

Plataforma Genérica para la Provisión de Servicios en Redes con Plano de Control IMS

José Luis Cantarero*, Iván Vidal*, Jaime García*, Francisco Valera*, Arturo Azcorra*†

*Departamento de Ingeniería Telemática

Universidad Carlos III de Madrid

Avda. de la Universidad 30, 28911, Leganés - Madrid, España

{jlcantarero, ivaldal, jgr, fvalera, azcorra}@it.uc3m.es

†IMDEA Networks

Avda. Mediterraneo 22, 28918, Leganés - Madrid, España

Resumen— Existe una tendencia hacia la convergencia entre los distintos dominios de redes de comunicación denominada Convergencia Fijo-Móvil, que promulga la consolidación de una red única en la cual sea posible la provisión de servicios a los que hasta ahora se ha accedido mediante distintos mecanismos de control. El presente artículo se centra en el diseño y desarrollo de una plataforma genérica para la provisión de servicios en redes con plano de control IMS (Subsistema Multimedia IP). Esta consistirá en un Sistema Cliente, capaz de realizar cualquier descripción de sesión e iniciar el establecimiento de dicha sesión y en un Sistema Servidor que proporcionará una interfaz genérica sobre la que será posible implementar servicios multimedia de valor añadido. Adicionalmente, se validará la solución presentada, mediante la implementación de un servidor de Video bajo Demanda.

Palabras clave—IMS, SIP, servicios de valor añadido, provisión de servicios.

I. INTRODUCCIÓN

EN la actualidad, podemos encontrarnos con dos grandes redes de comunicación, ya sea en cuanto a número de usuarios o la cobertura que alcanzan. Sin embargo, dichas redes pertenecen a dominios distintos: la red de telefonía dentro del dominio móvil e Internet en el dominio fijo. Mientras que la primera proporciona una cobertura prácticamente global a través de terminales cada vez más pequeños, Internet ha sido uno de los mayores avances en la sociedad de la información, y ha permitido el despliegue de una gran variedad de servicios (*World Wide Web*, *e-mail*, mensajería instantánea, etc.) gracias al uso de protocolos abiertos.

Sin embargo está surgiendo lo que parece ser la evolución lógica de estos dos grandes dominios de redes de comunicación, la tendencia hacia lo que se ha venido a denominar Convergencia Fijo-Móvil (*Fixed-Mobile Convergence*) [1] que se hace patente tanto en los diversos intentos por acercar servicios tradicionales de una red a la otra, como puede ser el despliegue de la llamada Web móvil.

Este cambio de escenario plantea un reto tecnológico claro: aunar las características de ambos dominios permitiendo al usuario que se aprovisione tanto de los servicios tradicionales de voz, como de servicios multimedia y de nueva generación o de los servicios tradicionalmente ofrecidos en las redes de paquetes (por ejemplo, los basados en la web). De este

planteamiento emergen los esfuerzos de estandarización de la arquitectura del Subsistema Multimedia IP (*IP Multimedia Subsystem*, IMS), para que dicha tecnología sea la arquitectura de referencia para las Redes de Próxima Generación (*Next Generation Networks*). [2]

La arquitectura IMS está pensada para ser capaz de ofrecer acceso al usuario final a un rango completo y heterogéneo de servicios de valor añadido - tanto servicios de voz clásicos como servicios Web, e incluso servicios creados a partir de la unión de otros servicios (concepto que enlaza directamente con la creación de *Mash-ups* y la evolución de la Web 2.0 en general) [3] - a partir de la definición genérica y estandarizada del servicio o servicios que se quieran proporcionar.

Además, esta arquitectura logra de una manera sencilla poder realizar las funcionalidades de control de sesión (establecimiento, mantenimiento y liberación) mediante la utilización del protocolo SIP (*Session Initiation Protocol*), [4] de manera que la red consigue entregar los servicios mencionados arriba de una manera muy sencilla; que es uno de los grandes problemas que se encuentran a la hora de la convergencia entre distintas redes [5].

En este contexto, y teniendo en mente el objetivo de analizar la provisión de servicios dentro de IMS, es donde se ha querido enmarcar el estudio del presente artículo. Resulta interesante desarrollar una implementación que pueda ser utilizada para el estudio y análisis de esta tecnología en cuanto al aprovisionamiento de servicios basados en el establecimiento de sesiones multimedia.

Para ello se propone la implementación de una plataforma genérica de provisión de servicios en IMS, consistente en un sistema cliente genérico, capaz de iniciar el establecimiento y terminación de sesiones multimedia a través del plano de control de IMS, y en un sistema servidor básico, capaz de gestionar el establecimiento y terminación de dichas sesiones, y sobre el que será posible construir un Servidor de Aplicación (*Application Server*) que proporcione servicios específicos.

Adicionalmente, para validar el diseño y la implementación de la plataforma se propone el desarrollo de un servicio de video bajo demanda que funcione sobre el sistema servidor de dicha plataforma.

El presente artículo se estructura de la siguiente manera; en

el apartado II se realizará un breve estudio del trabajo ya realizado dentro de este marco. Posteriormente en el apartado III se analizará detalladamente el diseño tanto del sistema cliente como del sistema servidor que conforman la plataforma. En el apartado IV se recoge el diseño del servidor de video bajo demanda implementado para validar la funcionalidad de la plataforma así como los distintos escenarios de prueba realizados con el sistema completo. Finalmente se termina el artículo en el apartado V con las conclusiones extraídas del diseño y las líneas de trabajo futuro.

