

Un sistema P2P-VoD tolerante a fallos de red

Javier Balladini¹, Eduardo Grosclaude¹, Remo Suppi², and Emilio Luque²

¹ Departamento de Ciencias de la Computación, Universidad Nacional del Comahue, Argentina, {jballadi, oso}@uncoma.edu.ar

² Departamento de Arquitectura de Computadores y Sistemas Operativos, Universidad Autónoma de Barcelona, España, {remo.suppi, emilio.luque}@uab.es

Resumen La mayoría de los sistemas P2P de video bajo demanda (VoD, Video-on-Demand) no consideran cuidadosamente todos los problemas que pueden surgir al desplegar sus servicios en una red con alta probabilidad de fallos como Internet. El resultado es que ellos no pueden garantizar servicios de visualización de videos libre de interrupciones. Para solucionar este problema, proponemos una arquitectura y esquema de tolerancia a fallos de red, fácilmente adaptable a los sistemas P2P-VoD ya desarrollados. La solución utiliza técnicas de: gestión de transmisiones basada en el estado de las comunicaciones, recepción de datos desde múltiples orígenes, selección adecuada de nodos servidores, mecanismos de migración de servicios, y reservación de recursos. A partir de los buenos resultados obtenidos con la propuesta previa de un sistema de VoD cliente-servidor tolerante a fallos de red, este trabajo adapta y extiende la funcionalidad para sistemas de VoD con arquitecturas P2P.

1. Introducción

Con el actual estilo de vida digital, la industria de entretenimientos y medios de comunicación ha pasado a ser un sector económicamente relevante. El *streaming* de video se ha vuelto una de las actividades más populares de Internet, favorecido por las mejoras tecnológicas de las redes de comunicaciones. La gran cantidad de clientes y el alto requerimiento de ancho de banda de red de estos servicios han impulsado un elevado número y variedad de estudios en el área.

El *streaming* de video puede ser clasificado en: *streaming* en vivo y video bajo demanda (VoD, *Video on Demand*). En *streaming* en vivo, los servidores transmiten programas en vivo o televisivos, y los usuarios visualizan los contenidos secuencialmente desde el momento en que acceden al servicio. Dentro del *streaming* en vivo se encuentra el *streaming* en vivo interactivo, tal como telefonía por Internet y video conferencia, con restricciones temporales aún mayores. A diferencia del *streaming* en vivo, el VoD permite a los usuarios reproducir un video, seleccionado de un gran conjunto de videos prealmacenados, en cualquier momento y usando comandos interactivos típicos de un reproductor de DVD.

Los sistemas *peer-to-peer* (P2P), donde los nodos actúan a la vez como clientes y servidores, han sido exitosos en distribuir archivos a una gran cantidad de usuarios. Los sistemas P2P son también ampliamente usados para distribución de video, incluyendo descargas de video completas antes de su reproducción

y *streaming* de video en vivo (tal como Coolstreaming). Recientemente, se han diseñado nuevos sistemas para soportar VoD usando P2P en Internet.

En Internet, los anchos de banda de las comunicaciones fluctúan debido a la congestión. La congestión puede producirse ya sea por un aumento del tráfico o por una falla física como la caída de enlaces o encaminadores. Cuando un enlace se cae, los algoritmos de encaminamiento derivan el tráfico hacia otros enlaces, pudiendo producir congestión en los enlaces que han recibido la derivación del tráfico. Es normal que, a pesar de la protección del encaminamiento IP, las caídas de enlaces y encaminadores sean seguidas por largos períodos de inestabilidad en el encaminamiento de paquetes, produciendo múltiples descartes de los mismos a causa de reenvíos por caminos incorrectos [1]. Por lo tanto, para que el sistema de VoD provea un servicio de *alta calidad*, es decir, reproducción libre de interrupciones y calidad de video constante³, debe tolerar fluctuaciones del ancho de banda de las comunicaciones y destinos temporalmente inaccesibles desde ciertos orígenes. Dado cierto video y una determinada comunicación entre un nodo servidor y uno cliente, decimos que ha ocurrido un *fallo de red* cuando el ancho de banda de la comunicación se ha vuelto insuficiente para garantizar un servicio (de entrega del video) de alta calidad.

