

## *Simulador Educativo de un Servicio de Audio/Vídeo Bajo Demanda*

David Melendi, Xabiel G. Pañeda, Ángel Neira, Roberto García, Víctor G. García

Departamento de Informática, Universidad de Oviedo,  
Campus Universitario de Viesques, Sede Departamental Oeste,  
33204, Xixón, Asturias, España  
{melendi, xabiel, neira, garciaroberto, victor}@uniovi.es

**Resumen:** Este artículo presenta el diseño e implementación de una herramienta para facilitar la comprensión del funcionamiento de un servicio de audio/vídeo *streaming*. Su objetivo serán alumnos de carreras técnicas (informática y telecomunicaciones) que se adentran por primera vez con un servicio de características muy especiales, entre las que se encuentran: fuerte consumo de recursos, elementos para garantizar la calidad de servicio, protocolos independientes para control y transporte de la información, protocolos con estados, entre otras.

**Palabras clave:** Simulador Educativo, streaming, audio/vídeo bajo demanda.

**Abstract:** This paper presents the design and the implementation of a tool whose aim is to show the behaviour of an audio/video streaming service. The target is students of technical degrees (computer science and telecommunications), who are trying to obtain a general vision to this type of services, featured by having a strong resource consumption, techniques to guarantee quality of service, specialized protocols for transport and flow control, among other characteristics.

**Keywords:** Educational Simulator, streaming, audio/video on demand.

### 1. Introducción

La popularización de Internet y la mejora de las redes de acceso de los usuarios, ha permitido en los últimos años la proliferación de nuevos tipos de servicio. Entre los servicios de mayor desarrollo tecnológico y de mercado están aquellos que proveen contenidos de audio/vídeo en Internet, debido principalmente a la fuerte apuesta de los medios de comunicación por este tipo de difusión de la información. Estos servicios están basados, en su mayor parte, en la tecnología de *streaming*, que se caracteriza principalmente por permitir la reproducción de los contenidos mientras la información está siendo enviada.

Detrás de la tecnología de *streaming* hay una alta complejidad que hace necesaria la utilización de

múltiples elementos y protocolos. Esta complejidad dificulta la asimilación de este tipo de servicios por estudiantes noveles en el campo de las tecnologías de la información. En este entorno, carreras universitarias como informática y telecomunicaciones no pueden quedarse al margen y deben incorporar a sus planes docentes temas relacionados con este tipo de servicios.

Para facilitar el aprendizaje de este tipo de tecnologías, resulta muy interesante disponer de elementos capaces de mostrar a los alumnos el funcionamiento interno de cada uno de los elementos necesarios para proveer el servicio. Sin embargo, durante el primer acercamiento, los conceptos que se presentan al alumno no deberán ser excesivamente complejos ya que podrían hacer que la visión general

del servicio quedara oculta por complejos conceptos teóricos.

Con este propósito, en el *Área de Ingeniería Telemática* del *Departamento de Informática* de la *Universidad de Oviedo* se ha diseñado un simulador capaz de mostrar a los estudiantes de una manera sencilla e intuitiva las interioridades de un servicio de audio/vídeo *streaming*. Este simulador pretende ser la primera piedra de un proceso de aprendizaje amplio que será posteriormente completado con clases teóricas y prácticas sobre la materia.

El simulador ha sido diseñado de tal forma que su manejo sea lo más similar posible al de un cliente convencional de este tipo de servicios. De esta forma, se consigue que los estudiantes tengan unas nociones básicas de utilización de la herramienta y puedan así realizar simulaciones sin ninguna formación previa sobre el uso de la misma. Para facilitar la difusión del simulador y posibilitar su ejecución en cualquier tipo de entorno, la implementación se ha realizado utilizando un *Applet Java*, permitiendo la integración de la herramienta en entornos Web.

El resto del artículo está organizado de la siguiente forma: en la sección 2 se comentan trabajos relacionados con el presentado en este artículo, mientras que en la sección 3 se describe el modelo teórico subyacente que caracteriza la tecnología de *streaming* y que se pretende que el alumno adquiera mediante el uso del simulador. Los objetivos del simulador se presentan en la sección 4. En la sección 5 se especificará el diseño del simulador. La sección 6 hablará sobre la herramienta que se ha desarrollado y la 7 presenta las conclusiones obtenidas. Por último, los trabajos futuros serán expuestos en la sección 8.

## 2. Trabajos Relacionados

Uno de los objetivos de la enseñanza en comunicaciones debe ser la de transmitir a los alumnos el conocimiento sobre las posibilidades y restricciones de cada medio. De esta forma, podrán seleccionar las herramientas y técnicas de comunicaciones más apropiadas para resolver un problema determinado, de entre el amplio abanico de soluciones disponible en la actualidad. Diseñar

entornos de enseñanza orientados a este fin es una forma idónea de alcanzar este objetivo [Collins et al. 00].

En esta línea, una técnica muy ampliamente utilizada para el aprendizaje es la del *estudio de casos*. Por medio de escenarios altamente intuitivos, se presentan al alumno problemas reales buscando la motivación del mismo en su comprensión inicial de lo que se pretende resolver. Esto permite un estudio posterior en profundidad, en la búsqueda de los conocimientos necesarios para proceder a su desarrollo y resolución. Sin pretender realizar una enumeración detallada de los numerosos trabajos de aplicación en este campo, sí que es necesario referenciar por su interés trabajos como [Wilson 96] en la identificación y caracterización de estos entornos educativos.

