLCJ**
 - **Relación con otras tareas:** comienza después de la tarea A.1.
 - **Duración:** 1 semana.
 - **Recursos:** Ingeniero 0.25 ingenieros/mes.
- **Tarea B: Documentación y análisis del estado del arte**
 - **Subtarea B.1: Estudio de las peticiones y respuestas SIP**
 - **Relación con otras tareas:** comienza después de la tarea A.
 - **Duración:** 1 semana.
 - **Recursos:** Ingeniero 0.25 ingenieros/mes.
 - **Subtarea B.2: Estudio del modelo de oferta/respuesta para la carga SDP**
 - **Relación con otras tareas:** comienza después de la tarea B.1.
 - **Duración:** 1 semana.
 - **Recursos:** Ingeniero 0.25 ingenieros/mes.
 - **Subtarea B.3: Estudio de las diferentes opciones para la distribución de IPTV por la red**

- **Relación con otras tareas:** comienza después de la tarea B.2.
 - **Duración:** 1 semana.
 - **Recursos:** Ingeniero 0.25 ingenieros/mes.
- **Tarea C: Identificación de los requisitos del sistema a implementar**
 - **Subtarea C.1:** Estudio de los requisitos del cliente
 - **Relación con otras tareas:** comienza después de las tareas A y B.
 - **Duración:** 1 semana.
 - **Recursos:** Ingeniero 0.25 ingenieros/mes.
 - **Subtarea C.2:** Estudio de los requisitos del servidor
 - **Relación con otras tareas:** comienza después de la tarea C.1.
 - **Duración:** 1 semana.
 - **Recursos:** Ingeniero 0.25 ingenieros/mes.
 - **Subtarea C.3:** Estudio de los requisitos del IPTV-AS
 - **Relación con otras tareas:** comienza después de la tarea C.2.
 - **Duración:** 1 semana.
 - **Recursos:** Ingeniero 0.25 ingenieros/mes.
 - **Subtarea C.4:** Estudio de los requisitos del plano de datos
 - **Relación con otras tareas:** comienza después de la tarea C.3.
 - **Duración:** 1 semana.
 - **Recursos:** Ingeniero 0.25 ingenieros/mes.
 - **Tarea D: Implementación del sistema**
 - **Subtarea D.1:** Desarrollo del establecimiento de sesión SIP entre los diferentes elementos del sistema
 - **Relación con otras tareas:** comienza después de la tarea C.
 - **Duración:** 1 mes.
 - **Recursos:** Ingeniero 1 ingenieros/mes.
 - **Subtarea D.2:** Desarrollo del plano de datos entre el cliente y el servidor
 - **Relación con otras tareas:** comienza después de la tarea D.1.
 - **Duración:** 1 mes.
 - **Recursos:** Ingeniero 1 ingenieros/mes.
 - **Subtarea D.3:** Desarrollo del interfaz del applet del cliente
 - **Relación con otras tareas:** comienza después de las tareas D.1 y D.2.
 - **Duración:** 2 semanas.
 - **Recursos:** Ingeniero 0.5 ingenieros/mes.
 - **Subtarea D.4:** Desarrollo de la desconexión de los elementos del sistema
 - **Relación con otras tareas:** comienza después de la tarea D.3

- **Duración:** 1 semana.
- **Recursos:** Ingeniero 0.25 ingenieros/mes.
- **Tarea E: Pruebas y medidas**
 - **Subtarea E.1:** Pruebas realizadas
 - **Relación con otras tareas:** comienza después de la tarea D.
 - **Duración:** 1 semana.
 - **Recursos:** Ingeniero 0.25 ingenieros/mes.
 - **Subtarea E.2:** Medidas obtenidas
 - **Relación con otras tareas:** comienza después de la tarea E.1.
 - **Duración:** 0.5 semanas.
 - **Recursos:** Ingeniero 0.25 ingenieros/mes.
- **Tarea F: Memoria**
 - **Subtarea F.1:** Organización y estructura de la memoria
 - **Relación con otras tareas:** comienza después de la tarea E.
 - **Duración:** 1 semana.
 - **Recursos:** Ingeniero 0.25 ingenieros/mes.
 - **Subtarea F.2:** Redacción de la memoria
 - **Relación con otras tareas:** comienza después de la tarea F.1.
 - **Duración:** 6 semanas.
 - **Recursos:** Ingeniero 0.75 ingenieros/mes.
 - **Subtarea F.3:** Redacción del resumen
 - **Relación con otras tareas:** comienza después de la tarea F.2.
 - **Duración:** 1 semana.
 - **Recursos:** Ingeniero 0.25 ingenieros/mes.

Tarea	Duración (semanas)	Recursos (Ing/m)
Estudio de los framework a utilizar en el desarrollo		
A.1 Interfaz JAIN SIP	1	0.25
A.2 Framework VLCJ	1	0.25
Total		0.5
Documentación y análisis del estado del arte		
B.1 Estudio de las peticiones y respuestas SIP	1	0.25
B.2 Estudio del modelo de oferta/respuesta para la carga SDP	1	0.25
B.3 Estudio de las diferentes opciones para la distribución de IPTV por la red	1	0.25
Total		0.75
Identificación de los requisitos del sistema a implementar		
C.1 Estudio de los requisitos del cliente	1	0.25
C.2 Estudio de los requisitos del servidor	1	0.25
C.3 Estudio de los requisitos del IPTV-AS	1	0.25
C.4 Estudio de los requisitos del plano de datos	1	0.25
Total		1
Implementación del sistema		
D.1 Desarrollo del establecimiento de sesión SIP entre los diferentes elementos del sistema	5	1
D.2 Desarrollo del plano de datos entre el cliente y el servidor	5	1
D.3 Desarrollo del interfaz del applet del cliente	2	0.5
D.4 Desarrollo de la desconexión de los elementos del sistema	1	0.5
Total		3
Pruebas y medidas		
E.1 Pruebas realizadas	1	0.25
E.2 Medidas obtenidas	0.5	0.25
Total		0.5
Memoria		
G.1 Organización y estructura de la memoria	1	0.25
G.2 Redacción de la memoria	6	0.75
G.3 Redacción del resumen	1	0.25
Total		1.25
Total		7

Tabla A.1: Resumen descomposición en tareas

A.2. Planificación con el diagrama de fases de ejecución detallado

En la figura A.1 se muestra el diagrama de Gantt con las tareas principales. Además en la figura A.2 se presenta el diagrama de Gantt con todas las tareas que se han realizado en este Trabajo Fin de Grado.