Para evitar estos problemas de la red, pueden utilizarse diversas técnicas: gestión de transmisiones basada en el estado de las comunicaciones, recepción de datos desde múltiples orígenes, selección adecuada de nodos servidores, mecanismos de migración de servicios, y reservación de recursos. Sin embargo, los sistemas P2P-VOD actuales, como [2,3], presentan diseños deficientes de dichas técnicas que impiden tolerar adecuadamente los fallos de red descriptos.

Inicialmente, nuestro grupo ha trabajado en tolerancia a fallos de red para sistemas de VoD de alta calidad basados en arquitecturas cliente-servidor. La decisión de adoptar una arquitectura simple permitió acotar el problema, facilitando el diseño y verificación de las técnicas utilizadas. Basándonos en la experiencia y los buenos resultados obtenidos, nos hemos planteado adaptar y extender la propuesta para soportar arquitecturas P2P. De esta manera, nuestra línea de investigación actual se centra en el diseño de un sistema P2P-VoD en Internet, tolerante a fallos de red, que gestione adecuadamente la degradación de las comunicaciones para brindar un servicio de alta calidad. El presente trabajo describe el diseño de una arquitectura P2P-VoD con un esquema de tolerancia a fallos de red, que puede ser implantado en la mayoría de los sistemas P2P-VoD (incluyendo los más actuales), permitiendo un aumento sustancial de sus prestaciones, que se traducen en una mayor satisfacción del usuario del sistema.

El resto de este artículo se organiza como sigue: en la sección 2 se discuten los trabajos relacionados. En la sección 3 se describe la arquitectura del sistema y se discuten los conceptos básicos y resultados de las técnicas propuestas. Finalmente, en la sección 4, se exponen las conclusiones y trabajos futuros.

³ la calidad del video es mantenida durante el tiempo de servicio, sin producirse una degradación de la calidad del video visualizado.

2. Trabajos relacionados

Algunas técnicas propuestas en relación a la tolerancia a fallos de sistemas de VoD son la recuperación de datos usando redundancia de datos, la transmisión de datos desde múltiples servidores con reasignación de la tasa de envío, y la disminución de la tasa de envío degradando la calidad de la imagen. Utilizando únicamente estas técnicas, si existiesen caídas de servidores o las comunicaciones se deteriorasen considerablemente, la visualización del contenido multimedia sería de todas formas interrumpida. La solución es utilizar la migración de servicio, técnica que permite variar el punto de origen de las transmisiones y así evitar fallos de red. Se han propuesto mecanismos de continuación de servicio para conexiones TCP, por ejemplo [4], que realizan migraciones del servicio cambiando el punto de origen de la transmisión ante problemas de la red. Aunque este esquema es muy relevante en el área del comercio electrónico, su mecanismo de detección de fallos no tiene en cuenta las restricciones temporales y los altos requerimientos de ancho de banda de red de los videos.

Hay varios trabajos en detección de fallos de red para migración de servicios en sistemas multimedia. Algunos trabajos, como [5], proponen técnicas de detección de fallos basadas en el retardo de paquetes, adecuadas para *streaming* en vivo pero inadecuadas para VoD; en VoD, la medida que importa es el ancho de banda y no el retardo de paquetes. En [6,7] se proponen esquemas de detección de fallos para *streaming* en vivo basados en anchos de banda. Los clientes reciben transmisiones desde múltiples nodos servidores y, usando un protocolo dirigido por el cliente, cada servidor entrega una parte de la tasa de transmisión total del video. La asignación de tasas es dinámica y se basa en las condiciones de los caminos de comunicación. Aunque estos trabajos no proponen la migración de servicio, su mecanismo de detección de fallos dirigido por el cliente tampoco es aplicable a VoD. Para evitar interrupciones en la visualización, un sistema de VoD de alta calidad planifica la entrega de los datos de video a los clientes, priorizando la atención de clientes con mayor urgencia de recepción de datos del video. Así, un cliente no puede saber si la comunicación es deficiente o el servidor ha decidido priorizar las transmisiones a otros clientes con mayor urgencia. Los esquemas implantados en los sistemas de VoD [8,9], tampoco son adecuados porque, al igual que los anteriores, son los clientes los encargados de detectar los fallos de red (controlando disminuciones de la tasa de recepción de datos).

