

# Diseño e implementación de un Sistema Interactivo de Video bajo Demanda \*

Emilio Luque <sup>1</sup>, Ana Ripoll <sup>1</sup>, Remo L. Suppi <sup>1</sup>, Porfidio Hernández <sup>1</sup>, Tomás Díez <sup>1</sup>, José A. Marco <sup>1</sup>, Fernando Cores <sup>1</sup>, Bahjat M. Qazzaz <sup>1</sup>, Rodolfo del Castillo <sup>2</sup>

<sup>1</sup>Departamento de Informática, Universitat Autònoma de Barcelona, España  
caos@uab.es

<sup>2</sup>Departamento de Informática, Universidad Nacional del Comahue, Argentina  
rolo@uncoma.edu.ar

## Resumen

El proyecto que presentamos muestra el diseño e implementación de un sistema interactivo de vídeo bajo demanda, que nos servirá como base en el desarrollo de modelos válidos que nos permitan representar sistemas multimedia, a efectos de definir un entorno donde mediante técnicas de simulación avanzadas, podamos analizar y diseñar este tipo de sistemas.

Es usual, al afrontar el diseño de este tipo de sistemas, orientar los esfuerzos únicamente, en una de las múltiples áreas que implican su resolución; nuestro objetivo integra el estudio del sistema en su conjunto: el software de sistema, la red de interconexión, el servidor de almacenamiento, servidor de procesamiento y el diseño/development del cliente. A partir de un subconjunto de las especificaciones que debe cumplir el sistema, construiremos una primera plataforma de pruebas e iremos diseñando y sintonizando los modelos en paralelo.

La construcción de prototipos, monitorización y simulación, serán las herramientas que nos permitirán afrontar con garantías el proyecto que pretendemos asumir.

Palabras clave: Multimedia, Video bajo Demanda (VoD) Calidad de Servicio (QoS)

---

\*Este trabajo fue soportado por la CICYT bajo contrato TIC98 - 0433

## 1. Introducción

El proyecto que se pretende abordar está centrado en el desarrollo de modelos que permitan representar el comportamiento de sistemas de "Video bajo Demanda (VoD)" para definir un entorno donde, mediante técnicas de simulación avanzadas, se pueda analizar y diseñar este tipo de sistemas.

Especial énfasis se realizará en los procesos de diseño y construcción de un prototipo totalmente operativo desarrollado a partir de componentes hardware no específicos y software propio y de dominio público. Este prototipo servirá para mostrar la viabilidad de las técnicas y modelos a desarrollar.

El VoD es en la actualidad uno de los servicios multimedia más importante en redes de alta velocidad. Prueba de ello es la reciente evolución de estándares y protocolos para estas redes así como los rápidos avances en las tecnologías de los dispositivos de almacenamiento. Un aspecto clave en cualquier servicio de vídeo es proporcionar una calidad de servicio (QoS) aceptable al usuario [Ste]. Esta QoS implica la consideración de aspectos como la frecuencia de pérdida de ventana, la sincronización entre sonido y vídeo, la cantidad de imágenes por segundo, etc. con la complejidad añadida que algunos de estos parámetros son difícilmente cuantificables porque dependen de la percepción subjetiva del observador. La QoS en el cliente refleja la forma en como se suministra el flujo de vídeo original desde un servidor remoto, donde los conceptos de transparencia semántica y transparencia en el tiempo caracterizan el rendimiento de los servicios de vídeo. Estos servicios requieren restricciones específicas con respecto al retardo (transparencia en el tiempo) en garantizar el servicio, más específicamente la variación del retardo (jitter), a través de la conexión, así como restricciones con respecto a la frecuencia de errores permitida (transparencia semántica) desde el servidor al cliente del vídeo. Por lo tanto, una cuestión importante en VoD es *"cómo asignar una QoS específica requerida en el cliente en una especificación de QoS para el servidor de vídeo y la red"*. Para conseguir unas prestaciones aceptables que garanticen una QoS aceptable es necesario una coordinación entre todos los componentes del sistema (servidores de ficheros, dispositivos de visualización, componentes de red, etc.). Desde este punto de vista se propone un diseño unificado que tenga en cuenta todas las componentes del sistema para garantizar la QoS requerida por las diferentes peticiones de vídeo procedentes de los clientes.