El modelado en general y la simulación en particular han jugado desde siempre un papel fundamental en el razonamiento y la resolución de problemas. Es más, la simulación constituye uno de los recursos más útiles para la enseñanza de técnicas y es, en muchos casos, la única forma de formalizar y resolver problemas conectados con el análisis o la síntesis de sistemas de gran complejidad en su estructura y/o funcionamiento. Un clásico imprescindible en este terreno es [Futó et al. 90].

Existen gran cantidad de ejemplos y aplicaciones en todos los campos de la Ingeniería, como constatan trabajos como [Menczer et al. 01, Dormida et al. 97, Campos et al. 02], y se haría ilimitada su enumeración aunque solo pretendiésemos restringirnos a entornos basados en la enseñanza a distancia. Merece, sin embargo, ser citado un trabajo como [Buendía et al. 01] que contempla el modelado y diseño de entornos educacionales basados en la simulación y posibilita la evaluación de los recursos didácticos. Aunque por su perspectiva pedagógica, trabajos divulgativos como [Cattaneo 02] también deben ser referenciados.

Naturalmente, para el desarrollo de la tecnología de *streaming* y de los servicios de audio y vídeo bajo demanda se han utilizado referencias clásicas como [Schulzrinne et al. 98, Schulzrinne et al. 00] y otras más recientes como [Martín et al. 03] en las que se proporcionan de forma pormenorizada todos los

detalles relacionados con este tipo de servicios. En cuanto a la aplicación del *streaming* en la gestión de contenidos ofrecidos a los alumnos en entornos educativos a distancia se han tomado como fuente trabajos como [Pañeda et al. 05], en el que se ofrece una completa metodología para la evaluación de sistemas de tele-educación multimedia.

### 3. Modelo Teórico Subyacente

La tecnología de *streaming* se ha diseñado para proveer servicios multimedia en Internet. Así, está orientada al manejo de información de tipo continuo, es decir, audio y vídeo, y utiliza arquitecturas cliente/servidor de  $N$  capas.

Su característica más importante consiste en la forma de fraccionar la información para transmitirla de forma temporizada, es decir, el servidor almacena la información multimedia en unos formatos especiales que le permiten enviar la información a trozos en instantes determinados. Estos formatos permiten también que el servidor sea capaz de separar la información en flujos independientes, por ejemplo, transmitir el sonido por un canal y el vídeo por otro. El cliente recoge cada uno de los trozos de cada uno de los flujos y los reproduce sin necesidad de tener almacenada la totalidad de la información para comenzar a reproducirla, es decir, la reproducción puede comenzar instantes después del inicio de la transmisión.

La transmisión de la información y el control sobre la misma se realizan mediante protocolos específicos, como pueden ser el *RTP (Real-Time Transport Protocol)* [Schulzrinne et al. 98] y el *RTSP (Real*

*Time Streaming Protocol)* [Schulzrinne et al. 00], entre otros.

La tecnología de *streaming* también se caracteriza por permitir al cliente interactuar con el servidor de forma que éste puede arrancar o parar la transmisión de la información, o simplemente solicitar diferentes trozos en cada momento.

#### 3.1. Servidor

El servidor es uno de los elementos clave en los servicios basados en la tecnología de *streaming*. Su misión es la de recibir las peticiones emitidas por los clientes y transmitir la información adecuada en cada momento.

El servidor filtra las peticiones con una política de admisión determinada, para no comprometer la calidad de servicio. Generalmente, deberá proceder a rechazar peticiones cuando detecte que no va a ser capaz de transmitir los flujos con las temporizaciones adecuadas.

Cuando el servidor acepta una petición establece las conexiones con el cliente, en algunos casos teniendo en cuenta la capacidad de la línea a la que está conectado. Es decir, si el cliente está conectado a una línea de 56 kbps, el servidor deberá enviarle un flujo de información que no supere este ancho de banda para garantizar que el audio/vídeo se vea con calidad.

El servidor establece con el cliente tantos canales de comunicación como flujos tiene la información que debe transmitir, es decir, si la información solicitada tiene flujos independientes de audio y de vídeo, el

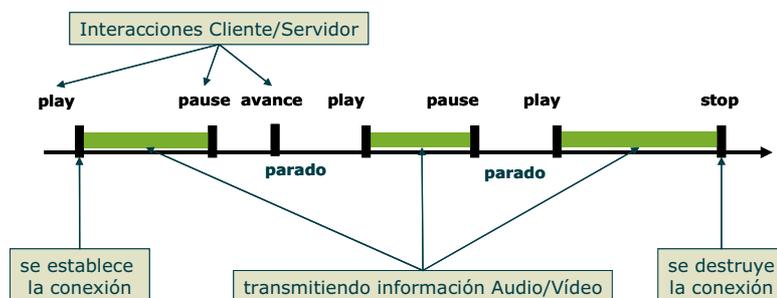


Figura 1. Estados en los que se encuentra el servidor durante una reproducción

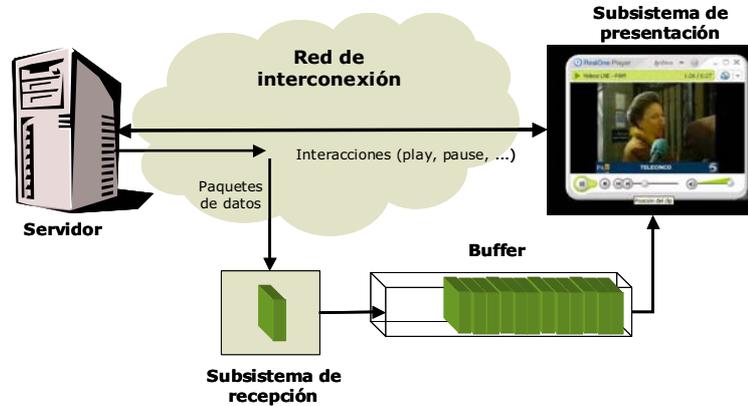


Figura 2. Subsistemas del cliente

servidor establecerá dos canales con el cliente.