Algo similar ocurre con trabajos, como [10], que utilizan mecanismos de *heart-beat* para proveer tolerancia a fallos de la red en un sistema de VoD. Mediante este esquema, se puede detectar la pérdida total de comunicación pero, sin embargo, no es posible detectar comunicaciones con anchos de banda insuficientes. Algunos trabajos en tolerancia a fallos de sistemas de VoD, como [11], se ocupan de las caídas de nodos pero no contemplan la degradación de las comunicaciones.

Para tolerar fallos de red también es importante la elección de los mejores nodos servidores, dentro de un número de posibles candidatos, para servir a un determinado nodo cliente. Sin embargo, los trabajos existentes no satisfacen todos los requerimientos de un sistema P2P-VoD tolerante a fallos de red. Por ejemplo, [3,2] no tienen una política definida que tenga en cuenta el estado de las

comunicación para preferir a un *peer* sobre otro. En [12] se presenta un esquema que selecciona a aquellos *peers* con más alto ancho de banda de comunicación. No obstante, no considera la elección de caminos de comunicación diversificados para evitar que un mismo fallo de red afecte a muchas conexiones.

3. Arquitectura del sistema

Nuestro sistema de VoD asume una arquitectura P2P en Internet, donde los *peers* reciben simultáneamente el contenido desde múltiples orígenes y transmiten contenido a múltiples destinos. Los *peers*-origen son *peers* que solo cumplen la función de servidor y son el origen de los contenidos; entre ellos se encuentra distribuido el catálogo completo de videos. Se forman grupos de colaboración de *peers*, de tamaño limitado, interesados en un mismo contenido y generalmente con puntos de reproducción muy próximos. Entre ellos se establecen conexiones con puntos de control y datos de video con el fin de intercambiar segmentos de video. Las descargas desde múltiples orígenes permite: balancear la carga entre nodos servidores, soportar más fácilmente la caída de alguno de ellos, y diversificar caminos de conexión para tolerar fallos de red. Además de estas conexiones fijas, hay conexiones temporales, que son normalmente transferencias de corta duración dedicadas a obtener segmentos de datos (posiblemente menos urgentes) que no se encuentran en sus nodos servidores con conexiones fijas. El tiempo de establecimiento de las mismas es menor, debido a que la elección del nodo servidor está simplificada por no tenerse en cuenta la diversificación de caminos. Para evitar una sobrecarga por gestión de datos de control, un nodo cliente puede tener un número máximo de conexiones fijas (que establecimos en 5 aunque podría ser otro) y temporales.

Las comunicaciones de datos de video tienen un control de congestión TCP-Friendly, asegurando que nuestra aplicación pueda coexistir adecuadamente con otras aplicaciones de Internet. Esto implica que los anchos de banda disponibles de las comunicaciones pueden cambiar significativamente durante una sesión. Nuestra solución se basa en reservación de recursos. Cada nodo cliente establece conexiones fijas y negocia una mínima tasa de envío de datos con sus nodos servidores. Cuando una comunicación deficiente impide a un nodo servidor cumplir con la tasa pactada, el nodo cliente podrá renegociar las tasas con sus nodos servidores o migrar el servicio, reemplazando al nodo servidor afectado por uno sustituto. Las comunicaciones de control, entre un nodo cliente y uno servidor, se realizan por una conexión TCP denominada *canal de control del stream*.

