

Minimizing the Impact of P2P-TV Applications in Access Links

I. Quintana, L. Sequeira, J. Fernandez, J. Ruiz, and J. Saldana

Abstract—Many studies are concentrated on provisioning certain levels of Quality of Service (QoS) employing different approaches related to different areas (control congestion algorithms, transport protocols, etc). This paper is focused on the improvement of the QoS of peer-to-peer TV (P2P-TV), and in particular, it proposes a methodology to minimize the impact of bursty traffic in the buffer of access devices using a traffic shaping technique. This paper shows how QoS can be improved by means of traffic shaping, specifically by using smoothing techniques. Some parameters such as packet loss, delay and throughput of a popular P2P-TV application are analyzed. The competition for the available bandwidth between the P2P-TV and some Transport Control Protocol (TCP) traffic flows is also studied. The results show a reduction in terms of loss percentage when the traffic of the P2P-TV application is smoothed, introducing small delays as a counterpart.

Index Terms—P2P-TV, Multimedia Streaming, Real-Time Communication, QoS, Traffic Smoothing.

I. INTRODUCCIÓN

LOS servicios multimedia en Internet (transmisión de video, juegos *online*, videoconferencia, VoIP, etc.) son utilizados por una gran cantidad de usuarios, lo que aumenta significativamente el volumen de tráfico en la red. Asimismo, los usuarios exigen que estos servicios multimedia en tiempo real estén disponibles con alta calidad, independientemente de las tecnologías de acceso. Sin embargo, cada servicio tiene su propio patrón de tráfico que depende de la naturaleza y el tamaño de la información transportada. En este contexto, algunas aplicaciones, como la videovigilancia y la transmisión de video, inyectan ráfagas de tráfico en la red, generando grandes volúmenes de información en períodos muy cortos.

Las aplicaciones P2P-TV han ganado popularidad, por lo que se han convertido en el foco de atención de muchos estudios de investigación [1]–[7]. Varios estudios han analizado y caracterizado el gran volumen de tráfico generado por estas aplicaciones, y sus implicaciones para los proveedores de servicios de Internet [3], [4]. Otros se han centrado en la optimización de algoritmos de distribución de contenido [5] o cómo mejorar la Calidad de la Experiencia (QoE) en función del número de usuarios [6], [7].

Usualmente, las aplicaciones P2P-TV generan ráfagas de paquetes pequeños, sobrecargando la red y generando picos de *throughput* muy altos. En estos casos, el *buffer* de los dispositivos de red puede saturarse dando como resultado

la ocurrencia de períodos de congestión que provocan una significativa pérdida de paquetes, con la posterior reducción de QoS para los usuarios. En este contexto, los enlaces de acceso pueden convertirse en puntos sensibles y producir cuellos de botella en la red debido a su capacidad limitada.

Para minimizar el impacto de estos picos de tráfico en la red, se han propuesto algunas soluciones, por ejemplo, multiplexar los paquetes pequeños en bloques más grandes [8], aumentando así la eficiencia de los flujos de tráfico y reduciendo el número de paquetes por segundo. Otra posibilidad es el uso de técnicas de conformado de tráfico, que establecen un rendimiento máximo para ráfagas de tráfico, y por lo tanto, pueden reducir el nivel de pérdida de paquetes. Como contrapartida, ambas técnicas introducen retardos, por lo que aparece una disyuntiva entre los nuevos retardos añadidos y la reducción de la pérdida de paquetes obtenida.

Otro aspecto que puede convertirse en una causa del deterioro de la QoS es la concurrencia de diversos flujos de datos en un solo enlace de acceso; de hecho, la combinación del tráfico de varios servicios puede dar lugar a ráfagas de paquetes que eventualmente conducirían a problemas de congestión en ciertos puntos de la red [9]. Este fenómeno de ráfagas puede ocurrir tanto para el tráfico UDP (User Datagram Protocol) como TCP o combinaciones de ambos.