El servidor se encarga de mantener el estado en el que se encuentra la sesión establecida con cada uno de los clientes de los que recibe peticiones. Según el estado, tendrá o no que transmitir datos a los clientes. Este estado se verá afectado por las interacciones que realizan los clientes con la información. Así, cuando los clientes hacen una pausa, el servidor podrá detener el flujo de datos multimedia, mientras que si se vuelve a solicitar la reproducción de los contenidos, la transmisión de información deberá comenzar de nuevo en el punto solicitado por el cliente correspondiente.

En la figura 1 se muestra la evolución temporal de la reproducción de un audio/vídeo donde las interacciones se van sucediendo. Algunas de ellas generarán un cambio de estado en la comunicación,

lo que generará periodos en los que se transmite información y otros en los que no.

En los periodos en los que el usuario demanda el envío de información, el servidor transmite el audio/vídeo de forma temporizada para cada uno de los flujos existentes, hasta que el usuario solicite una pausa o la finalización de la transmisión, tal y como se muestra en la figura 2.

### 3.2. Cliente

Un cliente o reproductor de un sistema basado en la tecnología de *streaming* está formado por dos subsistemas principales: el subsistema de recepción y el subsistema de presentación. El primero se encarga de la recepción de los paquetes de audio/vídeo enviados por el servidor o algún dispositivo

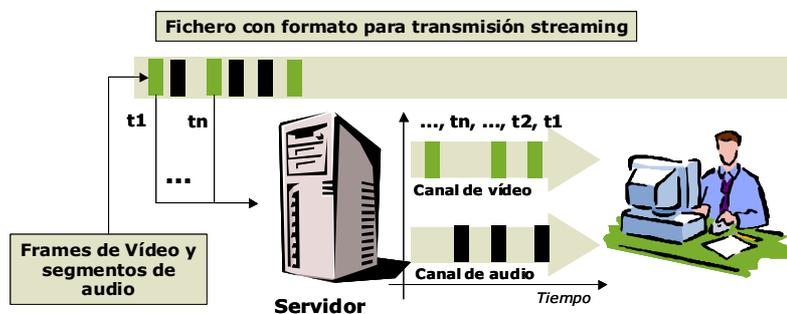


Figura 3. Transmisión temporizada de información multimedia con dos flujos

intermedio. Por otro lado, el segundo recibe las interacciones del usuario (enviándolas al servidor) y presenta la información multimedia que le proporciona el subsistema de recepción (en pantalla y en megafonía). Entre ambos subsistemas se encuentra un buffer para el almacenamiento temporal de la información, tal y como se muestra en la figura 3.

El cliente tiene varias misiones:

- Por un lado se encarga de recibir los paquetes que el servidor periódicamente le envía y almacenarlos en un *buffer*. El cliente mantiene este *buffer* para evitar deficiencias en la presentación del audio/vídeo debido a los posibles retrasos en la cadencia de llegada de los paquetes multimedia. El tamaño del *buffer* es configurable, y aunque en condiciones ideales no debería ser necesario, en los sistemas reales juega un papel muy importante debido a las subidas y bajadas de calidad de servicio de las redes por donde atraviesa la información. Inicialmente, y antes de que el usuario pueda visualizar o escuchar el audio/vídeo, el *buffer* es cargado con unos segundos de contenidos. A lo largo de la reproducción recibirá paquetes del subsistema de recepción y pasará paquetes al subsistema de representación.
- Por otro lado, el subsistema de presentación representa la información recibida. El programa cliente es el utilizado por el usuario del servicio para visualizar y escuchar la información de audio y vídeo. El subsistema de reproducción de la información lee del *buffer* la información que debe presentar en pantalla. Al instante de presentación de un paquete almacenado en el *buffer* se le denomina instante de *playout*. Cada paquete se reproduce en su instante de *playout*, y si en el instante de reproducción no hubiera ningún paquete almacenado en el *buffer*, el cliente haría una pausa en la reproducción lo suficientemente larga para volver al recargar el *buffer* totalmente.
- Además, el cliente proporciona una interfaz para que el usuario pueda interactuar con el servidor. Debe permitirle parar la reproducción, hacer pausas, reiniciar la reproducción, y en determinados casos, dependiendo del tipo del servicio, permitirle hacer avances en la línea de reproducción, retrocesos rápidos, y

posicionamientos en puntos concretos de la información.

- En algunos casos el cliente envía información al servidor sobre el tipo de red al que el usuario está conectado, para que éste envíe mayor o menor volumen de información y conseguir con ello que la reproducción no sufra interrupciones.

