



Universidad
Carlos III de Madrid

DEPARTAMENTO DE INGENIERÍA TELEMÁTICA

TRABAJO FIN DE GRADO

**DESARROLLO DE UN SERVICIO DE
TELEVISIÓN IP BASADO EN SIP Y
TECNOLOGÍAS DE ENTREGA
MULTICAST**

Autor: Cristina González Carrasco

Director: Iván Vidal Fernández

*También quería creer, quería que creyéramos.
Porque sabía que solo hace falta creer para conseguirlo todo.
(Un Intruso en mi cuaderno, David Fernández Siffes)*

Agradecimientos

En primer lugar, quiero agradecer su paciencia, apoyo y esfuerzo realizado a mis padres, sin ellos no hubiera llegado hasta donde he llegado. Por todos los momentos en los que me caía y ellos me ayudaban a levantarme. Por todos los buenos consejos que me han dado y el no permitirme nunca la idea de tirar la toalla, gracias. También agradecer a mi hermano, por estar ahí siempre. A mis tíos y primos, por ayudarme y apoyarme, por aguantar mis risas y llantos.

A todos mis amigos, que han sabido tenerme paciencia, cuando era casi imposible tenerla conmigo, por intentar despejarme en épocas de exámenes, así como durante esta última etapa intentando poner interés en mi proyecto, cuando quizá no se enteraban de nada, pero ahí han estado.

A toda la gente que he conocido durante estos 5 años, por los momentos buenos y también malos que hemos tenido. No quiero nombrar a nadie, porque probablemente alguien se me olvidará, pero las que sí que tienen que tener mención especial, son ellas, "mis titis", ¿Qué no hemos vivido juntas?. Más que compañeras de universidad, se han convertido en amigas, hermanas. Gracias a ellas este duro proceso se ha hecho más llevadero.

También agradecer a mi tutor Iván Vidal, por la comprensión y paciencia tenida conmigo.

EL TRIBUNAL

Presidente:

Vocal:

Secretario:

Realizado el acto de defensa y lectura del Trabajo Fin Grado el día..... de de en, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de:

VOCAL

SECRETARIO

PRESIDENTE

Resumen

En la actualidad, el desarrollo continuo de Internet ha permitido la aparición en el mercado de una gran diversidad de servicios multimedia, disponibles para los usuarios a través de la red. Entre los principales catalizadores que han propiciado el éxito de estos servicios destacan: el acceso ubico que ofrece Internet, que posibilita el acceso a estos servicios prácticamente en cualquier momento y desde cualquier lugar; el desarrollo de nuevas tecnologías de acceso de banda ancha, que permiten la entrega de dichos servicios con ciertas garantías de calidad; y la aparición de nuevos equipos de comunicaciones y de terminales de usuario avanzados, que posibilitan la entrega y reproducción del contenido multimedia por parte del usuario. A modo de ejemplo, entre los servicios multimedia que actualmente cuentan con una gran aceptación por parte del mercado, podemos citar los servicios de voz sobre IP (VoIP), video-conferencia, vídeo bajo demanda (*video on demand- VoD*) o televisión IP (IPTV).

En este contexto, el objetivo principal de este TFG es implementar un servicio de televisión IP, basado en las especificaciones del grupo de trabajo TISPAN [[TIS](#)] de ETSI. Dicho servicio soportará el acceso y distribución del vídeo correspondiente a un conjunto de canales de televisión a un número potencialmente elevado de usuarios. La implementación hará uso de protocolos estándares de Internet, que tradicionalmente han sido desarrollados por el IETF (*Internet Engineering Task Force*). Entre los protocolos utilizados, el protocolo SIP (*Session Initiation Protocol*) soportará las funcionalidades de control necesarias para proporcionar acceso al usuario al conjunto de canales de televisión, mediante el establecimiento y terminación de sesiones multimedia.. Por otro lado, la entrega de la información del vídeo correspondiente a los distintos canales se basará en tecnologías de entrega multicast. Las operaciones de cambio de canal se implementarán mediante el protocolo IGMP (Internet Group Management Protocol).

Abstract

Currently, the continuous development of the Internet has allowed the appearance on the market a wide range of multimedia services, available to users through the network. Among the main catalysts that have led to the success of these services include: the I locate Internet access, allowing access to these services virtually anytime and anywhere provided; the development of new technologies for broadband access, enabling the delivery of such services with certain quality assurance; and the emergence of new communications equipment and advanced user terminals that enable the delivery and playback of multimedia content by the user. For example, between multimedia services currently have a great acceptance by the market; we can cite the voice over IP (VoIP), video conferencing, video on demand (video on demand- VoD) and IPTV (IPTV).

In this context, the main objective of the TFG is to implement an IP TV service, based on the specifications of the working group TISPAN ETSI. This service will support the access and distribution corresponding to a television set to a potentially large number of users video. The implementation will use standard Internet protocols, which have traditionally been developed by the IETF (Internet Engineering Task Force). Among the protocols used, SIP (Session Initiation Protocol) bear the necessary control functions to provide user access to all television channels, by establishing and terminating multimedia sessions .. On the other hand, delivering information corresponding to different video channels will be based on multicast delivery technologies. Forex trading channel is implemented by IGMP (Internet Group Management Protocol).

Índice Figuras

2.1. UA	24
2.2 Forking Proxy	25
2.3 Redirect	26
2.4 Register	27
2.5 B2BUA	28
2.6 Cabecera SIP de un INVITE	32
2.7 Sesión SIP	33
3.1 Carga SDP	37
5.1 Componentes Servidor IPTV	45
5.2 Plano de Control	45
5.3 Plano de Datos	46
5.4 Establecimiento de sesión	47
5.5 Liberación de la sesión	51
6.1 Sistema a implementar	53
6.2 Arquitectura JAIN SIP	54
6.3 Esquema	55
6.4 Interfaz Gráfica	56
6.5 Esquema de proceso	57
6.7 Esquema de generación del INVITE	58
6.8 Procesamiento del OK y generación del ACK	59
6.9 Esquema reproducción de contenido multimedia	59
6.10 Mensaje mostrado por consola BYE	60
6.11 Desconexión	60
6.12 Procesado OK del BYE	61
6.13 Procesado INVITE en el Servidor	62
6.14 Generación del ACK	62
6.15 Envío del contenido multimedia	63
6.16 Liberación de la sesión	64
7.1 Escenario de Pruebas	66
7.2 Captura Tráfico inicio de sesión SIP	67
7.3 Captura Tráfico cambio de canal	68
7.4. Imagen del entorno de pruebas (Lab. Telemática)	70
7.5 Captura Tráfico liberación de sesión SIP	72
A.1 Diagrama de Gantt	78
F.1 Formato mensaje IGMP	94
G.1 Interfaz VLC	97
G.2 Emitir	97
G.3 Abrir medio	97
G.4 Abrir medio con vídeo cargado	98
G.5 Salida de emisión	99
G.6 Configuración dirección multicast y puerto	99
G.7 Emisión de vídeo	100
G.8 Visión del vídeo en el reproductor del cliente	100
H.1 Arquitectura HLS	101
H.2 División Streaming en distintas calidades	101
I.1. Actual picture of the test environment	102

Índice Tablas

2.1 Respuestas SIP	29
3.1 Campos SDP	37
4.1 Diferencias IPTV y TV por Internet	39
7.1: Elementos del Sistema	67
A.1 Descomposición de Tareas	77
A.2 Presupuesto Recursos Software	79
A.3 Presupuesto Recursos Hardware	80
A.4 Presupuesto Final	80
F.1 Tipos de mensaje IGMP	94

Índice General

1. Introducción	18
1.1. Motivación	18
1.2. Objetivos	19
1.3. Estructura de la memoria	20
2. SIP.....	23
2.1. Introducción.....	23
2.2. Identificación de usuarios	23
2.3. Elementos de SIP.....	24
2.3.1. SIP UA	24
2.3.2. Servidores SIP	25
2.3.2.1. Servidores Proxy	25
2.3.2.2. Servidores Redirectores	26
2.3.2.3. Servidores Registradores (Registrar Server).....	27
2.3.3. B2BUA	28
2.4. Mensajes de SIP	29
2.4.1. Respuestas SIP	29
2.4.1.1. 100 TRYING.....	30
2.4.1.2. 180 RINGING	30
2.4.1.3. 200 OK	30
2.4.2. Solicitudes SIP	30
2.4.2.1. INVITE	31
2.4.2.2. ACK	31
2.4.2.3. BYE.....	31
2.4.2.4. REGISTER	31
2.5. Campos de las cabeceras SIP	31
2.6. Establecimiento y terminación de sesión.....	33
2.7. Beneficios de SIP frente a otros protocolos	34
3. SDP.....	35
3.1. Modelo oferta/respuesta	37
4. IPTV.....	38
4.1. Introducción.....	38
4.2. Multicast IP	39
4.3. Servicio IPTV de TISPAN.....	40
4.3.1. Arquitectura de TISPAN	40
5. Diseño del sistema y especificación de los requisitos.	43
5.1. INTRODUCCIÓN	43
5.2. PLANO DE CONTROL O SEÑALIZACIÓN	46
5.2.1. Cliente IPTV.....	48
5.2.2. Servidor SIP	48
5.3. Plano de datos	49
5.3.1. Cliente IPTV	49

5.3.2 Servidor de streaming de vídeo	50
5.4 Marco regulatorio	50
6. Implementación de los elementos del sistema.....	51
6.1. Introducción.....	51
6.2. Cliente IPTV.....	54
6.2.1. Gestión de la señalización SIP	54
6.2.2. Interfaz gráfica	55
6.2.3. Procedimientos en el cliente	56
6.2.3.1. Establecimiento de sesión	56
6.2.3.2. Reproducción del contenido multimedia	58
6.2.3.3. Liberación de la sesión	59
6.3. Servidor IPTV	61
6.3.1. Procedimientos en el servidor	61
6.3.1.1. Establecimiento de sesión	61
6.3.1.2. Envío del contenido multimedia	63
6.3.1.3. Liberación de la sesión	63
7. Escenario de pruebas	65
7.1. Introducción.....	65
7.2. Establecimiento de sesión	66
7.3. Envío de contenido multimedia	67
7.4. Liberación de sesión	68
8. Conclusiones y futuras implementaciones	72
8.1. Conclusiones	72
8.2. Futuras implementaciones.....	72
Anexo A: Gestión de proyecto y presupuesto.....	75
A.1. Descomposición en tareas	75
A.2. Planificación con el diagrama de fases	79
A.3. Presupuesto.....	80
A.3.1. Recursos Humanos	80
A.3.2. Recursos Informáticos	80
A.3.2.1. Recursos Software.....	80
A.3.2.2. Recursos Hardware	81
A.3.1. Presupuesto Final	81
Anexo B: Manual de usuario	82
Anexo C: Solicitudes SIP	84
Anexo D: Campos SDP	87
Anexo E: Establecimiento de sesión SIP	89
Anexo F: IGMP.....	90
Anexo G: Manual de Configuración del VLC.....	92
Anexo H: HLS.....	97
Anexo I: Introduction, Objectives, Result and Conclusions.	99

I.1. Introduction.....	99
I.2. Objectives	100
I.3. Result	100
I.4. Conclusions.....	101
<i>Bibliografa</i>	103

INTRODUCCIÓN

1. Introducción

En este 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

En la actualidad, el desarrollo continuo de Internet ha permitido la aparición en el mercado de una gran diversidad de servicios multimedia. Estos servicios, que son accesibles a través de la red, no son más que la evolución natural de las herramientas multimedia y servicios que tradicionalmente hemos utilizado, como puede ser la televisión, la radio, la telefonía fija o sistemas de grabación y reproducción de vídeo (DVD). Son varios los motivos que han propiciado la gran aceptación de estos nuevos servicios multimedia en red. Por un lado, Internet se ha convertido en una red de masas, proporcionando cobertura a nivel global. De este modo, se facilita el acceso de los usuarios a estos servicios, que se encuentran disponibles prácticamente en cualquier momento y desde cualquier lugar; el desarrollo de nuevas tecnologías de acceso de banda ancha, que permiten la entrega de dichos servicios con ciertas garantías de calidad; y la aparición de nuevos equipos de comunicaciones y de terminales de usuario avanzados, que posibilitan la entrega y reproducción del contenido multimedia por parte del usuario. A modo de ejemplo, entre los servicios multimedia que actualmente cuentan con una gran aceptación por parte del mercado, podemos citar los servicios de voz sobre IP (VoIP), video-conferencia, vídeo bajo demanda (*video on demand*- VoD) o televisión IP (IPTV). Todos los dispositivos de los que disponemos actualmente, requieren que el usuario interactúe con ellos en la mayoría de los casos, por no decir todos, se hace a través de una interfaz gráfica.

En el ámbito empresarial está cobrando gran importancia, ya que empieza a suponer un alto grado de ingresos para empresas privadas, ya que los usuarios no solo demandan canales de TV convencionales, sino que demandan temporadas de series completas y su propio videoclub. Estos servicios los ofertan grandes compañías, aquí en España dos de ellas son Telefónica con su producto "Movistar Fusión" y ONO con "ONO TV". Según un reporte de Cisco, en 2013 el tráfico de vídeo es predominante en Internet, y pronostica que en 2018 el tráfico total de vídeo se aproximará al 80-90% del tráfico de usuario global [CISCO].

La tecnología IPTV es un servicio relevante que ha recogido esfuerzos de estandarización, el más relevante es el grupo de trabajo de TISPAN [TIS] de ETSI. Por este motivo, el objetivo principal de este TFG es implementar un servicio de televisión IP, basado en las especificaciones del grupo de trabajo TISPAN. Dicho servicio soportará el acceso y distribución del vídeo correspondiente a un conjunto

de canales de televisión a un número potencialmente elevado de usuarios. La implementación hará uso de protocolos estándares de Internet, que tradicionalmente han sido desarrollados por el IETF (*Internet Engineering Protocol*) soportará las funcionalidades de control necesarios para proporcionar acceso al usuario al conjunto de canales de televisión, mediante el establecimiento y terminación de sesiones multimedia. Dicha implementación no tiene un número máximo de usuarios a soportar, por lo que esto no supone una limitación al desarrollo. Por otro lado, la entrega de la información del vídeo correspondiente a los distintos canales se basará en tecnologías de entrega *multicast*. Las operaciones de cambio de canal se implementarán mediante el protocolo IGMP (*Internet Group Management Protocol*). Todos los detalles de la implementación realizada se comentarán en los siguientes capítulos.

1.2. Objetivos

El objetivo principal de este TFG es desarrollar un servicio de televisión IP, basado en las especificaciones del grupo de trabajo TISPAN [TIS] de ETSI. Dicho servicio tiene que soportar el acceso y distribución del vídeo correspondiente a un conjunto de canales de televisión a un número potencialmente elevado de usuarios.

Profundizando en detalle en este objetivo, se puede destacar los siguientes puntos a tener en cuenta en el desarrollo de este sistema:

- El diseño se basará en protocolos estándares de Internet para el acceso y entrega de contenido multimedia.
- El plano de control se implementará mediante el protocolo SIP (Session Initiation Protocol), definido por el IETF en [IR02a].
- En el plano de datos, la transmisión eficiente de información de vídeo a los receptores de cada canal se realizará mediante el empleo de tecnologías de entrega multicast.
- Las operaciones de cambio de canal se basarán en el protocolo de gestión de grupos de internet (IGMP- Internet Management Group Protocol [IGMP] [IGMP.2]).
- La implementación será extensible para soportar fácilmente la inclusión de nuevos canales de televisión.

Como objetivos secundarios del presente TFG, se definen los siguientes:

- Estudio de las tecnologías y protocolos recogidos en los estándares actuales de televisión IP para redes de operador.
- Aplicación de dichas tecnologías en el desarrollo de un servicio IPTV, al objetivo de comprender mejor su funcionamiento y su facilidad de implementación desde un punto de vista práctico.

1.3. Estructura de la memoria

En este apartado se describe la división de las partes y capítulos en los cuales se estructura esta memoria.

A continuación se describe brevemente el contenido de cada parte y de cada capítulo.

Primera parte: Introducción: contiene la motivación y los objetivos del presente TFG.

- Se explica la motivación y los objetivos del TFG, además de la estructura de la memoria.

Segunda parte: Estado del arte: En esta parte se describen los protocolos utilizados para el desarrollo del servicio IPTV, tanto para ejecutar los procedimientos de control del servicio, como para entregar el contenido de vídeo a los usuarios.

- Esta segunda parte consta de los siguientes capítulos:

Capítulo 2. SIP

Capítulo 3. SDP

Capítulo 4. IPTV

Tercera parte: Descripción del trabajo realizado: En esta parte, se definen los requisitos del servicio que se ha desarrollado, así como los distintos elementos que componen su diseño. Finalmente, se describe la implementación realizada así como las pruebas que han permitido validar dicha implementación. Los capítulos de los que consta esta parte son los siguientes:

Capítulo 5. Diseño del sistema y especificación de los requisitos

Capítulo 6. Implementación de los elementos del sistema

Capítulo 7. Escenario de pruebas

Cuarta parte: Conclusiones y futuras implementaciones. Se explican las conclusiones obtenidas al término del trabajo y futuras implementaciones para la posible ampliación de este TFG.

Quinta parte: Anexos. Es la última parte de este TFG, y consta de:

Anexo A: Gestión de Proyecto y Presupuesto

Anexo B: Manual de usuario

Anexo C: Solicitudes SIP

Anexo D: Campos SDP

Anexo E: Establecimiento de sesión SIP

Anexo F: IGMP

Anexo G: Manual de configuración del VLC

Anexo H: HLS

*Anexo I: Introduction, Objectives, Result and
Conclusions*

ESTADO DEL ARTE

2. SIP

2.1. Introducción

SIP [[Allan.B.09](#)] [[RFC 3261](#)] es un protocolo de control de la capa de aplicación que permite establecer, modificar y terminar sesiones multimedia. SIP también permite invitar a participantes a sesiones ya existentes y a conferencias multimedia. Soporta servicios de redirección de forma transparente, permitiendo así movilidad, por lo que el usuario puede mantener un identificador persistente de SIP independiente de su localización en la red (que puede variar).

SIP es un protocolo de señalización desarrollado para establecer, modificar y liberar sesiones multimedia. Dicho protocolo ha sido desarrollado por el IETF (Internet Engineering Task Force) [[IET](#)]. SIP soporta cinco tareas relacionadas con el establecimiento y la terminación de una comunicación multimedia, estas tareas se enumeran a continuación:

- Determinar la localización de "puntos finales de comunicación" (usuarios).
- Contactar "puntos finales" para determinar su disponibilidad para establecer una sesión de comunicación multimedia.
- Permite intercambiar las capacidades de comunicación multimedia de los "puntos finales" para determinar los parámetros a utilizar en una sesión.
- Modificar sesiones de comunicación multimedia, incluyendo la transferencia de sesiones, así como la modificación de los parámetros de la sesión.
- Terminar sesiones de comunicación multimedia

SIP no es un sistema completo de comunicaciones, ya que debe de ser ampliado por otros protocolos de la IETF [[IET](#)], para completar una arquitectura multimedia. Las arquitecturas más típicas incluyen protocolos como *RTP* (Real-time Transport Protocol)[[RTP](#)] para transportar datos de tiempo real y proporcionar realimentación sobre la calidad de la distribución de los datos, Real-Time Streaming Protocol (RTSP) [[RTSP](#)] para controlar el transporte de las sesiones de streaming y Session Description Protocol (SDP) [[Capítulo 3](#)] para la descripción de las sesiones multimedia.

2.2. Identificación de usuarios

SIP identifica a los usuarios mediante URIs SIP (Uniform Resource Identifier). Una URI SIP, es el esquema de direccionamiento SIP para indicar su identificación. De forma comparativa, se puede indicar que es similar a una dirección de correo electrónico. Su formato se indica a continuación:

sip: [userinfo] [hostport] [parameters]

Donde,

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

Un ejemplo de una URI SIP es:

sip:Alice.Smith@domain.com
sip:22444032@phonesystem.3cx.com

El estándar SIP URI ha sido definido en [RFC 3261](#).

2.3. Elementos de SIP

SIP es un protocolo cliente-servidor, ya que en la utilización de dicho protocolo, es necesario que haya dos participantes. Los elementos que se definen en la utilización del protocolo SIP se explican a continuación.

2.3.1. SIP UA

Los usuarios del protocolo SIP pueden ser seres humanos o aplicaciones de software, utilizadas para establecer sesiones multimedia. Un UA (User Agent) es un dispositivo capaz de establecer una sesión multimedia con otro UA ([figura 2.1](#)). Ejemplos de estos dispositivos son los teléfonos móviles, ordenadores, TV, etc. Cabe destacar que se pueden comportar como clientes (*UAC: User Agent Clients*) y como servidores (*UAS: User Agent Servers*). Son UAC cuando realizan peticiones y UAS cuando la reciben.

El protocolo SIP no se ocupa de la interfaz de estos dispositivos con el usuario final, sólo se interesa por los mensajes que estos generan y cómo se comportan al recibir determinados mensajes.