II. SOLUCIONES PREVIAS RELACIONADAS

En este apartado se muestra una recopilación de los prototipos e implementaciones existentes dentro del marco de IMS para ser tomados como referencia tanto a la hora de la implementación de nuestra solución como para tomar uso de ellas a la hora de realizar los escenarios de prueba para la plataforma.

A. Servidores SIP

Se han encontrado dos implementaciones de los servidores SIP que son interesantes para este estudio. La primera es *SIP Express Router* [6], una implementación con licencia GNU GPL centrada en la escalabilidad y la seguridad de la compañía iptel (aunque fue desarrollado en un principio por el instituto Fraunhofer FOKUS), y la segunda es *openSER* [7], una implementación comenzada por varios de los colaboradores de la implementación anterior (que permanecieron en el instituto Fraunhofer FOKUS) que se centra en la seguridad y la estabilidad y que también está sujeta a licencia GNU GPL.

Pese a que ambas implementaciones han seguido distintos métodos de desarrollo, para el marco de nuestro análisis son equivalentes puesto que ambas implementaciones cumplen los requisitos que queremos para el *core* de IMS, y es que implementen las tres configuraciones de los servidores SIP: registrar, proxy y servidor de redirección con una gran capacidad de computación de llamadas por segundo.

B. Open IMS Core

Esta implementación del instituto Fraunhofer FOKUS se inscribe dentro del proyecto “*Open IMS Playground @ FOKUS*”, un marco de desarrollo de aplicaciones para la tecnología IMS en el que participan tanto el instituto como distintas compañías para probar funcionalidades IMS con financiación del Ministerio Alemán de Educación e Investigación. Dentro de este marco de investigación se pueden encontrar diferentes tecnologías de acceso además de componentes de la infraestructura IMS y herramientas de gestión de la infraestructura. FOKUS implementa todos los componentes del *core* de IMS y enriquece esta implementación mediante componentes de implementaciones comerciales, principalmente plataformas de servicios que implementan algunos de los diferentes servidores de aplicaciones [8].

Dentro del marco de pruebas, se han utilizado tanto servidores multimedia y *streaming* basados en código libre e

implementaciones propietarias, así como servidores de aplicación basados en diversas tecnologías (*SIP servlets*, CPL, OSA/Parlay, Parlay X) por lo que presumiblemente esta implementación es capaz de funcionar dentro de diversos entornos.

C. CAMPARI

El proyecto CAMPARI (*Configuration, Architecture, Migration, Performance Analysis and Requirements of 3G IMS*) se inscribe dentro del grupo de proyectos del centro de investigación de telecomunicaciones de Viena de estudio de la tecnología IMS. Los resultados de este proyecto son complementarios a los del proyecto SIMS (*Services in IMS*) y el proyecto CAIPIRINA (*Converging towards All-IP: IMS Realization Issues for NGN Applications*).

El proyecto trata de, a partir de implementaciones de código libre, desarrollar una configuración mínima-óptima de la arquitectura de IMS con respecto a la calidad de servicio [9].

Con respecto a las implementaciones utilizadas, los CSCF (*Call Session Control Function*) se basan en servidores SIP *openSER* [7] implementados para cumplir con los requerimientos de IMS según las especificaciones [10], [11] y [12]. También se utilizan un servidor DNS, así como un *Home Subscriber Server* reducido y una implementación del protocolo DIAMETER.

D. Clientes IMS

Open IMS client es un nuevo cliente IMS desarrollado por el propio Instituto Fraunhofer para que sea plenamente funcional con *Open IMS Core*. Propone un entorno configurable donde desarrollar diversas aplicaciones IMS, alineado con las especificaciones 3GPP, IETF y TISPAN. Se divide en el interfaz configurable, donde el usuario tiene acceso a los servicios construidos, la capa de servicios donde se esconde la complejidad del cliente para diversos servicios IMS (VoIP, Presencia, Registro, Notificación de Eventos, Mensajería...) que puede ser extensible y finalmente la capa del motor IMS, donde se encuentran las pilas de protocolos a utilizar.

Por otro lado, el *IMS Communicator* es un proyecto de software libre de Portugal Telecom Inovação, basado en el *softphone SIP Communicator*, que tiene como objetivo probar nuevos servicios y escenarios de convergencia dentro del marco de las especificaciones del 3GPP y el IETF para las redes IMS, aunque se centra específicamente en servicios multimedia peer-to-peer (audio y video llamadas). Hace uso de la pila JAIN SIP para el plano de control y de la pila JMF (*Java Media Framework*) para el flujo multimedia.

III. PLATAFORMA GENERICA DE PROVISION DE SERVICIOS

Una vez establecido el marco tecnológico donde se inscribe este artículo además de las diversas implementaciones que se han realizado en dicho marco, en este apartado se va a tratar de detallar las decisiones de diseño que han sido implementadas, así como las

funcionalidades del sistema para conseguir la plataforma genérica.

A continuación en la Fig. 1 se muestra un esquema con la arquitectura tanto del sistema cliente como del sistema servidor, indicando los diversos módulos que posteriormente se explicarán.