#### 4. Objetivos

Los objetivos del simulador giran entorno a la idea de mostrar a los estudiantes cómo es el funcionamiento de un servicio de audio/vídeo *streaming* a alto y medio nivel, siguiendo los conceptos teóricos presentados en el capítulo anterior. El simulador se empleará como introducción a la materia, para posteriormente, realizar un análisis más detallado de la tecnología en clases guiadas con el profesor de la asignatura correspondiente.

Mediante un entorno que pueda resultarles familiar, se pretende que los estudiantes adquieran conocimientos sobre:

- Arquitectura básica de un servicio.
- Coordinación entre los diferentes elementos de la arquitectura (cliente, servidor).
- Coordinación entre los protocolos de control y de transporte.
- Funcionamiento y utilidad del buffer del cliente o reproductor.

Para ello, el simulador deberá proporcionar las siguientes funcionalidades:

- **Mostrar la evolución del servidor a través del estado en el que se encuentra.** Presentar en todo momento cuál es el estado en el que se encuentra el servidor. Los estados contemplados podrán ser dos: *Transmitiendo* o *Parado*.
- **Mostrar la transmisión de paquetes de control por la red.** Se presentarán los paquetes que se transmiten por la red como respuesta a las interacciones del usuario.
- **Mostrar la transmisión de paquetes de transporte por la red.** Se mostrará cómo se transmiten los paquetes que transportan los contenidos de audio/vídeo entre el servidor y el cliente.

- **Mostrar la evolución del reproductor del usuario.** El reproductor del usuario puede estar reproduciendo los contenidos que le envía el servidor o parado.
- **Permitir la interacción del usuario con el reproductor.** El usuario podrá realizar interacciones con el reproductor al igual que lo haría un usuario en un servicio real. Será posible comenzar la reproducción, detener la misma, hacer pausas intermedias, y avanzar o retroceder en la línea de reproducción.
- **Mostrar el nivel de carga del buffer del cliente.** En todo momento se presentará la cantidad de contenidos de audio/vídeo almacenados en el *buffer* del cliente.
- **Permitir al usuario la configuración de la cantidad de información que se carga inicialmente en el buffer del cliente.** El usuario deberá ser capaz de modificar la cantidad de tiempo que se carga en el *buffer* inicialmente y la cantidad mínima que debe haber en el *buffer* en cada instante de la reproducción. En caso de que los contenidos almacenados en el *buffer* del cliente sean insuficientes, será necesario realizar una recarga del mismo y el usuario experimentará una pausa no deseada.
- **Ofrecer estadísticas finales sobre la simulación.** Una vez que se haya concluido la simulación, el usuario podrá disponer de estadísticas detalladas de la misma en las que se le informará sobre la evolución en el intercambio de paquetes entre servidor y cliente.

## 5. Diseño

Uno de los objetivos que se pretenden conseguir con este simulador es que el alumno tenga una visión de conjunto de un servicio de audio/vídeo *streaming*. Percibir que los componentes que participan en su funcionamiento están coordinados a través de los mensajes que se intercambian es fundamental a la hora de entender el servicio.

Por todo ello, el simulador se ha estructurado en 3 partes bien diferenciadas que se corresponden con los elementos funcionales de un servicio real: servidor, cliente y red de comunicaciones. En la figura 4 se muestran las relaciones entre estos tres elementos.

El simulador mostrará el funcionamiento del servicio para un único cliente, que puede realizar peticiones sobre un servidor. Para ello, el cliente o reproductor tendrá un conjunto de controles en forma de botones desde los que el usuario podrá realizar las interacciones básicas de inicio o *play*, pausa o *pause* y parada o *stop*. Se proporcionará una barra de navegación horizontal que permitirá desplazarse a voluntad en la línea de reproducción, realizando saltos hacia atrás o hacia delante. También se dispondrá de una pantalla en la que poder ver la información que se desea reproducir. Para que el alumno pueda observar cuándo y porqué la reproducción se inicia o se detiene por problemas de calidad, el simulador mostrará el estado del *buffer* del reproductor en todo momento.

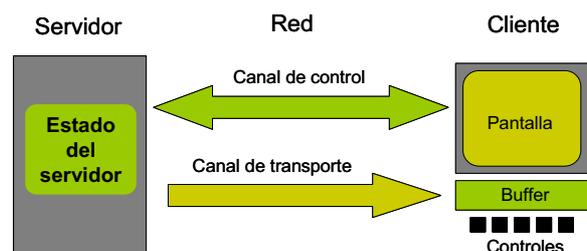


Figura 4. Diseño del simulador

En lo que se refiere a la red de comunicaciones, se representarán los dos canales de comunicación que utilizan los servicios de *streaming*, uno para el control del servicio, de tipo bidireccional, y otro para el transporte de datos de tipo unidireccional (de servidor a cliente). Cada vez que el cliente o el servidor envíen un paquete, éste se mostrará en el canal correspondiente en función del tipo del que se trate.

Por último, el simulador mostrará el estado en el que se encuentra el servidor. A medida que el usuario va haciendo interacciones, el servidor irá recibiendo paquetes de control que deberán visualizarse en el canal de control de la red de comunicaciones. En función de los paquetes de control recibidos, el servidor deberá adaptar su estado tal y como se mostraba en la figura 1. Así, un servidor podrá estar en dos estados principales: *Transmitiendo* o *Parado*. En el estado *Transmitiendo*, los paquetes que envía el servidor multimedia al cliente deberán mostrarse en

el canal de transporte de la red de comunicaciones, mientras que en el estado *Parado* el canal de transporte deberá permanecer inactivo.