En este artículo se propone una metodología que brinda soluciones basadas, principalmente, en la modificación de la forma de cómo el tráfico es enviado a la red. Inicialmente, se analiza el comportamiento de una aplicación P2P-TV en tiempo real en términos de parámetros objetivos de QoS (pérdida de *bytes*, pérdida de paquetes, retardo y *throughput*), cuando se le aplica una técnica de alisado. Posteriormente, se estudia cómo las aplicaciones con tráfico de diferente naturaleza se influyen mutuamente cuando compiten por la capacidad del enlace, analizándose el *throughput* logrado por la aplicación P2P-TV cuando comparte el enlace con un tráfico TCP.

El resto del documento está organizado de la siguiente manera: en la Sección II se discuten los temas relacionados con las limitaciones existentes en las redes de acceso, las principales características del tráfico de SopCast¹ (aplicación P2P-TV utilizada en este trabajo) y las técnicas que se emplean para mejorar la utilización del enlace. En la Sección III se presenta una descripción detallada de la metodología propuesta y su validación. La Sección IV detalla las pruebas y resultados obtenidos al aplicar una técnica de conformado de tráfico a

I. Quintana, J. Fernandez, J. Ruiz and J. Saldana, Universidad de Zaragoza, Dpto. de Ingeniería Electrónica y Comunicaciones, Zaragoza, España, e-mail: {idelkysq, navajas, jrui, jsaldana}@unizar.es

L. Sequeira, King's College London, Centre for Telecommunications Research, Londres, Reino Unido, email: luis.sequeira@kcl.ac.uk

¹<http://www.sopcast.com/>

través de dos casos de estudios y finalmente, las conclusiones se resumen en la Sección V.

II. ESTADO DEL ARTE

A. Limitaciones en las Redes de Acceso

Es bien sabido que Internet, en sus comienzos, fue diseñado para transferir una gran cantidad de datos de servicios tales como el correo electrónico, la navegación web o la transferencia de archivos (File Transfer Protocol (FTP)). Una de las principales características de estos servicios es la transmisión de tramas con el tamaño máximo permitido por diferentes tecnologías de comunicaciones (tamaño máximo de trama = unidad de transmisión máxima (MTU)), aumentando así la eficiencia y reduciendo el procesamiento en *routers* y *switches* [10].

Por otro lado, en las redes de acceso, es común encontrarse con *routers* de gama media y baja. Estos dispositivos utilizan *buffers* como mecanismo de control de tráfico, por lo que el tamaño y el comportamiento del *buffer* se convierten en parámetros de diseño importantes [9], [11]. En los últimos años, se han publicado algunos estudios relacionados con el problema del tamaño del *buffer*, su comportamiento e influencia en diferentes tipos de tráfico multimedia, utilizando características de QoS basadas en parámetros de red objetivos (como *jitter*, pérdida de paquetes, etc) [12].

Sin embargo, el uso de servicios en tiempo real está creciendo en Internet, generando altas tasas de paquetes pequeños. Por lo tanto, debido a que estos flujos generan gran cantidad de paquetes por segundo, incluso si mantienen la misma tasa medida en *bits* por segundo, la capacidad de procesamiento de los dispositivos de red puede convertirse en un cuello de botella [13].

B. Tráfico de Aplicaciones P2P-TV

Las aplicaciones P2P-TV permiten la disponibilidad de contenido de video, perdiendo cierto grado de QoS con respecto a otros servicios como TV bajo demanda o el modelo cliente-servidor utilizado por IPTV. Las aplicaciones P2P-TV generalmente generan tráfico que consiste en paquetes grandes, asociados con paquetes de video, y altas tasas de paquetes pequeños, asociados con el tráfico de control (por ejemplo, ACK del nivel de aplicación) [14], [15]. Esto da como resultado una mezcla de tráfico que incluye paquetes pequeños y grandes a través de Internet. En estos casos, la capacidad de procesamiento del *buffer* puede constituir un cuello de botella cuando tiene que gestionar demasiados paquetes por segundo [13]. Existen estudios [16] en los que se demuestra que los paquetes de video pueden ser penalizados por altas cantidades de paquetes pequeños, aumentando la probabilidad de ser descartados. En consecuencia, el comportamiento de los *peers* dentro de una estructura P2P no será el esperado.