Id	Nombre de tarea	Duración	Comienzo	Fin	jul '12	sep '12	nov '12	ene '13	mar '13	may '13	jul '13
1	Tarea A: Estudio de los frameworks a utilizar en el desarrollo	14 días	lun 03/09/12	jue 20/09/12							
4	Tarea B: Documentación y análisis del estado del arte	21 días	vie 21/09/12	vie 19/10/12							
8	Tarea C: Identificación de los requisitos del sistema a implementar	28 días	lun 22/10/12	mié 28/11/12							
13	Tarea D: Implementación del sistema	81 días	jue 29/11/12	jue 21/03/13							
18	Tarea E : Pruebas y medidas	7 días	vie 22/03/13	lun 01/04/13							
21	Tarea F: Memoria	57 días	vie 05/04/13	lun 24/06/13							

Figura A.1: Diagrama de Gantt con la planificación del proyecto resumida

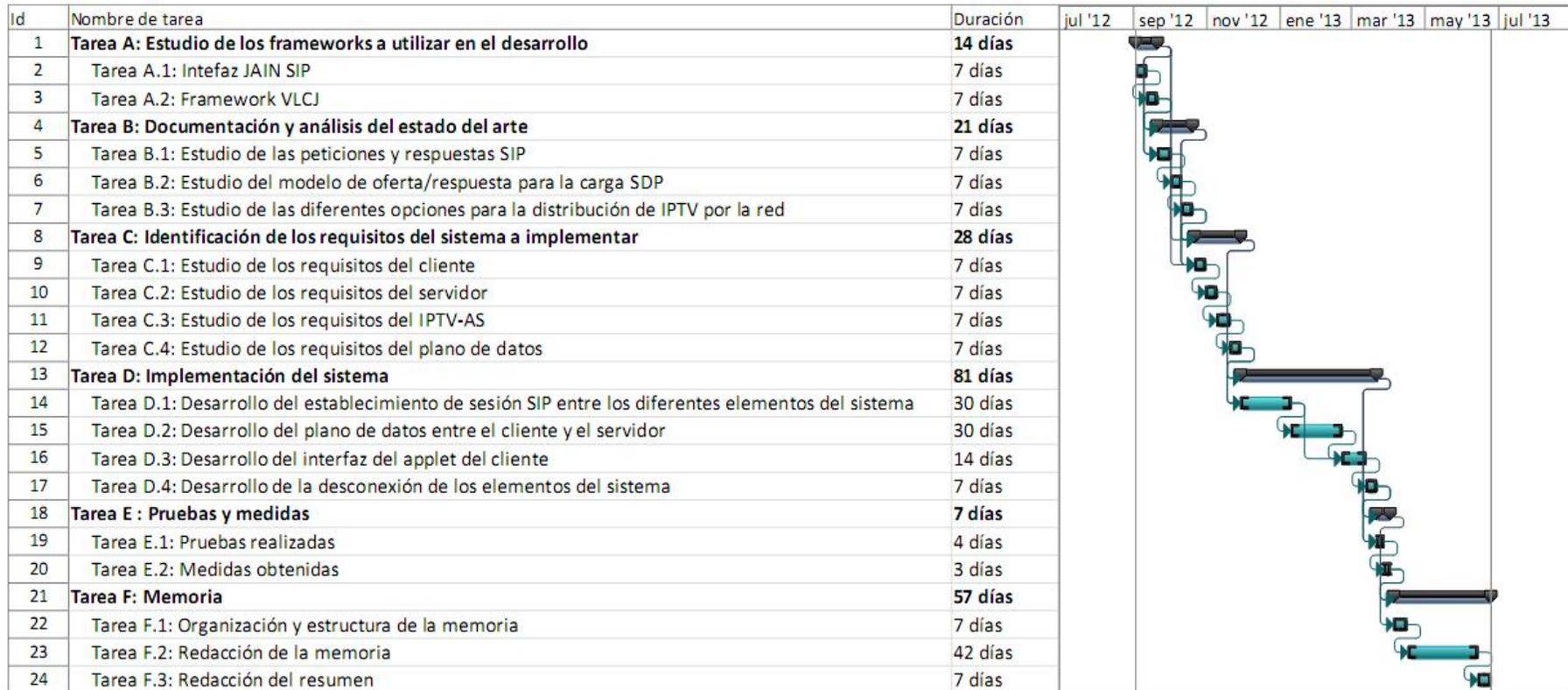


Figura A.2: Diagrama de Gantt con la planificación detallada del proyecto

A.3. Recursos

En esta sección se distinguen los diferentes recursos usados para la realización del Trabajo Fin de Grado:

- Recursos materiales:
 - Ordenadores de sobremesa.
 - 3 PCs *Intel*[®] *Pentium*[®] 4 3.00GHz, 2GB RAM, Sistema Operativo Microsoft Windows XP Service Pack 3
 - Ordenador portatil.
 - 1 *MacBookPro*[®] *Intel*[®] *Core*[™] i7 2.7GHz, 4GB RAM, Sistema Operativo Mac OS X 10.6.8 Snow Leopard
- Recursos de trabajo:
 - 1 Graduado en Ingeniería Telemática: 7 ingenieros/mes
- Otros costes:
 - Conexión a internet durante 10 meses.

A.4. Presupuesto de Proyecto

1. Autor: Sara Jiménez Carbajal
2. Departamento: Ingeniería Telemática
3. Descripción del Proyecto:
 - Título: Servicio IPTV para dar soporte a docencia basado en P2P y SIP
 - Duración: 10 meses
 - Tasa de costes indirectos: 20 %.
4. Presupuesto total del Proyecto (valorado en Euros): 19.765,4 euros. Ver tabla [A.2](#)
5. Subcontratación de tareas: no se especifican.
6. Otros costes directos del proyecto: no se especifican.

Concepto	Cantidad (€)	Coste €	% Proyecto	Dedicación (meses)	Depreciación (meses)	Total €
Recursos materiales						
Ordenadores de sobremesa	3	800	100	10	60	400
Ordenador portatil	1	1228	100	10	60	204,67
Total						604,67
Recursos de trabajo						
Graduado en Ing. Telemática	1 (7 ing/mes)	2.694,39	-	-	-	18.860,73
Total						18.860,73
Otros costes						
Conexión a internet	1	30	-	10	-	300
Total						300
Total						19.765,4 €

Tabla A.2: Tabla presupuesto

Anexo B

Manual de usuario

Es esta parte vamos a comentar como ejecutar la aplicación desarrollada. Podemos hacerlo de dos formas, bien desde un terminal de comandos o bien desde Eclipse (para ésta necesitaremos crear un proyecto dentro del programa que tendremos que tener en nuestro equipo). Para ambas ejecuciones vamos a tener una jerarquía de carpetas, con una carpeta principal y otras 3 dentro de ésta (src, bin y lib).

- **Antes de lanzar la aplicación.** Necesitaremos tener instalado una versión de JAVA 1.6 o posterior. Además tendremos las siguientes librerías .jar en la carpeta *lib*: *concurrent.jar*, *jain-sip-ri-1.2.90.jar*, *jain-sip-api-1.2.jar*, *log4j-1.2.8.jar*, *jna.jar*, *platform.jar* y *vlcj-1.2.2.jar*. Por último necesitaremos dos librerías más que vienen dentro del programa *VLC* (las pruebas se han realizado con la versión 2.0.2) que son *libvl.dll* y *libvcore.dll*.
- **Pasos para arrancar la aplicación.** En primer lugar tendremos que compilar la aplicación (si utilizamos Eclipse, este paso no es necesario ya que este programa se encarga de hacerlo por nosotros) para así poder llevar a cabo su correcta ejecución generando los correspondientes ficheros *.class*. En un terminal de comandos, estando en la carpeta *src* de nuestro proyecto, ejecutaremos la siguiente sentencia:

```
<ruta de la versión de JAVA que estamos usando>/javac -d ../bin/  
-classpath ../lib/concurrent.jar:../lib/jain-sip-ri-1.2.90.jar:../lib/jain-sip-  
api-1.2.jar:../lib/log4j-1.2.8.jar:../lib/jna.jar:../lib/platform.jar:../lib/vlcj-  
1.2.2.jar:. ActualStateDialog.java ElementHeap.java Heap.java  
IPTV_AS.java PendingOperations.java Professor.java Student.java Th-  
readMedia.java VideoRedirection.java VideoSockets.java
```

A continuación lanzaremos la aplicación que hace de UAS que se compone de dos partes. Para su ejecución haremos lo siguiente:

- En un terminal ejecutamos el siguiente comando:


```
<ruta de la versión de JAVA que estamos usando>/java -classpath
../bin:../concurrent.jar:../jain-jain-sip-ri-1.2.90.jar:../jain-sip-api-
1.2.jar:../log4j-1.2.8.jar:../jna.jar:../platform.jar:../vlcj-1.2.2.jar:
Professor <nombre_profesor>
```

 - nombre_profesor: nombre con el que identificar al profesor durante la sesión.
- En un navegador o en su defecto Eclipse, lanzamos el aplicativo del applet que será el encargado de hacer de servidor(profesor) en primera instancia.