## 6. Herramienta Desarrollada

### 6.1. Descripción General

Para llevar a cabo la implementación del simulador se ha escogido el lenguaje de programación *Java*, ya que permite la implementación de programas capaces de ser integrados en una página Web en forma de *applets*, lo cual resulta muy interesante a la hora de conseguir una amplia distribución de la herramienta y una masiva utilización por parte de los alumnos interesados.

Para su diseño se han seguido las especificaciones presentadas en el apartado anterior, al que se le ha incorporado un elemento para generar la traza de la simulación. Además, como elementos auxiliares el simulador, se dispone de dos pantallas que permiten la configuración de las simulaciones y la generación de informes detallados sobre las mismas.

En cuanto a los pormenores de implementación, debe indicarse que para la construcción del simulador educativo presentado en este trabajo se ha utilizado básicamente una estructura basada en agentes concurrentes interrelacionados y cooperando en los procesos de comunicación cliente-servidor. Se ha implementado para ello un sistema orientado a procesos distinguiendo explícitamente procesos e intercambio de información, encapsulando datos y métodos en objetos de acuerdo con la metodología presentada en [Splunter et al. 03].

En la figura 5 se muestra un modelo simplificado de clases del simulador desarrollado. Aunque se han omitido algunas clases, atributos y métodos, esta figura permite apreciar la complejidad de la herramienta desarrollada.

La herramienta recibe un conjunto de parámetros de entrada que mantiene en una instancia de la clase *CurrentConfig*. Aunque existen unos valores por defecto para cada uno de estos parámetros, el usuario

podrá alterar estos valores para modificar a placer el comportamiento deseado del simulador.

Durante las simulaciones se van almacenando listados con los paquetes utilizados por la herramienta en cada momento. Cada uno de estos listados es una especialización de uno de tipo genérico implementado mediante la clase *PacketList*. Estas especializaciones se emplearán para llevar un control exhaustivo sobre los paquetes de datos enviados por el servidor y recibidos por el cliente –terminadas en *Data-*, así como los paquetes de control enviados y recibidos tanto por cliente como por servidor – terminadas en *Ctrl-*. Además, existe otra especialización que pretende controlar los paquetes que hay en el buffer del cliente en cada momento – *ClientBuffer-*, que permite establecer los tiempos de carga iniciales y las recargas intermedias en caso de que se produzcan problemas en la comunicación. Teniendo en cuenta que los paquetes de datos se envían utilizando un protocolo que en principio no es fiable, cabe la posibilidad de que existan pérdidas que requieran reenviar alguno de los paquetes. De esta forma podemos encontrar paquetes de diferentes tipos: perdidos, reenviados, correctos, etc.

El simulador también mantiene un listado con las evoluciones en el estado general del sistema – *SystemStatusList-* y en el buffer del cliente – *BufferStatusList-*. Estos listados de estado van recogiendo los eventos ocurridos durante las simulaciones. Permiten generar un completo informe final en el que se detallan cada uno de los eventos ocurridos durante la ejecución, el estado anterior de cada elemento del sistema y el estado final al que se llega. Estos listados se gestionan a través de una instancia de la clase *SystemStatistics* que va recogiendo información que posteriormente será mostrada a los usuarios.

Para un correcto funcionamiento del simulador, se han empleado técnicas de programación con múltiples hilos. Estas técnicas permiten la ejecución concurrente de cada uno de los elementos que componen el sistema. Así, se dispone de diferentes hilos en los que se ejecuta el programa principal que controla las simulaciones y mantiene la sincronización del resto de elementos –*MainThread-*, el interfaz gráfico que muestra el estado de cada uno de los elementos –*ScreenThread-*, el reloj que