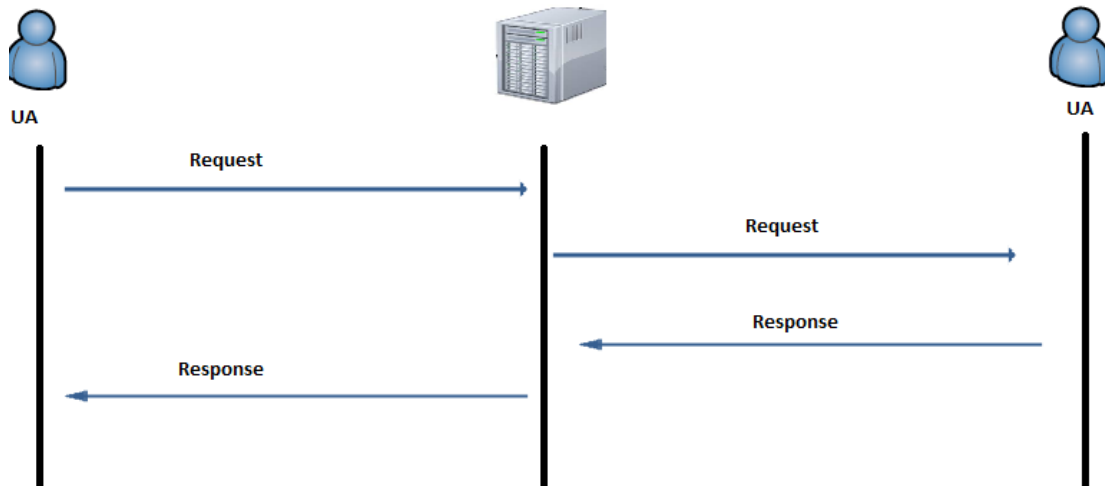


Figura 2.1: UA (User Agent)

2.3.2. Servidores SIP

Un servidor SIP es el dispositivo encargado de aceptar peticiones SIP y dar respuestas a ellas soportando TCP y UDP como protocolos de transporte. Se debe destacar que un servidor SIP no es lo mismo que un UA, ya que son entidades distintas. Existen distintos tipos de servidores SIP. A continuación se procede a describirlos.

2.3.2.1. Servidores Proxy

Un servidor Proxy, es un servidor SIP que actúa encaminando solicitudes hacia los *UAS* o encaminando respuestas hacia los *UAC*. Pueden existir varios proxies a través de los que una determinada solicitud deba pasar para ir del UAC origen al UAS destino. La respuesta a una solicitud determinada sigue exactamente el mismo camino (a la inversa) que siguió la solicitud original; es decir, que atraviesa los mismos proxies. Un proxy puede tomar decisiones sobre el encaminamiento de una solicitud, o incluso puede llegar a modificarla antes de reenviarla al siguiente paso.

En la [figura 2.2](#) se observa un ejemplo de cómo 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 procedentes de un UA.

- Un servidor proxy no tiene la capacidad de intercambiar información multimedia (ej. RTP).
- Un proxy no procesa cuerpos de mensaje, basa su procesamiento únicamente en campos de cabecera de SIP

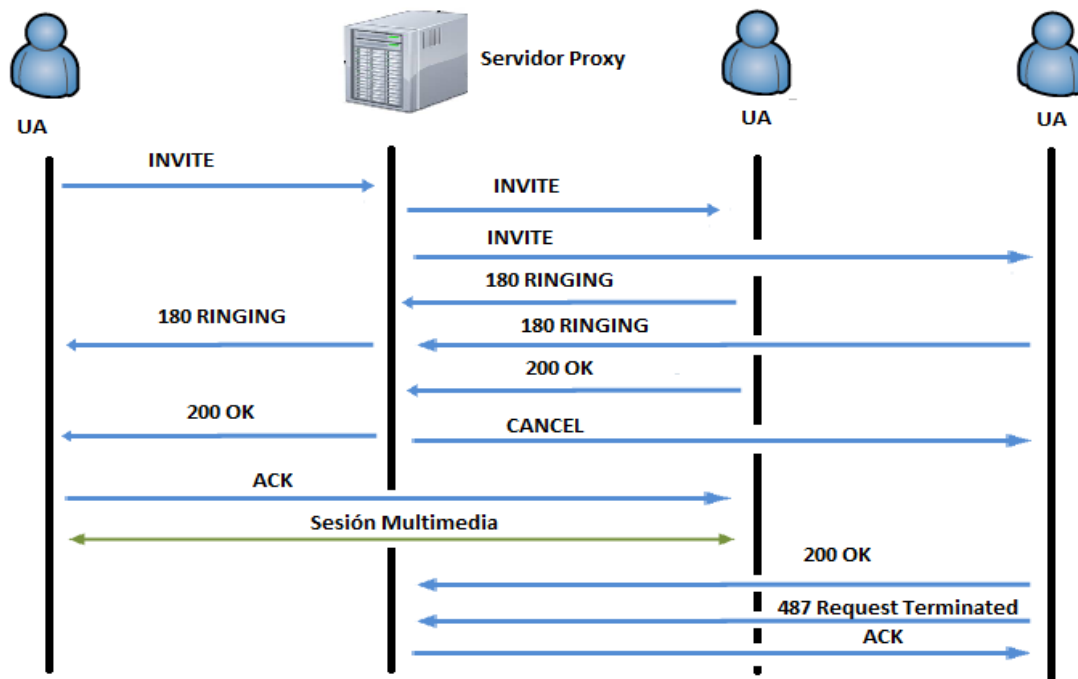


Figura 2.2: Forking Proxy [Allan.B.09]

2.3.2.2. Servidores Redirectores

Los servidores redirectores, pueden responder a peticiones SIP, pero no tienen capacidad de reenvío como los servidores comentados anteriormente. Los servidores redirectores suelen utilizar una base de datos para obtener la localización del usuario. En la [figura 2.3](#) se observa un ejemplo de señalización de llamada que incluye un servidor redirector.

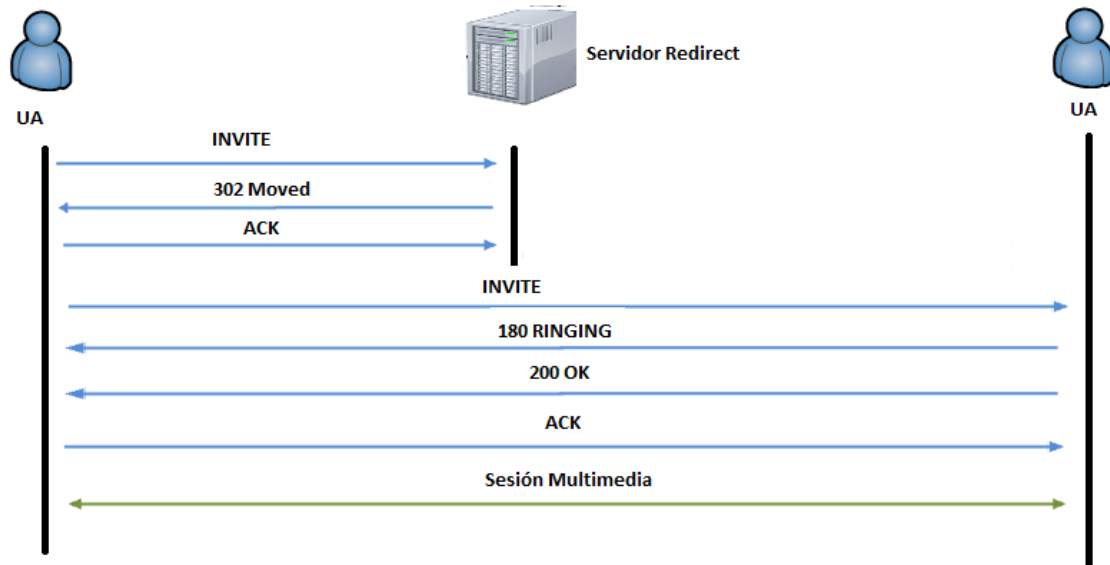


Figura 2.3: Redirect [Allan.B.09]

2.3.2.3. Servidores Registradores (Registrar Server)

Un Servidor Registrador es el encargado de gestionar toda la información de localización de usuarios. Los usuarios en SIP, pueden indicar un agrupamiento entre su URI SIP pública y las direcciones en las que dichos usuarios desean ser contactados. Para ellos se debe de realizar lo siguiente:

- Se realiza mediante una solicitud de SIP de tipo *REGISTER*.
- Los usuarios pueden modificar las asociaciones de la URI SIP y las direcciones de contacto, mediante solicitudes de tipo *REGISTER*.

Un *Servidor Register*, usa un servicio de almacenamiento de información de localización. Dicho servicio puede ser interno del propio servidor o externo a dicho servidor. En la [figura 2.4](#) se puede ver un ejemplo de este tipo de servidores.



Figura 2.4: Register

2.3.3. B2BUA

B2BUA (Back to back user agent), actúa tanto como UAC o UAS. Es un tipo de SIP UA. Dicho dispositivo recibe peticiones SIP y las procesa, las puede reformular o simplemente las reenvía. Genera solicitudes para otros servidores, puede modificar la carga de los mensajes o contactar con otros usuarios tras recibir una petición INVITE.

En la [figura 2.5](#) se observa un ejemplo de este tipo de UA.

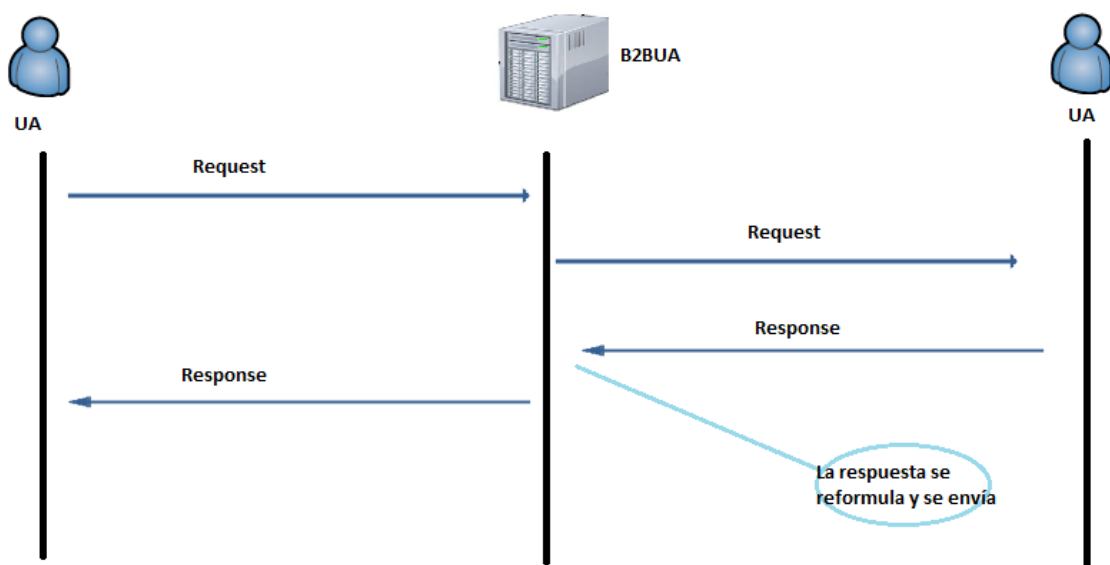


Figura 2.5: B2BUA

2.4. Mensajes de SIP

SIP es un protocolo basado en texto que usa caracteres UTF-8. Los mensajes SIP pueden ser solicitudes de un cliente a un servidor (*request*) o respuestas de un servidor a un cliente (*response*). A continuación se muestra el formato que tienen los mensajes SIP:

<start-line>
Campos de cabecera
Línea en blanco
<Cuerpo de mensaje> (body)

2.4.1. Respuestas SIP

Las respuestas SIP, son mensajes que generan los Servidores SIP o UAS, para responder a las peticiones que han generado los usuarios o UAC. Pueden contener cabeceras de información adicional, esta información puede ser necesaria para el UAC. Hay 6 tipos de respuestas SIP que son mostradas en la [tabla 2.1](#).

Código	Significado
1xx	Mensajes provisionales.
2xx	Respuestas de éxito.
3xx	Respuestas de redirección
4xx	Respuestas de fallo de método
5xx	Respuestas de fallos de servidor
6xx	Respuestas de fallos globales

Tabla 2.1: Tabla Respuestas SIP

Puede ocurrir que un código de respuesta SIP no sea entendido por un UAC, debe de ser interpretado por la clase respuesta. Todas y cada una de las respuestas SIP se caracterizan por su *Status-Line*. El **Status-Line** es la primera línea de cualquier respuesta, según la terminología indicada en la RFC 3261 [\[RFC 3261\]](#) cuyo formato es el siguiente:

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

- **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 dígitos mostrados en la [tabla 2.1](#).

- **Reason-Phrase:** descripción del Status-Code.

2.4.1.1. 100 TRYING

Es un tipo de respuesta provisional. Dicha respuesta indica que la solicitud ha sido recibida por el servidor. Esta respuesta es generada por cualquier servidor proxy o por un agente de usuario.

2.4.1.2. 180 RINGING

Dicha respuesta se envía para indicar que se ha recibido un INVITE y de esta forma se alerta al usuario.

2.4.1.3. 200 OK

Cuando se recibe dicha respuesta, indica que la solicitud ha sido cursada correctamente. Cuando se envía para aceptar una solicitud INVITE, normalmente el cuerpo del mensaje incluye una carga SDP.

2.4.2. Solicitudes SIP

Se distinguen por tener una *Request-Line* como línea de comienzo. Ésta contiene el nombre del método, un *Request-URI* y la versión del protocolo separados por un espacio. La *Request-Line* termina con una línea en blanco, no permitiéndose ni tabulaciones ni espacios en blanco en ninguno de los elementos que la conforman. Los mensajes INVITE, REGISTER, BYE, ACK, CANCEL y OPTION son las seis solicitudes SIP originales. La estructura de dichas solicitudes se muestra a continuación:

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

Donde,

- **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.

A continuación se explican las solicitudes SIP más importantes.

2.4.2.1. INVITE

Dicha solicitud es usada para establecer sesiones multimedia entre diferentes UA. El cuerpo del mensaje INVITE suele contener información que permite describir la sesión multimedia que se desea establecer. Dicha información se describe de acuerdo al formato definido por el IETF en SDP [[SDP](#)], el cual se describe en el [capítulo 3](#).

2.4.2.2. ACK

Cuando se envía dicho mensaje, es para la confirmación de las respuestas a una petición INVITE. Este tipo de respuestas son definidas por los códigos 2xx, 3xx, 4xx, 5xx o 6xx, tal y como se observa en la [tabla 2.1](#).

2.4.2.3. BYE

Dicha petición se usa para terminar una sesión multimedia previamente establecida.

2.4.2.4. REGISTER

Es utilizada por un UA para informar de la localización del usuario.

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 poseen las siguientes cabeceras:

- **To:** especifica el receptor de la solicitud.

To: Alice <sip:alice@biloxy.com>;tag=54268

- **From:** indica el origen de una solicitud.

From: Bob <sip:bob@atlanta.com>;tag=4df45s6

- **CSeq:** contiene un número de secuencia y un nombre de método. Se asocia este campo con solicitudes y respuestas.

Cseq: 5 INVITE

- **Call-ID:** identifica una solicitud INVITE, y los registros realizados por un cliente.

Call-ID: f41d8fee-7dec-12d0-a785-00b10c9c6bf@atlanta.com

- **Max-Forwards:** Este campo indica el tiempo de vida de la solicitud. Cada proxy que gestiona la solicitud decrementa el valor de este campo. Cuando dicho valor llega a cero, se descarta la solicitud. Se utiliza para evitar los posibles bucles de encaminamiento.

Max-Forward: 09

- **Via:** es 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 host.wcom.com:5060;branch=z3eGdbK78aseds7

- **Contact:** contiene una URI con la información del contacto que origina el mensaje.

Contact: sip:bob.sweeney@wcom.com

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

Expires: 60

```
⊞ User Datagram Protocol, Src Port: 5060 (5060), Dst Port: 5060 (5060)
⊞ Session Initiation Protocol
  ⊞ Request-Line: INVITE sip:107@10.172.0.2 SIP/2.0
  ⊞ Message Header
    ⊞ Via: SIP/2.0/UDP 10.172.0.101:5060;branch=z9hG4bK59fab8a8a649810a
    ⊞ From: "101" <sip:101@10.172.0.2>;tag=0374a1343263be14
    ⊞ To: <sip:107@10.172.0.2>
    ⊞ Contact: <sip:101@10.172.0.101:5060>
      Supported: replaces, timer
      Call-ID: d61d626db1c1d19de10.172.0.101
    ⊞ CSeq: 1660 INVITE
      User-Agent: Grandstream GXP2000 1.1.0.14
      Max-Forwards: 70
      Allow: INVITE, ACK, CANCEL, BYE, NOTIFY, REFER, OPTIONS, INFO, SUBSCRIBE, UPDATE, PRACK
      Content-Type: application/sdp
      Content-Length: 307
  ⊞ Message body
    ⊞ Session Description Protocol
      Session Description Protocol Version (v): 0
```

Figura 2.7: Cabecera SIP de un INVITE

2.6. Establecimiento y terminación de sesión

Para realizar la explicación de este apartado, se observa en la [figura 2.8](#) un establecimiento de sesión SIP completo, basado en una figura de la RFC 3261 [[RFC 3261](#)].

Se observa como uno de los usuarios, Alice, pretende establecer una sesión con otro usuario, Bob, y para ello genera una petición INVITE. 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 que inicio la sesión y después éste asiente la respuesta con un ACK.

Una vez que la sesión queda 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, quien envía una petición BYE indicando que quiere abandonar la sesión. Cuando Alice recibe esta solicitud, responde a ella con un 200 OK para finalizar la sesión.

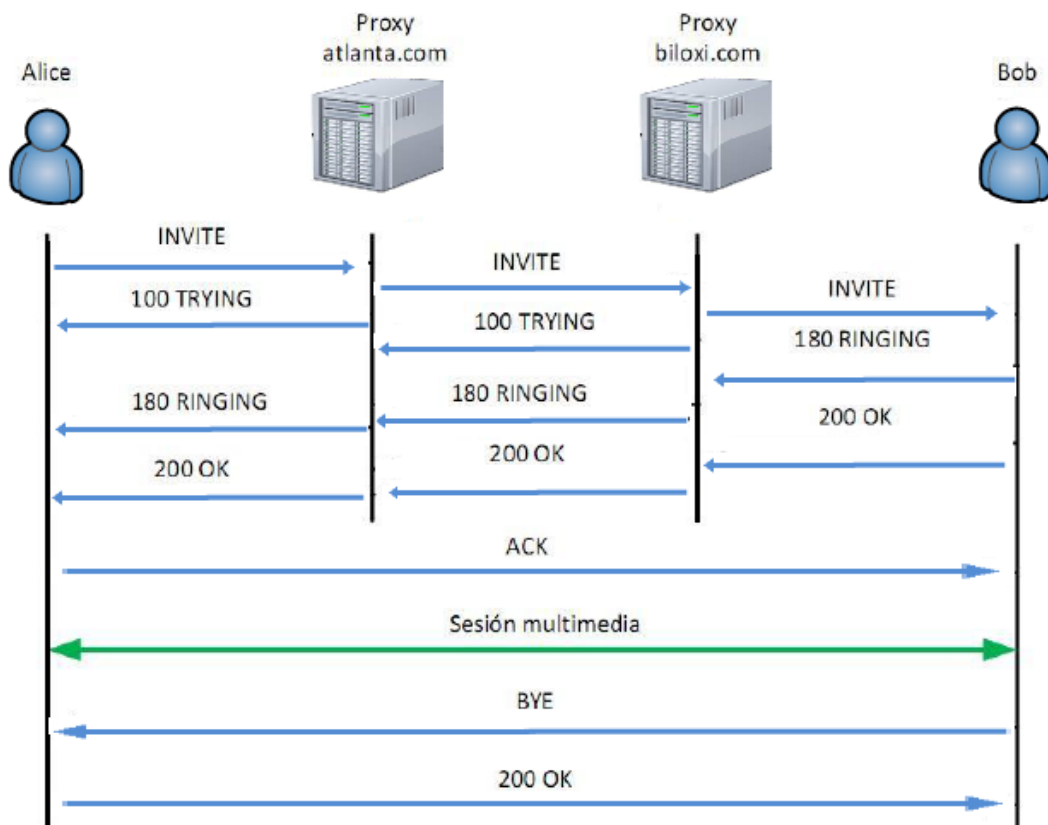


Figura 2.8: Sesión SIP [[Allan.B.09](#)]

2.7. Beneficios de SIP frente a otros protocolos

En la actualidad, los protocolos más usados para comunicaciones multimedia son tres: SIP, H.323 y IAX2.

H.323 es un estándar de la ITU que provee especificaciones para ordenadores, sistemas y servicios multimedia por redes que no proveen QoS (calidad de servicio).

Como principales características de H.323, destacan:

- Implementa QoS de forma interna.
- Control de conferencias

IAX2 (Inter Asterisk eXchange) es un protocolo creado y estandarizado por Asterisk. Unas de sus principales características son: *Media y señalización* viajan en el mismo flujo de datos.

- Trunking
- Cifrado de datos

Una de las ventajas de este protocolo es que al enviar el “streaming” y la señalización por el mismo flujo de datos, se evitan problemas derivados del NAT. Así pues, no es necesario abrir rangos de puertos para el tráfico RTP. Por último, IAX2 permite hacer trunking de forma que se puede enviar varias conversaciones por el mismo flujo, lo cual supone un importante ahorro de ancho de banda.

SIP es un protocolo que cada día se hace más sólido. Aspectos importantes referentes a dicho protocolo se enumeran a continuación:

- El control de llamadas es *stateless* o sin estado, y proporciona escalabilidad entre los dispositivos telefónicos y los servidores.
- SIP necesita menos ciclos de CPU para generar mensajes de señalización de forma que un servidor podrá manejar más transacciones.
- Una llamada SIP es independiente de la existencia de una conexión en la capa de transporte.

3. SDP

SDP [[SDP](#)] es un protocolo que describe los parámetros necesarios para la inicialización de los flujos multimedia. SDP es un modelo oferta/respuesta, ya que dicho protocolo permite la negociación de la descripción de la sesión multimedia. Normalmente, la carga SDP se transporta en el cuerpo de un mensaje SIP. La estructura de la información almacenada en la carga SDP se muestra a continuación.

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

Se puede observar un ejemplo de una carga SDP en la [figura 3.1](#), así como los campos SDP definidos en la RFC 4566 [[SDP](#)] son los mostrados en la [tabla 3.1](#).

En este TFG se centra en algunos campos, por ese motivo se explican con mayor detalle dichos campos. El resto de los campos aparecen con detallados en el [Anexo D](#).

- **version:** contiene el número de versión de SDP.
- **origin:** contiene la información sobre el origen de la sesión. Se usa para la identificación de la sesión.
Su formato es el siguiente:

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

Los parámetros de dichos campos están explicados más avanzado el apartado (véase [parámetros](#)).