En la actualidad hay muchos estudios que se han centrado en la caracterización del comportamiento y el tráfico de diferentes aplicaciones de transmisión de video. Las principales motivaciones para estos estudios son el impacto de este tráfico en las redes de comunicaciones [3], las mejoras que aporta a los operadores de red IPTV [4], la QoE proporcionada [6],

[17], algoritmos de distribución alternativos [5] y su creciente número de usuarios [7].

Para las pruebas realizadas en este trabajo se ha utilizado SopCast, siendo una de las aplicaciones P2P-TV más estudiadas. En [15], [17] se estudiaron los mecanismos básicos de su funcionamiento y se determinó que el tráfico de video y señalización es exclusivamente UDP. Su sistema de *streaming* se sustenta en una arquitectura de distribución no estructurada de tipo *mesh*, donde el cliente solicita los fragmentos de video (conocidos también como *chunks*) de uno de sus vecinos (*peers*). Para establecer y mantener esta estructura de red, el control de tráfico es necesario y el mismo está compuesto por paquetes UDP pequeños (≤ 100 bytes). En [7], [17] los autores presentaron una estimación del retardo tolerado por un cliente SopCast (\approx entre 30 y 45 segundos), detectando la presencia de dos *buffers*: el primero en la aplicación, relacionado con el intervalo entre la selección de un canal y el inicio de la transmisión del video; y el segundo en el reproductor de video del cliente.

En general, el comportamiento del tráfico de SopCast es similar al de otras aplicaciones P2P-TV ampliamente utilizadas en Internet, por lo que los resultados obtenidos en este estudio pueden, en cierta medida, extrapolarse a otras aplicaciones con características similares.

C. Técnicas para Mejorar la Utilización del Enlace

Diversas técnicas pueden ser aplicadas para mejorar la utilización del enlace. En [18] se ha introducido y utilizado una técnica de control de la congestión para sistemas de videoconferencia codificados con *H.264*, con el fin de mejorar la QoE en escenarios con limitaciones de capacidad del enlace. Sin embargo, el tamaño y el comportamiento de los *buffers* en los *routers* debe tenerse en cuenta para utilizar estas técnicas de forma adecuada. En [8] los autores presentaron un estudio de la influencia de un método de multiplexión en los parámetros que definen la calidad subjetiva de los juegos *online*. Los resultados mostraron que los *buffers* pequeños presentan mejores características para mantener el retardo y *jitter* en niveles adecuados, a costa de aumentar la pérdida de paquetes. Además, se puede obtener una reducción del 38% en la utilización del enlace, agregando retardos debido a la retención de los paquetes.

A pesar de las ventajas de la multiplexión, no siempre su uso es apropiado, por ejemplo, cuando una aplicación genera paquetes grandes. En estos casos, no se pueden multiplexar en un paquete más grande y es conveniente usar otras técnicas de optimización del tráfico. La metodología propuesta en este trabajo hace uso del alisado como mecanismo de conformado de tráfico. Esta técnica tiene como objetivo optimizar y garantizar el rendimiento de la red, mejorando así el uso de los recursos, a expensas de introducir pequeños retardos bajo ciertas circunstancias y un mínimo costo computacional. Los niveles de retardo introducido deben seleccionarse de tal manera que la calidad de la aplicación no se vea alterada. La correspondiente implementación se detallará en la Sección III.

III. METODOLOGÍA

La metodología propuesta permite minimizar la congestión en los *buffers* de los dispositivos de acceso. Dicha congestión usualmente es causada por la llegada del tráfico a ráfagas, provocando pérdida de paquetes. Se propone la utilización de una técnica de conformado de tráfico que previene que dichos dispositivos se conviertan en cuellos de botella, reduciéndose la pérdida de paquetes y aumentando la QoS.