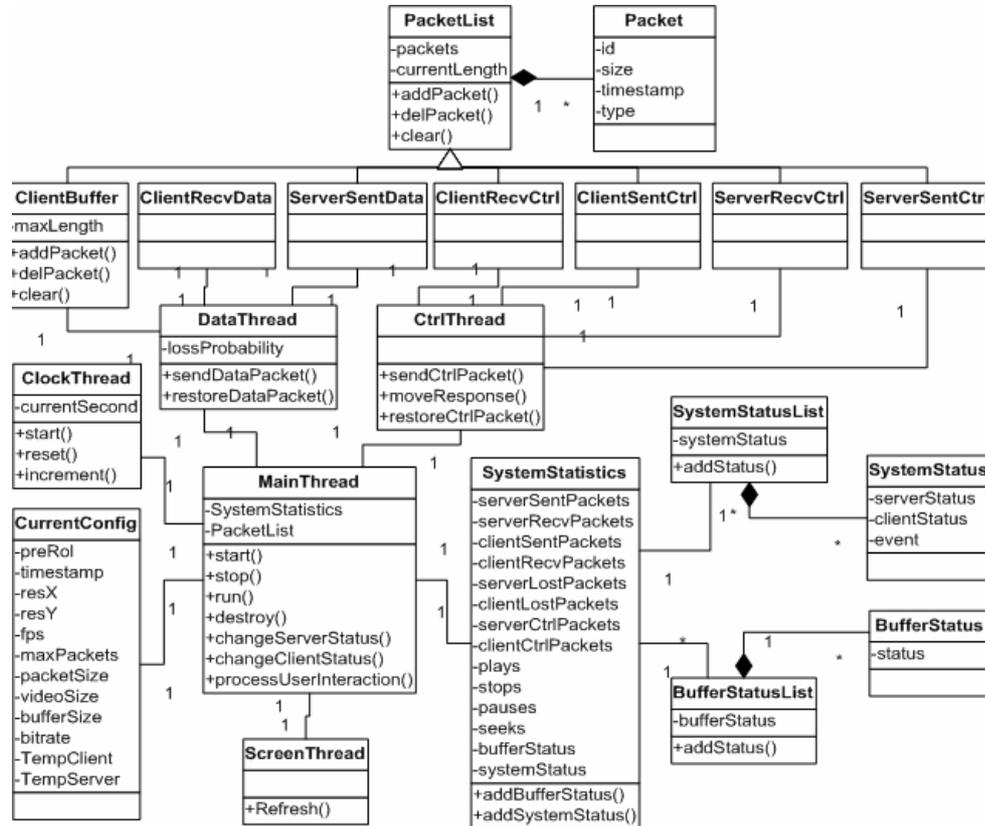


Figura 5. Modelo simplificado de clases del sistema

permite controlar y visualizar los tiempos de simulación *-ClockThread-*, un proceso que controla el intercambio de paquetes de datos entre cliente y servidor *-DataThread-* y un último proceso que gestiona el intercambio de paquetes de control correspondiente *-CtrlThread-*.

## 6.2. Configuración

La herramienta dispone de un módulo de configuración en el que se pueden ajustar los parámetros que regirán la ejecución de las simulaciones. Estos parámetros se resumen a continuación:

- **Tamaño del buffer del cliente.** Número máximo de paquetes de audio/vídeo que el buffer del cliente es capaz de almacenar.
- **Ancho de banda.** Ancho de banda medio disponible en la red en kilobits por segundo. Este elemento condicionará la velocidad de llegada de

los paquetes al cliente. En el caso que el ancho de banda disponible sea menor que el necesario para la transmisión de los contenidos, los retardos se irán acumulando hasta que la reproducción tenga que ser detenida por falta de paquetes en el *buffer* y sea necesario realizar una recarga del mismo.

- **Tamaño de los paquetes y número de fotogramas por segundo.** Estos parámetros condicionarán la cantidad de información que envía el servidor al cliente en cada momento. De este modo, podrá calcularse el ancho de banda necesario para llevar a cabo la transmisión del audio/vídeo. Comparando el ancho de banda necesario con el disponible, podrán simularse problemas relacionados con una capacidad de descarga insuficiente por parte del cliente.

Se proporcionan valores por defecto para todos y cada uno de estos parámetros, evitando que los usuarios tengan que determinar la configuración de la

herramienta en sus primeras simulaciones, ya que probablemente desconozcan claramente el significado de los mismos.

Una vez que los usuarios entiendan la utilidad de cada uno de estos parámetros, podrán realizar los ajustes deseados para generar nuevas simulaciones en las que reproducir efectos interesantes en las reproducciones de un audio/vídeo.

### 6.3. Ejecución de las Simulaciones

Una vez que se ha completado el paso inicial de configuración de la herramienta, las simulaciones pueden llevarse a cabo utilizando los controles provistos en la misma. En la figura 6 se muestra una captura del interfaz gráfico de la herramienta durante una simulación. En esta figura se aprecia cómo la herramienta se ha dividido en dos secciones principales: arquitectura del servicio y controles de simulación.

La sección de arquitectura de servicio está situada en

la parte superior de la herramienta, y en ella se muestra la evolución de cada uno de los elementos que intervienen en uno de estos servicios. En todo momento se muestra el estado en el que se encuentran tanto servidor como cliente. En cuanto a la red de comunicaciones, se muestra cómo los paquetes se intercambian, diferenciando entre paquetes de control y paquetes de datos. Para el cliente también se ofrece información detallada sobre el estado del buffer de reproducción, así como los botones de reproducción, parada y pausa y una barra de navegación horizontal que permite realizar saltos hacia delante y atrás.

La sección de controles de simulación ocupa el resto de la interfaz gráfica, y en ella se ofrece información de tipo detallado sobre la simulación, además de un conjunto de botones que permiten controlar las partes principales de la herramienta. En la parte central se ofrecen varias pestañas en las que se muestra el estado en el que se encuentra el simulador en cada momento, y la evolución e incidencias en el envío de paquetes de datos del servidor al cliente: número total de paquetes enviados por el servidor, recibidos en el cliente y perdidos durante la transmisión. Por otro lado, los botones que se ofrecen en la parte inferior