Para flexibilizar la adopción de nuestra propuesta a los sistemas de VoD existentes, no se impone una estrategia de conformación de grupos de colaboración, ni de planificación de segmentos, ni de replicación y descubrimiento de contenidos. A continuación se presenta la arquitectura de módulos de los *peers*, se describe la técnica de detección y recuperación de fallos, la gestión y sincronización del *stream* multi-origen, y el establecimiento de conexiones dentro de los grupos de colaboración.

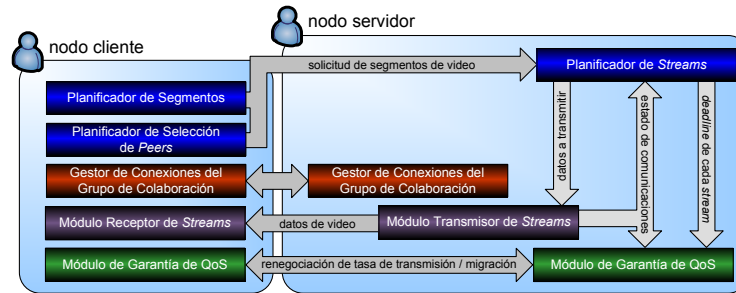


Figura 1. Arquitectura de un peer: módulos intervinientes cuando actúa como servidor y cliente

3.1. Peers

La arquitectura interna de los *peers* en sus funciones de servidor y cliente es mostrada en la figura 1. Cuatro módulos implementan la funcionalidad de un nodo servidor: un *módulo transmisor de streams* (STM, *Stream Transmission Module*), un *planificador de streams* (SS, *Stream Scheduler*), un *módulo de garantía de la calidad del servicio* (QoSAM, *Quality of Service Assurance Module*), y un *gestor de conexiones del grupo de colaboración* (NCM, *Neighborhood Connection Manager*). Estos módulos interactúan con módulos del nodo cliente: un *módulo receptor de streams* (SRM, *Stream Receiver Module*), un *planificador de segmentos* (SeS, *Segment Scheduler*), un *planificador de selección de peers* (PSS, *Peer Selection Scheduler*), y la contraparte del QoSAM y NCM.

El NCM establece las conexiones dentro de un grupo de colaboración. El SeS utiliza estrategias de *prefetching* (y posiblemente también de *batching* y *patching*) para determinar los segmentos de video a solicitar a determinados nodos servidores. Estos nodos servidores se seleccionan mediante el PSS. Normalmente se selecciona al nodo servidor que tiene menos segmentos útiles para el nodo cliente, prefiriendo a aquellos con los que tiene conexiones fijas. Esta estrategia permite no sobrecargar a aquellos *peers* con muchos segmentos útiles, como por ejemplo los *peers*-origen.

Una técnica de *prefetching* provee video a los nodos clientes por adelantado cuando los caminos de comunicación están infrautilizados, permitiendo a estos nodos soportar momentos de bajo ancho de banda de las comunicaciones [13]. Cuando el tráfico de datos hacia un nodo cliente se decrementa, el nodo servidor puede usar este ancho de banda extra con otros nodos clientes. El módulo SS (basado en nuestro trabajo previo [14]) recibe las solicitudes de segmentos y realiza las transmisiones mediante una planificación que tiene en consideración el ancho de banda de las comunicaciones y que prioriza a aquellos nodos clientes con mayor urgencia de datos de video. Así, se intenta evitar interrupciones en la visualización de los contenidos multimedia. Sin embargo, si el ancho de banda de comunicación con un nodo cliente permanece insuficiente durante cierto tiempo, dicho nodo podría consumir todos los datos del *buffer* provocando una

interrupción de la reproducción del video por *buffer underflow*.