Estudios anteriores en este campo se concentran mayoritariamente en garantizar la QoS analizando los componentes del sistema individualmente, fundamentalmente en aspectos relacionados con la jerarquía de almacenamiento [Özd]. Una desventaja de este enfoque radica en que la solución óptima para una componente no garantiza la mejor solución para todo el sistema y por tanto se requiere un análisis integrado. Puesto que el objetivo de nuestro trabajo es proporcionar una QoS de extremo a extremo, se debe desarrollar un marco para examinar y evaluar cómo *los puntos débiles* en el servidor de vídeo y en la red afectan a la calidad del vídeo percibido por el cliente. Se desea además responder a cuestiones relacionadas con los requerimientos de los recursos tales como la capacidad de la red y del servidor, el número de servidores de archivo, el número de servidores de peticiones, etc. que tienen un efecto crucial sobre la viabilidad de un sistema real ya que existe un compromiso con el cliente que debe ser cumplido independientemente de la situación particular en el momento de la conexión. Estas respuestas permitirán mejorar la utilización de la capacidad de almacenamiento y/o maximizar el número de peticiones simultáneas que pueden ser soportadas por el

servidor o bien, analizar soluciones de la arquitectura de almacenamiento a través de configuraciones jerárquicas que distribuyan la carga entre los diferentes niveles.

En el diseño del servidor [Gha], se han realizado importantes esfuerzos encaminados a encontrar técnicas de almacenamiento eficientes que minimicen las sobrecargas (overheads) relacionados con la recuperación de los datos multimedia y que permitan maximizar el número de peticiones concurrentes. Por ello es necesario actuar sobre las técnicas de administración (scheduling) de disco, la colocación de los datos y las políticas de control de admisión de peticiones que ejercen un papel crítico en la reducción de estas sobrecargas.

En el desarrollo de este marco es fundamental definir un modelo del sistema de VoD que contemple aspectos complejos tales como:

- Diferentes clases de peticiones
- Diferentes estructuras de la arquitectura
- Identificación de medidas de evaluación apropiadas para realizar un correcto análisis de rendimiento del sistema
- Desarrollo de técnicas eficientes para llevar a cabo estas medidas
- Validación del diseño a través de simulación
- Comparación de los resultados del modelo y de los experimentos.

El proyecto en cuestión pretende desarrollarse en una doble vertiente:

- El desarrollo de modelos, que sirvan para representar sistemas VoD y que permitan decidir cómo diseñar e implementar este tipo de sistemas a través de entornos avanzados de simulación. La dificultad radica en que este tipo de proyecto incluye aspectos de la arquitectura de computadores, el procesamiento de sonido y vídeo, vinculación con el sistema operativo, los protocolos de red, las interfaces de los clientes, etc. Es decir aspectos de diversas disciplinas que requieren que el proyecto sea diseñado y desarrollado por personas expertas en diferentes campos de la informática.
- Mostrar la viabilidad de estas propuestas a partir del desarrollo de una aplicación real que servirá como prototipo de estudio. En este sentido, el prototipo se orientará a la introducción de los sistemas multimedia en las instancias educativas de la Universidad Autónoma de Barcelona, que sirvan como herramienta para mejorar la educación en cursos tradicionales, a la vez que incrementen la efectividad en la preparación y enseñanza de los mismos.

En la sección 2, se plantea la metodología de trabajo a llevar a cabo para la consecución de nuestro objetivo; la especificación funcional y características del sistema a desarrollar se presentan en la sección 3, siendo en la sección 4 donde se muestra la premisas adoptadas en la construcción del prototipo y finalmente las conclusiones cerrarán el trabajo presentado.