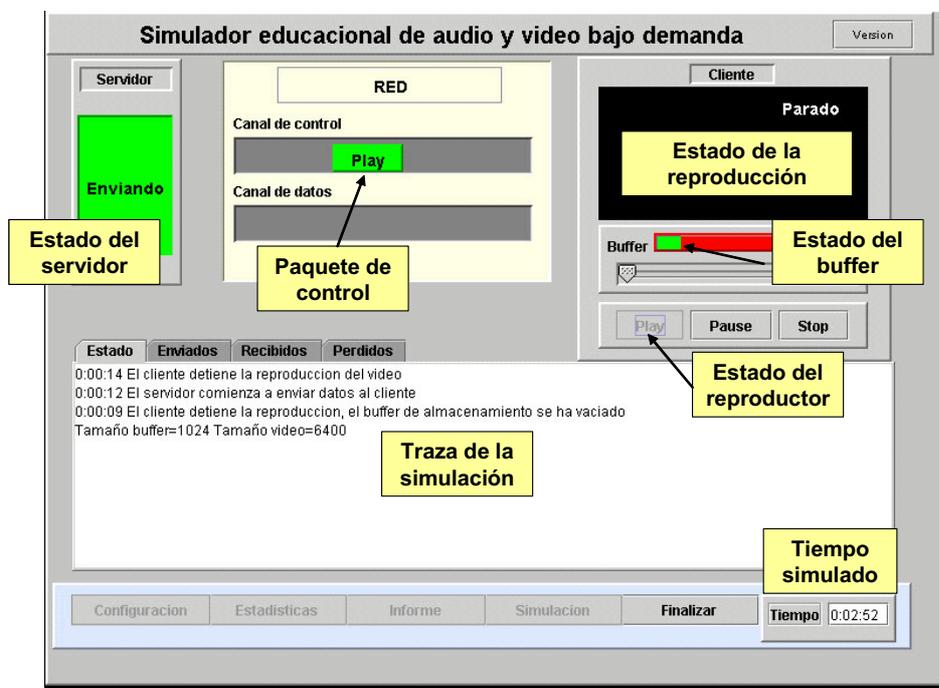


Figura 6. Interfaz gráfica de usuario

permiten acceder al módulo de configuración, a las estadísticas e informes finales, iniciar una simulación y detener la simulación en curso. Finalmente, también se ofrece un contador en el que se calcula la duración de las simulaciones con una precisión de segundos.

Una vez que el usuario ha accedido a la herramienta a través de una página web, puede comenzar las simulaciones con solo apretar el botón *Simulación* que se encarga de inicializar el sistema y arrancar los temporizadores utilizados por el sistema.

Después de este paso inicial el usuario puede proceder a ejecutar la primera interacción *play*. El cliente envía entonces por el canal de control un mensaje *play* al servidor. A partir de ese momento el servidor cambia de estado, envía un asentimiento al cliente y comienza a enviarle paquetes de datos (audio/vídeo). El cliente realizará entonces la carga inicial de su buffer, tras lo que sufrirá un cambio de estado y comenzará la reproducción. Esta reproducción se mantendrá hasta que el usuario realice una interacción, bien utilizando los botones de *pausa* o *stop*, o la barra de desplazamiento horizontal. En el momento en el que el usuario proceda a interactuar con los contenidos, se enviarán los mensajes correspondientes al servidor provocando los cambios de estado correspondientes y la ejecución de las acciones oportunas.

En función de los parámetros que se hayan seleccionado para realizar la simulación, el buffer del cliente podrá quedarse sin paquetes de datos. Esta situación puede alcanzarse cuando la velocidad de descarga es menor a la velocidad de reproducción de los contenidos. En ese caso el cliente detiene la reproducción hasta que el buffer está completamente recargado, tras lo que se reinicia el curso normal de simulación.

#### 6.4. Resultados Presentados

Una vez que se ha alcanzado el final de la simulación, bien por falta de contenidos, bien por haberse provocado una interacción de tipo stop, se ofrece al usuario información detallada sobre la simulación:

- La evolución en el estado de cliente y servidor y los eventos transcurridos durante la simulación,

que se muestra en la ventana principal de la herramienta en la pestaña titulada como *Estado*.

- En la pestaña *Enviados* se muestra un informe detallado sobre los paquetes enviados tanto por cliente como servidor hacia el otro extremo, en el que se proporciona información sobre el componente que originó el paquete, el instante en el que se envió el mismo, el tipo de paquete y el impacto que este tiene en el sistema general del simulador (porcentaje de carga en el buffer, tiempo de precarga en el cliente, etc.).
- Por otro lado, en la pestaña *Recibidos* se ofrece información similar a la anterior, pero para los paquetes recibidos por ambos extremos.
- Idéntica información se muestra en la pestaña *Perdidos*, pero únicamente para los paquetes de datos enviados por el servidor que no han llegado correctamente al cliente.

Además, se ofrecen estadísticas detalladas sobre el intercambio de paquetes de datos y control entre servidor y cliente presionando el botón *Estadísticas*: paquetes de datos y control enviados, recibidos, perdidos, etc.

Finalmente, el botón *Informe* permite generar un documento en formato *XML* que se puede copiar y pegar en un archivo para su posterior tratamiento por cualquier otra herramienta. Este informe incluye detalles temporizados sobre la evolución general del sistema, del cliente y su buffer, además del servidor.

## 7. Conclusiones

El simulador presentado en este artículo posibilita, mediante un entorno sencillo de manejar, que los usuarios puedan ver las interioridades de un servicio de alta complejidad a un nivel comprensible y de forma muy intuitiva.

Además, la herramienta se ha construido para que los alumnos modifiquen los parámetros básicos de funcionamiento y provoquen problemas de forma intencionada. Gracias a estos problemas se pueden vislumbrar los efectos que sobre los servicios de *streaming* tienen aspectos como el tamaño del buffer del cliente, el ancho de banda disponible por los

usuarios de uno de estos servicios, políticas erróneas de codificación de contenidos, etc.

