

Selección de los mejores peers servidores en un sistema P2P-VoD

Cristian Ilabaca, Javier Balladini, Remo Suppi¹

Facultad de Informática, Universidad Nacional del Comahue, Argentina
{cilabaca, jballadi}@uncoma.edu.ar

¹ Departamento de Arquitectura de Computadores y Sistemas Operativos,
Universidad Autónoma de Barcelona, España
remo.suppi@uab.cat

Resumen La mayoría de los sistemas peer-to-peer (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. De esta manera, hemos propuesto previamente un esquema de tolerancia a fallos fácilmente adaptable a los sistemas P2P-VoD ya desarrollados. Como parte de este esquema, en el presente trabajo se propone un nuevo mecanismo de selección de los mejores peers servidores para un peer cliente dado. La selección adecuada de nodos servidores tiene la finalidad de que un mismo problema de comunicación (congestión o fallo) de la red afecte a la menor cantidad posible de conexiones, minimizando el impacto en el servicio de VoD. Nuestro esquema de selección de peers servidores tiene en cuenta la diversificación de caminos y el ancho de banda de los mismos. Los resultados experimentales demuestran la ventaja de nuestra propuesta frente a los mecanismos existentes.

1. Introducción

En la actualidad, la industria de entretenimientos y medios de comunicación conforman un sector de gran relevancia económica. 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.

El *streaming* de video se puede clasificar en: *streaming* en vivo y video bajo demanda (VoD, Video-on-Demand). En el *streaming* en vivo, los servidores transmiten programas en vivo, 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, como telefonía por Internet y video conferencia. A diferencia del *streaming* en vivo, el VoD permite a los usuarios reproducir un video, seleccionado de un gran conjunto de videos prealmacenados, facilitando un control completo sobre la visualización del contenido como si se estuviera utilizando un reproductor de DVD o VHS.

A diferencia del esquema cliente-servidor, los sistemas peer-to-peer (P2P) resultan más económicos debido a que aprovechan los recursos libres de los usuarios

(ancho de banda, cómputo y almacenamiento), que se denominan *peers*. En los sistemas P2P, los *peers* actúan simultáneamente como clientes y servidores, lo que aumenta considerablemente la capacidad de servicio del sistema. Un sistema P2P-VoD implementa un servicio de VoD en una arquitectura P2P.

El diseño de este tipo de sistemas es altamente complejo, fundamentalmente debido a las características de la información gestionada (videos de alta calidad) y el entorno Internet, al cual orientamos nuestro sistema. Tanto el video como el audio de alta calidad tienen un gran tamaño y una elevada tasa de bits, que implica transmisiones de larga duración y de una gran cantidad de información por unidad de tiempo. En caso de tener un ancho de banda insuficiente, la visualización del video sería interrumpida. Estos requisitos se vuelven un inconveniente debido a la falta de facilidades que Internet ofrece para este tipo de contenido.

En Internet los datos se transmiten en fragmentos o paquetes IP (*Internet Protocol*), los que son encaminados independientemente a través de la red por dispositivos denominados encaminadores. Todos los paquetes son tratados de la misma manera sin que se los pueda clasificar, marcar o darles una prioridad, con lo que la calidad de servicio no se puede garantizar. Además, como la red es compartida con otras aplicaciones, el ancho de banda disponible varía dependiendo del tráfico en cada momento. Esta variación del ancho de banda disponible se da porque la red está sobrecargada. En tal caso, se dice que la red está congestionada. El aumento del tráfico también puede ocurrir por causa de 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].

Para solucionar las limitaciones de esta red, nuestro grupo de investigación ha propuesto en [2] una arquitectura y esquema de tolerancia a fallos de red, fácilmente adaptable a los sistemas P2P-VoD ya desarrollados. En dicho trabajo se plantea la necesidad de incluir un esquema de selección de los mejores peer servidores para un determinado peer cliente. Este esquema tiene la finalidad de seleccionar aquellos peer servidores cuyos caminos de comunicación con el peer cliente tengan el mayor ancho de banda posible y que, si se produce un caso de congestión o fallo de red, el mismo afecte a la menor cantidad posible de caminos. El presente trabajo propone un esquema de selección específico que cumple con tales requerimientos. Se detalla su implementación y se describen los experimentos realizados que demuestran la ventaja de su uso.