- **connection:** contiene información sobre la conexión.
Su formato se muestra a continuación:

```
c=<network-type> <address-type> <connection-address>
```

Los parámetros de dichos campos están explicados más avanzado el apartado (véase [parámetros](#)).

- **username:** nombre del usuario que inicia la sesión
- **session-id:** indica un identificador único de la sesión.
- **version:** identifica la versión de la sesión.

- **network-type:** identifica, en formato texto, el tipo de red que se está utilizando. (IN para Internet).
 - **address-type:** define el tipo de dirección. Utiliza IP4 para direcciones IPv4 e IP6 para direcciones IPv6.
 - **address:** dirección desde la cual se creó la sesión.
 - **connection-address:** Para sesiones unicast indica dirección IP y puerto del envío del contenido. 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=<media> <port> <transport> <format-list>

Los parámetros de dichos campos son los siguientes:

- **media:** puede tener varios valores en función del flujo de datos a transmitir: audio, video, text, application,...
- **port:** 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:** todos los formatos en los que el flujo multimedia puede ser codificado.

Un ejemplo podría ser el siguiente:

m=audio 49430 RTP/AVP 0 6 8 99

- **attributes:** campo para extender SDP. Los atributos son definidos para ser utilizados tanto a nivel de sesión como a nivel multimedia.

```
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 [Allan.B.09]

Campo	Nombre
v	<i>Version: versión</i>
o	<i>Origin: creador/identificador de la sesión</i>
s	<i>Session Name: nombre de la sesión</i>
i	<i>Information: información de la sesión</i>
u	<i>URI: identificador uniforme del recurso</i>
e	<i>Email: dirección de correo electrónico</i>
p	<i>Phone: número de teléfono</i>
c	<i>Connection: información de conexión</i>
b	<i>Bandwidth: información de ancho de banda</i>
t	<i>Time: información de comienzo y fin de sesión</i>
r	<i>Repeat times: número de repeticiones</i>
z	<i>Time zones: ajustes de las zonas horarias</i>
k	<i>Encryption key: clave encriptada</i>
a	<i>Attributes: línea con atributos</i>
m	<i>Media: información multimedia</i>

Tabla 3.1: Campos SDP [Allan.B.09]

3.1. Modelo oferta/respuesta

Está definido en la RFC 3264 [SDP.2]. Este modelo consiste un usuario u ofertante, que es el que genera una carga SDP, lo llamado oferta SDP. En la cual se incluye sus preferencias multimedia y se la envía al otro usuario. El otro usuario tiene dos opciones, o genera una respuesta o rechaza dicha oferta. Si finalmente genera la respuesta, la respuesta SDP la genera de nuevo y se la envía al usuario ofertante. La respuesta debe de contener el mismo número de líneas m= que se recibieron en la oferta.

4. IPTV

4.1. Introducción

IPTV, o televisión basada en el protocolo IP, trata la entrega segura de emisiones de televisión de alta calidad y/o video bajo demanda y contenido de audio a través de una red de banda ancha. IPTV es un término que generalmente se aplica a la entrega de contenidos de canales tradicionales de televisión, video bajo demanda o contenido de televisión en diferido sobre una red privada.

Cabe destacar cual es la definición oficial aprobada por el grupo de enfoque en IPTV de la Unión Internacional de Telecomunicaciones [UIT06], que es la siguiente:

“IPTV se define como servicios multimedia, tales como televisión, video, audio, texto, gráficos y datos ofrecidos sobre redes basadas en IP, gestionadas para proveer el nivel requerido de calidad de servicio (QoS) y experiencia, seguridad, interactividad y confiabilidad”

Desde la perspectiva de un proveedor del servicio, IPTV incluye la adquisición, procesamiento, y distribución segura de contenido de video a través de una infraestructura de red basada en IP.

IPTV tiene una serie de características que la hacen sobresalir por sobre otras tecnologías similares [G.007]:

- **Soporte para televisión interactiva:** Los tipos de servicios prestados a través de un servicio de IPTV pueden incluir televisión en directo estándar, televisión de alta definición (HDTV), juegos interactivos, y navegación por Internet de alta velocidad.
- **PVR o sistema de grabación interactivo:** IPTV en combinación con un grabador de video digital permite el cambio de tiempo o la pausa en directo de contenido de programación – un mecanismo de grabación y almacenamiento de contenido IPTV para su posterior visualización.
- **Personalización:** Un sistema IPTV de extremo a extremo soporta comunicaciones bidireccionales y permite a los usuarios finales personalizar sus hábitos televisivos dejándolos decidir lo que quieren ver y cuando lo quieren ver.
- **Facilidad de acceso en múltiples dispositivos:** La visualización de contenido IPTV no está limitada a los televisores. Los consumidores a menudo utilizan sus PCs y dispositivos móviles para acceder a servicios IPTV.

Cabe destacar la diferencia existente entre IPTV y TV por internet, estas diferencias se muestran en la [tabla 4.1](#).

IPTV	TV por Internet
Se Basa en una arquitectura de red cerrada.	Arquitectura de red abierta y disponible.
Sistema de TV propietarios (Pay TV principalmente)	Gran número de productores medianos y pequeños contribuyen con creación innovadoras en contenidos y canales de distribución.
Distribución y entrega en paquetes de Internet (IP) que permiten un alto grado de control y seguridad.	Distribución y entrega en paquetes de Internet (IP) que permiten un alto grado de control y seguridad.

Tabla 4.1: Diferencias IPTV y TV por Internet [Comp]

4.2. Multicast IP

Comprende un conjunto de tecnologías que permiten el envío de datagramas IP, de manera eficiente, a grupos de destinatarios que han mostrado su interés en recibir dichos datagramas. Cuando un usuario quiera recibir paquetes de un grupo multicast, debe de subscribirse al grupo al que quiera pertenecer.

Todo dispositivo o equipo que quiera recibir información que se está transmitiendo por multicast a una dirección IP multicast, debe de pertenecer al grupo correspondiente a dicha dirección. Para llevar a cabo esta suscripción los equipos interesados utilizan el protocolo **IGMP** (Internet Group Management Protocol [IGMP.2] [Adams]). IGMP es un protocolo que establece un mecanismo para el intercambio y actualización de información sobre la pertenencia de los equipos a un grupo multicast. La explicación de forma detallada del protocolo IGMP se encuentra en el [anexo F](#).

Para la identificación de grupos multicast se usan direcciones IP de clase D (224.0.0.0 a 239.255.255.255). Para cada dirección multicast puede no existir o existir varios destinatarios activos, que recibirán una copia del datagrama enviado a la dirección del grupo multicast.

Hay dos tipos de grupos multicast que se detallan a continuación:

- **Grupo permanente:** tiene asociada una dirección IP multicast, 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.

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

Cabe destacar que se considerará que el grupo multicast constará siempre de receptores, no de emisores. Esto implica que origine 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.

En la actualidad, la mayoría de aplicaciones existentes que usan multicast lo hacen sobre UDP (User Datagram Protocol) [J80]. En cambio, otras aplicaciones, en mayor medida las 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]. También existen otros protocolos como DVMRP, PIM-DIM, CBT o PIM-SM [A97].

4.3. Servicio IPTV de TISPAN

En la actualidad existen diferentes soluciones para sistemas IPTV. Este TFG se ha basado en la arquitectura propuesta por el grupo de trabajo TISPAN de ETSI. En la versión 2 de sus especificaciones de la red NGN (Next Generation Networks), TISPAN añade IPTV como un nuevo servicio en la arquitectura de la red, también se define un sistema IPTV integrado en las NGN, con el objetivo de fomentar los requisitos y permitir la integración de soluciones IPTV definidas por otras organizaciones, tales como DVB (Digital Video Broadcasting), ATIS IIF (ATIS IPTV Interoperability Forum), o ITU (International Telecommunication Union) [D11]., por ese motivo se procede a detallarla a continuación.

4.3.1. Arquitectura de TISPAN

En la actualidad existen varios organismos de estandarización internacionales trabajando en IPTV, TISPAN es el más relevante de todos ellos. 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.

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). En versiones posteriores, en concreto en la versión 3 de TISPAN, añade un conjunto de servicios de valor añadido. Algunos ejemplos son la televisión interactiva, la interacción entre los usuarios y los canales personalizados [[ETS11](#)].

El modelo TISPAN define una gestión de dos capas: la capa de servicios, IMS y el control de la capa de transporte, no basado en IMS. IMS es una tecnología desarrollada con el fin de proporcionar servicios multimedia basados en IP. Su arquitectura está organizada en 3 capas: la capa de aplicación, la capa de control y la capa de acceso. IMS soporta múltiples tecnologías, por ejemplo, tecnologías para el control de sesiones como SIP y SDP.

DESCRIPCIÓN DEL TRABAJO REALIZADO

5. Diseño del sistema y especificación de los requisitos.

5.1. INTRODUCCIÓN

Este Trabajo Fin de Grado se basa en el desarrollo de un servicio de televisión IP, basado en las especificaciones del grupo de trabajo TISPAN [TIS] de ETSI. Este servicio combina, un plano de control basado en el protocolo SIP, con un plano de datos basado en tecnologías de entrega multicast [TIS.027] [TIS.063].

El servicio implementado consta de dos componentes software principal, un cliente IPTV, que es accesible al usuario, y un servidor IPTV que gestiona el acceso al servicio. Éste a su vez consta de dos sub-componentes: un servidor SIP, que implementa un subconjunto de las funcionalidades definidas para las entidades SCF (*Service Control Function*) del servicio IPTV de TISPAN, y un servidor de vídeo en modo streaming, que proporciona la funcionalidad correspondiente a la entidad MDF (*Media Delivery Function*) de dicho servicio (Véase [figura 5.1](#)). El cliente y servidor IPTV implementan todas las funcionalidades del plano de control y del plano de datos que son necesarias, de modo que se garantice acceso a los clientes a los canales IPTV que conforman el servicio.

El plano de control correspondiente al servicio desarrollado se basa en el protocolo SIP [RFC-3261], y permite establecer, modificar y terminar las sesiones multimedia que son necesarias para acceder al vídeo de los distintos canales disponibles. Dicho plano de control se esquematiza en la [figura 5.2](#).

Como parte de los procedimientos del plano de control, en cuanto el usuario activa el cliente IPTV (solicita el acceso al servicio), es el propio Cliente IPTV, quién de forma transparente al usuario y sin necesidad de realizar ningún tipo de configuración, intercambia de forma automática una serie de mensajes de señalización SIP que serán con el Servidor SIP, al objeto de solicitar autorización para acceder al servicio y de establecer una sesión multimedia para la recepción del vídeo de los canales IPTV.

Una vez que ya está establecida la sesión, la entrega del vídeo se realiza sobre el plano de datos. El plano de datos ([figura 5.3](#)) comprende el procesamiento y el envío del contenido multimedia de vídeo mediante protocolos de encaminamiento multicast. Para ello, una vez establecida la sesión multimedia, entra en funcionamiento el Servidor de streaming de vídeo. Este servidor, es el servidor encargado de difundir los contenidos multimedia. Dicha difusión de dichos contenidos se realiza mediante tecnologías de entrega multicast, como se ha comentado anteriormente.

Finalmente, para la selección del canal que se desea visionar, el usuario dispondrá de una interfaz gráfica que le mostrará el Cliente IPTV.

A continuación se explica con más detalle en el diseño de los distintos componentes que forman parte del sistema implementado. Así como los requisitos que deben de cumplir, tanto el servidor como el cliente IPTV.

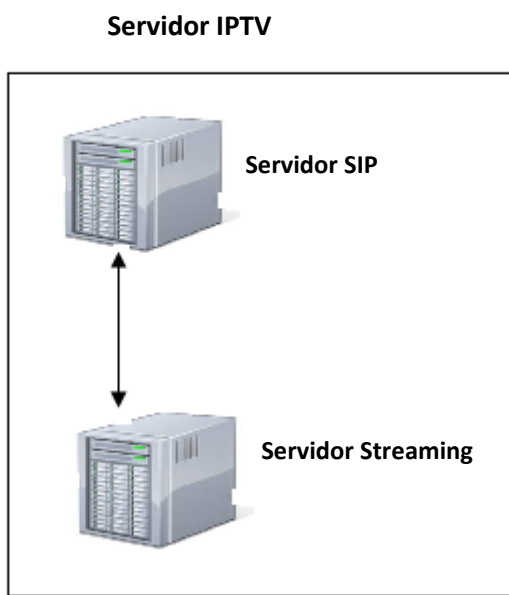


Figura 5.1: Componentes de Servidor IPTV

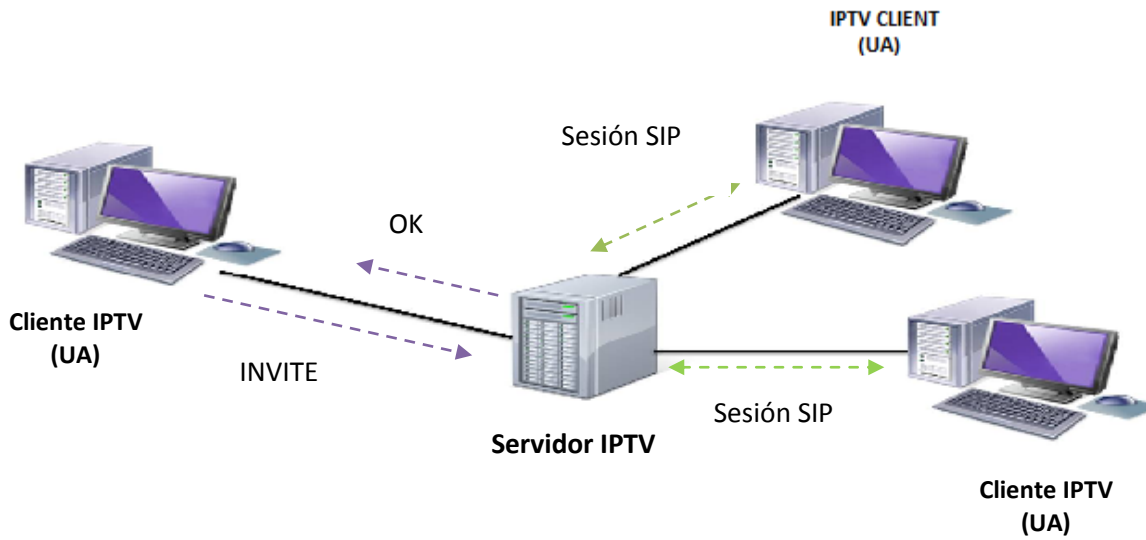


Figura 5.2: Plano de Control

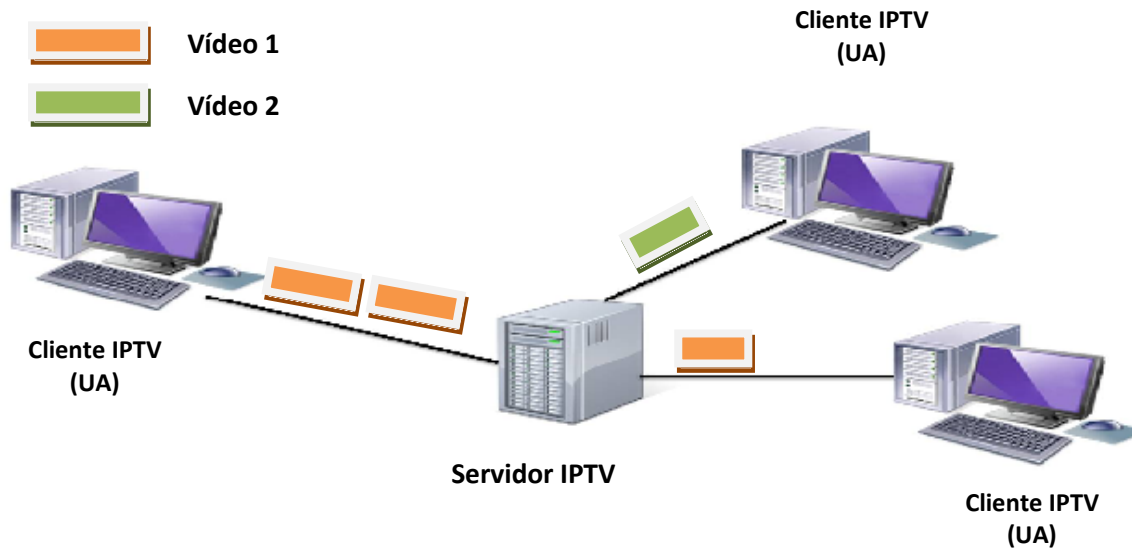


Figura 5.3: Plano de Datos

5.2. PLANO DE CONTROL O SEÑALIZACIÓN

Esta parte se encarga del desarrollo del servidor SIP. Este servidor, actuará, en lo referente a la realización del establecimiento y liberación de la sesión multimedia. Por otro lado, también se desarrolla el cliente IPTV. El cliente IPTV es el encargado de iniciar y terminar la sesión multimedia. El cliente IPTV es el primero en ejecutar el proceso de señalización, enviando un mensaje INVITE al servidor SIP. El servidor SIP enviará un mensaje TRYING y OK para indicar al cliente IPTV que acepta la sesión multimedia que el cliente IPTV ha propuesto.

Antes de continuar con la definición de los requisitos, tenemos que entender el marco en el que se encuentran los mismos. Como se mostraba en la [figura 5.1](#), el Servidor IPTV está formado por dos servidores a su vez. Uno de ellos es el Servidor SIP, espera hasta que el Cliente/Usuario mande una solicitud para establecer una sesión multimedia y en caso de recibirla, Servidor SIP manda su respuesta. El Servidor SIP sólo participa en la fase de establecimiento y liberación de la sesión, ya que el protocolo SIP no interviene en el envío de los datos multimedia.

En la [figura 5.4](#) se muestra un ejemplo con los mensajes de señalización SIP intercambiados entre un cliente y el servidor SIP. El usuario (Cliente IPTV, que también consta de dos partes una con referencia a la sesión multimedia y la otra al visionado del contenido) y el servidor SIP, serán los encargados de generar las solicitudes y respuestas SIP.



Figura 5.4: Establecimiento de la sesión

Inicialmente, el usuario accede al servicio. De este modo, su cliente IPTV genera una solicitud INVITE de SIP que es dirigida al Servidor SIP, y este es el encargado de autorizar si el usuario puede acceder al servicio y, en caso afirmativo, generar una respuesta OK de SIP. En este caso, se establece una sesión multimedia basada en multicast y el cliente comienza a recibir el contenido multimedia que está ofertando el servidor. En el [Anexo E](#) se puede 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 (de modo que el usuario podrá acceder a un canal de televisión, y cambiar de canal cuándo lo estime oportuno), y será el propio cliente el que indicará en una oferta SDP la información correspondiente a los canales que potencialmente querría visualizar, así como información de direccionamiento correspondiente a los canales (ej. dirección IP multicast del canal que desea comenzar a visionar en primer lugar). Asimismo, en la carga SDP también se indican el listado de códec de vídeo soportado. Esta negociación es la que se ha comentado en el [capítulo 3](#), el modelo de oferta/respuesta de SDP.