Para resolver el problema de *buffer underflow* y lograr un servicio de reproducción libre de interrupciones, se incluye el componente QoSAM. Este módulo se encarga de determinar si el ancho de banda de comunicación con un determinado nodo cliente es suficiente o no para continuar garantizando la tasa de transferencia de datos pactada. Cuando el QoSAM detecta un fallo de red, éste se comunica al nodo cliente, quien puede renegociar la tasa de transmisión o acordar la migración del servicio. La migración de servicio tiene como fin cambiar el origen de la transmisión para intentar recibir el contenido multimedia por un camino alternativo que eluda la congestión o la interrupción total de la comunicación.

El STM (parcialmente publicado en [15]) informa al SS y QoSAM el ancho de banda de cada conexión, que obtiene de su protocolo de control de congestión. Con esta información (y más), el SS planifica la entrega de los videos a los nodos clientes y el QoSAM detecta fallos de red en las comunicaciones. El SS utiliza el transporte de datos de los videos ofrecido por el STM y cuya recepción es gestionada por el SRM. Las características de este transporte de datos son: confiable, TCP-Friendly, y con soporte de selección de datos tardíos (algunos paquetes enviados a nodos clientes pueden ser eliminados si aún no están en el cable). Para soportar muchos nodos clientes/conexiones y alta carga de tráfico, el STM usa un mínimo de recursos por conexión y hace un uso máximo del ancho de banda de salida de red disponible (evitando el desperdicio de este recurso crítico). Este módulo se implementó en un sistema operativo de propósito general (Linux) y, para independizarse de una versión de kernel específica, se desarrolló completamente en espacio de usuario.

3.2. Detección y recuperación de fallos

La *detección de fallos* es efectuada por el QoSAM y utiliza dos mecanismos de detección que colaboran para brindar tolerancia ante fallos de red o caídas de nodos servidores. El *mecanismo principal* de detección está ubicado en el nodo servidor y se encarga de detectar cualquier tipo de fallo de red. Ante la detección de un fallo, éste se comunica al nodo cliente. Un *mecanismo de protección*, ubicado en el nodo cliente, detecta la caída del nodo servidor o una pérdida total (o casi total) de la comunicación (que impide, al mecanismo principal del nodo servidor, comunicar el fallo al nodo cliente).

El mecanismo de protección consiste en enviar, cada cierto tiempo y por el canal de control del *stream*, mensajes (*heartbeat*) al otro extremo preguntando si se encuentra “vivo”. El mecanismo principal funciona de la siguiente manera. Cada *peer* que sirve a otro *peer* acuerda garantizar un mínimo ancho de banda de comunicación. Las muestras de ancho de banda, suministradas por el STM, son utilizadas para estimar el ancho de banda de la comunicación. Cuando este ancho de banda baja del acordado, se emite un preaviso al nodo cliente para que inicie la planificación de su recuperación. Una vez transcurrido un tiempo prudencial, el nodo servidor confirma el fallo al nodo cliente o envía una negativa del fallo según corresponda.

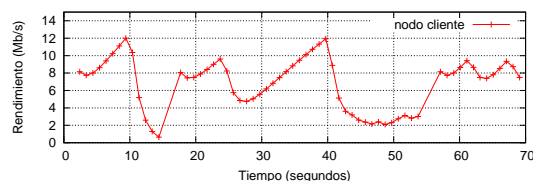


Figura 2. Comportamiento del mecanismo de detección y recuperación de fallos de red

El proceso de *recuperación de fallos* se encuentra solapado al de detección, ya que el mismo comienza cuando el nodo cliente recibe el preaviso de fallo, y no cuando el fallo ha sido confirmado. Este solapamiento produce una ganancia de tiempo determinante para poder mantener la calidad de servicio en la entrega del contenido. Al momento de arribar un preaviso, el nodo cliente busca y reserva recursos en un nodo servidor alternativo por si debe migrarse el servicio. Si luego llega una negativa del fallo, el nodo cliente cancela la reserva de recursos efectuada al nodo servidor alternativo. En caso de confirmación del fallo, el nodo cliente renegocia una tasa de transmisión menor con el nodo servidor afectado por el fallo, y una tasa de transmisión mayor con los otros nodos servidores. Sin embargo, una renegociación sólo podría ser considerada si el nuevo mínimo garantizado por el nodo servidor afectado por el fallo es mayor a cierto valor límite (por ejemplo, un 10 % de la tasa máxima de bits del video) y, el resto de los nodos servidores aceptan una renegociación por una tasa de transmisión mayor. Si se llega a un acuerdo, se cancela la reserva de recursos realizada al nodo servidor alternativo; por el contrario, si no hay acuerdo, se migra el servicio.