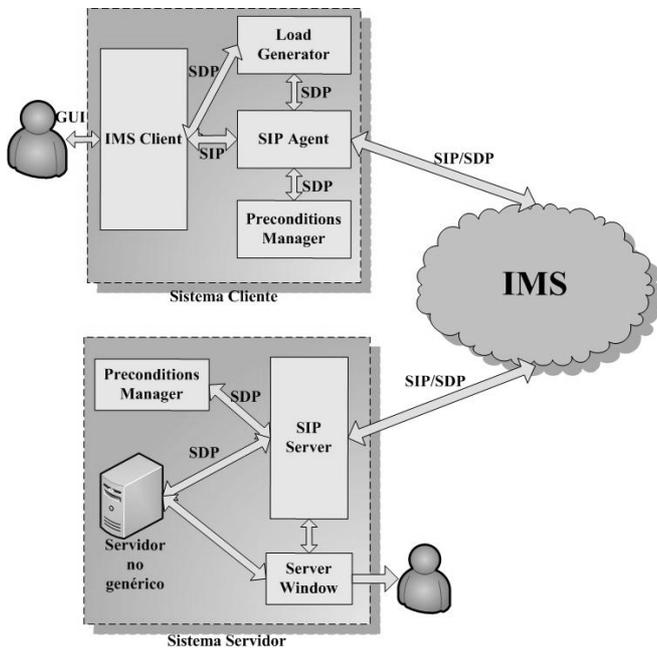


Fig. 1. Sistema cliente y sistema servidor completos

A. Sistema Cliente

El sistema cliente es principalmente una aplicación gráfica que implementa la funcionalidad de los protocolos SIP y SDP (además de otros mecanismos propios del establecimiento de sesión SIP tales como el modelo oferta/respuesta de SDP). Los módulos que conforman este cliente son los siguientes:

IMSClient: contiene toda la interfaz gráfica que se usa

LoadGenerator: es la encargada de generar todas las descripciones multimedia y de sesión que manda el cliente

SIPAgent: implementa el procesamiento de peticiones/respuestas de SIP

PreconditionsManager: se encarga de gestionar los parámetros locales y remotos de los atributos de SDP relacionados con la QoS y las precondiciones [13]

La interfaz gráfica que utiliza el usuario en el sistema cliente se ha dividido en varias pestañas dependiendo de los parámetros que se necesiten especificar. En primer lugar, la pestaña principal (*UAC Parameters*) es la que permite indicar los parámetros SIP necesarios para que el sistema cliente pueda iniciar el establecimiento de sesión, mientras que la segunda pestaña (*SDP Payload*) permite especificar la oferta SDP que se incluirá en el mensaje INVITE como parte del establecimiento de la misma. En la Fig. 2 se muestran dichas pestañas.

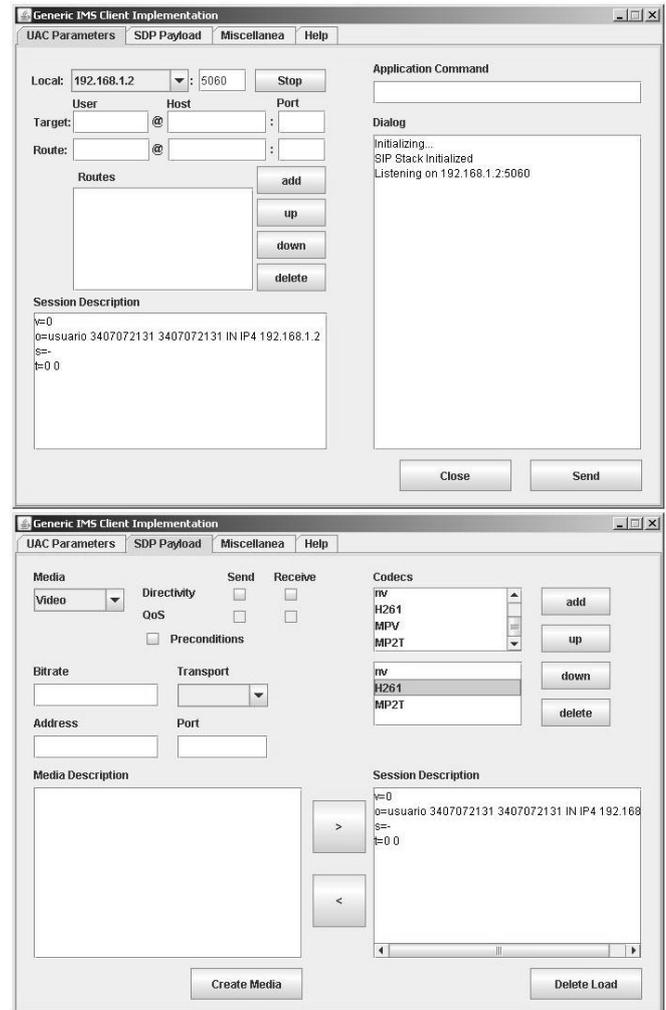


Fig. 2. Pestañas *UAC Parameters* (arriba) y *SDP Payload* (abajo)

La tercera pestaña (*Miscellanea*) contiene parámetros de configuración propios del establecimiento de sesión de IMS pero que no forman parte de ninguno de los protocolos anteriores. Finalmente, la última pestaña (*Help*) contiene un breve documento de ayuda sobre la utilización del cliente. Las dos primeras pestañas son las que ofrecen un mayor grado de interacción para el usuario, y por tanto son las más importantes, pues permiten la configuración de gran cantidad de parámetros para la descripción y el establecimiento de la sesión, mostrándose todos ellos en la Tabla 1 junto con el campo que hay que rellenar en la interfaz gráfica.