Así, en la figura anterior, es decir, en la [figura 5.4](#), el mensaje INVITE incluye una oferta SDP, en la cual se indican los parámetros necesarios correspondiente a los canales que se desean visionar, y la información de direccionamiento correspondiente al primer canal con el que se arrancará el servicio (esto es, la dirección IP multicast y el puerto de transporte correspondiente a este canal).

Cabe destacar que no se pueden ver varios vídeos a la vez, si no que para ver otro vídeo es necesario cerrar un vídeo para que la aplicación permita la opción de que se vuelva a elegir otro vídeo. De esta forma el cliente IPTV se unirá al canal multicast que corresponda con el vídeo que desee ver el usuario. También se propone una liberación de la sesión multimedia en caso de que el usuario decida abandonar la sesión. Se puede observar en la [figura 5.5](#).

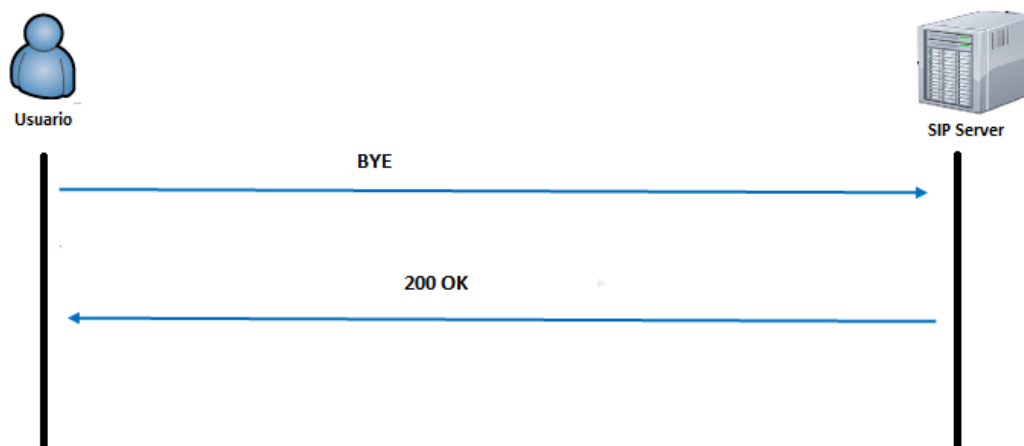


Figura 5.5: Liberación de la sesión

5.2.1. Cliente IPTV

El cliente IPTV es el que se encarga de iniciar el establecimiento de sesión, así como su liberación, para la recepción del tráfico de vídeo. Los requisitos necesarios que debe tener el desarrollo del cliente IPTV se enumeran a continuación:

- La implementación del cliente IPTV se realizará mediante una aplicación JAVA. El cliente dispondrá de una interfaz gráfica, donde el usuario podrá visualizar el contenido multimedia recibido.
- La implementación soportará los mecanismos correspondientes al establecimiento y terminación de sesiones basadas en el protocolo SIP.
- El cliente IPTV generará las ofertas SDP necesarias para el establecimiento de la sesión multimedia multicast que es necesaria para la provisión del servicio IPTV. Dichas ofertas se crearán de acuerdo a las especificaciones del servicio IPTV de TISPAN, y contendrán información de direccionamiento para el acceso al servicio (ej. la dirección IP multicast del primer canal que se desea visionar) información sobre el listado de los canales para los que el cliente solicita autorización de acceso.
- El cliente IPTV tendrá pre-configurada la URI SIP correspondiente al servicio IPTV. Esta URI, que contiene un nombre de usuario así como el nombre de dominio correspondiente al servicio IPTV, le permitirá intercambiar mensajes de SIP con el servidor SIP del servidor IPTV.

5.2.2. Servidor SIP

Este servidor, es el encargado de ejecutar, conjuntamente con el cliente IPTV, todos los procedimientos de establecimiento de sesión, así como la liberación de dicha sesión. Como se comento al principio del capítulo, servidor SIP sólo participa en la fase de establecimiento y liberación de la sesión, y una vez está establecida ésta, no interviene en el envío de los datos multimedia.

A continuación se indican los principales requisitos de diseño que se han definido para el servidor SIP:

- La implementación del servidor IPTV se realizará mediante una aplicación JAVA. El servidor SIP no dispone de ninguna interfaz gráfica, solamente es un desarrollo de una aplicación JAVA (ya que no es el objetivo de este TFG).
- La implementación soportará los mecanismos correspondientes al establecimiento y terminación de sesiones basadas en el protocolo SIP.
- El servidor SIP negociará las ofertas/respuestas SDP necesarias para el establecimiento de la sesión multimedia multicast que es necesaria para la provisión del servicio IPTV. Dichas ofertas se crearán de acuerdo a las especificaciones del servicio IPTV de TISPAN, y contendrán información de direccionamiento para el acceso al servicio (ej. la dirección IP multicast del

primer canal que se desea visionar) información sobre el listado de los canales para los que el cliente solicita autorización de acceso.

- El servidor IPTV tendrá pre-configurada la URI SIP correspondiente al servicio IPTV. Esta URI, que contiene el nombre del servidor así como el nombre de dominio correspondiente al servicio IPTV, le permitirá intercambiar mensajes de SIP con el cliente IPTV.

5.3. Plano de datos

El plano de datos está formado por todo lo relacionado con el contenido multimedia, es decir, con el envío del vídeo. En el plano de datos contribuye al servidor streaming de vídeo. Antes de pasar al plano de datos, se debe de tener establecida la sesión multimedia. Ésta se lleva a cabo en el Plano de control ([Véase 5.3](#)). A continuación se procede a realizar una explicación de cómo afectan al plano de datos de la aplicación desarrollada.

5.3.1. Cliente IPTV

Una vez que ya se tiene establecida la sesión multimedia entre el Servidor y el Cliente, al usuario se le mostrará por pantalla una interfaz gráfica. Este cliente IPTV permite reproducir más de un solo contenido multimedia. Cada contenido multimedia (vídeo), será recibido por un canal multicast diferente. El cliente IPTV de forma transparente al usuario se une al grupo multicast perteneciente al contenido multimedia que desea visionar, utilizando para ello el protocolo de gestión de grupos multicast en Internet (IGMP [[IGMP](#)]).

Los requisitos del diseño del cliente se detallan a continuación:

- El cliente IPTV debe de soportar el visionado de múltiples canales.
- Debe de soportar la recepción basada en multicast. Ya que los vídeos que son transmitidos por el servidor de streaming se hacen por canales multicast. Para ello deberá soportar el protocolo IGMP, utilizado para la gestión de grupos multicast en internet.
- El cliente IPTV deberá de auto-configurarse en relación con las librerías de reproducción de vídeo con las mismas direcciones IP y puertos indicados en la carga SDP.
- El cliente IPTV deberá de permitir al usuario que cambie de vídeo en el momento que lo desee. Para ejecutar un cambio de canal, debe tenerse en cuenta que cada contenido multimedia se va a emitir en un canal multicast diferente, y de modo que será el cliente IPTV, a indicación del usuario, se una al grupo multicast deseado en cada caso.

5.3.2 Servidor de streaming de vídeo

Esta parte del Servidor IPTV es muy simple y sencilla. No se ha desarrollado nada con referencia al servidor de streaming de vídeo. Es la aplicación VLC[VLCa]. Al lanzar dicha aplicación, lo primero que se debe de hacer es configurarla, tal y como se explica en el [anexo G](#). Los requisitos que debe cumplir dicho servidor, se comentan a continuación, aunque cabe destacar, que todos estos requisitos el programa VLC los cumple.

- Soportar varios códec para la emisión del video.
- Emisión de vídeo por canal multicast.
- Emitir vídeos por el mismo puerto y por distinto canal multicast.

5.4 Marco regulatorio

La implementación de este sistema debe de adaptarse a la legislación vigente sobre la Ley Orgánica 15/1999 de 13 de diciembre de Protección de Datos de Carácter Personal, (LOPD) [\[MR10\]](#), puesto que las empresas privadas tanto como administraciones públicas tendrían acceso a unos datos e información personales.

6. Implementación de los elementos del sistema

6.1. Introducción

En este capítulo se describe la implementación realizada de los elementos que forman del sistema IPTV, es decir, se describe la implementación tanto del cliente IPTV, cómo del servidor IPTV. Por conveniencia, en la [figura 6.1](#) se describe de forma visual los distintos componentes del servicio IPTV que se han implementado.

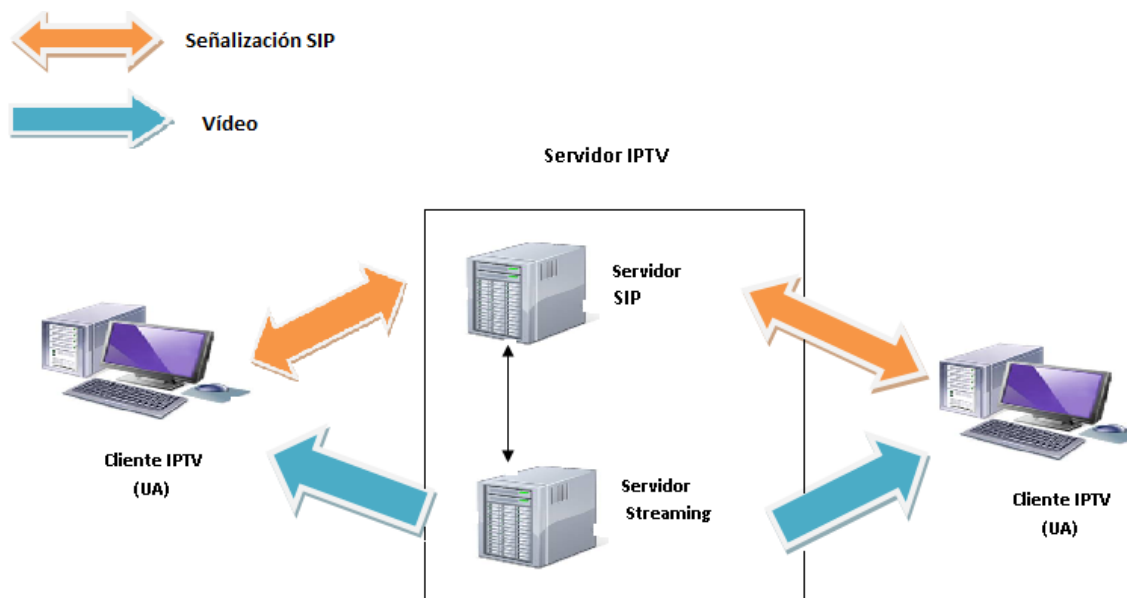


Figura 6.1: Sistema a Implementar

Tal y como se observa, el servicio implementado opera en dos planos: un plano de señalización, que está basado en el protocolo SIP, y que soportará el establecimiento de la sesión multimedia necesarias para recibir el vídeo correspondiente a los canales IPTV; y un plano de datos, sobre el que se realiza la entrega del vídeo mediante tecnología de entrega multicast. Este plano soporta también los cambios de canal mediante el uso del protocolo IGMP.

En el diseño que se va a implementar, el cliente IPTV y el servidor SIP implementan las funcionalidades correspondientes al plano de control, de modo que intercambiarán los mensajes de señalización SIP necesarios para el establecimiento y liberación de las sesiones multimedia. Una vez establecida la sesión, los paquetes con contenido multimedia de vídeo, del canal IPTV solicitado

por el cliente, serán recibidos directamente desde el servidor de streaming de vídeo, en la dirección IP multicast correspondiente al canal IPTV. En esta parte el servidor SIP no intervendrá.

Una vez finalizada la señalización y establecida la sesión, los paquetes con contenido multimedia serán enviados directamente desde el servidor streaming al puerto del cliente que se estableció en la fase de señalización. En esta parte el Servidor SIP no intervendrá.

El lenguaje de programación sobre el que se ha desarrollado esta aplicación, es en JAVA. La solución está formada por varias clases desarrolladas en el lenguaje de programación indicado anteriormente. Para la señalización de SIP en JAVA existe una API llamada JAIN SIP [[JAIN1](#)] [[JAIN2](#)]. Con esta API se gestiona toda la señalización de SIP, de modo que se simplifica la programación el código sin la preocupación de la complejidad de la sintaxis de los mensajes SIP y del transporte de los mismos, ya que esto se realiza a través de una pila de protocolos SIP.

La API JAIN comprende un conjunto de librerías basadas en tecnología Java que permiten el desarrollo de productos de telecomunicaciones y servicios sobre la plataforma Java. Inicialmente JAIN consta de dos desarrollos de API's (SIP Listener y SIP Provider, véase [figura 6.2](#)). La API que maneja lo concerniente a señalización SIP y el desarrollo de la API que maneja lo referente al direccionamiento. La API JAIN SIP puede ser fácilmente integrada en una variedad de plataformas con acceso a numerosas herramientas y utilidades. La API JAIN SIP encapsula las funciones del protocolo SIP con una interfaz normalizada de Java, de manera que el inicio de sesión está totalmente disponible. La implementación se desarrolla con las dos partes principales en las que se divide el API JAIN SIP, SIP Listener y SIP Provider, que interactúan con la pila SIP. 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.

Del API JAIN SIP se utilizan algunas librerías para la gestión de señalización SIP, dichas librerías son:

- **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.

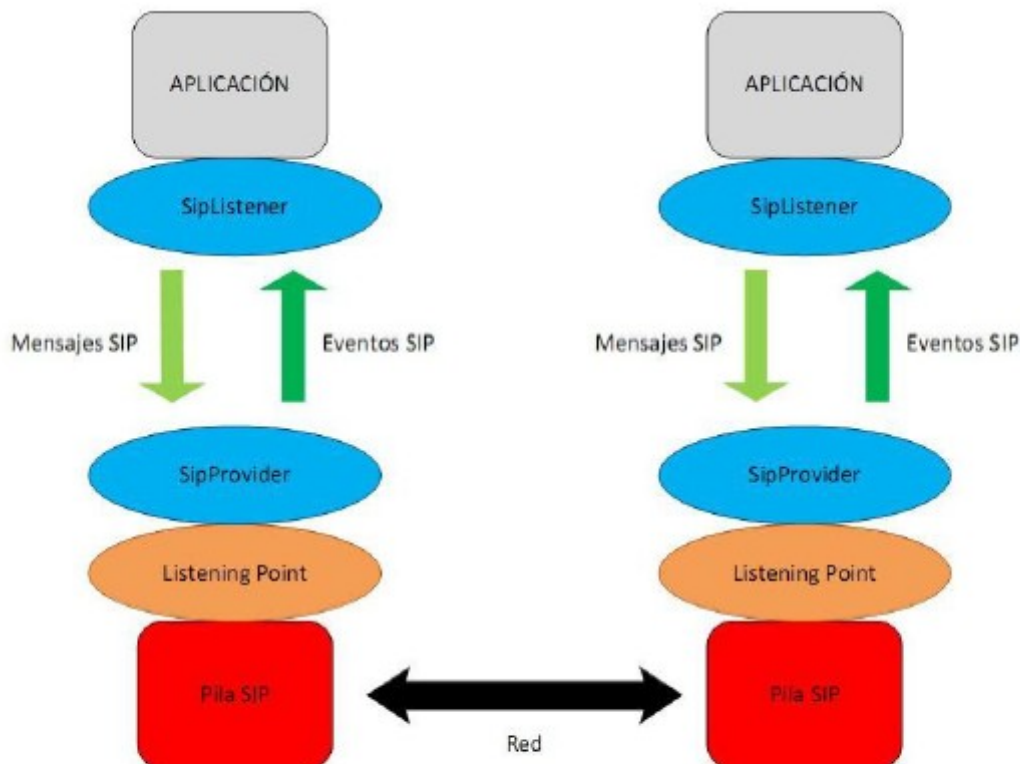


Figura 6.2: Arquitectura JAIN SIP [JAIN2].

Además de la API JAIN SIP, para el procesamiento del tráfico de vídeo en el cliente IPTV, y soportar la reproducción del mismo, es necesario la utilización de otro tipo de librerías. Estas librerías utilizadas para la aplicación es el framework VLCj [VLC] de JAVA:

- **vlcj-1.2.2.jar**: contiene la implementación de las clases de VLCj.
- **jna.jar**: permite el acceso a las librerías nativas.

En la implementación realizada, el usuario ejecutará una aplicación desarrollada en JAVA (cliente IPTV). Una vez lanzada dicha aplicación, se establece la sesión multimedia de forma satisfactoria con el servidor. Así, de forma automática, al arrancar el cliente IPTV, este inicia el establecimiento de la sesión SIP, generando y procesando las peticiones y respuestas SIP con el Servidor SIP. Con respecto al Servidor IPTV, cabe destacar que está dividido a nivel lógico en dos sub-componentes, el servidor SIP y el servidor de Streaming de vídeo (véase [figura 5.1](#)). El servidor SIP se encarga de gestionar las solicitudes y respuestas SIP y una

vez se tenga establecida la señalización, será el servidor de streaming de vídeo el encargado de la transmisión del contenido multimedia. Dicho servidor, es la aplicación de emisión de vídeo para que ésta, a su vez, retransmita el contenido multimedia al Cliente IPTV.

A continuación se detalla la implementación de cada uno de los elementos que forman parte de la solución adoptada. Así en la [figura 6.3](#) se puede ver un esquema de la implementación realizada.

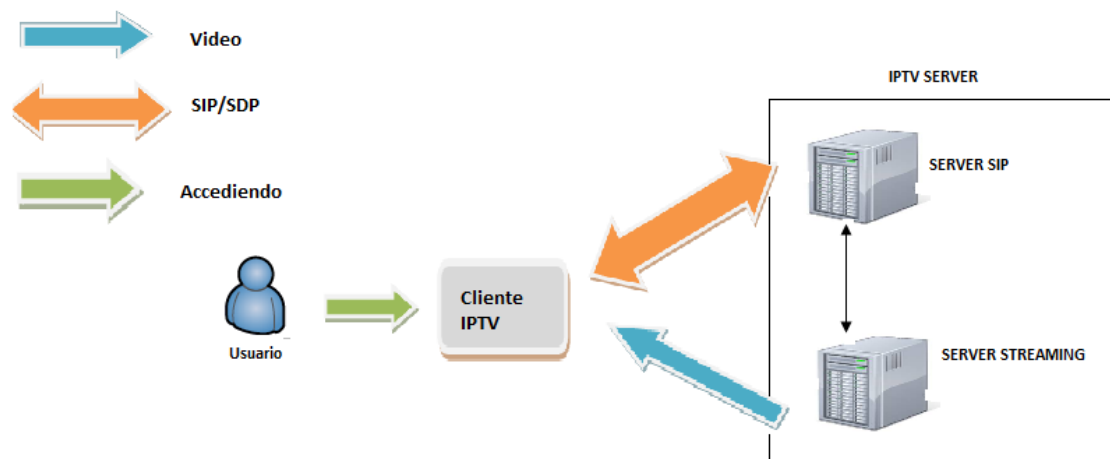


Figura 6.3: Esquema

6.2. Cliente IPTV

La implementación del Cliente IPTV está basada en el establecimiento de sesión SIP, el procesamiento y reproducción de vídeo, cambio de canal mediante IGMP y terminación de sesiones SIP. El establecimiento y liberación de sesión se realiza de forma automática y transparente al usuario, por lo tanto, no requiere de ningún tipo de configuración adicional por parte del usuario.

6.2.1. Gestión de la señalización SIP

Una vez que el usuario activa el cliente IPTV lanzado la aplicación JAVA, dicho cliente inicia, de forma automática, el establecimiento de una sesión multimedia basada en SIP con el servidor SIP. Así mismo, cuando el usuario cierre la aplicación JAVA, de forma automática se libera la sesión SIP. En apartados posteriores, se explicará de forma detallada dicho establecimiento de sesión.

6.2.2. Interfaz gráfica

La interfaz gráfica con la que el usuario podrá interactuar se muestra en la [figura 6.4](#). Se trata de una interfaz muy sencilla, ya que el objetivo de este TFG no se centra en la interfaz gráfica sino en las funcionalidades del plano de control y del plano de datos anteriormente mencionadas.

Es muy sencilla de utilizar, por lo que el usuario no tendrá problemas para interactuar con ella. Consta de varios botones, uno por cada canal IPTV disponible. Inicialmente, la aplicación cuyo funcionamiento se ha probado consta de dos botones, que permiten visionar dos canales, si bien el número de botones (y de canales) es fácilmente ampliable. Al pulsar en cada botón se abrirá una nueva ventana, en la cual se mostrará el vídeo del canal correspondiente. Mientras se está viendo un vídeo, el usuario no podrá abrir el otro canal para visionar el vídeo, si no se ha cerrado la ventana del canal que haya pulsado primero. Tras cerrar la ventana del vídeo que se está visionando, y abrir la ventana correspondiente al otro canal, se ejecutan los procesos correspondientes al otro canal, se ejecutan los procesos correspondientes al cambio de canal, basados en el protocolo IGMP.

Para el correcto visionado del contenido multimedia, se requiere de la **Ruta*** del sistema operativo dónde estén almacenadas las librerías de la aplicación VLC.



Figura 6.4: Interfaz Gráfica

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

6.2.3. Procedimientos en el cliente

6.2.3.1. Establecimiento de sesión

Para una mejor comprensión del establecimiento de sesión, se realiza por conveniencia un esquema mostrado en la [figura 6.5](#), en el cual se describe el proceso.

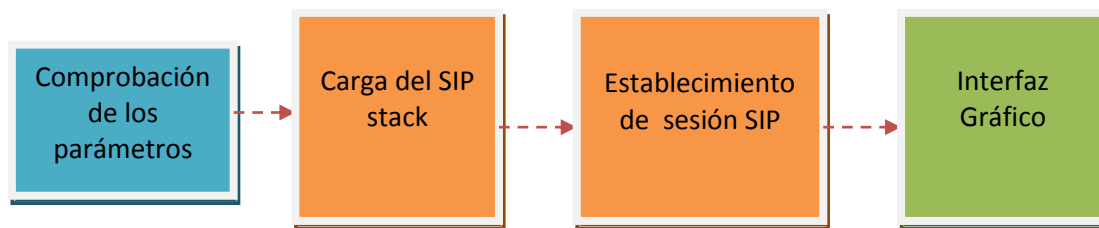


Figura 6.5: Esquema del proceso

Una vez que se han comprobado todos los parámetros (URI SIP, puerto), se procede a realizar la señalización SIP de forma automática, es decir, el usuario no tiene que realizar nada para realizar el establecimiento de sesión, ya que como se ha comentado, se realiza de forma automática y transparente al usuario. Por tanto, se genera una petición INVITE dirigida al servidor SIP, ya que es el encargado de generar el proceso de señalización y será el encargado de aceptar la sesión multimedia.

En la [figura 6.6](#) se observa el mensaje INVITE que envía el Cliente IPTV mostrado por consola. Así mismo en la [figura 6.7](#) se observa de forma esquemática como se realiza este proceso.