## **2. Metodología de trabajo**

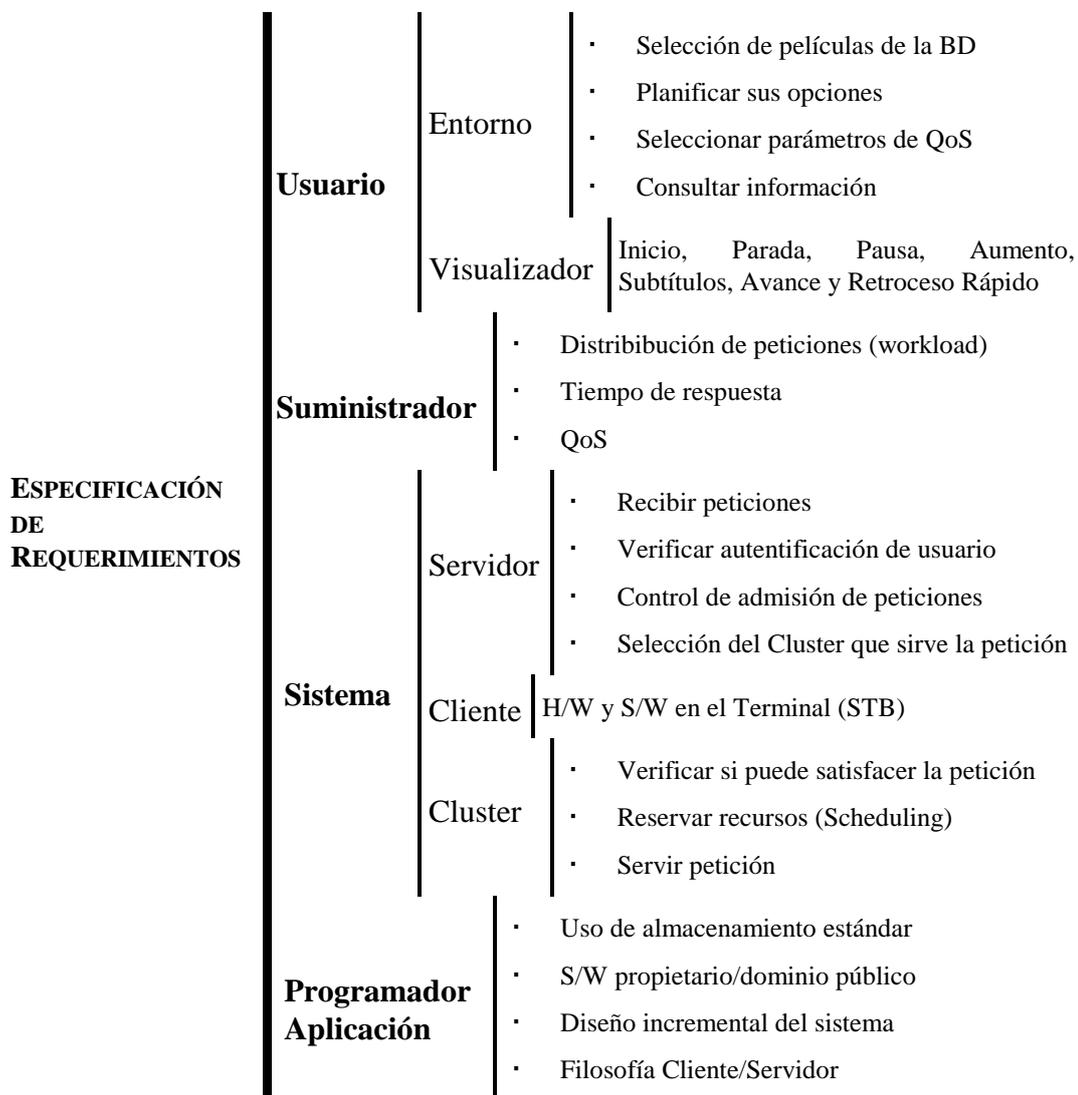
En todo proceso de diseño, una vez presentados los objetivos a conseguir mencionados en el punto anterior, se necesita establecer una cierta metodología científica que permita ir acercándose al objetivo final. Dada la complejidad del proyecto en marcha, éste se desarrollará por etapas, algunas de las cuales se realizarán simultáneamente en el tiempo y se muestran en la figura 1.

Esta figura muestra un conjunto de fases involucradas en el proceso de realización del proyecto así como la compleja interacción entre las mismas. El punto inicial del proyecto es la especificación de las características del sistema que implica definir todas



(cluster) debe determinar si la tarea encomendada por el servidor de peticiones es posible llevarla a cabo y en caso afirmativo gestionar los recursos para que dicha solicitud pueda ser satisfecha. Desde el punto de vista del cliente es necesario especificar el hardware y el software que serán necesarios para que el usuario pueda darse de alta en el servicio. Esta parte es especialmente sensible ya que generalmente el usuario dispone de un hardware sobre el cual ejecutará el software cliente de la aplicación.

Para el programador de la aplicación será necesario definir bajo que filosofía de diseño se construirá la aplicación, como se reducirá el tiempo de desarrollo (por ejemplo software de dominio público), como llevará a cabo este diseño (fases, incremental, gradual, etc.) y que criterios se tendrán en cuenta a la hora seleccionar el hardware donde se ejecutarán los servidores.