Experimentación. Para mostrar la efectividad de las técnicas propuestas de detección y recuperación de fallos de red, se realizó un experimento simple (por cuestiones de espacio y claridad) utilizando el simulador de red Ns-2. El experimento se conforma de dos nodos servidores y un nodo cliente, donde el nodo cliente recibe contenido desde un único nodo servidor a la vez. En la figura 2 se muestra la curva del rendimiento supuesto por los nodos servidores para el nodo cliente. En el segundo 10, una caída de un enlace (de 20 segundos de duración) impide la comunicación del nodo cliente con su nodo servidor. El mecanismo de protección del nodo cliente detecta el fallo y el servicio se migra; note que el mecanismo principal del nodo servidor nunca pudo notificar el fallo al cliente debido a la pérdida total de comunicación. La recuperación se hace efectiva en el tiempo 15,25. En el tiempo 40 la comunicación del nodo cliente es afectada por congestión de la red. El nodo servidor espera un tiempo prudencial y, como la comunicación no mejora, ordena migrar el servicio antes de que el cliente sufra interrupciones en la visualización del video.

3.3. Gestión y sincronización del *stream* multi-origen

Los *peers*, a través del SeS, periódicamente determinan los segmentos de video a descargar de nodos servidores, de acuerdo a alguna política adopta-

da de *prefetching* como: secuencial, aleatoria, local-rarest, híbrida “secuencial y local-rarest” [3], híbrida “secuencial y asociaciones basadas en contenidos entre diferentes segmentos de video” [2], etc. Para evitar desperdiciar el ancho de banda de comunicación reservado, el nodo cliente debe entregar nuevas solicitudes de segmentos antes de que los nodos servidores finalicen la transmisión de los anteriores. Las conexiones temporales no deberían ser extremadamente cortas para evitar sobrecarga por control (establecimiento de conexión, negociación de ancho de banda, etc); así, si el tamaño de los segmentos es pequeño, deberían solicitarse más de uno por vez.

Cuando un *peer* cambia su punto de reproducción o su estado (reproducción-normal, reproducción-rápida, prefetch y pausa), debe indicarlo a sus nodos servidores⁴. El módulo SS de un nodo servidor calcula el *deadline* de reproducción del siguiente segmento a transmitir a un nodo cliente, basándose en la siguiente ecuación: $deadline = tpmss - (upr + ttdupr)$. En esta ecuación, *tpmss* es el tiempo de reproducción asociado al primer marco del siguiente segmento de video a transmitir al nodo cliente, *upr* es el último punto de reproducción informado, y *ttdupr* es el tiempo transcurrido desde que se estableció el último punto de reproducción informado. Basándose en el estado de reproducción y en el *deadline*, el SS determina la urgencia de datos de sus nodos clientes y planifica la entrega del contenido multimedia (ver resultados experimentales en [14]). A medida que se van efectivizando las transmisiones, el SS actualiza la lista correspondiente de segmentos a transmitir eliminando los segmentos ya entregados.

Cuando se realiza una renegociación del ancho de banda o migración del servicio, debido a que la comunicación de un nodo servidor con un nodo cliente se deterioró, el nodo cliente redistribuirá la solicitud de segmentos de video (aún no entregados) a todos sus nodos servidores. Esta nueva distribución de solicitudes de segmentos tiene como fin redistribuir los segmentos asignados al nodo servidor cuya comunicación fue afectada.