```

Problems @ Javadoc Declaration Console
<terminated> CLIENT [Java Application] C:\Program Files (x86)\Java\jre1.8.0_25\bin\javaw.exe (12/01/2015 19:37:49)
/n SDP: INVITE sip:PruebaServidor@127.0.0.1:7060 SIP/2.0
Call-ID: 986c1c149aad6e8a573442b92a644c49@127.0.0.1
CSeq: 1 INVITE
From: <sip:PruebaCliente@proyecto.com>;tag=12345
To: <sip:PruebaServidor@proyecto.com>
Via: SIP/2.0/UDP 127.0.0.1:7070
Max-Forwards: 70
Contact: <sip:PruebaCliente@127.0.0.1:7070>
Content-Type: application/sdp
Content-Length: 160

v=0
o=Prueba 1057165447541 1057165447541 IN IP4 127.0.0.1
s=-
c=IN IP4 239.0.0.1
t=0 0
m=video 1234 udp OSA
b= BW: 13Mga=rtpmap:0 PCMU/8000+ a=rtpmap:8000

INVITE ENVIADO
    
```

Figura 6.6: INVITE enviado mostrado por consola

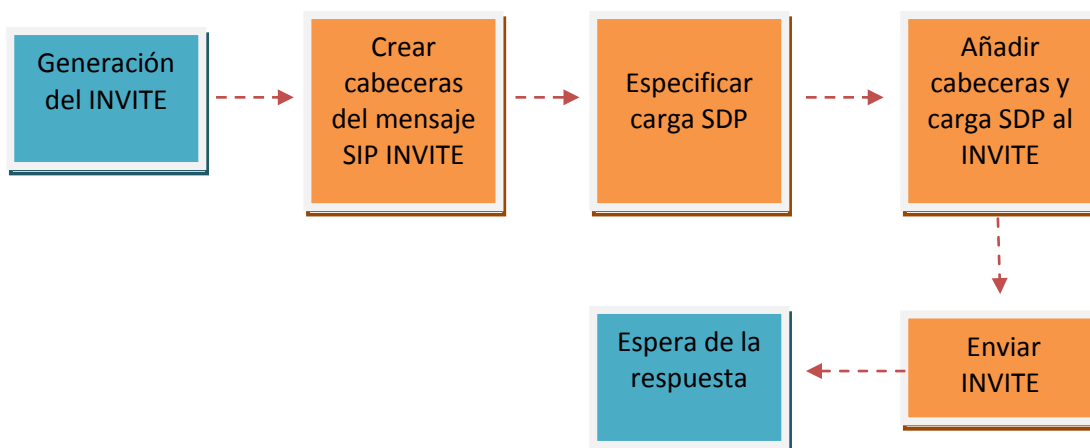


Figura 6.7: Esquema de generación del INVITE

Una vez que se ha realizado el envío del INVITE, la aplicación se queda esperando para recibir la respuesta del servidor SIP. Cuando le llega la respuesta 200 OK, el cliente IPTV, procesa dicha respuesta y genera el correspondiente ACK, de forma esquemática se puede observar en la [figura 6.8](#). Si por cualquier motivo, la respuesta del servidor IPTV no es un 200 OK, si no que es una respuesta de error, no se ha contemplado en la implementación de este TFG.

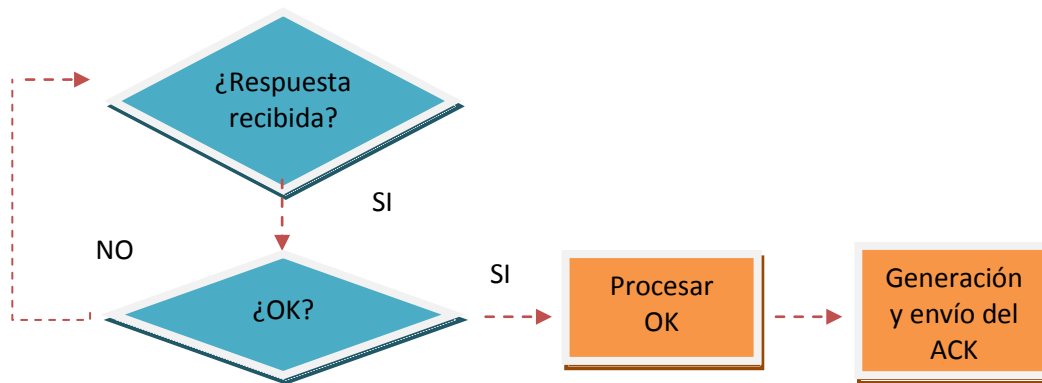


Figura 6.8: Procesamiento del OK y generación del ACK

Una vez realizado todo lo comentado anteriormente, el usuario ya tiene establecida la sesión multimedia.

6.2.3.2. Reproducción del contenido multimedia

Una vez que está establecida la sesión multimedia, se mostrará por pantalla el interfaz gráfico del cliente IPTV (mostrado en la [figura 6.4](#)) y será el usuario el que elija el canal que va a visionar. Dicho contenido procederá del servidor de Streaming de vídeo. El contenido llegará a través del puerto que se haya establecido en la carga SDP, generada por el cliente IPTV e incluida en la solicitud INVITE, para la recepción del canal correspondiente. Cada vídeo se va a emitir en una dirección IP multicast distinta, pero por el mismo puerto, según se indica en las especificaciones de TISPAN, para facilitar así la configuración del servidor de Streaming de vídeo.

En la [figura 6.9](#), se puede observar de forma esquemática los pasos seguidos para la reproducción del vídeo.

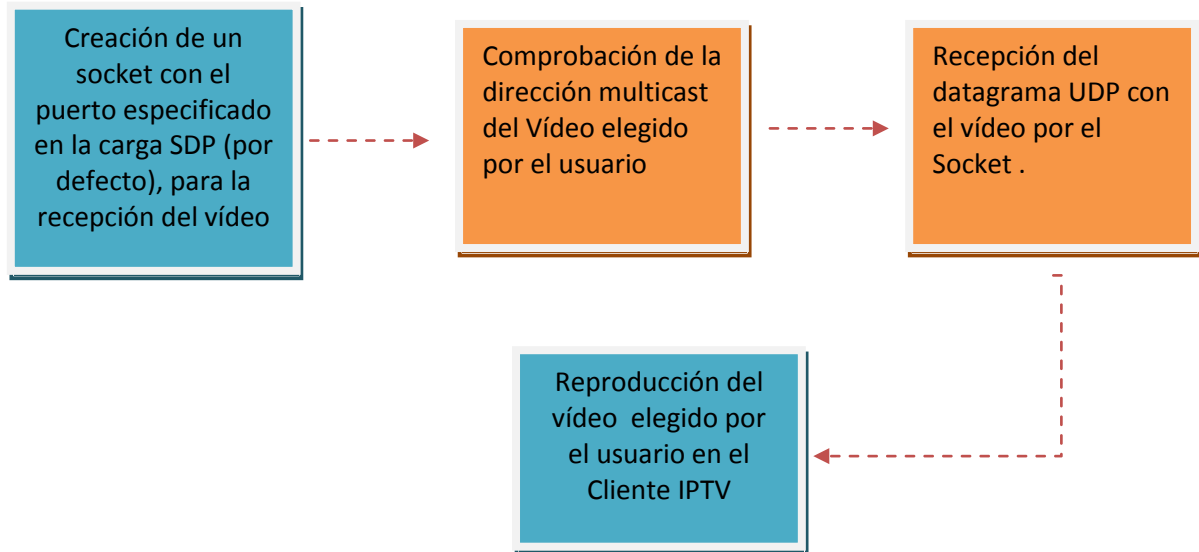


Figura 6.9: Esquema Reproducción de contenido multimedia

Cuando el usuario desea cambiar de vídeo, se produce un cambio de canal. Este cambio se basa en el protocolo IGMP. El soporte para IGMP lo ofrece VLCj. El protocolo IGMP permite conocer a que grupos multicast pertenece cada equipo, en este caso, a qué grupo multicast pertenece cada vídeo. En el [anexo E](#), se puede observar mayor información acerca de dicho protocolo.

6.2.3.3. Liberación de la sesión

Cuando el usuario decida abandonar la sesión multimedia, solamente tendrá que pulsar el botón de cerrar de las ventanas que se han abierto. Como es una interfaz muy sencilla, esto no supondrá un problema al usuario. Una vez hecho esto, el cliente IPTV generará un mensaje BYE de forma automática con destino el servidor SIP. En la [figura 6.10](#) se observa el mensaje BYE mostrado por la consola de Cliente IPTV y en la [figura 6.11](#) y [6.12](#) se ve de forma esquemática dicho proceso.

```
----- BYE SENT:
BYE sip:127.0.0.1:7060 SIP/2.0
Via: SIP/2.0/UDP 127.0.0.1:7070;branch=z9hG4bK1e4c4f8bc13b1fa9c621d24fad84d0da
CSeq: 2 BYE
Call-ID: 986c1c149aad6e8a573442b92a644c49@127.0.0.1
From: <sip:PruebaCliente@proyecto.com>;tag=12345
To: <sip:PruebaServidor@proyecto.com>;tag=4321
Max-Forwards: 70
Content-Length: 0

Response received : Status Code = 200 CSeq: 2 BYE
```

Figura 6.10: Mensaje mostrado por consola BYE



Figura 6.11: Desconexión

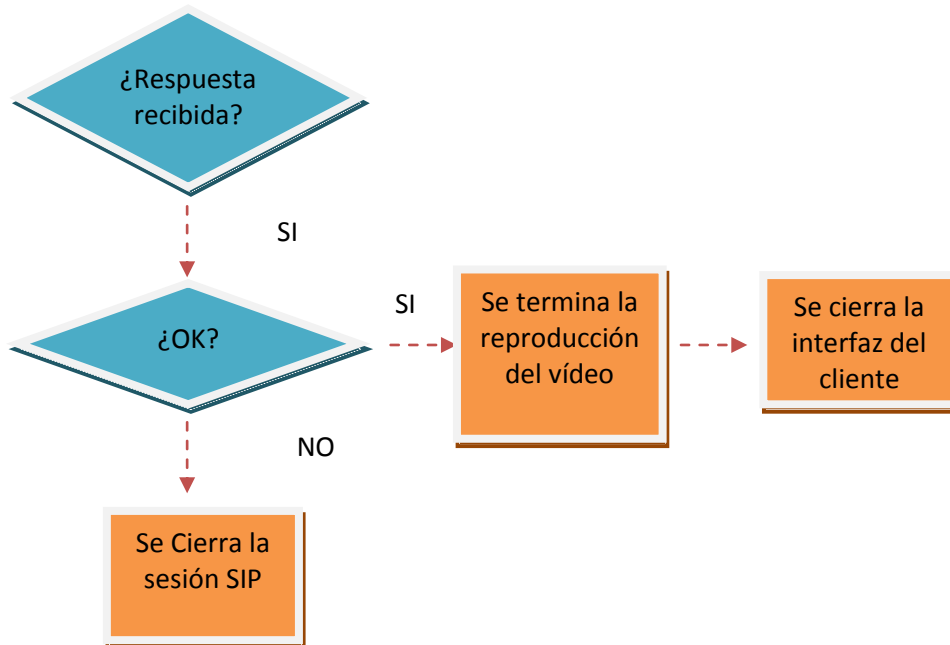


Figura 6.12: Procesado OK del BYE

6.3. Servidor IPTV

Como se ha comentado en apartados anteriores, el Servidor IPTV está compuesto por dos sub-componentes, el servidor SIP y el servidor de streaming de vídeo (véase [figura 5.1](#)).

Este TFG no tiene como finalidad la creación de un servidor de streaming de vídeo, por ese motivo se ha realizado un sencillo servidor SIP (aplicación JAVA de un servidor SIP), el cual no consta de interfaz gráfica, y se ha desplegado el servidor de streaming de vídeo, mediante el uso de la herramienta VLC [[VLC](#)]. Para la ejecución de ambos sub-componentes se utilizará un terminal de comandos (o en su defecto el entorno de desarrollo Eclipse).

6.3.1. Procedimientos en el servidor

A continuación, se procede a explicar 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.

6.3.1.1. Establecimiento de sesión

Una vez que las variables del Servidor IPTV están inicializadas de forma correcta, el servidor permanece en estado de espera hasta que recibe la primera solicitud, que será un INVITE procedente del Cliente IPTV. Una vez que el Servidor IPTV está configurado, permanece en estado de espera hasta que recibe alguna solicitud SIP. Lo que hace el Servidor SIP es almacenar ciertos datos de la sesión para tener en todo momento localizado los elementos que forman parte de ella. Para ello, realiza varias operaciones:

- Almacenar en un objeto dialog (pertenecientes de la clase Dialog.java) toda la información para poder identificar este diálogo iniciado por el cliente. Todos estos datos son necesarios para el establecimiento de la sesión. Por ejemplo la cabecera de contacto, número de secuencia del INVITE,...
- Comprobación de los códec que vídeo que se utilizarán durante la transmisión. Estos datos los recoge de la carga SDP que se recibe en el INVITE.

El servidor mira en la carga SDP (que se recibe en el INVITE) y en ella se conocen los canales multicast que potencialmente podrá solicitar el cliente para su visionado, y el puerto hacia donde irán dirigidos los paquetes con el contenido multimedia.

Una vez que reciba la solicitud, el servidor actúa tal y como se explica de forma esquemática en la [figura 6.13](#).

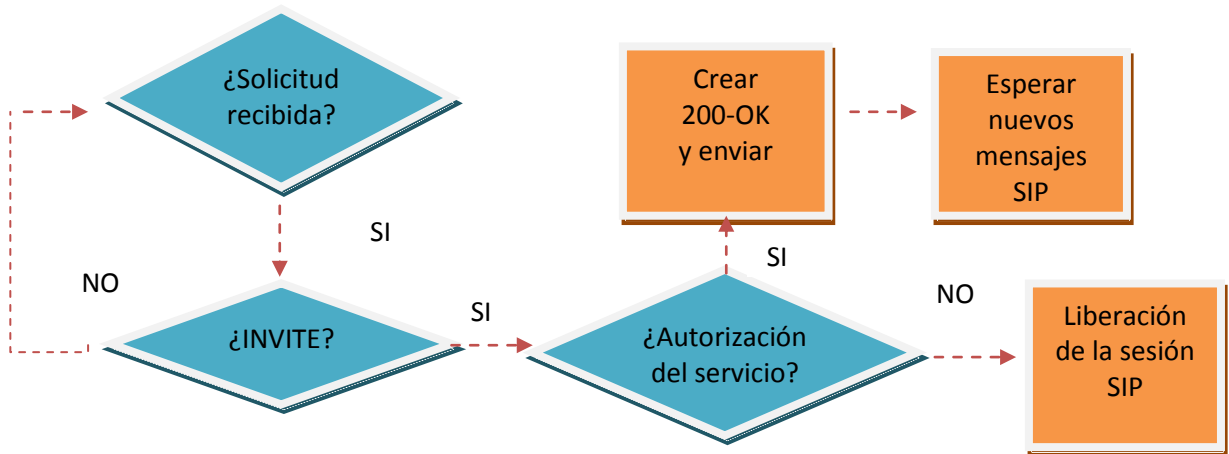


Figura 6.13: Procesado INVITE en el Servidor

Una vez que el servidor recibe el INVITE, generará, en el caso que la política de autorización del proveedor IPTV sea afirmativa, una respuesta 200 OK con su respuesta SDP correspondiente, en la que el códec de vídeo será el mismo que viniera. La autorización del acceso al servicio, no se ha implementado para que sea el propio proveedor IPTV el que la implemente a su conveniencia. 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, en el caso en que la autorización del proveedor IPTV acepte la sesión multimedia, es un ACK que le envía el cliente IPTV. Este ACK confirma que el destinatario ha aceptado establecer de la sesión, pudiendo recibir más peticiones y respuestas de otros usuarios. El ACK se envía como asentimiento de la respuesta 200 OK del cliente. En la [figura 6.14](#), se puede observar de forma esquemática el procesado del ACK.

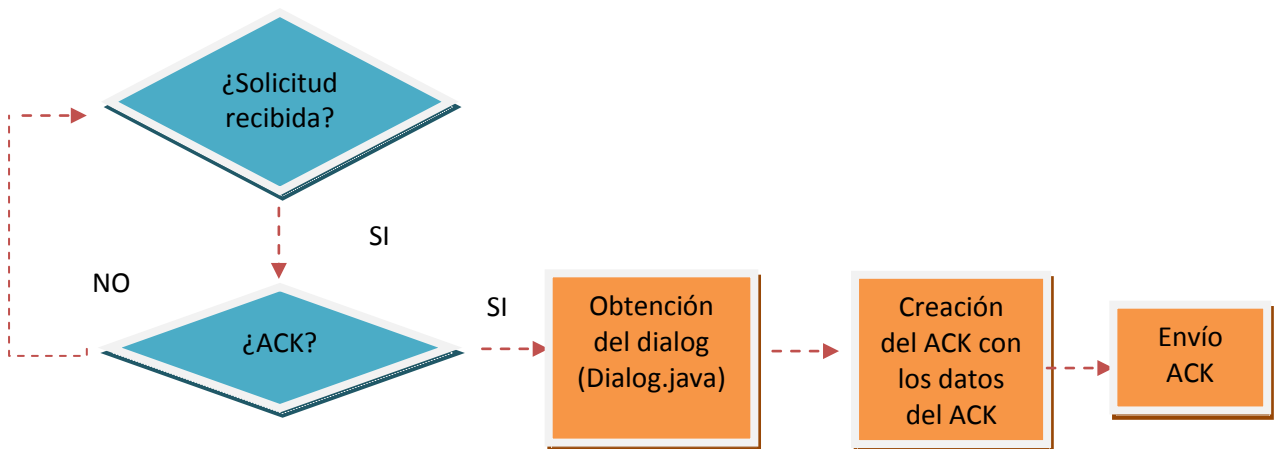


Figura 6.14: Generación del ACK

6.3.1.2. Envío del contenido multimedia

La parte encargada de realizar el envío del contenido multimedia dentro del Servidor IPTV, es el servidor de Streaming de vídeo (véase [figura 5.1](#)). Una vez que está establecida la sesión multimedia entre el cliente y el servidor, el cliente IPTV comienza a recibir el contenido multimedia que esté emitiendo el servidor IPTV, tal y como lo muestra la [figura 6.15](#).

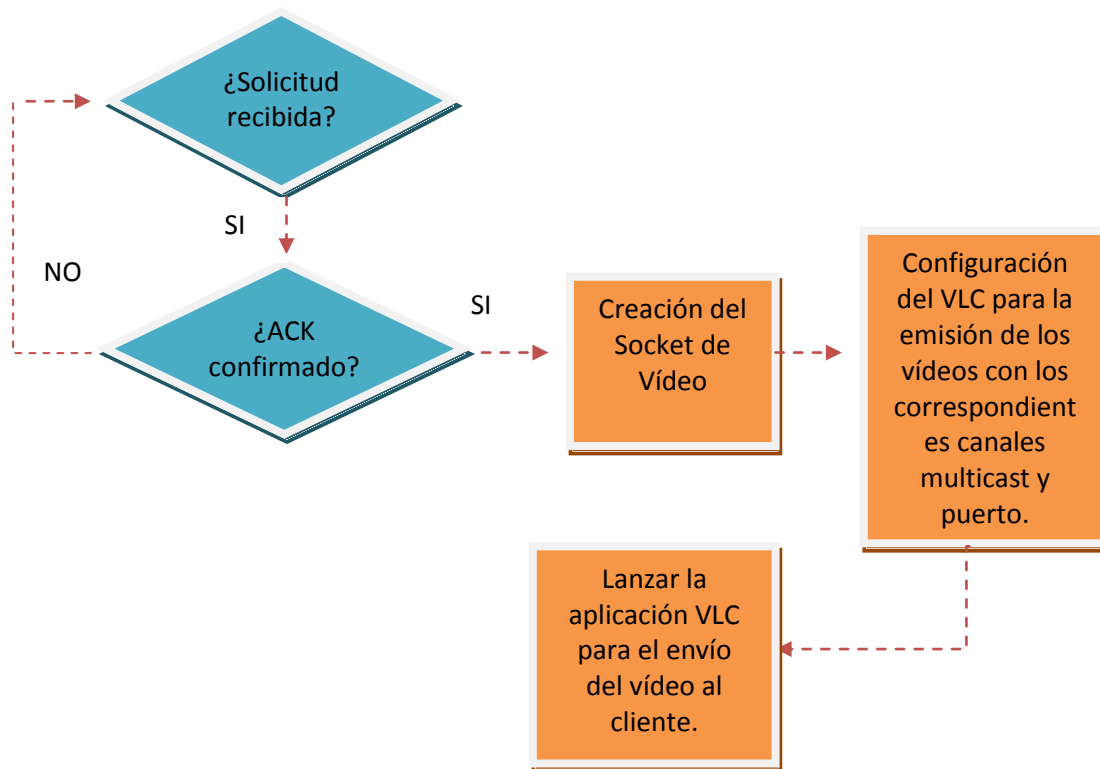


Figura 6.15: Envío de contenido multimedia

6.3.1.3. Liberación de la sesión

Cuando el servidor SIP recibe un mensaje BYE procedente del cliente IPTV, deja de realizar la transmisión de contenido multimedia al usuario, ya que decide abandonar la sesión multimedia. De forma automática se cierra el socket que se tenía establecido para la transmisión. Posteriormente se genera la respuesta 200 OK a la petición de BYE recibida por parte del Cliente IPTV, como se observa en la [figura 6.15](#).

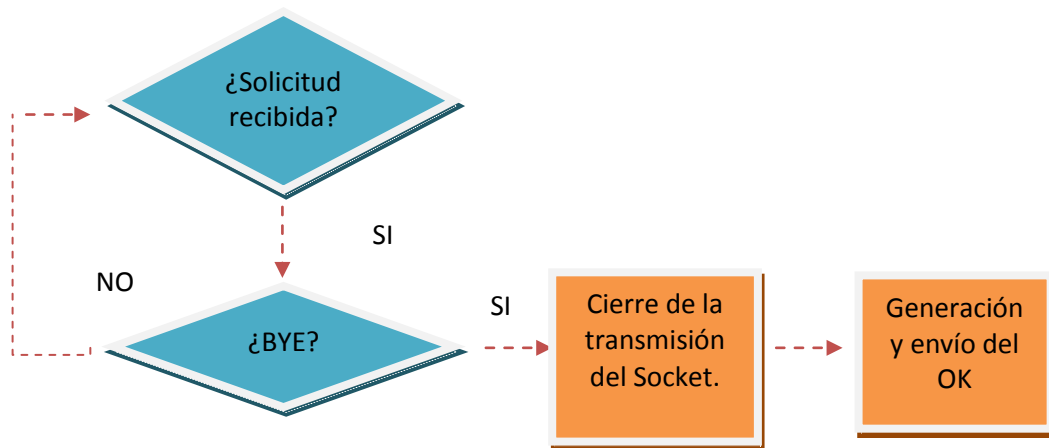


Figura 6.16: Liberación de sesión

Como respuesta a la petición BYE se debe de recibir un 200 OK como para abandonar la sesión multimedia. Al recibir dicha respuesta, todos los datos alojados en el objeto dialog (de la clase Dialog.java) serán eliminados y la sesión multimedia estará finalizada, tal y como se muestra en la [figura 6.16](#).

7. Escenario de pruebas

7.1. Introducción

Este desarrollo se ha probado en un entorno de red de área local, en el laboratorio de Telemática de la Universidad Carlos III de Madrid, con diez equipos. Esta red está conectada mediante un switch y por ese motivo soporta el envío de datagramas UDP.

Este capítulo se centra en la explicación del segundo escenario de pruebas probado, basta que es mucho más ilustrativo y se comprueba de mejor forma el funcionamiento la implementación de este TFG. De los diez equipos sometidos a las pruebas, cinco de ellos iniciaron el servicio accediendo a un canal IPTV, mientras que los cinco restantes lo hicieron accediendo a otro canal (las pruebas se realizaron para 2 canales de televisión). Así mismo, tras iniciar el servicio y visualizar el vídeo correspondiente durante un cierto tiempo, tres de los equipos realizaron un cambio de canal. Durante todas las pruebas se capturó todo el tráfico intercambiado desde uno de los equipos mediante la herramienta Wireshark [WIR]. En la [figura 7.1](#) se ve de forma esquemática como fue el escenario de pruebas realizado.

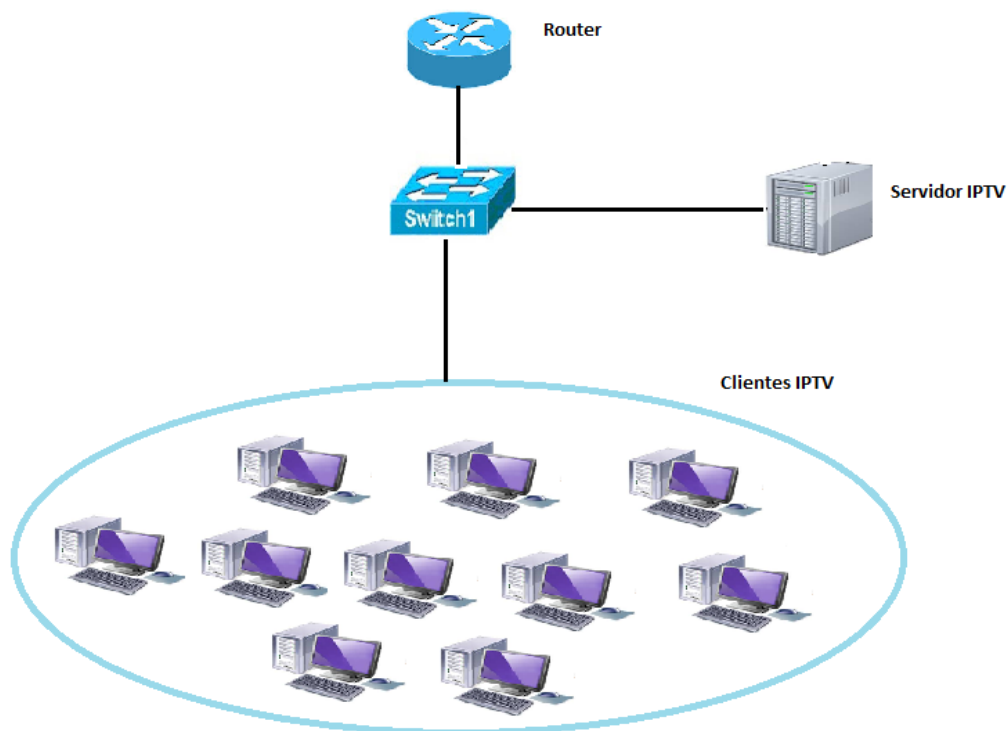


Figura 7.1: Escenario de Pruebas (Lab. Telemática)

Los datos que se han empleado en la configuración del escenario de pruebas se pueden observar en la [tabla 7.1](#) donde, para cada equipo o host utilizado, se indica su dirección IP unicast asignada, así como el puerto SIP, necesario para el establecimiento de la sesión multimedia, que utiliza cada equipo y el puerto de vídeo, es decir, el puerto por el cual va a recibir el contenido multimedia cada equipo.

HOST	Dirección IP	Puerto SIP	Puerto Vídeo
UA 1	163.117.144.102	7070	1234
UA2	163.117.144.103	7070	1234
UA 3	163.117.144.104	7070	1234
UA 4	163.117.144.107	7070	1234
UA 5	163.117.144.109	7070	1234
UA 6	163.117.144.113	7070	1234
UA 7	163.117.144.114	7070	1234
UA 8	163.117.144.115	7070	1234
UA 9	163.117.144.119	7070	1234
UA 10	163.117.144.120	7070	1234
Servidor SIP	163.117.144.108	5060	----
Servidor Streaming	239.0.0.1/239.0.0.2	----	1234

Tabla 7.1: Elementos del Sistema

A continuación se procede a explicar los pasos realizados en el escenario de pruebas, para concluir que el funcionamiento del desarrollo realizado es correcto.

7.2. Establecimiento de sesión

El servidor SIP está constantemente a la espera de que un cliente IPTV, inicie el establecimiento de una sesión SIP. Tras recibir una solicitud INVITE de un cliente IPTV, el servidor SIP autoriza al usuario a acceder al servicio (este es el comportamiento por defecto de la implementación desarrollada) y contesta la solicitud con una respuesta OK de SIP. Tras recibir dicha respuesta, el cliente permite al usuario seleccionar un canal para comenzar a visualizar. Tras la selección, el equipo del cliente IPTV se da de alta en el grupo multicast correspondiente al canal, mediante un reporte de pertenencia a grupos de IGMP. En este momento, el cliente comenzará a recibir el vídeo correspondiente al canal solicitado, y a reproducirlo en la ventana de la aplicación. El envío de este vídeo en

modo multicast se realiza con la herramienta VLC, que implementa las funciones de servidor de streaming de vídeo en las pruebas presentadas.

En la [figura 7.2](#), se puede observar una captura de tráfico de un mensaje SIP, un INVITE, más concretamente para el inicio de sesión.

No.	Time	Source	Destination	Protocol	Length	Info
515	2.84793100	163.117.144.107	163.117.144.108	SIP/SDP	637	Request: INVITE sip:PruebaServidor@163.117.144.108:7060

Frame 515: 637 bytes on wire (5096 bits), 637 bytes captured (5096 bits) on interface 0
Ethernet II, Src: DellInc_80:b9:33 (f8:bc:12:80:b9:33), Dst: DellInc_81:6e:30 (f8:bc:12:81:6e:30)
Internet Protocol Version 4, Src: 163.117.144.107 (163.117.144.107), Dst: 163.117.144.108 (163.117.144.108)
User Datagram Protocol, Src Port: 7070 (7070), Dst Port: 7060 (7060)
Session Initiation Protocol (INVITE)
Request-Line: INVITE sip:PruebaServidor@163.117.144.108:7060 SIP/2.0
Method: INVITE
Request-URI: sip:PruebaServidor@163.117.144.108:7060
Request-URI User Part: PruebaServidor
Request-URI Host Part: 163.117.144.108
Request-URI Host Port: 7060
[Resent Packet: False]
Message Header
Call-ID: 8f89326546cf2b12407c6d1b01e602fa@163.117.144.107
CSeq: 1 INVITE
From: <sip:PruebaCliente@proyecto.com>;tag=12345
To: <sip:PruebaServidor@proyecto.com>
Via: SIP/2.0/UDP 163.117.144.107:7070;branch=z9hg4bk0e8020181589b5259fa07e9438c54102
Max-Forwards: 70
Contact: <sip:PruebaCliente@163.117.144.107:7070>
Content-Type: application/sdp
Content-Length: 166
Message Body
Session Description Protocol
Session Description Protocol Version (v): 0
Owner/Creator, Session Id (o): Prueba 1057165447541 1057165447541 IN IP4 163.117.144.107

Figura 7.2: Captura tráfico inicio sesión SIP

En la captura anterior, vemos un mensaje INVITE (este es un mensaje mostrado por wireshark de uno de los 10 UA probados en el laboratorio de telemática), mandado al servidor SIP. Como se puede comprobar, el cliente IPTV envía la solicitud al servidor SIP, cuya dirección IP es 163.117.144.108, tal y como se puede observar en la Request-Line (la primera línea del mensaje SIP). Por otro lado, se observa que el mensaje contiene una carga SDP de acuerdo a la cabecera Content-Type. De dicha carga se puede extraer que, en la línea "c", se especifica la dirección IP donde se va a producir la conexión de vídeo. En la línea "m", se especifica el puerto por dónde se va a recibir el vídeo y el protocolo de transporte utilizado en la transmisión.

7.3. Envío de contenido multimedia

Una vez establecida la sesión, se procede al envío del contenido multimedia. Son los clientes IPTV los que se encargaran de elegir el contenido multimedia que desean visionar. Como se comento al principio de este capítulo, cinco de los diez equipos en pruebas eligen el vídeo correspondiente a un canal IPTV, mientras que

los otros cinco eligen un canal alternativo. Así mismo, tres de los equipos, en mitad del visionado del contenido, deciden realizar un cambio de canal. En la [figura 7.3](#), se puede observar el cambio de canal establecido por uno de los tres equipos que realizaron el cambio de canal.

No.	Time	Source	Destination	Protocol	Length	Info
723	11.5216240	163.117.144.102	239.0.0.1	IGMPv2	62	Membership Report / Join group 239.0.0.1 for any sources
724	11.5216240	163.117.144.102	239.0.0.1	IGMPv2	60	Membership Report / Join group 239.0.0.1 for any sources
763	15.3057340	163.117.144.102	239.0.0.1	IGMPv2	62	Membership Report / Leave group 239.0.0.1
764	15.3057340	163.117.144.102	239.0.0.1	IGMPv2	60	Membership Report / Leave group 239.0.0.1
815	22.4257220	163.117.144.102	239.0.0.2	IGMPv2	62	Membership Report / Join group 239.0.0.2 for any sources
816	22.4257220	163.117.144.102	239.0.0.2	IGMPv2	60	Membership Report / Join group 239.0.0.2 for any sources
887	27.3245260	163.117.144.107	239.0.0.2	IGMPv2	62	Membership Report / Leave group 239.0.0.2
888	27.3245260	163.117.144.107	239.0.0.2	IGMPv2	60	Membership Report / Leave group 239.0.0.2
893	28.4145900	163.117.144.107	239.0.0.2	IGMPv2	62	Membership Report group 239.0.0.2
894	28.4145900	163.117.144.107	239.0.0.2	IGMPv2	60	Membership Report group 239.0.0.2
931	32.1465950	163.117.144.107	239.0.0.2	IGMPv2	62	Membership Report group 239.0.0.2

<

Frame 723: 62 bytes on wire (496 bits), 62 bytes captured (496 bits) on interface 3

Linux cooked capture

Figura 7.3: Captura tráfico cambio de canal.

En la [figura 7.4](#), se puede observar una fotografía real del escenario de pruebas. Se observa cuatro de los diez equipos sometidos al escenario de pruebas. En dicha imagen, se observa a dos equipos visionando el vídeo del primer canal IPTV, y a los otros dos equipos, el vídeo de segundo canal.



Figura 7.4: Imagen del entorno de pruebas (Lab. Telemática)

7.4. Liberación de sesión

Partiendo de la sesión multimedia anteriormente establecida, cuando un usuario decide abandonar la sesión, simplemente al cerrar la ventana de la interfaz gráfica que se muestra en el Cliente IPTV (mostrado en [figura 6.4](#)), se termina la sesión de

forma automática y transparente al usuario. Para ello, el cliente IPTV manda una solicitud BYE al servidor SIP para indicarle su deseo de abandonar la sesión. Una muestra de prueba de los mensajes entre cliente IPTV y servidor SIP se muestra a continuación:

Mensajes de liberación de sesión desde cliente IPTV a servidor SIP