*Figura 2. Operatividad de los distintos niveles involucrados en la construcción del sistema*

La funcionalidad para el Proveedor (Suministrador) de la Conexión Servidor/Cliente estará basada en los requerimientos de la QoS mínima necesaria para que el sistema pueda funcionar con las prestaciones deseadas, el tiempo de respuesta máximo de

comunicación entre cliente/servidor y como se distribuirá la carga de trabajo entrante en el servidor de peticiones para impedir una saturación de mismo.

Una vez establecida la funcionalidad del sistema, se abren cuatro vías de actuación:

- ◆ El diseño de experimentos, orientado a obtener información de partes específicas del sistema. Esto implica adquirir experiencia en los procesos de transmisión de información comprimida a través de la red utilizando formatos del tipo MPEG y en técnicas de sincronización de sonido/vídeo para implementar funciones de avance/retroceso rápido o de "ir a" del programa de visualización.
- ◆ El diseño y la construcción de un prototipo que integre todos los elementos representativos del sistema final, desde que la película es almacenada de forma comprimida en el dispositivo de almacenamiento secundario hasta su visualización y posibilidades de interacción por parte del cliente.
- ◆ El modelado del sistema y sus componentes.
- ◆ La monitorización de los recursos del sistema y la correspondencia con los parámetros de QoS que solicita el usuario.[Lee]

Nuestra visión respecto a la función de la simulación, está en total consonancia con la idea original de predecibilidad más que de reproducibilidad del sistema, ya que interesa responder cuestiones relativas a la escalabilidad del sistema, tiempos de respuesta en situaciones críticas de fallo de nodos, en condiciones de carga baja/alta de peticiones de usuario, etc. Para el modelo de simulación del sistema VOD se ha utilizado, en una primera aproximación, una metodología de especificación "Arriba/Abajo" (top-down. Para ello se ha representado el sistema por un modelo de alto nivel como el mostrado en la figura 3. El Servidor cuenta con la CPU, su interfaz a red, un conjunto de discos y un dispositivo de intercambio, mientras que el cliente consta sólo de los 3 primeros dispositivos. El sistema se modela como un sistema de colas en la cual en una primera aproximación se utilizan modelos de colas exponenciales. Una de las características básicas de este sistema es que no es un sistema de colas abierto en la cual las tareas llegan a las colas, son procesadas y salen de la cola (en este sistema el tamaño de la cola puede crecer sin límite). En el presente caso, el sistema de colas es cerrado y existe una recirculación de las tareas que pasan por las diferentes colas y donde el tamaño máximo de las colas será igual número de tareas que existen en el sistema. Esta representación implica un grado de multiprogramación acotado, ya que el número de tareas en el sistema puede considerarse constante en un determinado intervalo de tiempo, y facilita considerablemente la simulación permitiendo además obtener resultado más cercanos a la realidad. Este tipo de modelo permitirá estimar diferentes parámetros necesarios a tener en cuenta en el diseño de software tales como: tamaño de las colas, tiempo de ciclo de una petición/tarea, escalabilidad del sistema, tiempos de respuesta y de espera, etc.

El primer prototipo del programa de simulación ha sido desarrollado basado en este modelo utilizando las librerías de SMPL [Mac]. SMPL es una extensión funcional de un lenguaje de propósito general -lenguaje host- (en nuestro caso "C"). Esta extensión toma la forma de un conjunto de funciones de librería (SMPL simulation subsystem) las cuales, junto con el lenguaje "host", configura un lenguaje de simulación orientada a eventos discretos. Un modelo SMPL es desarrollado como un programa en el lenguaje host en el que las operaciones de simulación son realizadas mediante llamadas a funciones de subsistema de simulación. Esta opción provee una capacidad de simulación para definir modelos de media/elevada complejidad.

Con respecto a la monitorización debe prestarse especial atención ya que es una fase crítica durante la especificación del proyecto. Asegurar la calidad de servicio (QoS) supone implicar a todos los elementos que participan en el transporte y procesamiento del flujo multimedia. Esto requiere pruebas de control de admisión y reserva de recursos seguidos de una cuidadosa coordinación de los planificadores de hilos (threads) y discos, control de flujo en la red, etc. y finalmente, una negociación de la QoS.

La monitorización y la simulación serán los elementos de juicio a la hora de definir y configurar el sistema final. El entorno de simulación a desarrollar, última fase a implementar de nuestro proyecto, pretende servir de ayuda al diseñador de sistemas de vídeo bajo demanda a efectos de analizar y diseñar este tipo de sistemas estudiando la viabilidad de las propuestas de requerimientos introducidas por el diseñador del sistema. [Nah].