La transmisión de datos de algunas aplicaciones y la congestión de dispositivos de red en ciertos puntos sensibles (por ejemplo, enlaces de acceso domésticos, dispositivos de red de gama media y baja con *buffers* limitados y manejo de grandes cantidades de paquetes) pueden causar muchos problemas en la información que se transmite por las redes. Para reducir la pérdida de paquetes causada por la congestión, se pueden considerar dos enfoques: modificar el código de la aplicación o considerar soluciones externas a la aplicación.

Con el primer enfoque, se asegura un mecanismo automático de control de la congestión para que el tráfico pueda adaptarse a las nuevas condiciones externas e implementar un mecanismo de QoS adecuado (como lo hace Skype [19]), sin embargo esto requiere que se tenga acceso al código fuente. En el segundo enfoque, se utiliza un dispositivo (o software) intermedio entre la aplicación y el *buffer* del *router*, que permita el procesamiento o tratamiento externo del tráfico generado [20], con el objetivo de adaptarlo a las situaciones críticas antes mencionadas. En este trabajo se ha adoptado el segundo enfoque debido a que la aplicación en estudio no es de código abierto.

A. Metodología Propuesta

La metodología propuesta se divide en dos grandes bloques (Fig. 1). En el primer bloque, denominado *Análisis de Tráfico*, se estudian las características del tráfico, con el propósito de definir y seleccionar los parámetros de QoS que influyen en su comportamiento. Lo primero que se requiere es implementar un mecanismo de monitorización que permita capturar el tráfico de la aplicación. Luego, se realiza un análisis de los parámetros objetivos de QoS, como por ejemplo: el tamaño de los paquetes, el tiempo entre paquetes y el *throughput*.

Mediante el *Selector de parámetros de calidad*, se definen los parámetros más restrictivos en la utilización del enlace de acceso por dicho tráfico (tamaño de paquetes, tiempo entre paquetes y *throughput*). En el caso del tamaño de los paquetes, la idea principal es determinar si el tráfico transmitido consiste de muchos paquetes grandes y/o pequeños, ya que como se mencionó en la sección II-C, es un factor a tener en cuenta al seleccionar la técnica de conformado de tráfico más conveniente. Otro de los parámetros útiles en el estudio del tráfico de una aplicación de tiempo real es el tiempo entre paquetes, ya que ayuda a identificar cómo es el comportamiento de la aplicación: (a) si presenta ráfagas de paquetes o (b) si los paquetes presentan un patrón de envío constante. El *throughput* se utiliza para poder determinar un umbral mínimo garantizable para la aplicación, de tal manera que tenga un funcionamiento adecuado.

En el bloque de *Conformado de Tráfico* se selecciona la técnica de conformado de tráfico más adecuada y se entrega como salida el tráfico optimizado. El objetivo de este bloque es minimizar el impacto causado por el tráfico a ráfagas en el *buffer* de los dispositivos de acceso. De esta manera se evita la pérdida de paquetes en los *buffers* durante los períodos de congestión, introduciéndose ciertos niveles de retardo que deben ser tolerables por las aplicaciones. Este bloque puede contener diferentes *Algoritmos* o técnicas de conformado de tráfico y se selecciona el que ofrezca la mejor relación de compromiso en cuanto al retardo soportado por la aplicación y la utilización del enlace. Con el uso de estas técnicas se pretende limitar los picos altos de *throughput* que podrían causar congestión.

El *Estimador de retardo* permite determinar el retardo añadido a cada paquete teniendo en cuenta las restricciones de tiempo real del tráfico, esto es especialmente importante para poder comparar con los niveles tolerables de retardo de cada aplicación. Por último, el *Selector de conformado de tráfico* permite determinar la técnica de conformado de tráfico que más se ajuste al comportamiento de la aplicación. Estos valores de retardo se deben seleccionar para controlar la pérdida de paquetes causada por la llegada del tráfico a ráfagas y para mejorar la calidad de la comunicación. Por lo tanto, se debe valorar una relación de compromiso entre los retardos añadidos y los niveles de pérdida de paquetes, según las características de la aplicación y sus requisitos de retardo.