```
----- BYE SENT:
BYE sip:163.117.144.108:7060 SIP/2.0
Via: SIP/2.0/UDP 163.117.144.107:7070;branch=z9hG4bKabb005bcacd32c223f3b6e4eb1280346
CSeq: 2 BYE
Call-ID: 8f89326546cf2b12407c6d1b01e602fa@163.117.144.107
From: <sip:PruebaCliente@proyecto.com>;tag=12345
To: <sip:PruebaServidor@proyecto.com>;tag=4321
Max-Forwards: 70
Content-Length: 0
```

El mensaje anterior, es el BYE que el cliente IPTV envía al servidor IPTV manifestando su interés en abandonar la sesión multimedia que tenían establecida.

Mensajes de liberación de sesión desde servidor SIP a cliente IPTV

```
Request BYE received at PruebaServer with server transaction id
gov.nist.javax.sip.stack.SIPServerTransaction@718dfe7
SERVER: got a bye sending OK.
```

Tras recibir el mensaje BYE enviado por el cliente IPTV en el servidor SIP, éste lo recibe, lo procesa y libera la sesión multimedia que tenían establecida. Asimismo, tras recibir el OK para la solicitud INVITE, el cliente IPTV se da de baja en el grupo multicast (esto no tiene por qué implicar necesariamente el envío de mensajes IGMP). Tras observar de forma detallada los mensajes, se deduce que la liberación de la sesión multimedia, se ha realizado de forma correcta. En la [figura 7.5](#) se observa una captura de tráfico de la liberación de la sesión SIP.

1104	55.0131530	163.117.144.107	163.117.144.108	SIP	377	Request: BYE sip:163.117.144.108:7060
1105	55.0154640	163.117.144.108	163.117.144.107	SIP	337	Status: 200 OK

<

- ⊞ Frame 1104: 377 bytes on wire (3016 bits), 377 bytes captured (3016 bits) on interface 3
- ⊞ Linux cooked capture
- ⊞ Internet Protocol Version 4, Src: 163.117.144.107 (163.117.144.107), Dst: 163.117.144.108 (163.117.144.108)
- ⊞ User Datagram Protocol, Src Port: 7070 (7070), Dst Port: 7060 (7060)
- ⊞ Session Initiation Protocol (BYE)
 - ⊞ Request-Line: BYE sip:163.117.144.108:7060 SIP/2.0
 - Method: BYE
 - ⊞ Request-URI: sip:163.117.144.108:7060
 - Request-URI Host Part: 163.117.144.108
 - Request-URI Host Port: 7060
 - [Resent Packet: True]
 - [Suspected resend of frame: 1103]
- ⊞ Message Header
 - ⊞ Via: SIP/2.0/UDP 163.117.144.107:7070;branch=z9hg4bkabb005bcacd32c223f3b6e4eb1280346
 - ⊞ CSeq: 2 BYE
 - Call-ID: 8f89326546cf2b12407c6d1b01e602fa@163.117.144.107
 - ⊞ From: <sip:PruebaCliente@proyecto.com>;tag=12345
 - ⊞ To: <sip:PruebaServidor@proyecto.com>;tag=4321
 - Max-Forwards: 70
 - Content-Length: 0

Figura 7.5: Captura tráfico liberación de sesión SIP

**CONCLUSIONES
Y
FUTURAS
IMPLEMENTACIONES**

8. Conclusiones y futuras implementaciones

8.1. Conclusiones

El objetivo de este Trabajo Fin de Grado ha sido la implementación de un servicio de televisión IP, para soportar el acceso a un conjunto de canales de televisión a un número potencialmente elevado de usuarios, utilizando como modelos de referencia el servicio de televisión IP definido por el grupo TISPAN de ETSI. El desarrollo se ha estructurado en dos planos o niveles: un *plano de control o de señalización*, donde se establece y se libera la sesión mediante el protocolo de señalización SIP, y un *plano de datos*, que soporta la entrega de la información de los canales del vídeo y las operaciones de cambio de canal mediante el uso de tecnologías multicast. La realización del sistema IPTV, implementado mediante el lenguaje de programación JAVA, se ha desarrollado con la ayuda de librerías ya existentes, como la API JAIN SIP o el framework VLCJ.

Con respecto a los objetivos secundarios marcados como punto de partida de este TFG, se puede concluir que dichos objetivos se han llevado a cabo con éxito. Además del desarrollo de un servicio IPTV, se ha llevado a cabo el estudio de las tecnologías y protocolos recogidos en los estándares actuales de televisión IP para redes de operador, así como la aplicación de dichas tecnologías en el desarrollo de un servicio IPTV, comprendiendo mejor su funcionamiento y la facilidad de su implementación. La implementación del servicio IPTV basado en las tecnologías SIP y multicast, ha resultado sencilla gracias al soporte existente por parte de la API JAIN SIP, así como por el framework VLCj.

8.2. Futuras implementaciones

Las futuras implementaciones que puede tener este TFG y no se han llevado a cabo, se comentan a continuación:

- Mejorar la interfaz gráfica del Cliente IPTV, ya que es una interfaz gráfica muy básica porque no era uno de los objetivos de este TFG. Dando a la interfaz gráfica un aire corporativo de la Universidad.
- Introducir mecanismos que permitan configurar automáticamente el número de canales disponibles en el cliente IPTV, de acuerdo con las especificaciones de TISPAN.
- Desarrollo de una interfaz gráfica para el servidor IPTV, de forma que sea más sencillo para el administrador del servidor IPTV y así pueda

retransmitir directamente el contenido multimedia sin necesidad de programas auxiliares como es el caso del VLC, utilizado en este TFG.

- Desarrollo de esta aplicación para que pueda ser ejecutable en un navegador WEB en distintas plataformas, de modo que se proporcione acceso a la aplicación desde cualquier dispositivo conectado a Internet con una aplicación de navegación Web. También, desarrollo de la aplicación para dispositivos móviles en iOS o Android.
- Ampliación del desarrollo para poder visionar clases en una mejor calidad de imagen con el protocolo HLS (Http Live Streaming), en el [anexo H](#), se encuentra la explicación de dicho protocolo.

ANEXOS

Anexo A: Gestión de proyecto y presupuesto

A.1. Descomposición en tareas

Para la correcta realización de este TFG, se ha dividido en tareas. En la [tabla A.1](#) se puede observar un resumen de todas las tareas realizadas.

De todos modos, a continuación se exponen cada una de las tareas que se han llevado a cabo.

- ❖ **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.
 - **Subtarea A.2:** Framework VLCJ
 - ◆ Relación con otras tareas: comienza después de la tarea A.1.
 - ◆ Duración: 1 semana.
- ❖ **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.
 - **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.
 - **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.
- ❖ **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.
 - **Subtarea C.2:** Estudio de los requisitos del servidor
 - ◆ Relación con otras tareas: comienza después de la tarea C.1.
 - ◆ Duración: 1 semana.
 - **Subtarea C.3:** Estudio de los requisitos del Servidor SIP.
 - ◆ Relación con otras tareas: comienza después de la tarea C.2.
 - ◆ Duración: 1 semana.
 - **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.
- ❖ **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
 - ◆ Duración: 1 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.
 - **Subtarea D.3:** Desarrollo de la interfaz del cliente
 - ◆ Relación con otras tareas: comienza después de las tareas D.1 y D.2.
 - ◆ Duración: 2 semanas
 - **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.
- ❖ **Tarea E: Pruebas**
- **Subtarea E.1:** Pruebas realizadas

- ◆ Relación con otras tareas: comienza después de la tarea D.
- ◆ Duración: 1 semana.

- **Subtarea E.2:** Comparación de resultados

- ◆ Relación con otras tareas: comienza después de la tarea E.1.
- ◆ Duración: 0.5 semanas.

- ❖ **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.

- **Subtarea F.2:** Redacción de la memoria

- ◆ Relación con otras tareas: comienza después de la tarea F.1.
- ◆ Duración: 6 semanas.

- **Subtarea F.3:** Redacción de los apartados en inglés.

- ◆ Relación con otras tareas: comienza después de la tarea F.2.
- ◆ Duración: 1 semana.

Tarea	Duración (semanas)
Estudio de los framework	
A.1. Interfaz JAIN SIP	1
A.2. Framework VLCj	1
TOTAL	2
Análisis del estado del arte	
B.1. Estudio de peticiones y respuestas SIP	1
B.2. Estudio de modelo oferta/respuesta SDP	1
B.3. Estudio de las opciones IPTV	1
TOTAL	3
Identificación de los requisitos del sistema	
C.1. Estudios requisitos del cliente	1
C.2. Estudios requisitos del Servidor	1
C.3 Estudios de los requisitos del SIP Server	1
C.4. Estudios de los requisitos del plano de datos	1
TOTAL	4
Implementación del sistema	
D.1 Desarrollo del establecimiento de sesión SIP	5
D.2 Desarrollo del plano de datos	5
D.3 Desarrollo del cliente	2
D.4. Desarrollo de la desconexión	1
TOTAL	13
Pruebas	
E.1. Pruebas	1
E.2. Comparativa de pruebas	0,5
TOTAL	1,5
Memoria	
F.1 Organización y estructura	1
F.2. Redacción de la memoria	6
F.3. Redacción de capítulos a inglés	1
TOTAL	8
TOTAL	31,5

Tabla A.1: Descomposición de Tareas

A.2. Planificación con el diagrama de fases

En la [figura A.1](#) se muestra el diagrama de Gantt con todas las tareas que se han realizado para llevar a cabo en este Trabajo Fin de Grado.

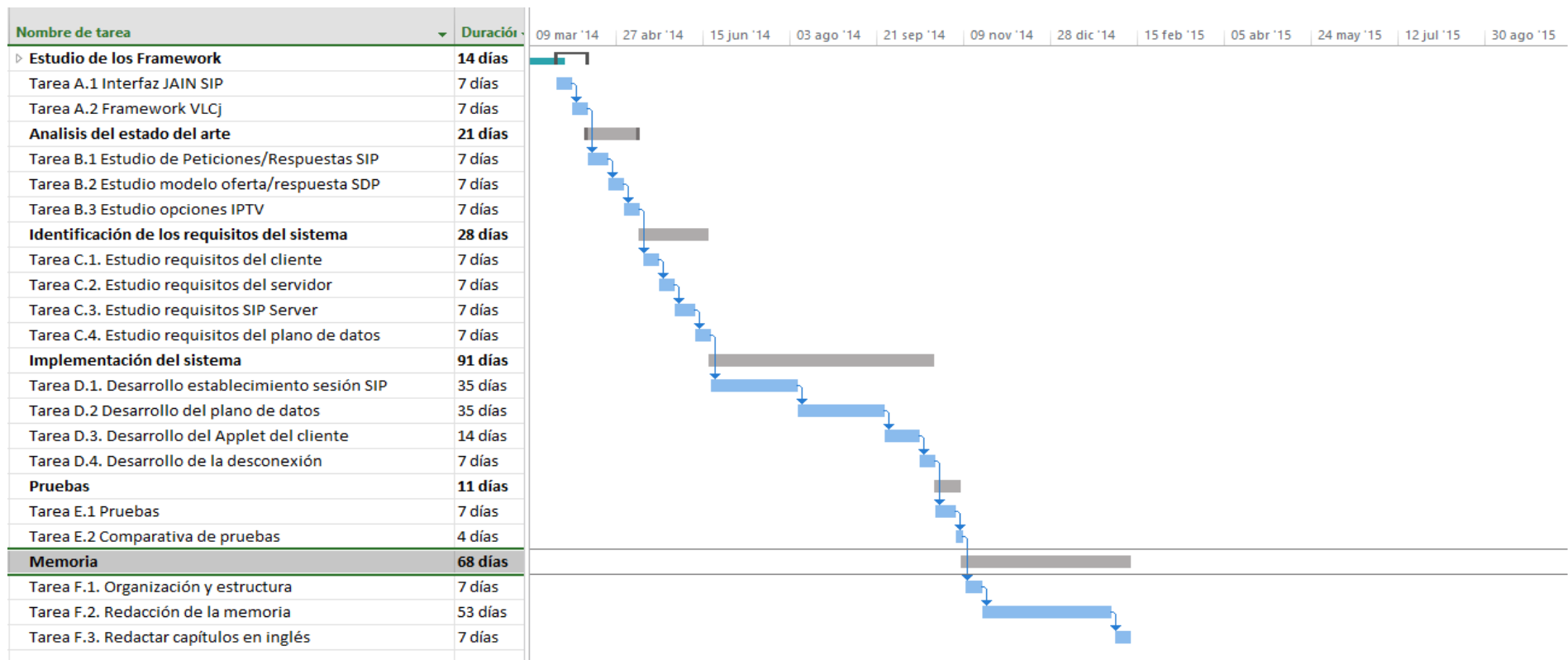


Figura A.1: Diagrama de Gantt

A.3. Presupuesto

A continuación se procede a detallar todos los recursos que se han necesitado para llevar a cabo este TFG.

A.3.1. Recursos Humanos

Para la realización de este TFG se han necesitado 2 Ingenieros:

- Estudiante de Ingeniería de Sistemas Audiovisuales.
- Profesor/Tutor del TFG.

A.3.2. Recursos Informáticos

Los recursos informáticos necesitados en la realización de este TFG se dividen en dos grupos: *recursos software* y *recursos hardware*.

A.3.2.1. Recursos Software

Los recursos Software utilizados para el desarrollo de este TFG se enumeran a continuación:

- Programa Eclipse
- Programa VLC
- Librerías específicas (JAIN SIP, VLCj)
- Microsoft Office
- Microsoft Project

En la [tabla A.2](#), se observa el precio de los Recursos Software utilizados:

Recursos Software	Precios
Eclipse	0 €
VLC	0 €
Librerías Específicas	0 €
Microsoft Office	119 €
Microsoft Project	1.369 €
TOTAL	1.488 €

Tabla A.2: Presupuesto Recursos Software

A.3.2.2. Recursos Hardware

Los recursos Hardware utilizados para el desarrollo de este TFG se enumeran a continuación:

- 10 PCs
- 1 router
- 1 Swich

En la [tabla A.3](#), se observa el precio de los Recursos Software utilizados:

Recursos Hardware	Precio Unidad	Precio Final
PC	581 €	5.812 €
Swich	25 €	25 €
Router	50 €	50 €
TOTAL		5.887 €

Tabla A.3: Presupuesto Recursos Hardware

A.3.1. Presupuesto Final

El presupuesto final, se detalla en la siguiente tabla ([tabla A.4](#)):

Recursos	Horas por día	Precio por día	Semanas	Precio	IVA (21%)	Precio Final
Estudiante	8h	50 €	31.5	11.025 €	2.315 €	13.340 €
Tutor PFG	4h	60 €	15.75	6.615 €	1.389 €	8.004 €
Recursos Software						1.488 €
Recursos Hardware						22.832 €
TOTAL						45.664 €

Tabla A.4: Presupuesto Final

Anexo B: Manual de usuario

Para ejecutar la aplicación, se puede hacer de dos formas distintas, bien desde un terminal de comandos o bien desde Eclipse. Si se decide por la segunda opción se deberá crear un proyecto dentro del programa que se tendrá en el equipo.

Para ambas ejecuciones se tiene una jerarquía de carpetas, con una carpeta principal y otras 3 dentro de ésta (src, bin y lib).

Requisitos:

- Se tendrá que tener instalado una versión de JAVA 1.6 o posterior.
- Se deberá tener 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*
 - *vlcj-1.2.2.jar*
- De forma adicional al VLC, pero obligatoria para el correcto funcionamiento de la aplicación, se necesitarán dos librerías más que vienen dentro del programa VLC:
 - *libvl.dll*
 - *libvcore.dll*

Pasos a seguir:

- Se compila la aplicación del servidor. Dependiendo de si se realiza por líneas de comando o mediante Eclipse, se realizará de forma diferente.
 - **Eclipse:** No será necesario realizar la compilación, ya que el propio programa lo hace de forma automática y transparente para el usuario. Genera los ficheros .class correspondientes.
 - **Terminal de comandos:** se debe de acceder a la carpeta src del proyecto. Una vez en ese directorio, se ejecutará la siguiente sentencia:

```
<ruta de la versión de JAVA >/javac -d ../bin/-classpath
../lib/concurrent.jar:../lib/jain-sip-ri-1.2.90.jar:../lib/jain-sipapi-
1.2.jar:../lib/log4j-1.2.8.jar:../lib/jna.jar:../lib/platform.jar:../lib/vlcj-
1.2.2.jar:. server.java UA.java
```

Para la ejecución del Cliente IPTV, se usará la siguiente línea:

```
<ruta de la versión de JAVA >/javac -d ../bin/-classpath  
../lib/concurrent.jar:../lib/jain-sip-ri-1.2.90.jar:../lib/jain-sipapi-  
1.2.jar:../lib/log4j-1.2.8.jar:../lib/jna.jar:../lib/platform.jar:../lib/vlcj-  
1.2.2.jar:. interfaceClient.java UA.java Client.java
```

- En un navegador o en su defecto eclipse se lanzará la interfaz gráfica del cliente. Primero se lanzará Servidor SIP (código desarrollado en JAVA, es la parte del Servidor IPTV encargada de hacer la señalización). Posteriormente se configurará el VLC (tal y como se indica en el [anexo G](#)).
- Posteriormente, se lanzará Cliente IPTV. Se hará de forma similar al [Servidor SIP](#). La interfaz mostrada al usuario se muestra en la [figura 6.4](#) de apartados anteriores.
- De forma automática el cliente se comunicará con el servidor y establecerán la señalización de la sesión.

Anexo C: Solicitudes SIP

CANCEL

Se utiliza para terminar búsquedas pendientes o intentos de establecimiento de llamadas. Puede ser generado por UAs o por proxies.

Este mensaje puede ser utilizado, si el servidor aún no ha emitido ninguna respuesta.

OPTIONS

Su utilización radica básicamente para conocer sobre cualquier UAC o UAS las capacidades que soportan y para saber su disponibilidad.

La respuesta a dicha 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.

SUBSCRIBE

Es usado por un UA para establecer una suscripción a un cierto servicio para recibir notificaciones (a través del método NOTIFY) sobre un evento en particular.

Dicha solicitud contiene la cabecera Expires cuya función es establecer 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 indica 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.

NOTIFY

Es utilizada por un UA para notificar la ocurrencia de eventos.

Dicho mensaje 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.

PRACK

Es usado para la confirmación de 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.

Si se tienen respuestas provisionales, como por ejemplo: 180 Ringing, es crítico determinar el estado de la llamada y confirmar la recepción de la respuesta provisional.

Esta solicitud la genera el UAC cuando se recibe una respuesta provisional que contiene las cabeceras RSeq (Reliable Sequence Number) y la cabecera Supported. 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.

UPDATE

Con este mensaje, se puede cambiar el estado de la sesión sin necesidad de modificar el diálogo. Puede ser utilizado antes de que se genere la respuesta a la solicitud INVITE y algunas de sus posibles utilidades son silenciar la llamada o poner la llamada en espera.

REFER

Este mensaje se utiliza para que un UA solicite 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 *ReferTo*.

MESSAGE

Con este mensaje, se realiza el transporte de 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.

En el caso de que el mensaje sea recibido de forma 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.

PUBLISH

Dicho mensaje 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.

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 INFO. Normalmente este mensaje contiene un cuerpo que almacena información de señalización y siempre incrementa el número de secuencia CSeq.

Anexo D: Campos SDP

Session name

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

Information

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

Email

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

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 y a continuación el número local.

URI

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

Bandwidth

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

b=modifier: bandwidth-value

El campo *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.

El parámetro *bandwidth-value* es el número de kilobytes por segundo.

Time

Este campo *t=* especifica los tiempos de comienzo y fin de la sesión. Los tiempos son especificados usando timestamps NTP (Network Time Protocol).

Su formato es el siguiente:

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

Repeat time

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

Time zones

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

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 (Cliente IPTV) véase la [figura 5.3](#).

Dicho INVITE contiene las siguientes cabeceras:

- La primera línea es una Request URI que contiene la URI SIP del Servidor IPTV, en concreto del Servidor SIP. Servidor SIP, es la parte del Servidor IPTV al que el Cliente IPTV debe conectarse.
- También se especifica cuál 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*.

En el momento que la petición INVITE llega al Servidor SIP , es el propio Servidor SIP el que se comunica con el Servidor Streaming para acceder al contenido multimedia que el usuario ha especificado en la cabecera *Contact* de este mensaje.

Una vez que el servidor (Servidor SIP) 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 Cliente IPTV.

Cuando el cliente recibe el 200 OK, responde con un mensaje ACK al Servidor SIP para notificar que ha recibido el 200 OK.

En el momento en que el servidor recibe el ACK empieza la transmisión directa, Servidor SIP se pone en comunicación con Servidor Streaming, se lanza el cliente y el usuario mediante la interfaz gráfica es el que indica que canal y que vídeo es el que quiere visionar, de las dos opciones que indica la interfaz gráfica mostrada en el Cliente IPTV.

Anexo F: IGMP

Previo al envío de los paquetes a la dirección IP multicast es necesario que los participantes de la sesión se hayan suscrito a ese grupo. Para llevar a cabo la suscripción de los participantes hacen uso del protocolo IGMP [Z.01] [IGMP]

Establece un mecanismo para el intercambio y actualización de información sobre la pertenencia de los equipos de un segmento a un grupo multicast. El dialogo se desarrolla entre los equipos miembros y los routers de su segmento y posibilita 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 sondan periódicamente el estado de dichas pertenencias.

Mensajes IGMP

Se transportan encapsulados dentro de los datagramas IP. Su estructura se muestra en la [figura F.1](#).

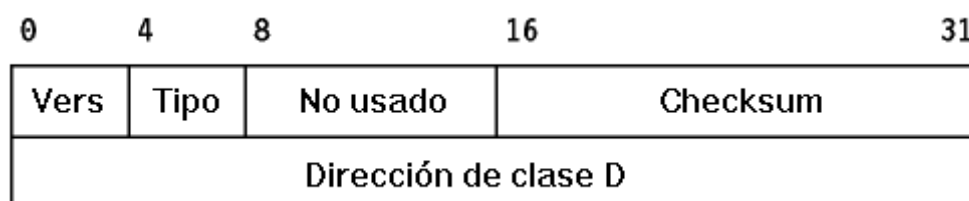


Figura F.1: Formato mensaje IGMP

El campo *tipo* mostrado en la figura anterior ([figura F.1](#)) establece los distintos tipos de mensaje reconocidos en IGMP. En la [tabla F.1](#) se observa los tipos de mensajes.

Tipo	Función
0x16: Informe de pertenencia de equipo (<i>membership report packet</i>)	Enviado por un equipo que se une al grupo multidifusión. También reenvía como respuesta a un mensaje de consulta del router <i>multicast</i> .
0x11: Consulta de pertenencia de equipos (<i>membership query packet</i>)	Enviado periódicamente por los routers para sondear la subred en busca de los miembros del grupo.
0x17: Dejar grupo (<i>leave Group packet</i>)	Enviado por un equipo cuando abandona el grupo multidifusión.

Tabla F.1: Tipos de mensaje IGMP

El tiempo máximo de respuesta indica el tiempo máximo que el router esperará un mensaje IGMP de tipo informe de pertenencia por parte de los equipos.

El campo *checksum* protege el mensaje, en cambio el campo dirección contiene la dirección multicast del grupo.

Los routers multicast mantiene una base de datos de grupos multicast para cada uno de sus interfaces, donde se recogen las direcciones multidifusión activas.

El router detecta ese mensaje y comprueba si ya existe dicho grupo asociado a la interfaz, y si no es así procede a registrar la nueva pertenencia. De forma periódica el router chequea la presencia de los miembros de los grupos registrados en cada interfaz y si deja de recibir respuestas eliminará la referencia a dicho grupo.

Anexo G: Manual de Configuración del VLC

El VLC es el llamado a lo largo de este documento Servidor Streaming, es una de las dos partes del Servidor IPTV tal y como se muestra en la [figura 5.1](#). Dicha aplicación soporta muchos formatos en los cuales se tiene que transmitir el vídeo, como por ejemplo .mp4, .mov o .mpg. Una vez que se tiene el vídeo con el requisito anterior, ya se puede proseguir a realizar los siguientes pasos dentro del VLC, para proceder al envío del vídeo.

Pasos a seguir:

- Una vez establecida la sesión SIP entre el servidor y el cliente, en el servidor se lanzará la aplicación VLC. En la [figura G.1](#) se observa la primera ventana que muestra el VLC.

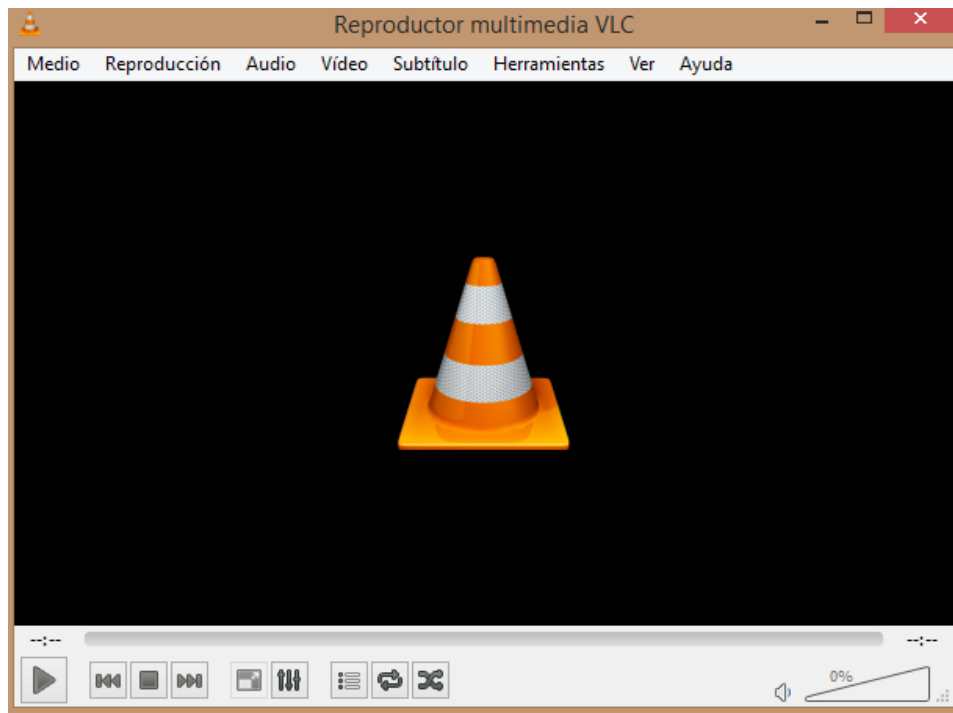


Figura G.1: Interfaz del VLC

- Una vez que se muestra la [figura G.1](#), se pulsa sobre el botón Medio de la barra de herramientas y en el menú desplegable, se selecciona la opción Emitir, tal y como se muestra en la [figura G.2](#) y [G.3](#).

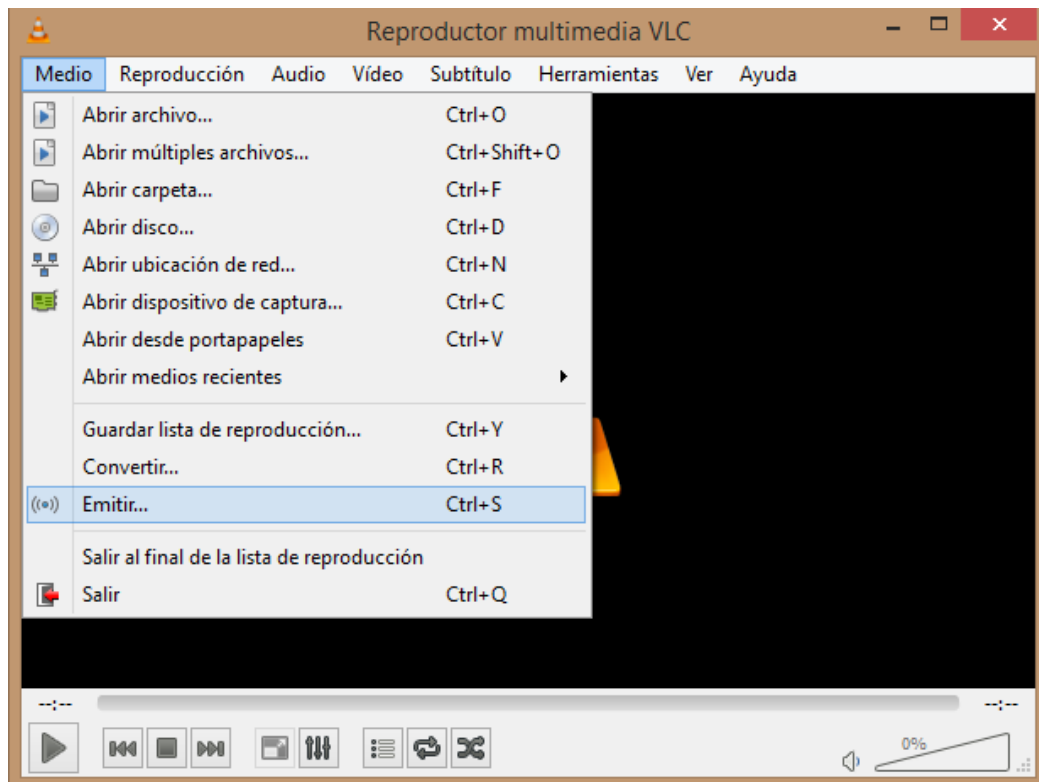


Figura G.2: Emitir

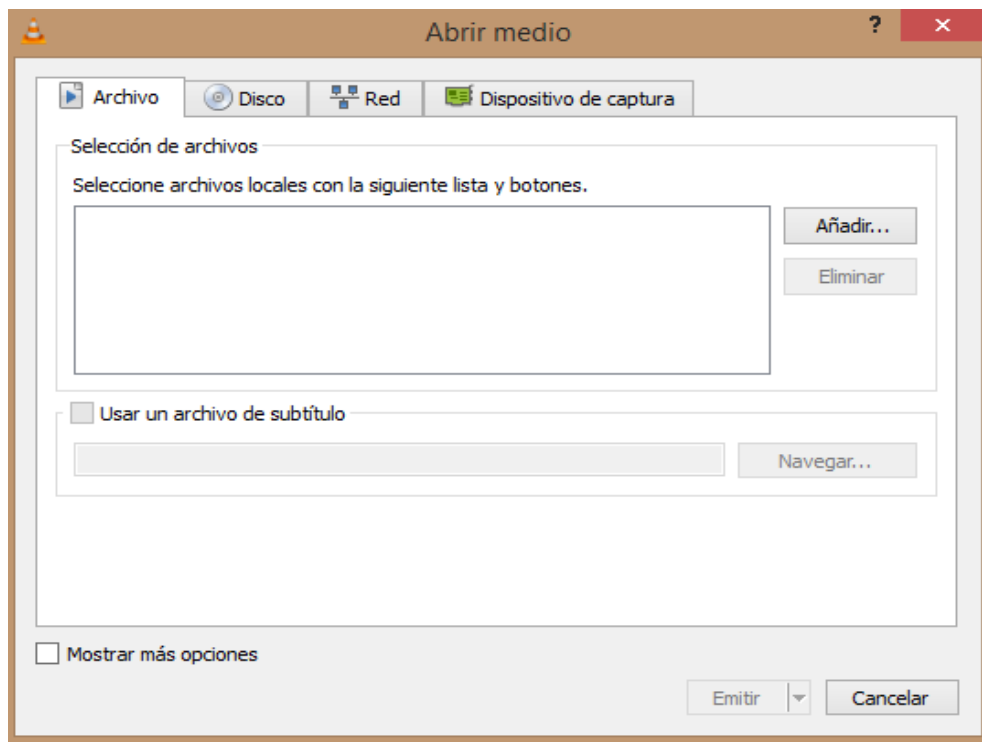


Figura G.3: Abrir medio

- Como se observa en la [figura G.3](#), debemos de pulsar el botón añadir en la pantalla que se muestra se debe de buscar el archivo de vídeo que vamos se quiere transmitir.
- Una vez que se haya seleccionado el archivo de video, se deberá pulsar el botón Abrir y mostrará el video cargado en la ventana del VLC, tal y como se muestra en la [figura G.4](#).

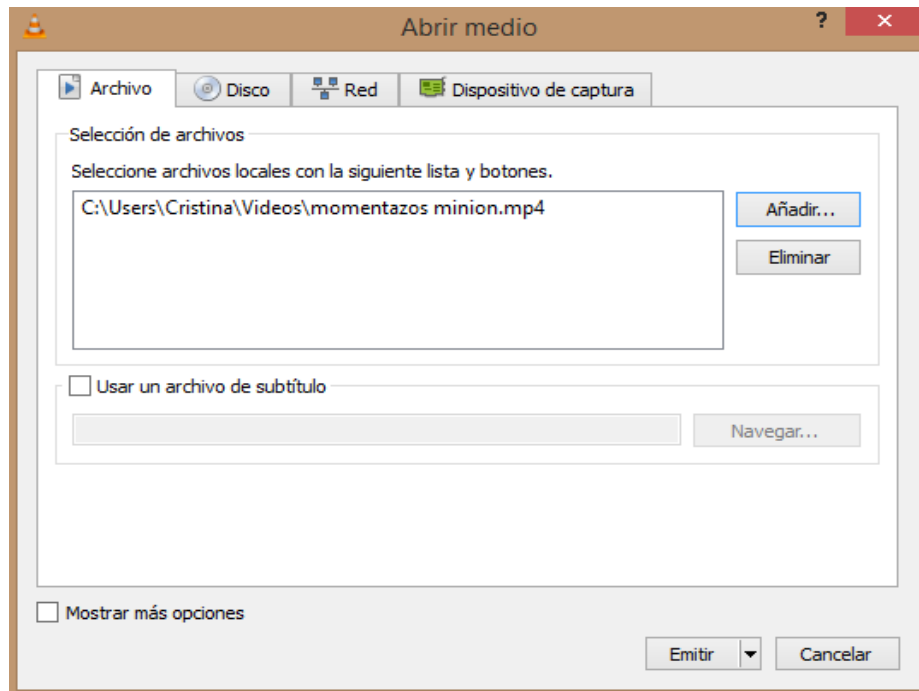


Figura G.4: Abrir medio con vídeo cargado

- Se debe de pulsar el botón de emitir y a su vez, en el desplegable que se muestra, se debe de volver a pulsar emitir. En la siguiente ventana que se muestra, es la ubicación del vídeo, por lo que se debe de pulsar next.
- En la [figura G.5](#) se observa cómo se escoge el protocolo. Para este caso se utilizará UDP, por lo que se selecciona dicha opción y se procede a añadir la dirección multicast y el puerto por el cual se va a emitir el vídeo*, [figura G.6](#).

* Para la emisión del segundo vídeo se debe de realizar los mismos pasos, la diferencia radica en que el canal multicast debe de ser diferente, pero el puerto a emitir puede ser el mismo (estos parámetros se incluyen en el carga SDP que lleva el INVITE).

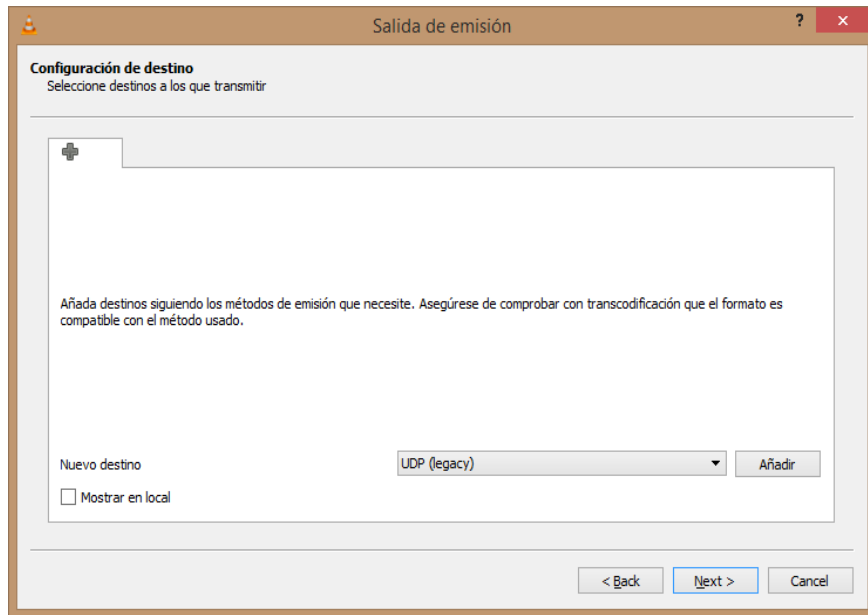


Figura G.5: Salida de Emisión.

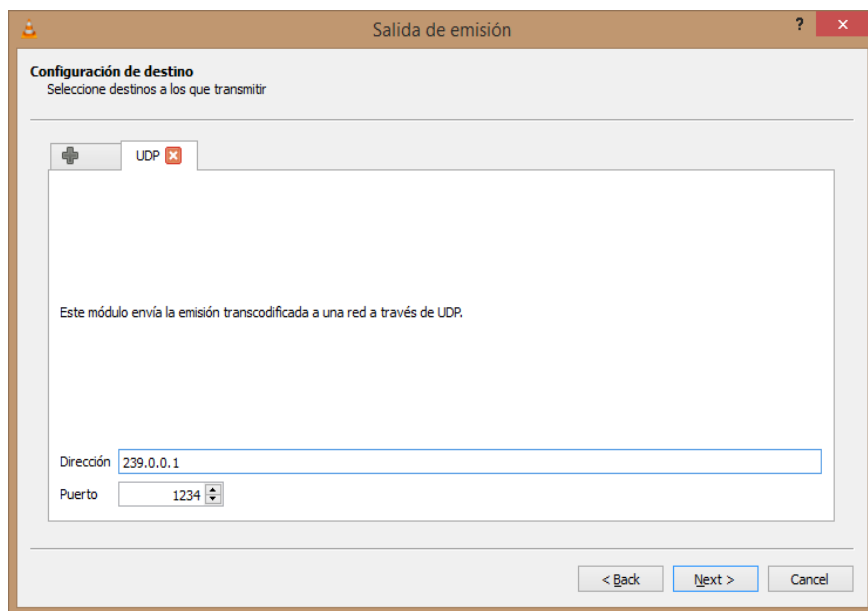


Figura G.6: Configuración dirección multicast y puerto

- A continuación se debe de seleccionar el formato que contiene el archivo de vídeo y se pulsa el botón de Next.
- Una vez que se han realizado todos los pasos anteriores, en el VLC se verá la pantalla mostrada en la [figura G.7](#), y una vez que el usuario elige el *Channel* a visualizar se verá tal y como se muestra en la [figura G.8](#).

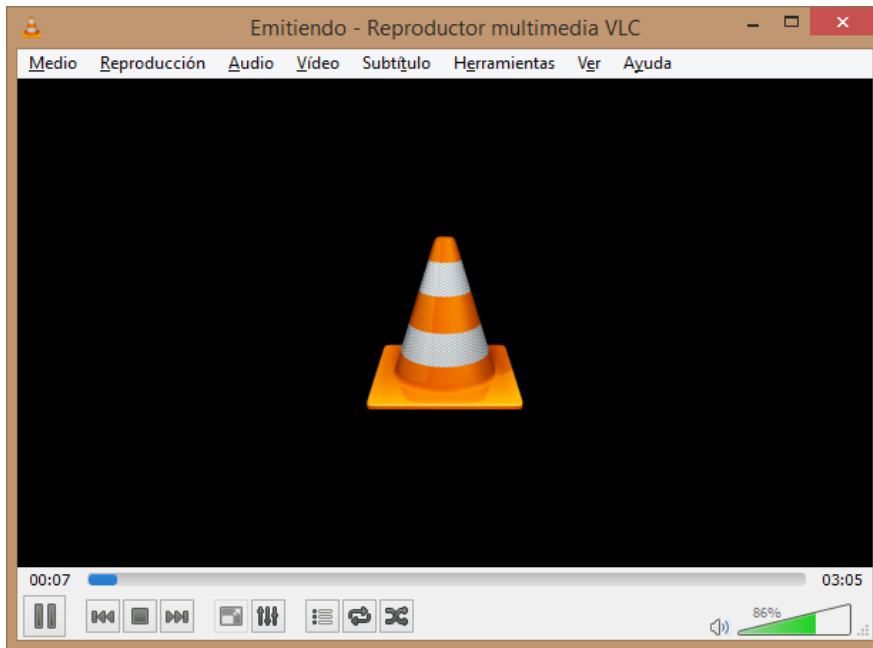


Figura G.7: Emisión del vídeo



Figura G.8: Visión del vídeo en el reproductor del Cliente.

Anexo H: HLS

HLS (HTTP Live Streaming) [\[HLS\]](#), es un protocolo de comunicación Streaming. Es un protocolo de redes no gestionado, es decir, redes públicas (internet). Este protocolo ha sido implementado por Apple INC.

Es un protocolo muy utilizado actualmente, ya que es un protocolo adaptativo, es decir, dependiendo del ancho de banda disponible, la calidad del vídeo enviado será mayor. En la [figura H.1](#) se puede observar su arquitectura básica.

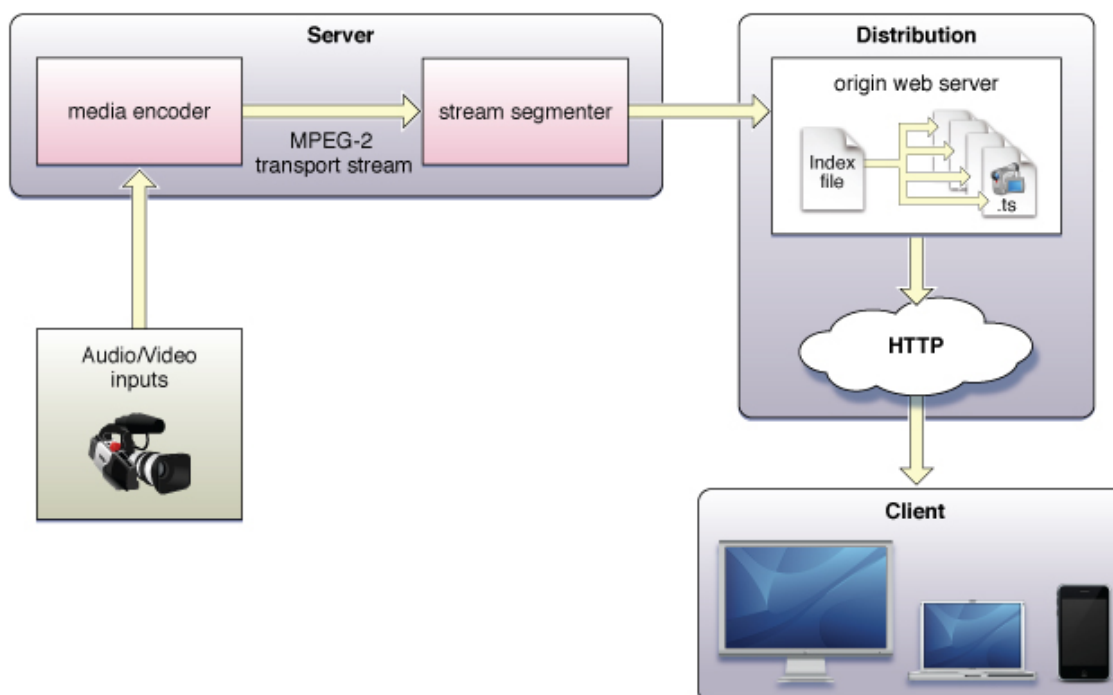


Figura H.1: Arquitectura HLS [\[HLS\]](#)

Utiliza vídeo H.264 MPEG-2 TS segmentado y archivos descriptores M3U8 para entregar vídeo con tasas de bits adaptativas en directo y a la carta.