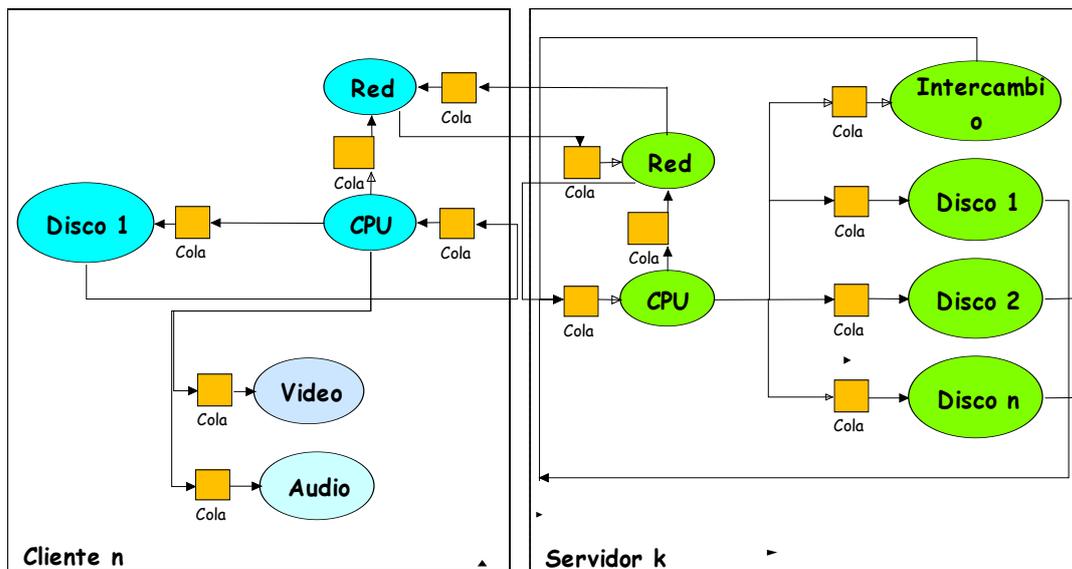


Figura 3: Modelo de simulación del Sistema VoD

#### 4. Construcción del prototipo

La figura 4, muestra una visión "End to End" del sistema prototipo a construir el cual incluirá como punto de partida los elementos que se encuentran dibujados en línea continua: servidor de peticiones/almacenamiento, dispositivos de almacenamiento secundario/terciario y clientes. Para esta definición se ha analizado diferentes opciones existentes tanto comerciales como de dominio público. [Lok][Dis]

El diseño y desarrollo de la aplicación VoD ha sido organizado como una estructura cliente servidor. Este modelo de procesamiento presenta un conjunto de ventajas que lo hacen más adecuado que otras alternativas de proceso distribuido de datos.

El objetivo principal de una aplicación de estas características es dividir la aplicación en tareas y poner cada una de estas en la plataforma más adecuada, es decir donde pueda ser manejada con mayor eficacia. Esto suele significar que se sitúe en la máquina del usuario el proceso necesario para la presentación, y en el servidor, la gestión y almacenamiento de la información. El elemento básico de este sistema es la red de interconexión. El software del servidor acepta peticiones de datos del software del cliente y los devuelve a través de la red.

Este modelo, por lo tanto, implica un procesamiento cooperativo de solicitudes emitidas por un cliente a un servidor que las procesa y devuelve los resultados al cliente. El proceso se inicia en realidad en el cliente y es controlado parcialmente por éste, pero