B. Validación de la Metodología

Con el propósito de validar la metodología, se ha seleccionado una aplicación P2P-TV para analizar su comportamiento y determinar sus principales parámetros y características con el objetivo de garantizar un mejor rendimiento en la transmisión y reducir la pérdida de paquetes en el enlace de acceso. Estas mejoras consisten en adaptar la tasa del tráfico, teniendo en cuenta las restricciones de los dispositivos de acceso. Este trabajo se ha enfocado en el estudio del enlace de subida, siendo el caso más restrictivo en una red de acceso asimétrica.

1) Análisis de Tráfico:

a) *Monitorización del tráfico*: Con el fin de capturar el tráfico y obtener una traza real de la aplicación P2P-TV seleccionada, se implementa el escenario mostrado en la Fig. 2. Se ha utilizado *Tshark* como herramienta de captura y análisis de tráfico, capturándose alrededor de 30 minutos de un partido de la *Champions League*. El cliente SopCast se encuentra ubicado en el campus de la Universidad de Zaragoza, con una dirección IP pública y utiliza una interfaz de red *Gigabit Ethernet*; está ejecutándose en una *host* Debian con Linux kernel 2.6.38 – 7, con CPU Intel® Core™ i3 a 2.4 GHz. Como se puede apreciar en la Fig. 2, se añade un *sniffer* para garantizar que las mediciones no interfieran en el rendimiento de la aplicación.

b) *Análisis del tamaño de los paquetes*: Una vez obtenida la traza, se procesa y se analiza, siendo el tamaño de los paquetes uno de los principales parámetros a tener en cuenta para una posterior caracterización del tráfico. La Fig. 3

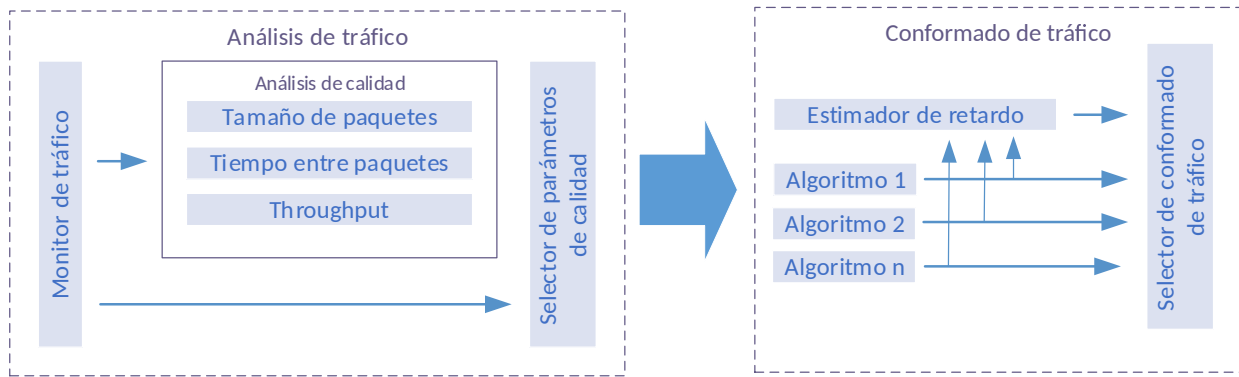


Fig. 1. Metodología propuesta.

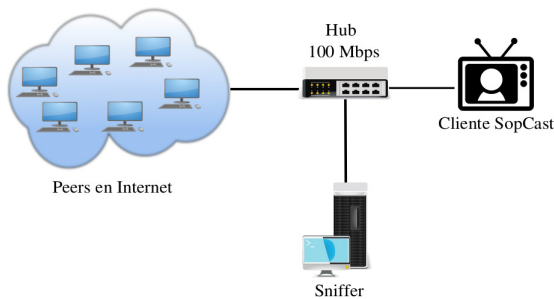


Fig. 2. Escenario para capturar el tráfico.