TABLA I
CABECERAS SIP Y ATRIBUTOS SDP ESPECIFICADOS EN LAS PESTAÑAS DE LA INTERFAZ GRÁFICA DEL SISTEMA CLIENTE

Parámetro	Protocolo	Campo	Obligatorio
From	SIP	Local	Si
To	SIP	Target	Si
Request-URI	SIP	Target	Si
Via	SIP	Local	Si
Route	SIP	Route	No

TABLA I (CONTINUACIÓN)

Contact	SIP	Local	No
Media	SDP	Media	Si
Media	SDP	Port	Si
Media	SDP	Transport	Si
Media	SDP	Codecs	En audio/video
Connection	SDP	Address	Si
Bitrate	SDP	Bitrate	No
Atributo rtpformat	SDP	Codecs	En audio/video
Atributo de directividad	SDP	Directivity: Send/Receive	Si
Atributo de QoS local deseada	SDP	Qos: Send/Receive	No

Una vez especificados los parámetros correspondientes, la generación de la carga SDP, que realizará el módulo *LoadGenerator*, consta principalmente de dos procesos: la creación de la descripción de sesión y la creación de cada una de las descripciones multimedia que puede contener la descripción de sesión SDP. Para permitir una mayor generalidad a la hora de describir la sesión que se quiere establecer, dichos procesos son completamente independiente uno del otro.

El primero de ellos se realiza cada vez que se quiera iniciar una sesión (cuando se inicia el cliente y cada vez que termina una sesión) de manera automática, siguiendo el proceso que se muestra en la figura siguiente:

Las descripciones multimedia se generarán y añadirán a la carga SDP cada vez que se pulse el botón ">" de la pestaña *SDP Payload*.

De manera análoga a la generación de la carga SDP, al pulsar el botón *Send* de la pestaña *UAC Parameters* el módulo *SIPAgent* generará la solicitud SIP (con el método INVITE) que dará comienzo al establecimiento de sesión. La creación de esta solicitud consta de tres procesos principales: la toma de los datos introducidos por el usuario, la generación de la solicitud INVITE y el comienzo del establecimiento de la sesión al mandar dicha solicitud.

Para el correcto establecimiento de sesión se han implementado los métodos que se encuentran disponibles en el interfaz *SipListener* de JAIN SIP. Mediante la herencia e implementación de éstos, el sistema cliente es capaz de manejar cualquier respuesta y eventualmente cualquier solicitud que pueda recibir (como un BYE generado directamente por el servidor). Además el interfaz *SIPListener* da la posibilidad de utilizar la gestión de transacciones a la hora de mandar solicitudes, lo que permite guardar el estado de los diálogos que se generen y que el establecimiento de la sesión se pueda realizar de una manera muy sencilla.

En la Fig. 3 se puede observar como se procesan las distintas solicitudes que puede recibir el cliente a lo largo del

establecimiento de sesión.

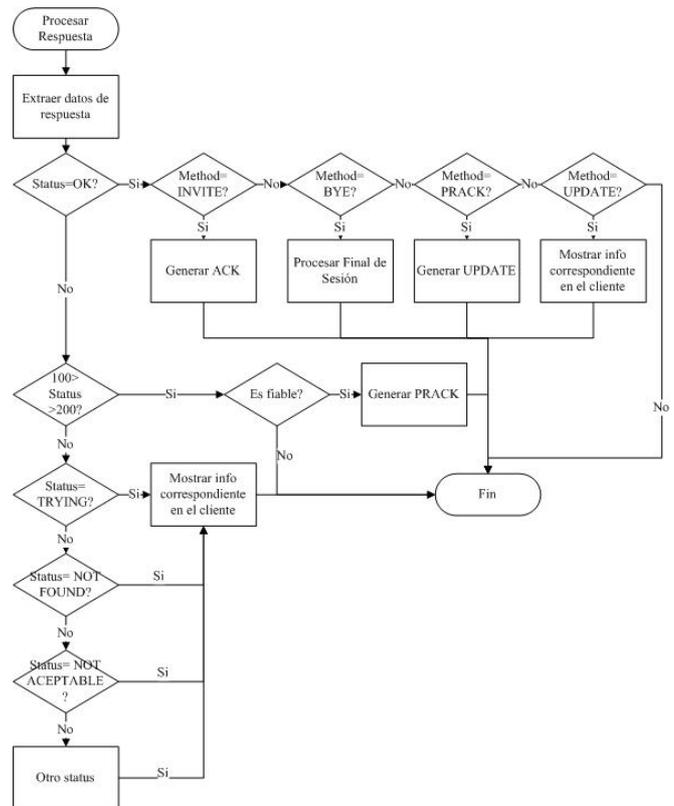


Fig. 3. Procesamiento de respuestas en el sistema cliente

B. Sistema Servidor

El sistema servidor da servicio a las peticiones que recibe de una manera genérica exponiendo un interfaz fácilmente implementable. De esta manera el servidor genérico se puede utilizar como servidor de aplicación de cualquier servicio dentro de la arquitectura IMS.