Una vez arrancado el UAS, ahora lanzaremos el IPTV-AS con el siguiente comando:

```
<ruta de la versión de JAVA que estamos usando>/java -classpath
../bin:../concurrent.jar:../jain-jain-sip-ri-1.2.90.jar:../jain-sip-api-
1.2.jar:../log4j-1.2.8.jar:../jna.jar:../platform.jar:../vlcj-1.2.2.jar: IPTV_AS
<nombre_IPTV-AS> <nombre_profesor> <IP_profesor>
```

- nombre_IPTV-AS: nombre con el que identificar al IPTV-AS en la sesión.
- nombre_profesor: nombre con el que identificar al profesor en la sesión.
- IP_profesor: dirección IP del profesor (será la dirección IP del applet lanzado como UAS).

Para ejecutar el estudiante, lanzaremos el applet (bien desde Eclipse o bien desde un navegador web¹) y nos aparecerá la interfaz mostrada en la figura B.1.

Deberemos de pulsar en el botón *Conexión* para que comience el intercambio de la señalización entre los diferentes elementos del sistema y así poder empezar a ver el contenido multimedia solicitado.

¹Para ejecutarlo desde un navegador web tendrá que estar el aplicativo del cliente en un servidor web.

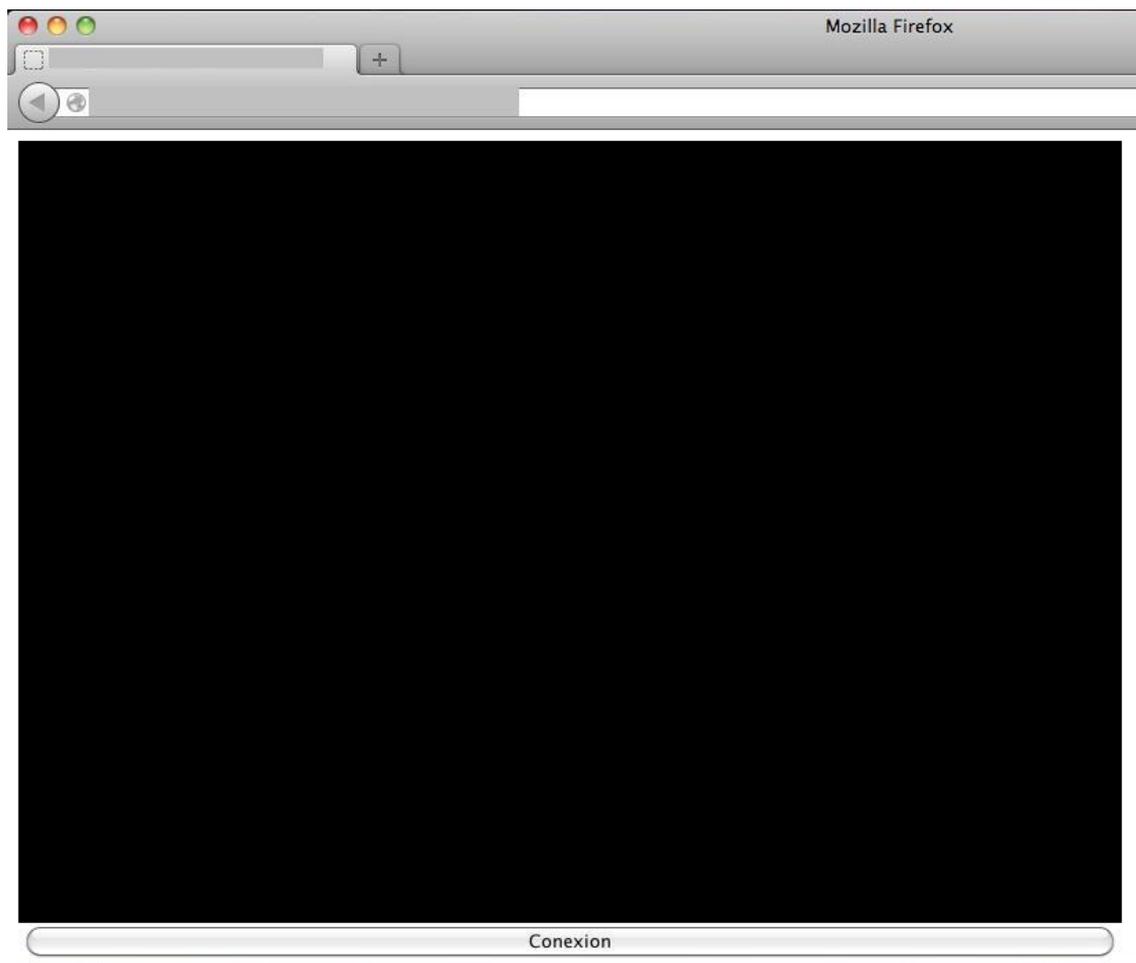


Figura B.1: Applet lanzado al estudiante

Anexo C

Solicitudes SIP

C.1. CANCEL

El método *CANCEL* se utiliza para terminar búsquedas pendientes o intentos de establecimiento de llamadas. Puede ser generado tanto por un UA como por proxies. Este método puede ser utilizado siempre que el servidor no haya enviado aún una respuesta final.

C.2. OPTIONS

El mensaje *OPTIONS* se utiliza para preguntar a cualquier UAC o UAS sobre las capacidades que soportan y para saber su disponibilidad. La respuesta a esta solicitud es una lista con las capacidades que soporta el UAC o UAS. Un proxy nunca genera una solicitud *OPTIONS*. El tipo de respuesta que recibe es similar a la que se recibe cuando se manda un *INVITE*.

C.3. SUBSCRIBE

El método *SUBSCRIBE* se usa por un UA para establecer una suscripción a un cierto servicio para recibir notificaciones (a través del método *NOTIFY*) sobre un evento en particular. Una vez que el usuario ha completado la suscripción se establece un diálogo entre UAC y UAS. La solicitud *SUBSCRIBE* contiene la cabecera *Expires* cuya función es fijar la duración de la suscripción. Una vez transcurrido este periodo, la suscripción caduca automáticamente.

El tipo de evento al que el usuario se suscribe se especifica en la cabecera *Event*. Cada evento define una serie de parámetros asociados a dicho evento. El servidor debe indicar que tipo de eventos soporta. Esta información se almacena en la cabecera *Allow-Events*.

C.4. NOTIFY

La solicitud *NOTIFY* es utilizada por un UA para notificar la ocurrencia de eventos. El *NOTIFY* siempre se envía dentro de un diálogo cuando existe una suscripción entre el usuario y el que notifica.

La respuesta al *NOTIFY* es un *200 OK* indicando que el mensaje ha sido recibido. Si se recibe un mensaje con el código *481 Dialog/Transaction Does Not Exist* la suscripción se termina automáticamente y no se envían más *NOTIFY*.

Estas peticiones contienen, a parte de las principales cabeceras, una cabecera llamada *Event* indicando el tipo de evento notificado y otra denominada *Subscription-State* que denota el estado actual de la suscripción.