3.4. Establecimiento de conexiones en grupos de colaboración

Dentro de un grupo de colaboración, la cantidad de conexiones fijas en paralelo que un nodo cliente puede tener, están limitas para evitar sobrecargas por gestión de datos de control. De esta manera, una vez determinado el grupo de colaboración al que pertenecerá cierto *peer*, éste debe seleccionar con qué nodos servidores establecerá conexiones fijas para descargar segmentos de video. Esta selección es llevada a cabo por el NCM. Este módulo también se encarga de implementar un control de admisión de solicitudes de conexiones temporales efectuadas por nodos clientes. El control de admisión acepta o rechaza las peticiones de acuerdo a una política que considera la cantidad de conexiones disponibles y el *deadline* de los segmentos solicitados.

Para establecer las conexiones fijas se define el siguiente esquema. Se seleccionan aquellos cinco *peers* con diversificación de caminos y máximo ancho de

⁴ Según la política específica de cada sistema de VoD, ciertos cambios de estado o punto de reproducción podrían hacer trasladar al peer a otro grupo de peers colaboradores.

banda de comunicación, prefiriendo *peers-no-origen* a *peers-origen*. Luego, en cada nodo servidor, se debe reservar una tasa de transmisión (menor o igual a la ofrecida) tal que la suma de las tasas individuales sea igual a la tasa máxima del video. Si son todos *peers-no-origen*, en cada uno se reserva una tasa de transmisión proporcional a la tasa que ofrece, respecto a la tasa total ofrecida por el grupo completo (de 5 *peers*). En cambio, si algunos son *peers-origen*, se reserva en cada *peer-origen* un 10 % de la tasa necesaria, y el resto se asigna a los *peers-no-origen*, siendo el valor asignado a cada uno proporcional a la tasa individual ofrecida respecto a la tasa ofrecida por todos los *peers-no-origen* juntos; si aún no se ha cubierto la demanda, se aumenta proporcionalmente la asignación realizada a cada *peer-origen*. En cuanto sea factible, es conveniente reemplazar los *peers-origen* por *peers-no-origen*.

Este esquema, necesita disponer de información del ancho de banda disponible de las comunicaciones. La misma es obtenida por el STM durante su uso, por lo tanto, la información será desconocida al inicio del sistema, pero conocida cuando el sistema esté en un estado estable.

Priorizar la selección de *peers* con caminos de comunicación diversificados tiene como fin, que un mismo fallo de red, afecte a la menor cantidad posible de conexiones. La selección de caminos diversificados es llevada a cabo utilizando el esquema presentado en [16], diseñado para implantarse en una red de distribución de contenidos (CDN, *Content Distribution Network*). Esta técnica utiliza *pings* para inferir la estructura de caminos a través de heurísticas.

4. Conclusiones y trabajos futuros

Hemos analizado el impacto de implementar un sistema de VoD en Internet, donde el servicio ofrecido por la red dificulta garantizar la calidad de servicio. Nuestra propuesta es un sistema P2P-VoD tolerante a fallos de red que incluye una arquitectura y mecanismos que ofrecen una solución integral de gestión de degradación de las comunicaciones, garantizando a los usuarios del sistema una visualización sin interrupciones y sin degradación de la calidad de imagen.

En esta arquitectura, los *peers* reciben el contenido multimedia desde múltiples *peers* transmisores que disminuye el impacto de los fallos de red y balancea la carga. Nuestra solución propone un novedoso esquema de detección de fallos principal ubicado en los nodos servidores, que da a ellos la libertad de aumentar o disminuir dinámicamente las transmisiones sin que los nodos clientes detecten fallos por baja tasa de recepción de datos. Esto permite planificar la entrega del contenido a los clientes para garantizar la calidad del servicio. La recuperación de un fallo de red se basa en la renegociación de tasas de transmisión, migración de servicios y un nuevo esquema de selección adecuada de nodos servidores. La detección de fallos es solapada con su recuperación para ganar tiempo y disminuir así la probabilidad de interrupciones en la visualización del video.