El estudio de su implementación lo vamos a dividir según sus partes constituyentes: interfaz gráfica, procesamiento de peticiones (y de respuestas) y comunicación con la parte no genérica.

Los módulos que conforman este servidor genérico son los siguientes:

SIPServer: contiene todo el manejo de solicitudes SIP correspondiente al servidor genérico

ServerWindow: contiene la interfaz gráfica de ayuda al usuario

GenericServer: interfaz con los métodos a implementar para crear el servidor de aplicación deseado

PreconditionsManager: se encarga de gestionar los parámetros locales y remotos de los atributos de precondiciones

Al contrario de lo que ocurría en el sistema cliente, aquí las únicas funcionalidades de la interfaz gráfica son escoger donde debe escuchar el servidor y mostrar mensajes informativos sobre las peticiones que se generan y el estado del servidor.

Al igual que en el sistema cliente, el módulo *GenericServer* del sistema servidor hace uso de los métodos que se encuentran

disponibles en el interfaz *SipListener* de JAIN SIP. El proceso que se sigue en este módulo para el establecimiento de la sesión es el que se muestra en la Fig. 4.

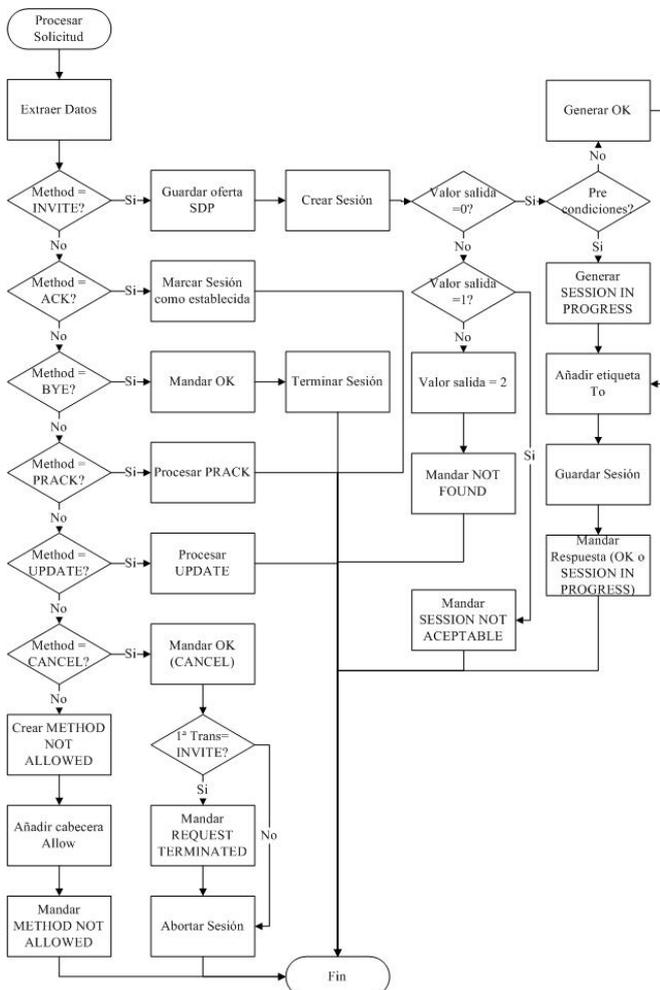


Fig. 4. Procesamiento de solicitudes en el sistema servidor

Como ya hemos dicho, la solución facilita la creación y utilización de cualquier servidor de aplicación que sea capaz de aprovisionar cualquier tipo de servicio mientras que en la solución haya un módulo *ApplicationServer* que herede el interfaz del módulo *GenericServer* y sobrescriba sus métodos. Estos métodos cubren todas las etapas del establecimiento de la sesión: la creación, la reserva de recursos, la actualización, la terminación y la cancelación, entre otras. Cuando llegue una solicitud de sesión, el servidor de aplicación se encargará de evaluar la solicitud INVITE que recibe para decidir si le da servicio o no. Tiene un estado de salida entero: 0 si se encuentra el recurso deseado (AVAILABLE_TARGET), 1 si se encuentra pero no se puede aprovisionar (NOT_POSSIBLE) y 2 si no se encuentra el recurso (TARGET_NOT_FOUND). En el caso positivo se encargará de guardar la sesión para identificarla en futuras solicitudes dentro del establecimiento de sesión.

En la Fig. 5 se muestra la comunicación entre el interfaz genérico y el servidor de aplicación que se debe implementar junto con el flujo de mensajes de un establecimiento de sesión IMS.