Un *NOTIFY* se envía siempre al comienzo y al final de de la suscripción. Si esta solicitud contiene un cambio respecto al estado del usuario, el cuerpo del mensaje deberá contener un número de estado de la versión que se incrementará en una unidad por cada *NOTIFY* enviado. De esta manera el receptor de esta solicitud puede identificar si algún *NOTIFY* se ha perdido al estar fuera de la secuencia.

C.5. PRACK

El mensaje *PRACK* se utiliza para confirmar la recepción de las respuestas provisionales (con código 1xx). La fiabilidad de las respuestas con código 2xx, 3xx, 4xx, 5xx o 6xx se obtiene asintiendo al mandar un *ACK*.

En caso de tener respuestas provisionales tales como *180 Ringing*, es crítico determinar el estado de la llamada y confirmar la recepción de la respuesta provisional. Por tanto, el *PRACK* puede ser aplicado a todas las respuestas provisionales 1xx salvo para el *100 Trying*.

Esta solicitud la genera el UAC cuando se recibe una respuesta provisional que contiene las cabeceras RSeq (*Reliable Sequence Number*) y la cabecera *Supported:100rel*. Entonces el UAC manda un *PRACK* asintiendo la respuesta provisional y añade una cabecera *RAck* con el número de secuencia que recibió en la respuesta temporal y el método que la generó. Además incrementa el número de secuencia CSeq.

Una vez que el *PRACK* llega al destinatario, este responde con un *200 OK* para informar de que el mensaje ha llegado correctamente.

C.6. UPDATE

El mensaje *UPDATE* es un método que se utiliza para cambiar el estado de la sesión sin modificar el del diálogo. Puede ser utilizado antes de que se genere

la respuesta final a la solicitud *INVITE* y algunas de sus posibles utilidades son silenciar la llamada, poner la llamada en espera y dejar de enviar el flujo de datos o realizar negociaciones de parámetros de calidad de servicio extremo a extremo antes de establecerse la sesión.

C.7. REFER

El método *REFER* se utiliza por el UA para solicitar a otro UA el acceso a un recurso localizado en una URI o URL. El recurso es identificado por una URI o URL en la cabecera *Refer-To*.

C.8. MESSAGE

La solicitud *MESSAGE* se utiliza para transportar mensajes instantáneos (IM) utilizando SIP. Un IM consiste en un mensaje corto intercambiado entre los participantes de una sesión. Este mensaje puede ser enviado tanto dentro como fuera de un diálogo y no crea un diálogo adicional entre los participantes en la sesión.

El contenido del mensaje se transporta dentro del cuerpo del *MESSAGE* como un adjunto de tipo MIME. Todos los UAs que sean capaces de soportar el método *MESSAGE* han de soportar también el formato en texto plano y opcionalmente otros formatos tales como HTML, cpim (Common Presence and Instant Messaging), etc...

Si el mensaje se recibe de manera correcta, el receptor manda un mensaje *200 OK* de respuesta al originador del mensaje. En el cuerpo de este mensaje no se incluirá ningún IM. En caso de que el destino quiera contestar al iniciador con otro IM le mandará un *MESSAGE*.

C.9. PUBLISH

El mensaje *PUBLISH* es utilizado por un UA para enviar información de estado sobre un evento a un servidor (*ESC - Event State Compositor*). Este mensaje normalmente es usado cuando hay múltiples fuentes de eventos de información, tales como un número de fuentes compartiendo el mismo recurso.

C.10. INFO

El método *INFO* sirve para mandar información de señalización a otro usuario con el que tiene establecido la sesión. Al ser un mensaje extremo a extremo, un proxy siempre reenvía una solicitud de *INFO*. Normalmente este mensaje contiene un

cuerpo que almacena información de señalización y siempre incrementa el número de secuencia CSeq.

Anexo D

Campos SDP

D.1. Session name

El campo $s=$ contiene un nombre para la sesión y no puede quedarse vacío, al menos ha de tener un carácter.

D.2. Information

El campo $i=$ proporciona información adicional acerca de la sesión. Puede contener cualquier número de caracteres.

D.3. Email

El campo $e=$ contiene la dirección de correo electrónico de la persona que es responsable del establecimiento de la sesión.

D.4. Phone number

El campo $p=$ almacena un número de teléfono. Este número debe estar en formato globalizado, es decir, tiene que empezar por un + seguido del código del país, un espacio o - y a continuación el número local.

D.5. URI

El campo $u=$ contiene una URI con más información sobre la sesión.

D.6. Bandwidth

El campo $b=$ tiene información sobre el ancho de banda requerido. Tiene la siguiente forma:

$$b=modifier:bandwidth-value$$

El *modifier* puede ser CT cuando se tienen varias sesiones en paralelo de manera que indica el ancho de banda total que se puede usar por todos los participantes de la sesión o AS cuando se refiere al ancho de banda utilizado por una única sesión. El parámetro *bandwidth-value* es el número de kilobytes por segundo.

D.7. Time

Este campo $t=$ especifica los tiempos de comienzo y fin de la sesión. Los tiempos son especificados usando *timestamps* NTP (Network Time Protocol). El aspecto de esta línea es el siguiente:

$$t=<start-time> <stop-time>$$

Si los tiempos de comienzo y fin son cero significa que la sesión es permanente.

D.8. Repeat time

Esta línea $r=$ representa cada cuanto se tiene que repetir una determinada sesión.

D.9. Time zones

El campo $z=$ contiene información sobre los desfases de las zonas horarias.

D.10. Encryption keys

El campo $k=$ contiene la clave utilizada para la encriptación de los datos.

Anexo E

Establecimiento de sesión SIP

El establecimiento de sesión comienza cuando se envía una solicitud *INVITE* desde un cliente (Estudiante1, por ejemplo) tal como se ve en la figura 5.3. Este *INVITE* contiene las siguientes cabeceras:

- La línea inicial es una *Request URI* que contiene la URI SIP del IPTV-AS con el que tiene que conectarse el estudiante para conocer quién será su servidor de contenidos multimedia. En esta primera línea también se especifica cual es el protocolo de transporte a usar y la versión de SIP.
- La cabecera *Call-ID* contiene una cadena alfanumérica única que identifica a esta solicitud.
- La cabecera *CSeq* indica que esta solicitud es de tipo *INVITE*.
- Las cabeceras *From* y *To* indican el originario y destinatario de esta solicitud.
- Una cabecera *Via* en la que se almacena la dirección del estudiante originador de esta sesión.
- La cabecera *Max-Forwards* contiene el número máximo de servidores que puede atravesar esta solicitud antes de llegar a su destino.
- La cabecera *Contact* indica en qué dirección y puerto el estudiante quiere recibir las respuestas de este diálogo.
- Una cabecera *Content-Type* donde se identifica que esta petición tiene una carga SDP con información multimedia ya que su valor es *application/sdp*.

Cuando esta petición *INVITE* llega al IPTV-AS, en la estructura de árbol interno en el sistema se busca quien es el servidor (en esta sesión el servidor será el profesor tal como podemos comprobar en la figura 5.3) que debe de proporcionar el contenido multimedia al usuario que está especificado en la cabecera *Contact* de este mensaje. Una vez se conoce este dato, el IPTV-AS genera una nueva solicitud *INVITE* destinada a dicho servidor.

Una vez que el servidor (profesor) recibe el *INVITE*, genera una respuesta *200 OK* en la que se aceptan los parámetros de la sesión propuestos por el usuario originario del primer *INVITE*. Esta respuesta irá dirigida al IPTV-AS que será el encargado de reenviarla al cliente (Estudiante1).