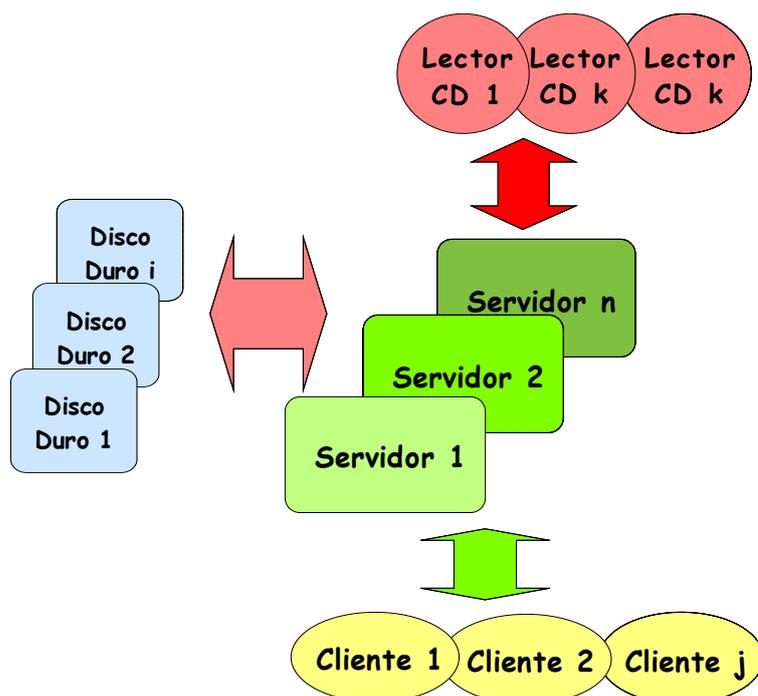


Figura 4: Visión del prototipo del sistema básico ("end to end")

no de una forma maestro-trabajador. Por el contrario, tanto cliente como servidor cooperan para ejecutar con éxito una aplicación.

Este tipo de modelo es el óptimo cuándo los usuarios no coinciden físicamente en un entorno cercano de utilización de los datos. La otra opción es replicación de la información con todos los problemas de gestión, administración y almacenamiento que esto genera. El proceso cliente/servidor es un proceso distribuido. Los usuarios, aplicaciones y recursos se encuentran distribuidos en respuesta a los requisitos del negocio y quedan enlazados por una LAN o WAN o por varias subredes. Existen una serie de características que hacen que esta organización sea adecuada para este tipo de aplicaciones:

- **Confianza:** al tener aplicaciones fáciles de usar para los usuarios en su propio sistema ya que existe un alto grado de control sobre la aplicación y su administración es muy simple.
- **Centralización de la información:** ya que habilita una gestión corporativa para mantener un control global de la inversión en sistemas de información e informática. Al mismo tiempo, alivia a los departamentos individuales y divisiones de gran parte de la carga de mantener servicios de computación sofisticados, permitiéndoles elegir cualquier tipo de máquina e interfaz que necesiten para acceder a los datos y la información.
- **Sistema abierto y modular.**

#### 4.1. Desarrollo

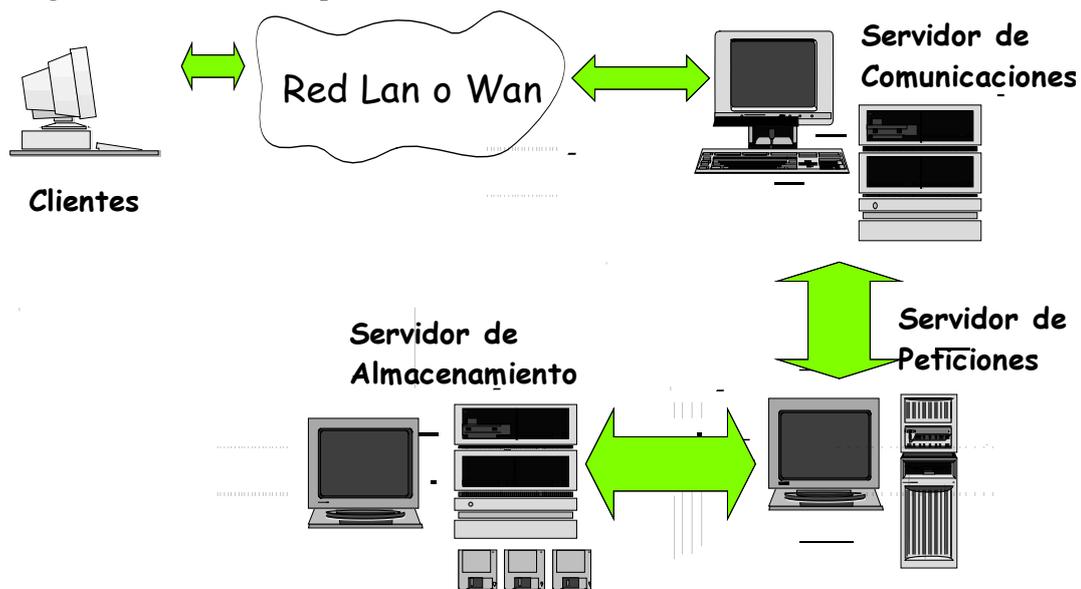
En nuestra aplicación tanto cliente como el servidor utilizan como software básico un sistema operativo (Linux - SuSe 6.4) ejecutándose sobre la plataforma hardware. Si bien las plataformas y los sistemas operativos del cliente/servidor pueden ser diferentes, en nuestra primera versión se ha tomado la decisión de diseñar la aplicación sobre un sistema Open Source debido a las características del proyecto y a la voluntad de ahorrar

costos en el diseño y desarrollo de las mismas. Como subsistema de comunicaciones se ha escogido TCP/IP para que la aplicación pueda ser analizada y evaluada dentro de un sistema heterogéneo de acceso a la información.