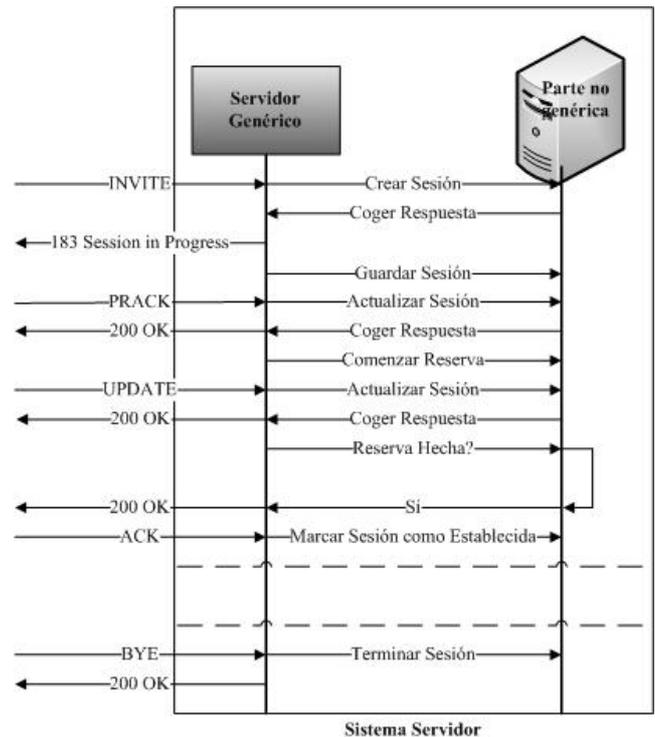


Fig. 5. Comunicación entre el interfaz genérico y el servidor de aplicación

Adicionalmente, para que el manejo de las sesiones que se establece en el servidor sea lo más sencillo a la hora de implementar el servidor de aplicación deseado, se han encapsulado las variables necesarias para facilitar la gestión de las sesiones.

IV. VALIDACIÓN Y RESULTADOS

Debido a la naturaleza genérica de la solución surge la necesidad de buscar un escenario real donde probar las funcionalidades del sistema. Para ello se propuso implementar un servidor de video bajo demanda (*Video on Demand, VoD*) que implementara la interfaz comentada en el apartado anterior para, de este modo, servir videos mediante *streaming*. En el presente apartado se muestra el diseño de dicho servidor.

También se recogen aquí las diferentes pruebas que se fueron haciendo durante el desarrollo del proyecto (dada la magnitud del mismo se fueron realizando diversas pruebas incrementales a medida que se iba desarrollando la implementación).

A. Servidor de Video bajo Demanda

El hecho de que el servidor genérico implemente la funcionalidad de la comunicación SIP permite tomar cualquier solución de diseño a la hora de implementar el servidor VoD en tanto en cuanto dicho servidor exponga correctamente la interfaz propuesta.

Por tanto, la implementación dispone de un servidor que aprovisiona videos en *streaming* al recibir una carga SDP con una descripción multimedia en la que se encuentren tanto la dirección y el puerto donde se desea recibir el flujo de video como los codecs soportados por el cliente.

Para identificar el recurso que se desea aprovisionar se decidió utilizar el parámetro *Request-URI* de la solicitud INVITE, utilizándose para ello el siguiente formato:

```
sip:identificador_fichero_video@direccion_servidor[:puerto]
```

Se decidió utilizar el campo User de la *Request-URI* para especificar el recurso que se desea consumir (mediante un identificador único), dejando el resto de los campos de la *Request-URI* disponibles para las tareas de encaminamiento.

Para lanzar y recibir el flujo de video se decidió utilizar la aplicación VLC [14] que además de poder funcionar como fuente y receptor de *streaming* de video puede ser manejado mediante la línea de comandos, lo que permite que de una manera sencilla se lancen los videos directamente desde el código del servidor de video bajo demanda y que mediante la inserción del comando correspondiente en el campo *Application Command* en la aplicación cliente se reproduzcan dichos videos.

Finalmente, para desplegar el servidor, se ha creado un archivo de configuración que se leerá al lanzar el servidor VoD.

Cuando se procese el fichero de configuración al iniciar el servidor VoD, se asignará de manera dinámica un identificador único a cada una de las líneas del archivo (es decir, a cada una de las duplas video-codec).

Ese código es el que se deberá introducir en la *Request-URI* de la solicitud SIP para identificar el video. Como es lógico, en un sistema comercial, dichos identificadores de recursos deberían ser pre-aprovisionados de alguna manera al cliente, por ejemplo, exponiendo el identificador de cada vídeo mediante un servicio web. Esto sería sólo una mejora y queda fuera del ámbito donde se inscribe este artículo, ya que no se pretende implementar un sistema servidor VoD completo, sino tener una implementación básica con la que poder validar nuestra plataforma. Por tanto, se optó por no pre-aprovisionar esos identificadores de manera automática.

La decisión de servir o no un vídeo se basa en dos decisiones; la primera es que el identificador recibido coincida con el identificador de alguno de los vídeos que se cargaron en el fichero de configuración (y que son por tanto los que se pueden servir) y lo segundo es que el códec que soporta el cliente coincida con el códec asociado a dicho video. En caso de no cumplirse la primera condición, el código de salida será un 2, que en la parte genérica del servidor se traducirá en mandar una respuesta 404 NOT FOUND. En el segundo caso, si no coinciden el códec, el valor de salida es un 1, que equivale a mandar un 606 SESSION NOT ACCEPTABLE.

B. Escenarios de pruebas

Durante el proceso de implementación del sistema se fueron realizando diversos test para comprobar que se estaban desarrollando las funcionalidades tal y como se pensaron durante las etapas de diseño. Estas pruebas se fueron realizando de manera incremental, desde el establecimiento de sesión más sencillo (el establecimiento SIP básico INVITE-OK-ACK) hasta el establecimiento de sesión de IMS con la gestión de precondiciones, la fiabilidad de respuestas provisionales y la

reserva de recursos.