Un archivo M3U8 es un índice que permite al cliente saber qué secuencias y segmentos están disponibles en un momento dado, en la [figura H.2](#) se puede observar este tipo de archivos. El dispositivo selecciona automáticamente la secuencia más adecuada desde el archivo de manifiesto primario teniendo en cuenta las limitaciones de ancho de banda y de CPU. HLS transmite datos a través de HTTP.

Ofrece diversas mejoras sobre los protocolos de transmisión tradicionales, como RTP o RTMP, entre las que se cuentan:

- Reducción de los costes de infraestructura.

- Posibilidad de almacenamiento en memoria caché en redes CDN y en otras infraestructuras de caché HTTP.
- Reducción de las amenazas procedentes de restricciones de proxy y cortafuegos.
- No hay problemas de “NAT”.
- Optimización en tiempo real mediante heurística en el equipo cliente (tasa de bits adaptativa).
- Implementación sencilla de reproductores HTML5.

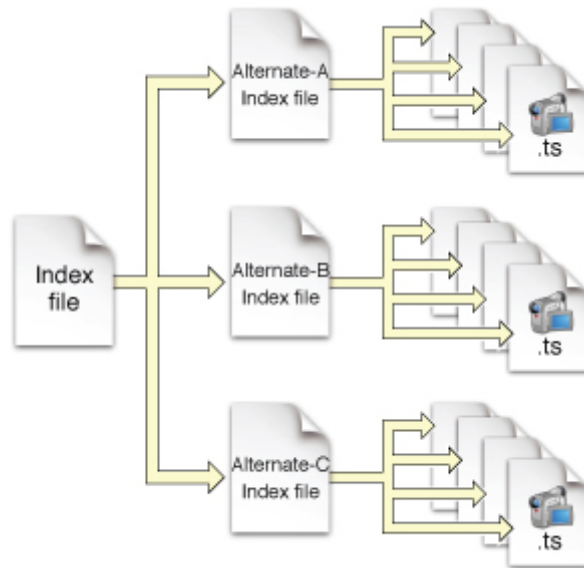


Figura H.2: División de Streaming en segmento (M3U8) [\[HLS\]](#)

Las fases por las que pasa un flujo para ser HLS son:

- **Codificación:** Se codifica la señal maestra. HLS requiere video progresivo y cuadros Intra de referencia.
- **Escalado:** Se decima el video para adaptarlo a las diferentes calidades.
- **Segmentación:** Se segmenta por GOP en las diferentes calidades cumpliéndose que el salto de una calidad a otra no debe conllevar falta de paquetes de video o audio, solo una merma o una mejora en la calidad de visualización.

El problema que tiene asociado este protocolo es el retardo en la visualización, ya que constantemente se está comprobando el estado del ancho de banda, para poder enviar los segmentos de la mayor calidad posible.

Anexo I: Introduction, Objectives, Result and Conclusions.

I.1. Introduction

This chapter shows the motivation and the objectives of the TFG and comments the structure of the document as well as a brief description of the different parts that it contains.

Nowadays everything turns around the multimedia world, as we are surrounded by many technological devices from PCs to mobiles. Most of the devices which we have, require the user's interaction which is done by GUI, (graphic interface), as well as an internet connection. Thanks to Internet, a wide door has opened for the distribution of information through the network. This way, the user can have access to a full range of applications and interesting services such as, access to information, communication via internet (video calls), video on demand (VoD), or digital Television.

Lately, and possibly with future anticipation, some mobile devices have entered the market due to users and business requests, as well as the display of the new broadband technologies.

The main objective of this TFG is to create an IPTV service based on the specifications of the working group TISPAN ETSI. This service must support access and distribution corresponding to a television set to a potentially large number of users' video. This requires developing an IPTV client to be able to watch different videos that the IPTV server is transmitting. It will develop an IPTV client and SIP (a sub-component of the IPTV server) server. The communication between both (client and server) will be done, firstly by the SIP protocol, and once the session is started, the multimedia contents will be sent via multicast from the server to the client also like to the user can make a change video display, the IGMP protocol is used.

IPTV technology (Internet Protocol Television) is starting to gain leadership in both the business and educational sphere. It is very important in the business scope, as it can provide private companies with great income, as users not only demand for conventional TV channels, but also complete seasons of series, and their own video club. These services are offered by big companies, here in Spain there are two, Telefónica with their product "Movistar Fusión" and ONO with "ONO TV".

I.2. Objectives

The main objective of this TFG is an IPTV service based on the specifications of the working group TISPAN ETSI. This service must support access and distribution corresponding to a television set to a potentially large number of users' video.

Delving into detail on this objective, we can highlight the following points to consider in the development of this system:

- The design is based on standard Internet protocols for access and delivery of multimedia content.
- The control plan is implemented using SIP (Session Initiation Protocol), defined by the IETF in [[JR02a](#)].
- In the data plane, the efficient transmission of video information to receivers of each channel is done by using multicast delivery technologies.
- The channel change operations are based on the group management protocol Internet (IGMP- Internet Group Management Protocol [[IGMP](#)] [[IGMP.2](#)]).
- The implementation is extensible to easily support the inclusion of new TV channels.

As secondary objectives of the TFG, the following are defined:

- Study of technologies and protocols contained in the current standards of IP television operator networks.
- Application of these technologies in the development of an IPTV service, the aim of better understanding their performance and ease of deployment from a practical standpoint.

All the goals listed above have been carried out. In the following sections, the success of the TFG is discussed.

I.3. Result

The result of this is positive TFG. Tests have been performed on two different scenarios in order to verify the correct operation of the implementation done.

The first stage has been in local mode, on the same PC IPTV server and client runs; on the other hand has been tried in an environment of local area network, in the laboratory of Telematics of the University Carlos III of Madrid, with ten teams. This network is switched and for that reason supports sending UDP datagrams.

Overcome with evidence satisfactory shape of the second stage, it is concluded that implementation has been carried out correctly. Of the ten teams undergoing tests,

five of them started in the video 1 and the other five in the video 2. Likewise in three of the ten teams, they were changing channel while the network traffic is captured.

All equipment properly established multimedia session and the proper display of the GUI, the two videos that were put on trial and the release of the multimedia session.

In the next picture you can see the end result of this TFG. It is seen that various videos are four, ten computers that were put into evidence. These tests were performed in the laboratory of Telematics. Showing only four of the ten simplicity and for better image detail.



Figure I.1: Actual picture of the test environment

I.4. Conclusions

The objective of this TFG is creating of an IPTV service for issuing and viewing multimedia content thereof. SIP has been developed in JAVA server. That server is responsible for establishing the multimedia session with the IPTV client developed. Similarly, we have developed an application in JAVA, IPTV client, to establish the correct viewing multimedia session thereof and.

It can be divided into two main parts: a control plane or signalling, which provides and the session is released by the protocol of SIP signalling and data plane technology-based multicast delivery, allowing the distribution of multimedia content a potentially large number of users. The realization of IPTV system implemented in JAVA was developed with the help of existing libraries such as JAIN SIP API or framework VLCj.

To verify the correct operation of IPTV system implemented, it has been proven in a test environment (performed in the laboratory of Telematics, Universidad Carlos III de Madrid), that the system works correctly in 10 teams.

For the data plane, sending the contents is made by datagrams UDP over IP. To do this, once the SIP session between the client and server is established, you own IPTV server that makes all processes transparent to the user.

It can be concluded that all major targets required in this TFG have been achieved, and is also simple implementation of an IPTV service based on SIP and multicast technologies, thanks to the existing support from the JAIN SIP API, as well as the framework VLCj.

Bibliografía

- [Adams] W. Siadak A. Adams, J. Nicholas. Protocol Independent Multicast - Dense Mode (PIM-DM): Protocol Specification (Revised). RFC 3973, January 2005.
- [Allan.B.09] Allan B. RFC 3261. SIP Understanding the Session Initiation Protocol.
- [A97] A. Ballardie. Core Based Trees (CBT version 2) Multicast Routing Protocol Specification. RFC 2189, September 1997.
- [CISCO] Cisco Visual Networking Index: Forecast and Methodology, 2013–2018.
- [Comp] <http://es.masternewmedia.org/iptv-vs-television-via-internet/diferencias-clave.htm>
- [D11] D. Kanellopoulos E. Mikóczy, I. Vidal. IPTV Evolution Towards NGN and Hybrid Scenarios. Informática, 2011.
- [ETS11] 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.
- [G.007] G. O’Driscoll. IPTV Network Distribution Technologies. June 2007.
- [HLS] <https://developer.apple.com/streaming/>
- [H. 03] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson. RTP: A Transport Protocol for Real-Time Applications. RFC 3550, July 2003.
- [H06] H. Holbrook I. Kouvelas B. Fenner, M. Handley. Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification. RFC 4601, August 2006.

- [IET] The internet engineering task force. <http://www.ietf.org>
- [IMS.11] IPTV Evolution Towards NGN and Hybrid Scenarios, Nov. 2011
- [IGMP] W. Fenner. Internet Group Management Protocol, Version 2. RFC 2236, November 1997.
- [IGMP.2] Z. Albanna, K. Almeroth, D. Meyer, M. Schipper. IANA Guidelines for IPv4 Multicast Address Assignments. RFC 3171, August 2001.
- [JM94] J. Moy. Multicast Extensions to OSPF. RFC 1584, March 1994.
- [J80] J. Postal. User Datagram Protocol. RFC 768, Ago. 1980.
- [JAIN1] <https://github.com/lagigliaivan/rvideo/blob/master/src/server/SipManager.java>
- [JAIN2] Emmanuel Proulx. An Introduction to the JAIN SIP API. <http://www.oracle.com/technetwork/articles/entarch/introduction-jain-sip090386.html>
- [MR10] http://www.agpd.es/portalwebAGPD/canaldocumentacion/publicaciones/common/Guias/GUIA_SEGURIDAD_2010.pdf
- [RFC 3261] G. Camarillo J. Peterson R. Sparks M. Handley Schooler J. Rosenberg, H. Schulzrinne. SIP: Session Initiation Protocol. RFC 3261, June 2002.
- [RTP] S. Casner, R. Frederick, V. Jacobson, H. Schulzrinne, "RTP: A Transport Protocol for Real-Time Applications," RFC 1889, Internet Engineering Task Force, Enero 1996.
- [SDP] C. Perkins M. Handley, V. Jacobson. SDP: Session Description Protocol. RFC 4566.
- [SDP.2] H. Schulzrinne J. Rosenberg. An Answer Model with Session Description Protocol (SDP). RFC 3264, June 2002.

- [TIS] Telecommunications and Internet converged Services and Protocols for Advanced Networking. <http://www.etsi.org/tispan/>, último acceso febrero 2015.
- [TIS.027] ETSI TS 182 027 V3.5.1 (2011-03) Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); IPTV Architecture; IPTV functions supported by the IMS subsystem; March 2011.
- [TIS.063] ETSI TS 187 003 V3.4.1 (2011-03) Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); NGN Security; Security Architecture; March 2011
- [VLC] 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>
- [W09] Wes Simpson; IPTV and Internet Video: expanding the reach of television broadcasting. ISBN-13: 978-0240812458, Focal Press, 2009.
- [WIRE] Wireshark. <https://www.wireshark.org/>, 10 Febrero 2015