Una visión más precisa de la aplicación indica que el Sistema Servidor debe ser dividido en un conjunto de tres servidores con funciones específicas:

- **Servidor de almacenamiento:** Permitirá administrar y gestionar la información almacenada en función a políticas que contemplen múltiples peticiones en tiempo real, bajo demanda y con gestión de múltiples dispositivos físicos de almacenamiento.
- **Servidor de peticiones:** analizará, gestionará y tramitará las peticiones de los clientes e intercederá sobre el servidor de almacenamiento para obtener la información solicitada. Este servidor tiene comunicación directa con el servidor de almacenamiento pero no con los clientes. La comunicación con estos se realiza a través de un servidor de comunicaciones.
- **Servidor de comunicaciones:** Analizará, gestionará y atenderá las comunicaciones entre los clientes y el servidor de peticiones. Su objetivo es centralizar y gestionar de forma eficiente todas las comunicaciones que lleguen(salgan) al(desde el) servidor de peticiones utilizando políticas de comunicaciones basadas en demanda bajo prioridades.

La figura 5 muestra un esquema del desarrollo realizado.



*Figura 5: Arquitectura del sistema VoD diseñado*

Para la construcción del prototipo se han especificado los siguientes requerimientos:

- Formato de la información: mpeg3
- Servidores multiproceso con comunicación externa TCP/IP-Streams e interna TCP/IP-Datagrams.
- Código Open Source desarrollado con GNU Compiler bajo Linux SuSe 6.4, Simple DirectMedia Layer Libraries (SDL<sup>®</sup> Library) y SMPEG<sup>®</sup> Library para la gestión de formato mpeg y Gtk<sup>®</sup> (Gnome Project) para la gestión de interfaces gráficas del cliente. [Lok]

En la figura 6 describe la estructura de la versión actual del prototipo diseñado en el cual no se incluye el Servidor de Comunicaciones de la figura anterior y donde le

Servidor de Peticiones y el de Almacenamiento coinciden en una misma máquina. El cliente está formado por 3 procesos independientes. El proceso *Control de Conexión* centraliza, para el cliente, toda las comunicaciones y es el responsable de negociar con el servidor las peticiones de usuario y/o enviar la información que proviene del servidor

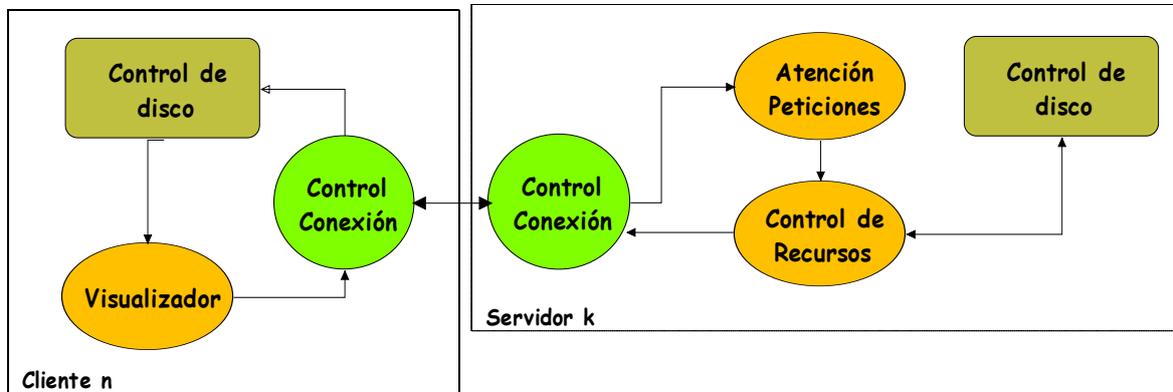


Figura 6: Estructura del software del sistema VoD

al proceso *Control de Disco*. Este proceso *Control de Conexión* es el encargado de crear, como procesos hijos, el proceso *Control de Disco* y el proceso *Visualizador*. El proceso *Control de Disco* controla y administra el espacio de disco donde se almacena la información que viene del servidor y gestiona las posibles peticiones de usuario que pueden ser resueltas con la información disponible. El proceso *Visualizador* es el encargado de descomprimir la información (sonido y vídeo) recuperada del disco y visualizarla/reproducirla en los dispositivos del cliente.