1) Establecimiento de Sesión SIP básico

El establecimiento de sesión básico de SIP es tan simple como se muestra en la Fig. 6. El cliente manda la descripción de sesión SDP (oferta SDP) dentro del primer INVITE, y el servidor lo acepta enviando un 200 OK confirmando la carga SDP, o bien lo puede rechazar. Si el servidor acepta la sesión, el cliente responderá con un ACK y se iniciará el flujo multimedia.

Finalmente, cuando uno de los dos extremos quiere terminar la sesión, mandará un BYE y el otro extremo le responderá con un OK, momento en el que se tomará la sesión como finalizada.

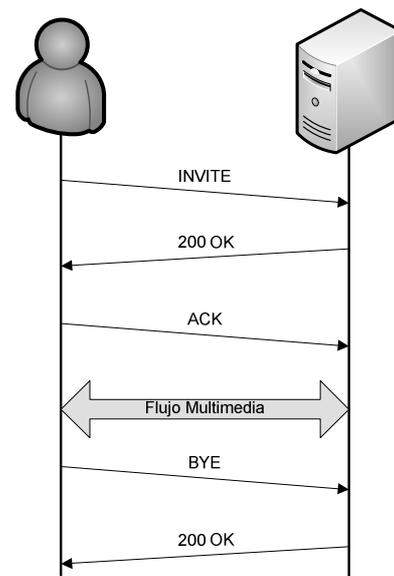


Fig. 6. Establecimiento de sesión SIP básico

Esta prueba se realizó en una subred privada donde se encontraban conectados tanto el cliente como el servidor. El puerto y el protocolo de transporte que van en la descripción de la sesión han de coincidir con los parámetros que se introducen en el comando de aplicación que ejecuta el VLC. En cuanto al Target (que corresponde con la Request URI del INVITE), especifica además de la dirección donde se encuentra el servidor, el identificador del recurso a servir, en la parte de usuario de la SIP URI. Se pueden especificar cuantos codecs se quieran, siempre y cuando aparezca el correspondiente al recurso deseado en esa lista.

Con esta prueba se verifican principalmente la capacidad que tiene el cliente de generar una carga SDP simple a través de los parámetros especificados en el interfaz gráfica y de establecer una intercambio de mensajes correcto con el servidor.

2) Establecimiento de Sesión IMS

Lo que se quiere realizar con este escenario de prueba es comprobar que la plataforma es operativa en un entorno basado en IMS. Para ello, en vez de comunicar directamente al cliente con el servidor vamos a simular el *core* de IMS con un servidor SIP SER en configuración proxy. Como ya se ha visto dichos nodos en soluciones como *Open IMS Core* se basan en implementaciones de estos servidores SIP. En la Fig. 7 se muestra dicho escenario.

En este caso si que se negocian los parámetros de la sesión y la reserva de recursos (en este caso se delega esta responsabilidad a la red). Por esa razón no se responde inmediatamente al INVITE con un OK como en el caso anterior, sino que se envía una respuesta provisional para indicar que se esta tratando de establecer la sesión.

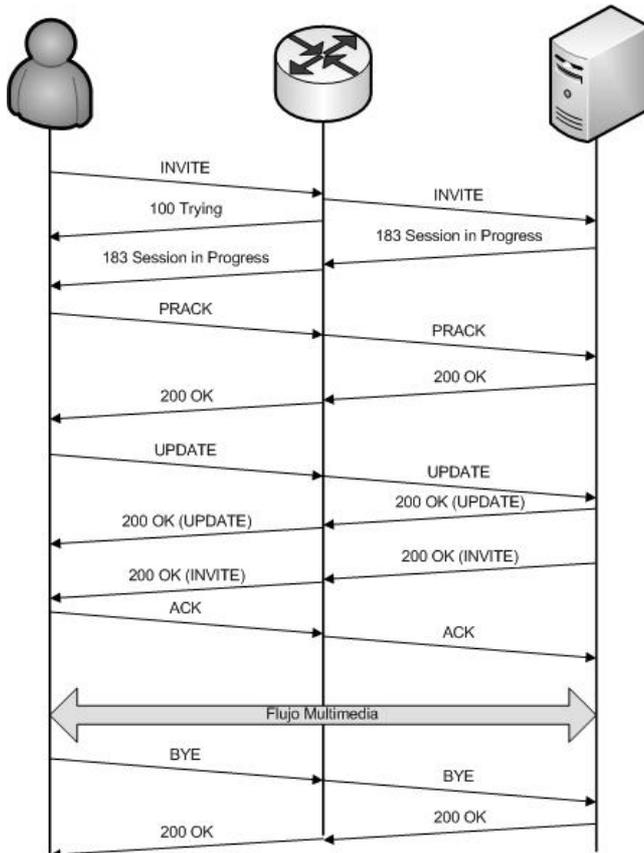


Fig. 7. Establecimiento de sesión IMS

En este entorno se utiliza la extensión de SIP “Fiabilidad de las respuestas provisionales” [15], el cliente debe enviar un PRACK (un ACK provisional equivalente al ACK de las respuestas finales) con carga SDP para confirmar la recepción del SESSION IN PROGRESS.

Además, como es posible que hayan cambiado los parámetros que especifican la sesión en la carga SDP, el cliente ha de mandar un UPDATE para confirmar la finalización del proceso de reserva de recursos, además de actualizar la descripción de la sesión con los parámetros finales de la sesión (que han sido obtenidos tras esa reserva).