El resto de este artículo se organiza de la siguiente manera. En la sección 2 se describen los trabajos relacionados. En la sección 3 se presenta el sistema de VoD bajo estudio y el esquema de selección de peers servidores. En la sección 4 se describe la implementación de un prototipo de simulación del esquema propuesto, y en la sección 5 los resultados experimentales. Finalmente, en la sección 6 se presentan las conclusiones y trabajos futuros.

2. Trabajos Relacionados

Algunos trabajos en tolerancia a fallos de sistemas de VoD, como [3], se ocupan de las caídas de nodos pero no contemplan la degradación de las comunicaciones. Los trabajos [4,5] no tienen una política definida que tenga en cuenta el estado de las comunicaciones para preferir a un peer sobre otro. En [6] 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. En [7] se propone un esquema para implementarse en una red de distribución de contenidos. Este trabajo proporciona la diversificación de caminos infiriendo la estructura de la red a través de heurísticas basadas en distancias entre nodos. Sin embargo, no tiene en cuenta el ancho de banda de las comunicaciones, de suma importancia para entregar contenido multimedia de alta calidad. Nuestra solución, tomando las bases de la diversificación de caminos propuesta por [7], presenta un nuevo esquema de selección de servidores que también considera el ancho de banda de las comunicaciones.

3. Un Sistema P2P-VoD tolerante a fallos de red

En esta sección se describe brevemente la arquitectura y funcionamiento de nuestro sistema P2P-VoD tolerante a fallos de red (que hemos propuesto previamente en [2]), y luego se describe la política de selección de peers que hemos definido y evaluado en este trabajo.

3.1. Generalidades de la arquitectura del sistema

Nuestro sistema [2] considera una arquitectura P2P en malla, la cual permite el establecimiento de conexiones arbitrarias entre los peers. Esta flexibilidad hace posible obtener una mayor ventaja al utilizar nuestro esquema de selección de peers servidores, mejorando la tolerancia a fallos del sistema.

Los peer que solo cumplen la función de servidor se llaman peer-origen, estos son el origen de los contenidos y el catalogo completo de video esta distribuido entre ellos. En nuestra arquitectura se forman grupos de colaboración de peers que están interesados en un mismo contenido y con puntos de reproducción muy próximos. Entre estos grupos de colaboración se establecen conexiones. Cada conexión se gestiona por medio de dos tipos de comunicaciones, una de control y otra de datos de video. Por esta última se intercambian los segmentos de video en donde la descarga de múltiples orígenes permite balancear la carga entre nodos servidores, permitiendo soportar la caída de uno de ellos mas fácilmente y además diversificar caminos de conexiones para tolerar fallos de red.

Estas comunicaciones utilizan un algoritmo de control de congestión TCP-Friendly¹, lo que permite a nuestra aplicación coexistir con otras aplicaciones de

¹ esto significa que el ancho de banda utilizado por un cierto protocolo no excede al utilizado por TCP (*Transport Control Protocol*).

Internet. El comportamiento de este tipo de algoritmos ante casos de congestión resulta en variaciones significativas del ancho de banda de las comunicaciones durante una sesión. Para salvar esto, se utiliza un mecanismo de reservación de recursos en donde cada nodo cliente establece conexiones y negocia una mínima tasa de envío de datos con sus nodos servidores. Cuando una comunicación 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.

Además de estos mecanismos de reservación de recursos, renegociación y migración del servicio, nuestro sistema incluye un innovador gestor de tráfico multimedia publicado en [8,9].