Cuando el cliente recibe el *200 OK*, ya sabe quién va a ser su servidor de contenidos multimedia y responde con un mensaje *ACK* al IPTV-AS para notificar que ha recibido el *200 OK*. El IPTV-AS, a su vez, retransmite el *ACK* al servidor de contenidos para confirmarle la recepción del *200 OK*.

En el momento en que el servidor recibe el *ACK* empieza la transmisión directa del contenido multimedia entre el servidor y el cliente.

Anexo F

Configuración del VLC para el envío del primer contenido multimedia

Antes de comenzar con el envío del contenido multimedia, tenemos que tener el vídeo en formato *.mpg*. Este formato es el que soporta dicha aplicación para la transmisión de vídeo a través de UDP.

Una vez tengamos este requisito, tendremos que seguir los siguientes pasos dentro de la aplicación VLC para proceder con el envío del vídeo:

1. Cuando la sesión SIP esté establecida, en el equipo del servidor se lanzará la aplicación VLC de manera automática y nos aparecerá una pantalla como la siguiente:

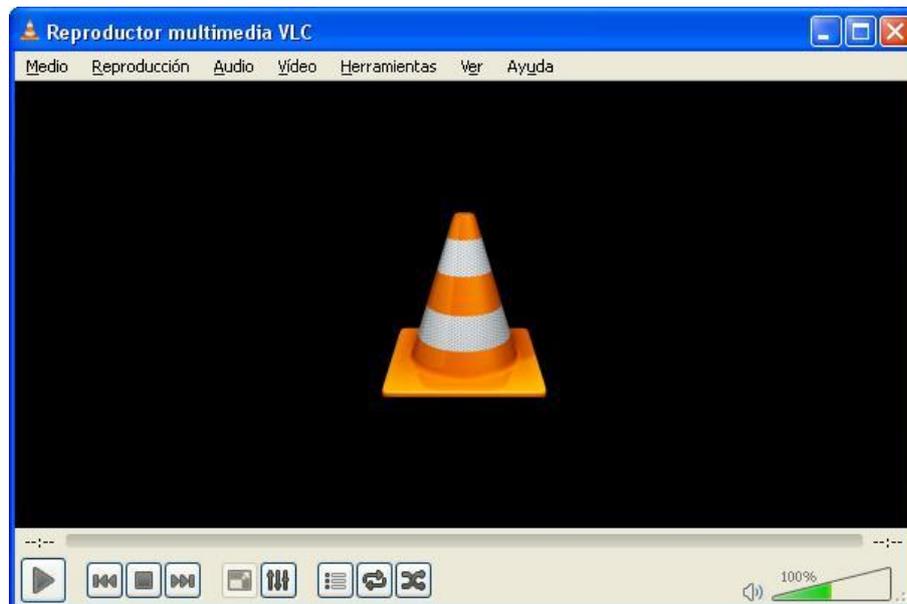


Figura F.1: Interfaz de la aplicación VLC

2. A continuación, pulsaremos sobre el botón **Medio** de la barra de herramientas

y en el menú desplegable seleccionaremos la opción de **Abrir archivo (avanzado)...** como podemos ver en la figura F.2.

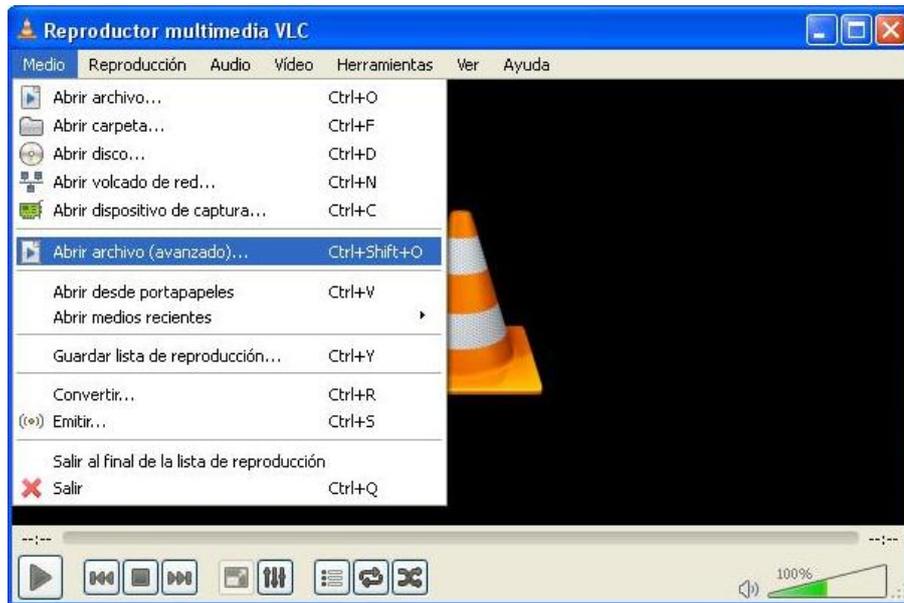


Figura F.2: Abrir archivo

3. Después de esto, nos aparecerá una interfaz nueva como la mostrada en la figura F.3. En esta pantalla tendremos que pulsar sobre el botón **Añadir** para buscar el archivo de vídeo que vamos a querer transmitir. Una vez hayamos encontrado el archivo, pulsamos sobre el botón **Abrir** para que dicho archivo nos aparezca como seleccionado tal y como podemos visualizar en la figura F.4.
4. A continuación pulsaremos sobre el desplegable del botón **Reproducir** y nos mostrará diferentes opciones, nosotros seleccionaremos **Emitir** e iremos a una nueva pantalla como la mostrada en la figura F.5.
5. En esta pantalla tendremos que pulsar sobre el botón **Siguiente** para visualizar las opciones de la pestaña **Configuración de destinos** mostradas en la figura F.6. En el desplegable de **nuevo destino** seleccionaremos la opción de **UDP** y pulsaremos sobre el botón **Añadir** que nos llevará a una nueva pestaña mostrada en la figura F.7. Aquí tendremos que establecer como dirección y puerto, la dirección IP y el puerto por la que nuestro cliente va a recibir el contenido multimedia en su equipo, es decir, la IP y el puerto que vienen en la carga SDP del mensaje *INVITE*.
6. Una vez que tengamos estos parámetros establecidos, pulsamos sobre el botón **Emitir** y comenzará la transmisión del vídeo a nuestro cliente como podemos ver en la figura F.8.
7. En el equipo del cliente comprobamos que se está recibiendo correctamente el contenido multimedia en el applet como podemos ver en la figura F.9