Para simular el *core* de IMS se escogió la última versión estable (0.9.6) del servidor SIP SER [6], en su configuración más simple, funcionando como proxy dentro de la subred de pruebas. En la Fig. 7 se muestra la configuración de red que se utilizó y el intercambio de mensajes correspondiente.

V. CONCLUSIONES Y TRABAJO FUTURO

Para finalizar este análisis, hay que extraer determinadas conclusiones tanto del resultado de la implementación como del

desarrollo de la misma, además de intentar buscar las distintas aplicaciones que puede tener la plataforma.

Se ha implementado una plataforma genérica para la provisión de servicios multimedia siguiendo los mecanismos del plano de control de IMS. El sistema cliente propuesto para la plataforma es capaz de realizar las labores de control (establecimiento, mantenimiento y liberación de sesión) tanto para redes basadas en el protocolo SIP clásico, como redes con el plano de control utilizado en IMS.

Hay que destacar también que la interfaz gráfica desarrollada permite iniciar sesiones multimedia genéricas, permitiendo de este modo acceder a cualquier tipo de servicio ofrecido por un proveedor a través de la arquitectura IMS. A pesar de lo laborioso que puede ser a priori generar una descripción de sesión basada en SDP debido al gran número de etiquetas y atributos necesarios, se ha conseguido esconder dicha complejidad gracias a los procedimientos utilizados por NIST SDP y mostrar sólo al usuario los parámetros que debe escoger para generar dicha carga.

Puesto que el público base de la aplicación se enmarca dentro de un entorno académico de pruebas (en redes IMS/TISPAN) la facilidad de especificar descripciones de sesión usando parámetros de bajo nivel hace que la aplicación sea de gran utilidad.

Los mecanismos adicionales para el establecimiento de la sesión, como puede ser la modificación de las precondiciones, también han sido implementados. Además, el sistema está preparado para soportar distintos mecanismos de reserva de recursos aunque en un principio no se ha implementado ninguno en particular, ya que se la plataforma ha sido diseñada a priori para el acceso desde redes fijas, siguiendo el modelo propuesto por el grupo de estandarización TISPAN, en las que la reserva de recursos se delega en la red.

Con respecto a la separación dentro del sistema servidor en parte genérica (que implementa la comunicación SIP) y en parte no genérica (que implementa un servidor tradicional), esto hace realmente fácil la creación de servidores de aplicación nativos en SIP, e incluso el adaptar servidores que de otra manera tendrían que adoptar configuraciones mucho más complejas. Esto vuelve a ser, dentro del entorno en el que se desea utilizar la aplicación, de gran utilidad al poder implementar de una manera sencilla y ligera, diversos servidores de aplicación a los que solicitar varios servicios dentro de una misma sesión.

A continuación, se identifican ciertas líneas de trabajo futuro, que no se han realizado en la implementación pero que se relatan a continuación:

La plataforma se centra en las comunicaciones con servidores de aplicación, pero también se podría poder proporcionar una manera sencilla de establecer comunicaciones de tipo peer-to-peer. En cuanto a mejoras técnicas, y a la vista de las diversas extensiones del protocolo SIP que se utilizan dentro de la arquitectura IMS, tales como las relacionadas con la compresión de los mensajes SIP, como línea futura se podría incluir en la implementación el soporte de estas extensiones, para de este modo disponer de una implementación completa de la plataforma válida para cualquier escenario de acceso.

REFERENCIAS

- [1] Recommendation Q.1762/Y.2802 "Fixed-mobile Convergence General Requirements"
- [2] Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN): NGN Terminology
- [3] N.Blum, T.Magedanz, A Jaokar "IMS and Web2.0/SOA" Workshop 3rd International FOKUS IMS Workshop 2007
- [4] IETF "SIP: Session Initiation Protocol", RFC 3261
- [5] G. Camarillo, M. García-Martín: "The 3G IP Multimedia Subsystem (IMS)", 2nd Edition, Wiley
- [6] SIP Express Router (SER), www.iptel.org (última visita: 6-06-08)
- [7] Open SIP Express Router www.openser.org (última visita: 6-06-08)
- [8] Dragos Vingarzan, Peter Weik, Thomas Magedanz: "Design and Implementation of an Open IMS Core", MATA 2005, 2nd International Workshop on Mobility Aware Technologies and Applications (Formerly Mobile Agents for Telecommunication Applications), Montreal, Canada, 17-19 October 2005
- [9] J. Fabini, P. Reichl, A. Poropatich, R. Huber, N. Jordan, "IMS in a bottle: Initial Experiences from an OpenSER-based Prototype Implementation of the 3GPP IP Multimedia Subsystem", Proc. International Conference on Mobile Business (ICMB) m>business2006, Copenhagen, Denmark, June 26-27, 2006
- [10] IETF "SIP Extension Header Field for Registering Non-Adjacent Contacts", RFC 3327
- [11] IETF "Private Header Extensions to the Session Initiation Protocol (SIP) for the Third Generation Partnership Project (3GPP)", RFC 3455
- [12] IETF "SIP Extension Header Field for Service Route Discovery During Registration", RFC 3608
- [13] IETF "Integration of Resource Management and Session Initiation Protocol" RFC 3312
- [14] VideoLAN www.videolan.org (última visita: 6-06-08)
- [15] IETF "Reliability of Provisional Responses in Session Initiation Protocol" RFC 3262