3.2. Selección de peers servidores

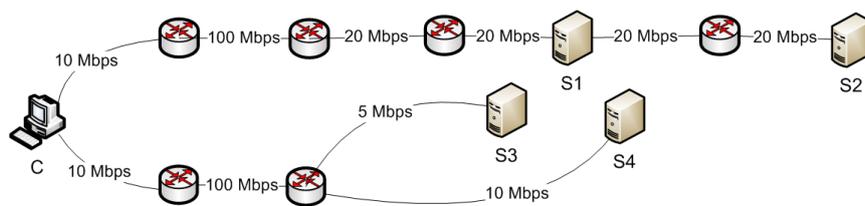


Figura 1: Ejemplo de selección de servidores

La selección de peers servidores cumple un rol muy importante en cualquier esquema tolerante a fallos de red. En la figura 1 se muestra un ejemplo de un sistema con un cliente *C* y los servidores *S1*, *S2*, *S3* y *S4*. Supongamos que el cliente debe seleccionar a dos servidores. Si el cliente *C* elige los nodos servidores *S1* y *S2*, al ocurrir cualquier fallo de red en el camino de comunicación ente *C* y *S1*, el cliente *C* dejaría de recibir datos desde ambos nodos servidores. En cambio, si se hubiesen elegido los servidores *S1* y *S3*, solo una conexión habría sido afectada. Por lo tanto, es necesario priorizar la selección de peers con caminos de comunicación diversificados y, de este modo, un mismo fallo de red afectaría a la menor cantidad posible de conexiones.

Sin embargo, para nuestro sistema, debido a la alta tasa de bits de los videos de alta calidad, es necesario que el proceso de selección de servidores también considere el ancho de banda de las comunicaciones. Por ejemplo, en el ejemplo anterior, sería más conveniente que el cliente *C* elija al servidor *S1* y *S4* por tener *S4* un mayor ancho de banda de comunicación con *C* que *S3*.

Si las comunicaciones tienen anchos de banda muy dispares, y un inconveniente en la red afecta a la comunicación de mayor ancho de banda, el servicio de VoD sería fuertemente perjudicado. Por lo tanto, también es necesario dar preferencia a anchos de banda poco dispares.

Para determinar la diversificación de caminos entre los nodos servidores utilizamos el esquema planteado por [7], diseñado para implementarse en una red de distribución de contenidos. Este algoritmo busca encontrar dos servidores cercanos a los clientes mientras se reduce al mínimo el número de enlaces compartidos entre los caminos. Para obtener la distancia entre dos nodos se utiliza una operación de ping, el cual calcula el número de saltos entre los nodos.

Con la información de la distancia entre el nodo cliente y los nodos servidores, y las distancias entre los nodos servidores, los autores definen una heurística cercana a la óptima. La estimación de la diversificación de caminos dc es calculada usando la siguiente ecuación:

$$dc_{s_i,s_j,c} = p_g^{d(c,s_i)} + p_g^{d(c,s_j)} - p_g^{\frac{d(c,s_i)+d(c,s_j)+d(s_i,s_j)}{2}} \quad (1)$$

donde p_g es la probabilidad de que un paquete sea transmitido correctamente.

En nuestro esquema definimos la selección de peers servidores priorizando una mayor diversificación de caminos y ancho de banda grandes y dispares. Con estos tres requisitos, definimos una nueva heurística que determina el valor db (*Diversification and Bandwidth*) para un par de servidores s_i y s_j y un cliente c , que se estima usando la siguiente ecuación:

$$db_{s_i,s_j,c} = \alpha \times \frac{dc_{s_i,s_j,c}}{divmax} + \beta \times \frac{(bw_{s_i,c} + bw_{s_j,c})}{bwmax} + \gamma \times \frac{1}{1 + abs(bw_{s_i,c} - bw_{s_j,c})} \quad (2)$$

donde los parámetros α, β y γ son valores entre 0 y 1, y $\alpha + \beta + \gamma = 1$. El parámetro α representa la prioridad (o peso) que se da a la diversificación de caminos, β es la prioridad que se da al ancho de banda, y γ es la prioridad que se da a los anchos de banda igualitarios (no dispares). El valor de diversificación de caminos $dc_{s_i,s_j,c}$ para los servidores s_i y s_j es el definido en la ecuación 1. $bw_{s_i,c}$ y $bw_{s_j,c}$ son el ancho de banda de comunicación entre los servidores s_i y s_j con el cliente c , respectivamente. $divmax$ es el valor máximo de $dc_{s_i,s_j,c}$ obtenido para cada par de servidores s_i y s_j de todo el grupo de colaboración. Similarmente, $bwmax$ es el valor máximo de la suma de los anchos de banda ($bw_{s_i,c} + bw_{s_j,c}$) obtenida para cada par de servidores (s_i y s_j) de todo el grupo de colaboración. El par de servidores con mayor db es seleccionado.