El simulador es una primera etapa en el aprendizaje que debe abrir la mente del alumno a los conceptos complejos que se manejan en este tipo de servicios. Una vez entendidas las ideas principales, el profesor podrá enseñar estos conceptos con detalle, teniendo la garantía de que el alumno tiene una visión general de la tecnología.

Se encuentra accesible a través de la sección de *Tutoriales* del sitio <http://www.it.uniovi.es>, y ha estado siendo utilizado por alumnos de varias asignaturas de las titulaciones de Ingeniería Técnica Informática, Ingeniería Técnica de Telecomunicación e Ingeniería de Telecomunicación de la Universidad de Oviedo durante el último año. No ha sido posible evaluar el impacto de la herramienta desde un punto de vista comparativo, debido principalmente a la juventud de los planes de estudio y de las asignaturas en las que se utiliza. No obstante, el interés despertado por los alumnos y las altas puntuaciones obtenidas en las preguntas asociadas a este tipo de servicios vienen a defender la tesis de la conveniencia de una herramienta de este tipo.

## 8. Trabajos Futuros

A pesar de que el simulador cubre perfectamente los objetivos para los que fue diseñado, podría ser interesante incorporar ciertos parámetros de configuración y nuevos elementos que permitieran al alumno dar un segundo paso en el proceso de aprendizaje:

- Permitir al usuario decidir el tipo de servicio que se desea simular: de entre servicios de audio/vídeo bajo demanda, en directo y casi bajo demanda.
- Dejar que el usuario seleccione el protocolo de transporte a utilizar, alterando el comportamiento del servicio según se emplee un protocolo u otro. Los alumnos podrían complementar su formación observando la influencia de estos protocolos sobre los servicios de nivel de aplicación. Adicionalmente, y en función de los conocimientos del usuario, puede permitirse la

ejecución de simulaciones desde el punto de vista del nivel de transporte, mostrando detalles del protocolo empleado en cada caso.

- Permitir alterar la arquitectura de servicio: incorporando nuevos elementos como *proxies* multimedia, servidores Web para ofrecer servicios de tipo *Server-less* o *fast-start*, fuentes de información como herramientas de producción o generadores de flujos continuos, servidores redundantes, etc.
- Dejar al usuario que decida la utilización de múltiples clientes simultáneos, para ilustrar la gestión de la concurrencia que se realiza en un servicio de este tipo.

## 9. Agradecimientos

Esta investigación ha sido financiada por el operador global de comunicaciones *Telecable de Asturias S.A.U.* y el periódico *La Nueva España* en el entorno de los proyectos *NuevaMedia*, *Telemidia* y *ModelMedia*, además de contar con el apoyo del proyecto *INTEGRAMEDIA* del **Plan Nacional de I+D** (TSI2004-00979).

## 10. Referencias

- [Collins et al. 00] A. Collins, P. Neville y K. Bielaczyc, "The role of different media in designing learning environments". *International Journal of Artificial Intelligence in Education*, Número 11. (2000).
- [Wilson 96] B.G. Wilson, "Constructivist learning environments: case studies in instructional design". Englewood Cliffs. NJ. EEUU Educational Technology Publications. (1996).
- [Futó et al. 90] I. Futó y T. Gergely, "Artificial Intelligence in Simulation". Ellis Horwood. (1990).
- [Menczer et al. 01] F. Menczer y A.M. Segre, "OAMulator: A teaching resource to introduce Computer Architecture Concepts". En *Journal on Educational Resources in Computing (JERIC)*, Volume 1, Issue 4. (2001).

- [Dormida et al. 97] G. Da Dormida, D. Ponta y G. Donzellini, "Methodologies and tools for learning digital electronics". IEEE Transaction in Education, Número 40. (1997).
- [Campos et al. 02] A.M. Campos, D.F. García, J. Entrialgo y J.L. Díaz, "Simulador Educativo de un Computador Elemental basado en la arquitectura Von Neumann". XII Jornadas de Paralelismo, Lleida. (2002).
- [Buendía et al. 01] F. Buendía, J.V. Belloch y J.M. Gómez, "Development of Didactic Resources for Distance Learning based on Simulation". Computers and Education, Kluwer Academic Publication. (2001).
- [Cattaneo 02] J. Cattaneo, "Entendiendo el Streaming Media". Artículos ICTNet, número 40, Disponible en la dirección <http://www.ictnet.es/novedades/articulos/112.htm>. (2002).
- [Schulzrinne et al. 98] H. Schulzrinne, A. Rao y R. Lanphier, "Real Time Streaming Protocol (RTSP)". The Internet Society. (1998).
- [Schulzrinne et al. 00] H. Schulzrinne, S. Casner, R. Frederick y V. Jacobson, "RTP: a Transport Protocol for Real-Time Applications". The Internet Society. (2000).
- [Martín et al. 03] I.V. Martín, J.J. Alins, M. Aguilar, R.A. Vallejos, J.L. de la Cruz y J. Mata, "Modelado y Análisis transitorio de un sistema de distribución de vídeo bajo demanda". JITEL2003, Gran Canaria. (2003)
- [Pañeda et al. 05] X.G. Pañeda, A. Neira, D. Melendi, R. García y V.G. García, "Metodología para la evaluación de un servicio de tele-educación multimedia". En IE Comunicaciones, Número 1. (2005).
- [Splunter et al. 03] S. Splunter, N.J.E. Wijngaards y F.M.T. Brazier, "Structuring Agents for Adaptation". En Adaptive Agents and Multi-Agent Systems. Springer. (2003)