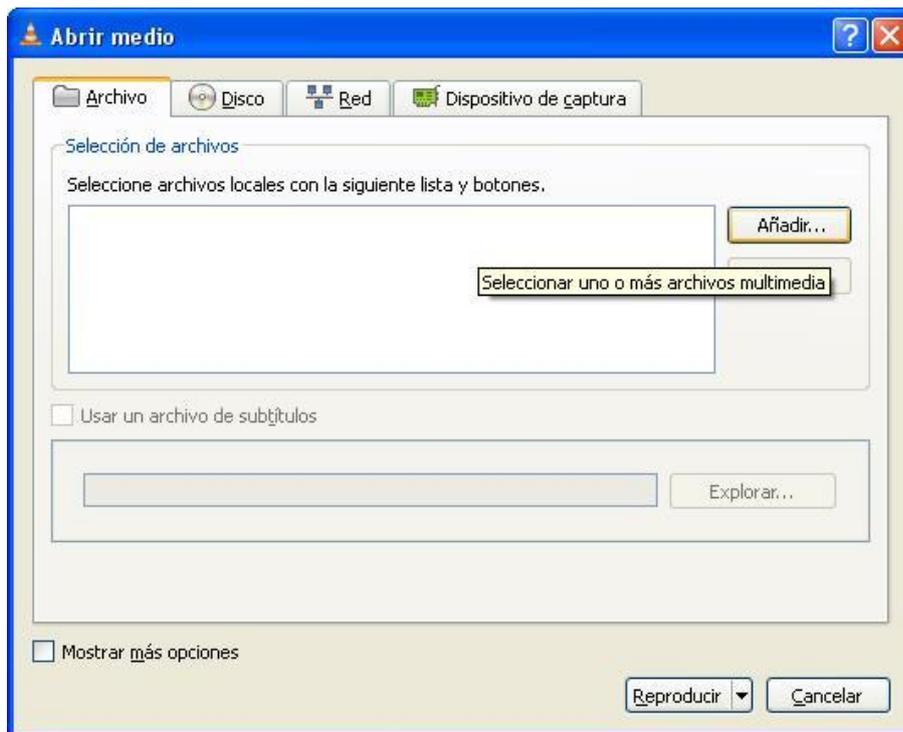


Figura F.3: Abrir medio

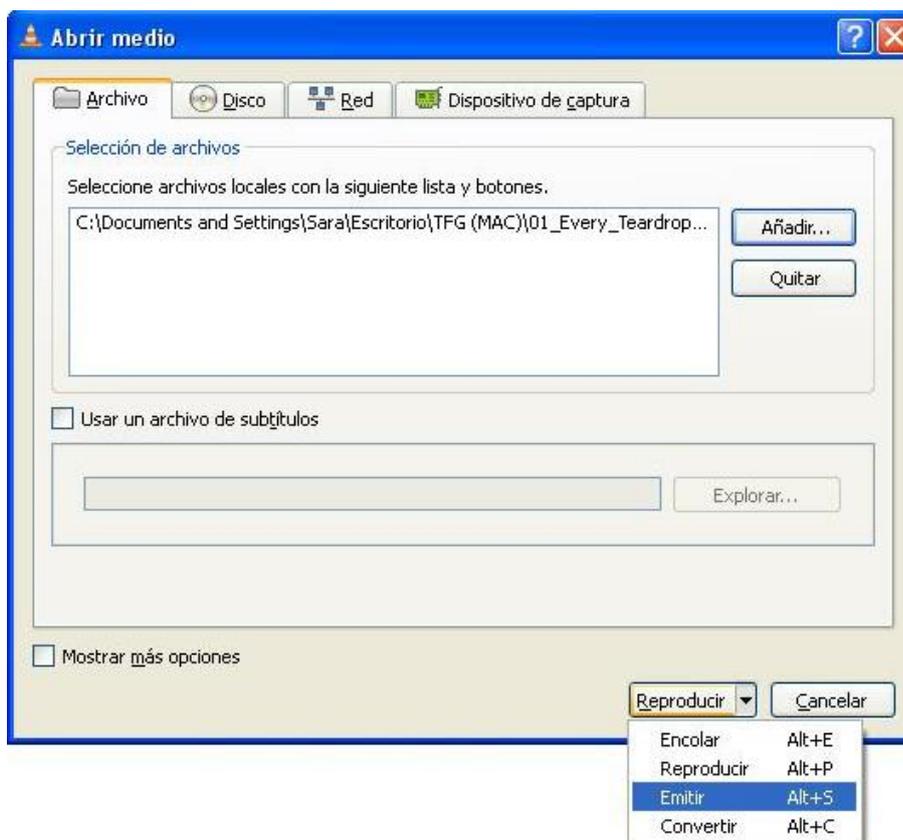


Figura F.4: Medio seleccionado

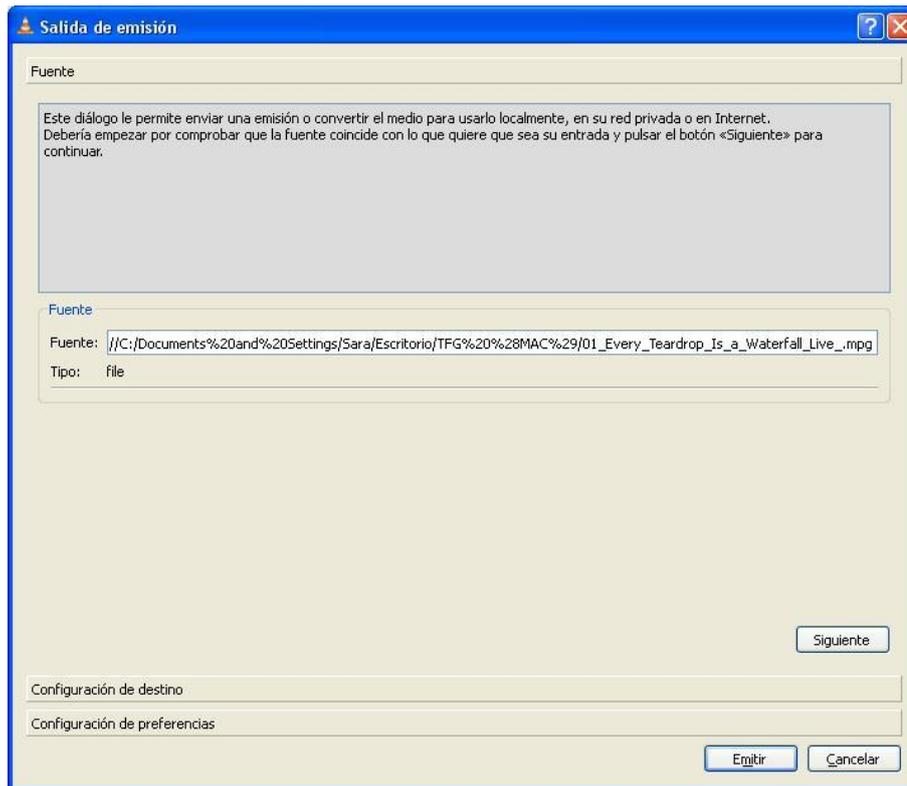


Figura F.5: Salida de emisión - Fuente

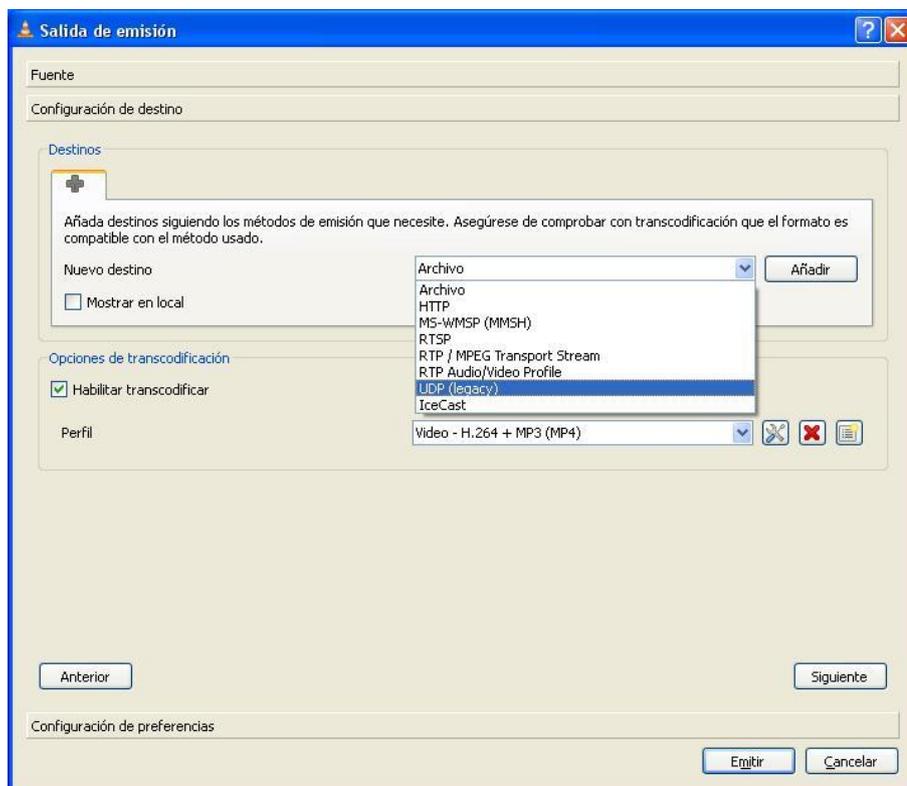


Figura F.6: Salida de emisión - Configuración de destinos