4. Implementación del prototipo de selección de servidores en el simulador NS-3

Para implementar el prototipo optamos por utilizar el simulador NS3 [10], orientado principalmente al uso educativo y de investigación. Para poder comparar la versatilidad de nuestra propuesta, hemos construido una arquitectura de prototipo genérica que nos permitió implementar tres mecanismos distintos de selección de servidores. Un mecanismo es el propuesto por nosotros, otro mecanismo es el propuesto por [7], y el restante es un mecanismo de selección aleatoria.

Para implementar nuestro prototipo en NS3 se agregaron dos nuevas aplicaciones al árbol principal de clases del simulador. La aplicación *Peer* implementa la funcionalidad de los peers. La aplicación *VodV4Ping* implementa la operación de ping que transmite un paquete IP. Este paquete tiene un encabezado con un campo denominado TTL (*Time To Live*) cuyo valor es disminuido en 1 cada vez que atraviesa un encaminador. Así, observando el valor del TTL de partida y el resultante, es posible calcular el número de saltos entre dos nodos.

Para utilizar la nueva aplicación *Peer*, inicialmente se configura la topología de la red deseada y se instala la aplicación a los nodos deseados. Para cada peer, es necesario especificar cuál será el tipo de selección de servidores que utilizará: aleatoria, con diversificación de caminos y con diversificación de caminos y ancho de banda. Además, es necesario definir para cada peer cuales serán sus vecinos (direcciones IP de los posibles peers servidores) y el ancho de banda que los comunica.

La definición del ancho de banda es manual solo para éste prototipo parcial que únicamente implementa la selección de servidores. En el sistema P2P-VoD completo, los valores de los anchos de banda es obtenida por uno de sus módulos durante su uso. Similarmente, la definición de los vecinos es realizada por otro de los módulos del prototipo completo del sistema.

Nuestro sistema P2P-VoD no transmite la información de los videos mediante el protocolo TCP sino que lo hace con uno propio. No obstante, este prototipo utiliza comunicaciones TCP debido a que aún no tenemos implementado nuestro protocolo en NS3 sino en NS2.

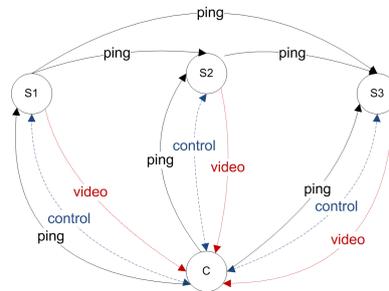


Figura 2: Comunicación entre cliente y peers servidores

La selección aleatoria simplemente elige dos peers cualesquiera de la lista de nodos vecinos y les solicita los datos multimedia. Entonces los servidores comienzan la transmisión de datos de video. Sin embargo, para los dos casos de selección restantes, se agrega la comunicación para recolección de datos de distancias entre peers. En la figura 2 se muestran estos diferentes tipos de comunicaciones. La aplicación *Peer* en su función de cliente hace un ping a cada uno de sus vecinos y, además, envía por medio de la comunicación de control, un listado de servidores (direcciones IP) a cada servidor con los cuales debe averiguar su distancia. Una

observan las velocidades de los enlaces. Se asigna una probabilidad de pérdida de paquetes de 0,004 a cada nodo de la topología. Utilizamos una topología de red acotada para facilitar el estudio del comportamiento de los métodos de elección de servidores.

A partir de la infraestructura de red descrita, se asumen 16 peer servidores S_i , vecinos del peer cliente C . Al cliente se le provee una lista con todos los peers servidores (vecinos) disponibles. Para el método de selección con diversificación de caminos y ancho de banda, la lista incluye un valor de ancho de banda que cada servidor tiene en su comunicación con el cliente. Para este método, cuya heurística define la ecuación 2, asumimos un valor $\alpha = 0,6$, $\beta = 0,30$ y $\gamma = 0,1$.