Nosotros creemos que nuestro esquema de tolerancia a fallos puede mejorar sustancialmente el servicio de sistemas P2P-VoD ya construidos, aumentando la satisfacción del usuario del sistema. Los trabajos futuros están centrados en

finalizar la implementación de un prototipo real y de simulación del sistema propuesto y realizar experimentación a mayor escala.

Referencias

1. Boutremans, C., Iannaccone, G., Diot, C.: Impact of link failures on voip performance. In: NOSSDAV '02: Proceedings of the 12th international workshop on Network and operating systems support for digital audio and video, New York, NY, USA, ACM (2002) 63–71
2. He, Y., Liu, Y.: Vovo: Vcr-oriented video-on-demand in large-scale peer-to-peer networks. *IEEE Trans. Parallel Distrib. Syst.* **20**(4) (2009) 528–539
3. Xiaoyuan Yang, P.R.: Kangaroo: Video seeking in p2p systems. In: Proc. of IPTPS 2009, Boston, MA, USA (2009)
4. Sultan, F., Bohra, A., Iftode, L.: Service continuations: An operating system mechanism for dynamic migration of internet service sessions. In: Proc. Symposium in Reliable Distributed Systems (SRDS). (oct 2003)
5. Karol, M., Krishnan, P., Li, J.: Voip network failure detection and user notification. *Computer Communications and Networks*, 2003. ICCCN 2003. Proceedings. The 12th International Conference on (20-22 Oct. 2003) 511–516
6. Nguyen, T., Zakhori, A.: Multiple sender distributed video streaming. *IEEE transactions on multimedia* **6** (2004) 315–326
7. Magharei, N., Rejaie, R.: Adaptive receiver-driven streaming from multiple senders. *Multimedia Systems* **11**(6) (2006) 550–567
8. Guo, Y., Suh, K., Kurose, J., Towsley, D.: P2cast: peer-to-peer patching scheme for vod service. In: WWW '03: Proceedings of the 12th international conference on World Wide Web, New York, NY, USA, ACM (2003) 301–309
9. Do, T.T., Hua, K.A., Tantaoui, M.A.: Robust video-on-demand streaming in peer-to-peer environments. *Comput. Commun.* **31**(3) (2008) 506–519
10. Maharana, A., Rathna, G.: Fault-tolerant video on demand in rserpool architecture. *International Conference on Advanced Computing and Communications (ADCOM)* (20–23 Dec. 2006) 534–539
11. Anker, T., Dolev, D., Keidar, I.: Fault tolerant video on demand services. *Proceedings of 19th IEEE International Conference on Distributed Computing Systems* (1999) 244–252
12. Kim, H., Kang, S., Yeom, H.: Node selection for a fault-tolerant streaming service on a peer-to-peer network. *Multimedia and Expo, IEEE International Conference on* **2** (2003) 117–120
13. Antoniou, Z., Stavrakakis, I.: An efficient deadline-credit-based transport scheme for prerecorded semisoft continuous media applications. *IEEE/ACM Trans. Netw.* **10**(5) (2002) 630–643
14. Ballardini, J., Souza, L., Suppi, R.: Un planificador de canales lógicos para un servidor de VoD en internet. *XII Congreso Argentino de Ciencias de la Computación (CACIC 2006)* (2006)
15. Ballardini, J., Souza, L., Suppi, R.: A network scheduler for an adaptive VoD server. In: *E-Business and Telecommunication Networks, Communications in Computer and Information Science. Volume 9.*, Springer-Verlag Berlin Heidelberg (2008) 237–251
16. Guo, M., Zhang, Q., Zhu, W.: Selecting path-diversified servers in content distribution networks. *Global Telecommunications Conference. IEEE GLOBECOM '03.* **6** (2003) 3181– 3185