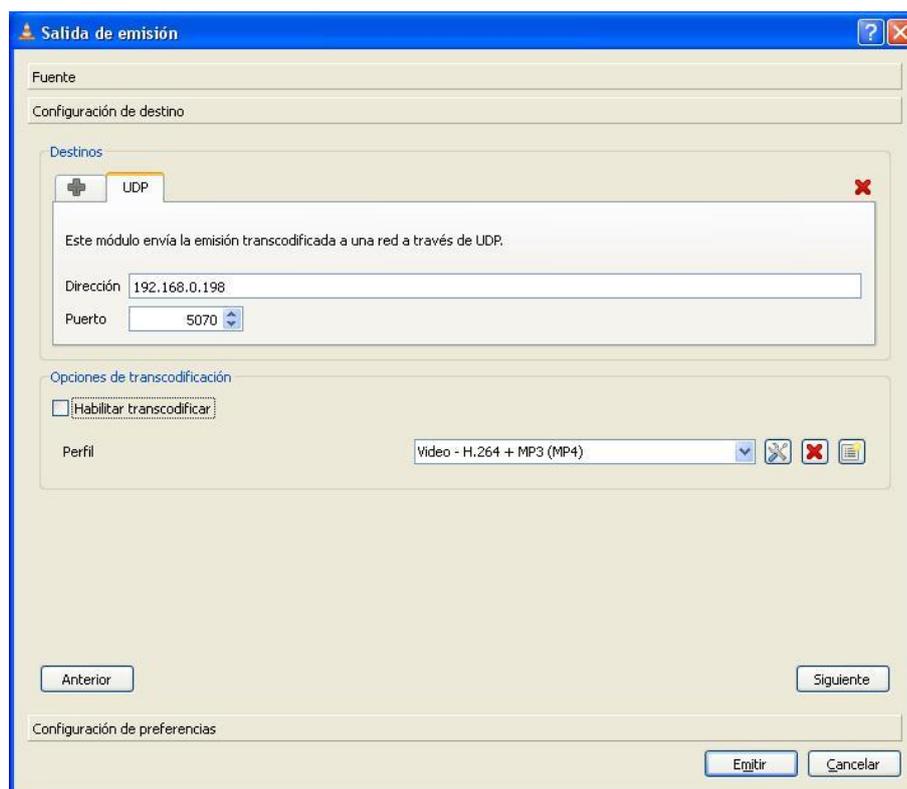


Figura F.7: Configuración de destinos - UDP

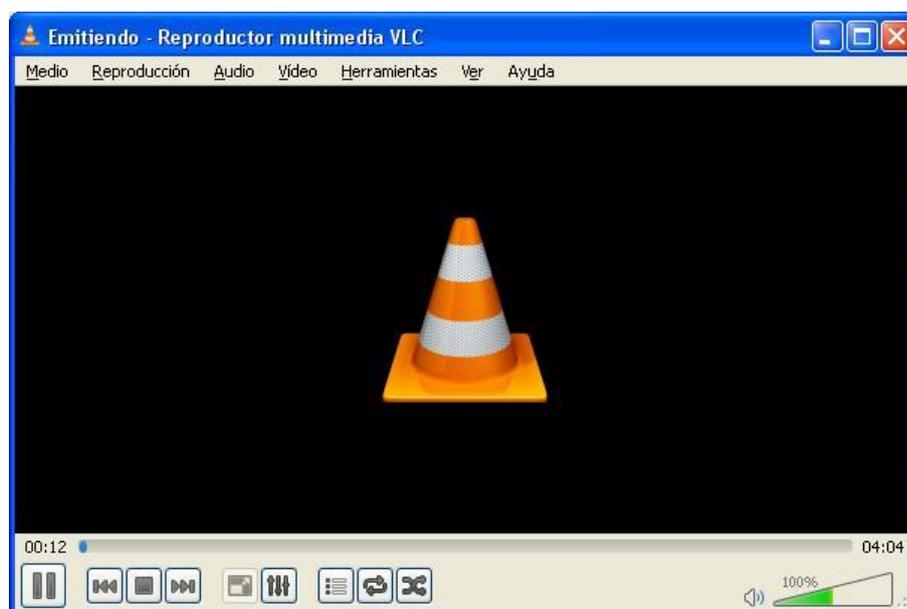


Figura F.8: Emisión del contenido multimedia

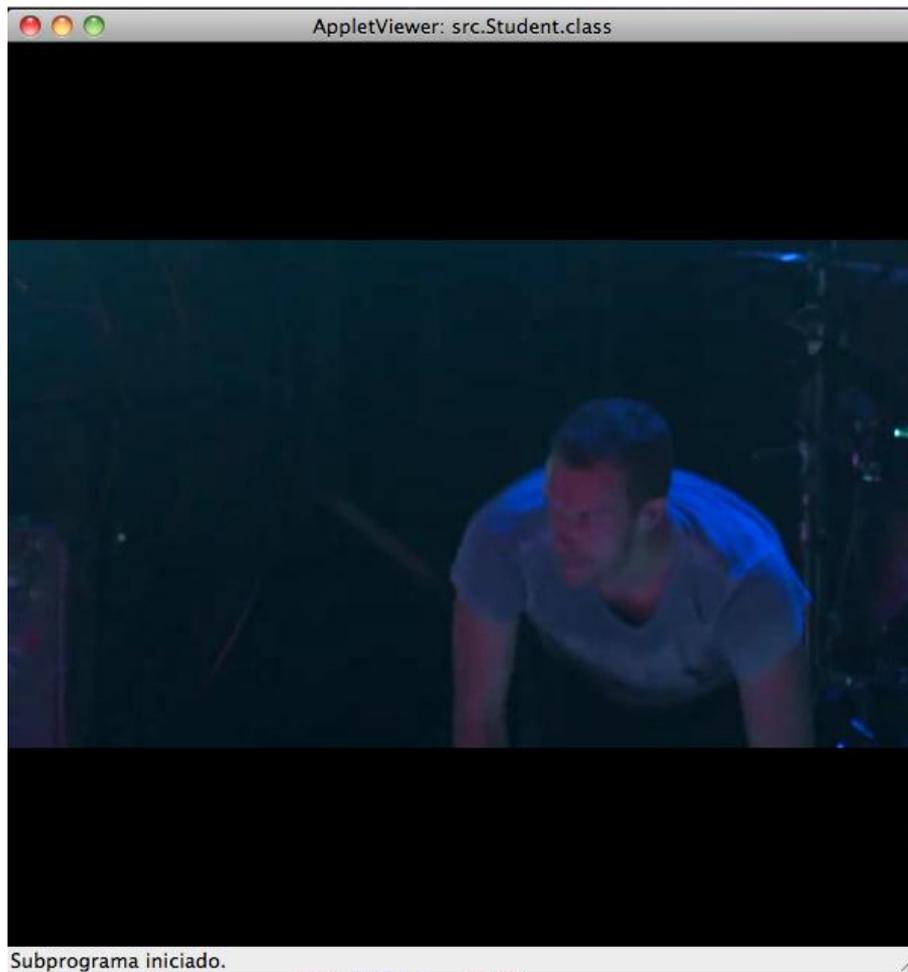


Figura F.9: Reproducción del contenido multimedia en el applet

Glosario

La lista de los acrónimos utilizados en este proyecto es la siguiente:

3GPP 3GPP 3rd Generation Partnership Project

AS Agent Server

ATIS IIF ATIS IPTV Interoperability Forum

B2BUA Back to Back User Agent

CBT Core-Based Trees

CoD Content on Demand

DNS Domain Name System

DVMRP Distance Vector Multicast Routing Protocol

DVB Digital Video Broadcasting

ETSI European Telecommunications Standards Institute

HTML HyperText Markup Language

HTTP Hypertext Transfer Protocol

IETF Internet Engineering Task Force

IGMP Internet Group Management Protocol

IM Instant Message

IMS IP Multimedia Subsystem

IPTV Internet Protocol Television

IPTV-AS Internet Protocol Television - Agent Server

ITU International Telecommunication Union

MIME Multipurpose Internet Mail Extensions

MOSPF Multicast Open Shortest Path First

- NGN** Next Generation Networks
- N-PVR** Network-Personal Video Recorder
- NTP** Network Time Protocol
- OSPF** Open Shortest Path First
- P2P** Peer to Peer
- P2PTV** Peer to Peer Television
- PIM-DM** Protocol Independent Multicast-Dense Mode
- PIM-SM** Protocol Independent Multicast-Sparse Mode
- RFC** Request For Comments
- RTCP** RTP Control Protocol
- RTP** Real Time Protocol
- RSVP** Resource ReSerVation Protocol
- SDP** Session Description Protocol
- SIP** Session Initiation Protocol
- SMTP** Simple Mail Transfer Protocol
- TCP** Transmission Control Protocol
- TISPAN** Telecommunications and Internet converged Services and Protocols for
Advanced Networking
- UA** User Agent
- UAC** User Agent Client
- UAS** User Agent Server
- UDP** User Datagram Protocol
- URI** Uniform Resource Identifier
- VoD** Video on Demand
- VoIP** Voice over Internet Protocol

Bibliografía

- [AA05] W. Siadak A. Adams, J. Nicholas. Protocol Independent Multicast - Dense Mode (PIM-DM): Protocol Specification (Revised). RFC 3973, January 2005.
- [AF09] I.Vidal A.Bikfalvi, J.García-Reinoso and F.Valera. A peer-to-peer IPTV service architecture for the IP multimedia subsystem. *International Journal of Communication Systems*, 2009.
- [APIa] Java™ Platform, Standard Edition 6 API Specification. <http://docs.oracle.com/javase/6/docs/api/>. última vez visitada Abril 2013.
- [APIb] The Reference Implementation for JAIN-SIP 1.2. <http://hudson.jboss.org/hudson/job/jain-sip/lastSuccessfulBuild/artifact/javadoc/index.html>. última vez visitada Abril 2013.
- [B. 02] B. Cain, S. Deering, I. Kouvelas, B. Fenner, A. Thyagarajan. Internet Group Management Protocol, Version 3. RFC 3376, October 2002.
- [Bal97] A. Ballardie. Core Based Trees (CBT version 2) Multicast Routing – Protocol Specification –. RFC 2189, September 1997.
- [BF06] H. Holbrook I. Kouvelas B. Fenner, M. Handley. Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised). RFC 4601, August 2006.
- [cis13] Cisco visual networking index: Forecast and methodology, 2012-2017. PDF, May 2013.
- [DW88] S. Deering D. Waitzman, C. Partridge. Distance Vector Multicast Routing Protocol. RFC 1075, November 1988.
- [EM11] D. Kanellopoulos E. Mikóczy, I. Vidal. IPTV Evolution Towards NGN and Hybrid Scenarios. *Informatica*, 2011.
- [ETS11a] ETSI. Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); IPTV Architecture; IIP TV functions supported by the IMS subsystem. ETSI TS 182 027 v3.5.1, 2011.
- [ETS11b] ETSI. Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); NGN integrated IPTV subsystem Architecture. ETSI TS 182 028 v3.5.1, 2011.

- [Gre09] Wes Simpson; Howard Greenfield. *IPTV and Internet Video: expanding the reach of television broadcasting*. Focal Press, 2009.
- [H. 03] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson. RTP: A Transport Protocol for Real-Time Applications. RFC 3550, July 2003.
- [IET] The internet engineering task force. <http://www.ietf.org> , última vez visitada Junio 2013.
- [Joh09] Allan B. Johnson. *SIP Understanding the Session Initiation Protocol*. Artech House, 2009.
- [JR02a] G. Camarillo A. Johnston J. Peterson R. Sparks M. Handley E. Schooler J. Rosenberg, H. Schulzrinne. SIP: Session Initiation Protocol. RFC 3261, June 2002.
- [JR02b] H. Schulzrinne J. Rosenberg. An Offer/Answer Model with the Session Description Protocol (SDP). RFC 3264, June 2002.
- [MH06] C. Perkins M. Handley, V. Jacobson. SDP: Session Description Protocol. RFC 4566, July 2006.
- [moo] Moodle. <http://moodle.org/> , última vez visitada Junio 2013.
- [Moy94] J. Moy. Multicast Extensions to OSPF. RFC 1584, March 1994.
- [PO03] Mudumbai Ranganathan Phelim O'Doherty. Jain sip tutorial, serving the developer community. PDF, 2003.
- [Pos80] J. Postal. User Datagram Protocol. RFC 768, 1980.
- [Pro07] Emmanuel Proulx. An Introduction to the JAIN SIP API. <http://www.oracle.com/technetwork/articles/entarch/introduction-jain-sip-090386.html>, última vez visitada Abril 2013, October 2007.
- [Roa02] A. Roach. Session Initiation Protocol (SIP) - Specific Event Notificacion. RFC 3265, June 2002.
- [RS05] Klaus Wehrle Ralf Steinmetz. *Peer-to-Peer Systems and Applications*. Springer Berlin Heidelberg, 2005.
- [Sav08] P. Savola. Overview of the Internet Multicast Routing Architecture. RFC 5110, January 2008.
- [tis] Telecommunications and Internet converged Services and Protocols for Advanced Networking.
- [VLCa] Videolan's wiki. <http://wiki.videolan.org>, última vez visitada Junio 2013,.
- [vlcb] vlcj: A framework for embedding the vlc media player in java applications. <http://vlcj.googlecode.com/svn/tags/vlcj-1.2.2/vlcj/javadoc/index.html>. última vez visitada Abril 2013.

-
- [W. 97] W. Fenner. Internet Group Management Protocol, Version 2. RFC 2236, November 1997.
- [Z. 01] Z. Albanna, K. Almeroth, D. Meyer, M. Schipper. IANA Guidelines for IPv4 Multicast Address Assignments. RFC 3171, August 2001.