Todos los peer comienzan su ejecución en el tiempo 0. El cliente comienza la descarga de los datos multimedia en el segundo 8. La simulación se detiene a los 170 segundos. A los 20 segundos de comenzar la simulación, se induce la caída del enlace entre los nodos 3 y 14, el cual se vuelve operativo 20 segundos más tarde, es decir, en el tiempo 40. El propósito de producir este fallo en la red es el análisis del comportamiento de los métodos de selección de servidores.

5.2. Resultados del experimento

En esta sección se presentan los resultados más representativos que permiten apreciar el funcionamiento y la bondad del método de selección de servidores con diversificación de nodos con ancho de banda en un sistema VoD-P2P frente al resto de los métodos de selección.

En la figura 4 vemos el rendimiento alcanzado por los tres métodos de selección, por medio de mediciones de los Mb/s que el cliente recibe de video a lo largo del tiempo. La curva verde y azul muestra el rendimiento diferenciando los datos entregados por cada uno de los dos servidores seleccionados. La curva roja muestra el rendimiento global (la suma de las curvas verde y azul).

En la figura 4 (a) suponemos un caso en donde el método de selección aleatoria ha elegido a los peer servidores $S9$ y $S10$ que comparten el enlace 3-14 y el mismo subdominio-stub. En el tiempo 20 se ve que la caída del enlace afecta a ambas transmisiones y se detiene el envío de datos multimedia por completo hasta reiniciarse en el tiempo 45. El envío total de los datos tarda en total 171 segundos.

En la figura 4 (b) vemos el método con diversificación de caminos que elige nodos que tengan la menor cantidad de enlaces en común, es decir, diversificación de caminos. Entre los pares de peers que cumplen con esta condición se eligen los peer servidores $S5$ y $S16$ en subdominios-stub diferentes. La caída del enlace 3-14 en el tiempo 20 afecta únicamente a la transmisión de un solo nodo ($S16$) de uno de los subdominio-stub. Luego la falla del enlace no afecta tanto como en el caso anterior, aunque genera un retraso en el envío de los datos que termina en el tiempo 110.

Finalmente en la figura 4 (c) vemos el método de selección de servidores con diversificación de caminos y ancho de banda. Entre los pares de peers elige los peers servidores $S7$ y $S16$ de diferentes subdominios-stub pues $S7$ tiene mayor ancho de banda (4 Mb/s) priorizando además de la diversificación de caminos

el ancho de banda. En este caso la caída del enlace 3-14 en el tiempo 20 no afecta en lo absoluto la transmisión de los datos y el cliente recibe los datos en forma normal siendo así la caída del enlace transparente para el peer cliente y terminando en el tiempo 71. En este caso, *S7* puede entregar la totalidad de los 4 Mb/s que puede recibir el cliente, por eso es que el cliente nunca deja de recibir el total de su capacidad (aunque en otro caso podría no ser así).

El experimento aquí presentado es uno de los tantos que podrían ocurrir. Por supuesto que nuestro mecanismo no es una solución para todos ellos, pero es una mejora significativa respecto a los mecanismos existentes.