El software del servidor ha sido diseñado en 4 procesos principales. Los procesos *Control de Conexión* y *Control de Disco* tienen básicamente la misma funcionalidad que en el proceso cliente. El proceso *Atención de Peticiones* controla y gestiona las peticiones del usuario e interactúa con el proceso *Control de Recursos* para planificar y administrar las peticiones recibidas. El proceso *Control de Recursos* debe gestionar los recursos del servidor para satisfacer todas las demandas de servicio realizadas teniendo en cuenta las disponibilidades de los dispositivos físicos.

Una de las consideraciones de diseño en las cuales se ha prestado una cuidadosa atención es en la arquitectura de software de visualización. La evolución en formatos/estándares y de los sistemas operativos con respecto a las funciones multimedia hacen que un diseño quede obsoleto en un intervalo muy corto de tiempo y/o existan incompatibilidades en versiones diferentes del sistema operativo por lo cual la decisión ha sido crear una arquitectura de visualización por capas donde es muy simple cambiar una capa para adecuarla a un nuevo formato/estándar o agregar nueva funcionalidad. Esto impide que el software pueda adecuarse automáticamente al hardware disponible (no existe una integración entre aplicación y sistema operativo) siendo el usuario el que debe ajustar el tamaño de la ventana, la velocidad de reproducción, la calidad y el factor de compresión a través de un conjunto de diálogos (donde existirán valores por defecto) pero como contrapartida el software desarrollado tiene una vigencia mayor y es fácilmente modificable para incluir nuevas características.

La arquitectura de la herramienta de visualización se compone básicamente de tres módulos con relación entre sí:

- Interfaz de usuario: encargada de controlar y gestionar la interacción con el usuario.

- Gestión de la compresión: responsable de obtener los streams de datos del disco, descomprimir la información de acuerdo al formato y generar los streams de audio y vídeo para su reproducción.
- Administración de dispositivos: tiene por fin controlar los dispositivos de audio y vídeo, gestionar las secuencias de datos que recibe del Gestor de Compresión y reproducirlas en los dispositivos físicos de acuerdo a las opciones indicadas por el usuario.

## Conclusiones

El proyecto presentado pretende ser una aproximación al diseño y desarrollo de sistemas en el campo de los sistemas multimedia. La experiencia del grupo en el diseño de modelos para simulación, y en los temas relacionados con las áreas involucradas en el proyecto de VoD nos permitirá afrontar el modelado de este sistema complejo, con las garantías de éxito demandadas por las entidades que financian el mismo. En la actualidad el proyecto ya cuenta con una versión operativa del software del cliente/servidor, un modelo de simulación básico y una definición preliminar de las medidas que afectan a la QoS y que deben ser consideradas en la instrumentación tanto del cliente como del servidor. El prototipo se ha construido a partir de componentes hardware que se pueden conseguir fácilmente en el mercado y con software propio y de dominio público.

## Referencias

- [Lee] C. Lee, M. Potkonjak and W. H. Mangione-Smith, " MediaBench: A Tool for Evaluating and Synthesizing Multimedia and Communications Systems" Proc. IEEE/ACM International Symposium on Microarchitecture, December 1997, pp. 330-335.
- [Gha] S. Ghandeharizadeh, "Design and Implementation of Multimedia Servers" Handbook of Multimedia Computing. CRC Press LLC. 1999, Pp.557-578.
- [Lok] Smpg Libraries. <http://www.lokigames.com/development/smpeg.php3>
- [Mac] M. S. MacDougall, "Simulating Computer Systems" Mit Press. 1990
- [Nah] K. Nahrstedt and L. Quiao, "A Tuning System for Distributed Media Applications" Technical Report UIUCDCS-R-96-1958, University of Illinois at Urbana Champaign, July 1996.
- [Özd] B. Özden, A. Biliris, R. Rastogi and A. Silberschatz, " A Disk- Based storageArchitecture for Movie on Demand Servers" Information Systems, Vol. 20, No. 6, 1995, pp. 465 - 482.
- [Dis] "A Distributed Real-Time MPEG Video Audio Player. Available via anonymous FTP from <ftp://ftp.cse.ogi.edu/pub/dsrg/Player>.

[Ste] R. Steinmetz and K. Nahrstedt, " Multimedia: Computing Communications and Applications", Prentice Hall, 1995.