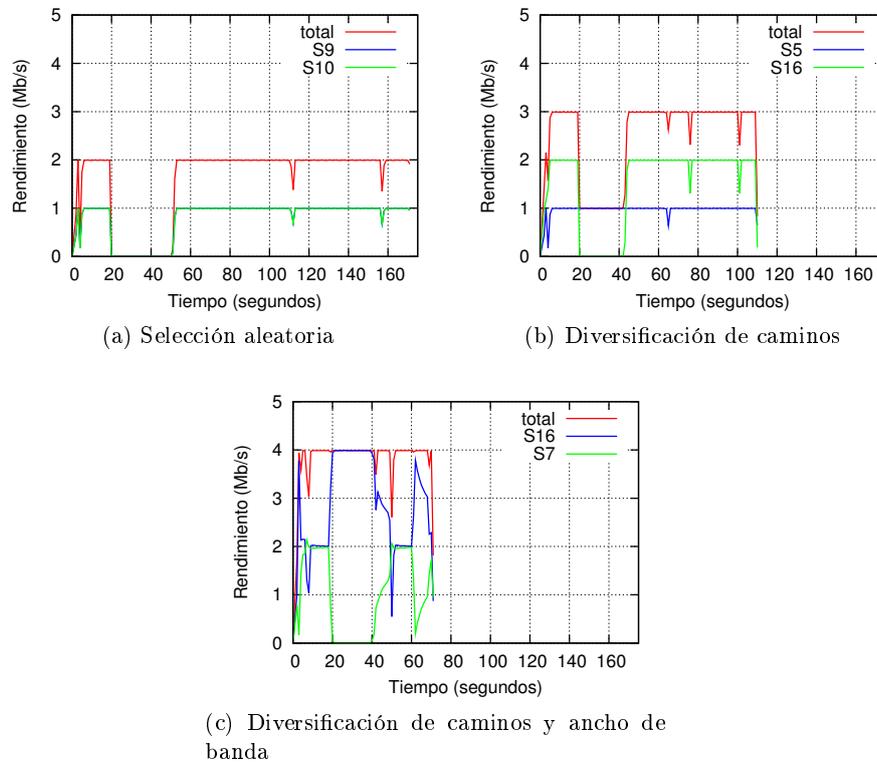


Figura 4: Resultados experimentales

6. Conclusiones y Trabajos Futuros

En un trabajo previo hemos propuesto 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, intentando garantizar a los usuarios del sistema una visualización sin interrupciones y sin degradación de la calidad de imagen. En el presente trabajo proponemos un nuevo mecanismo de selección adecuada de nodos servidores para ser incluido en tal sistema, basado en la diversificación de caminos y el ancho de banda de las conexiones. Se detalla la implementación de un prototipo de simulación que, además de incluir el mecanismo propuesto, incluye otros dos mecanismos existentes en la literatura. Los experimentos presentados demuestran la ventaja de nuestra propuesta frente a los dos mecanismos ya existentes. Creemos que nuestro nuevo esquema puede mejorar sustancialmente el servicio de los sistemas de VoD.

Los trabajos futuros se centran en extender la heurística de selección a n peers servidores entre m , en vez de únicamente 2 entre m . También está planificado integrar el prototipo de selección de peers servidores en el prototipo completo de nuestro sistema P2P-VoD.

Referencias

1. Boutremans, C., Iannaccone, G., Diot, C.: Impact of link failures on voip performance. In: Proceedings of the 12th international workshop on Network and operating systems support for digital audio and video. NOSSDAV '02, New York, NY, USA, ACM (2002) 63–71
2. Balladini, J., Grosclaude, E., Suppi, R., Luque, E.: A network failure-tolerant p2p-vod system. In: Computer Science and Technology Series, XV Argentine Congress of Computer Science. (2010) 73–83
3. Danny, T.A., Dolev, D., Keidar, I.: Fault tolerant video on demand services. In: Proceedings of the 19th International Conference on Distributed Computing Systems. (1999) 244–252
4. Yang, X., Gjoka, M., Chhabra, P., Markopoulou, A., Rodriguez, P.: Kangaroo: Video seeking in p2p systems. In: Proceedings of USENIX IPTPS '09, Boston, MA, USA (April 2009)
5. He, Y., Liu, Y.: Vovo: Vcr-oriented video-on-demand in large-scale peer-to-peer networks. *IEEE Trans. Parallel Distrib. Syst.* **20** (April 2009) 528–539
6. Kim, H., Kang, S., Yeom, H.Y.: Node selection for a fault-tolerant streaming service on a peer-to-peer network. In: Proceedings of the 2003 International Conference on Multimedia and Expo - Volume 1. ICME '03 (2003)
7. Guo, M., Zhang, Q., Zhu, W.: Selecting path-diversified servers in content distribution networks. In: Global Telecommunications Conference, 2003. GLOBECOM '03. IEEE. Volume 6. (2003) 3181 – 3185 vol.6
8. Balladini, J., Souza, L., Suppi, R.: A network scheduler for an adaptive vod server. In Filipe, J., Obaidat, M.S., eds.: E-Business and Telecommunication Networks. Volume 9 of Communications in Computer and Information Science. Springer Berlin Heidelberg (2008) 237–251
9. Balladini, 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)
10. McCanne, S., Floyd, S.: NS3 - Network Simulator. <http://www.nsnam.org/> (July 2011)