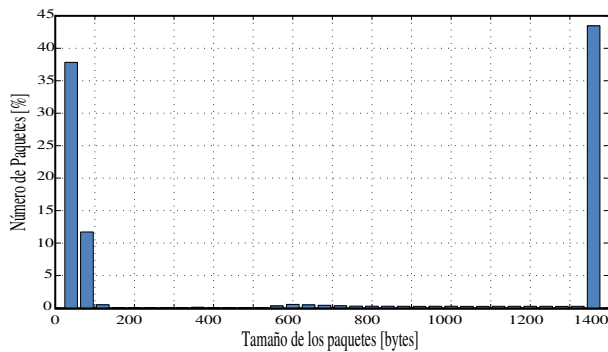


Fig. 3. Histograma del tamaño de los paquetes.

muestra el histograma del tamaño de los paquetes generados por el cliente SopCast, donde se puede apreciar que casi el 50% de los mismos se corresponden a paquetes pequeños (con tamaños ≤ 100 bytes) y cerca de un 45% a paquetes grandes (≈ 1300 bytes). Los paquetes grandes transmitidos transportan la información del video intercambiado con otros peers en Internet. Los paquetes pequeños son utilizados por la capa de aplicación para gestionar el estado de cada peer presente en la comunicación, pero también para monitorizar, controlar y reorganizar los paquetes de video [21]. Además, por cada paquete de video recibido se envía un ACK del nivel de aplicación, generando muchos paquetes pequeños con redundancia en muchos campos de sus cabeceras y, por tanto, una baja eficiencia en términos de uso de recursos de red.

c) *Análisis del tiempo entre paquetes:* De la Fig. 4 se puede inferir que el tráfico de SopCast presenta una amplia dis-

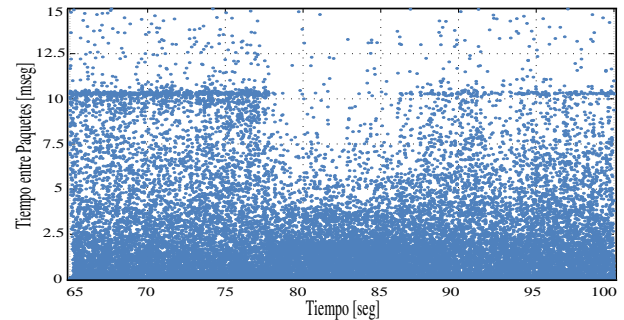


Fig. 4. Tiempo entre paquetes para el tráfico utilizado.

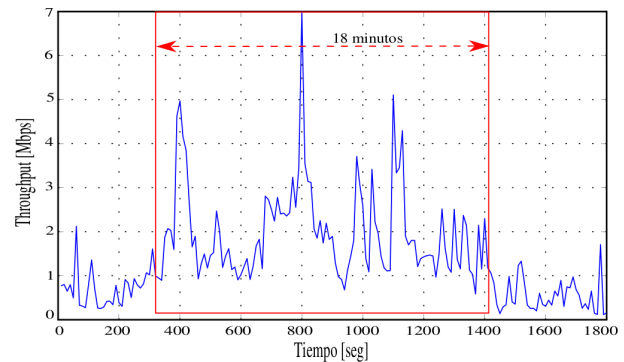


Fig. 5. Muestra de tráfico utilizada en las pruebas.

persión en cuanto al tiempo entre paquetes. Esta aleatoriedad dificulta la búsqueda de un modelo estadístico para este tipo de tráfico, por lo cual en este trabajo se utiliza una traza real. Por otra parte, esta aplicación no presenta una tasa uniforme de entrega de paquetes.

d) *Análisis del throughput:* Para estudiar el intervalo de tiempo más crítico en el comportamiento de la aplicación en cuanto a la demanda de recursos de red, se selecciona una muestra representativa de su tráfico, donde se concentran la mayor cantidad de picos de throughput. La muestra seleccionada tiene una duración de 18 minutos (recuadro rojo en la Fig. 5) y el throughput promedio de la aplicación es de aproximadamente 1.84 Mbps.

e) *Selección de parámetros de calidad:* Debido a la alta presencia de paquetes grandes, la técnica de alisado parece ser

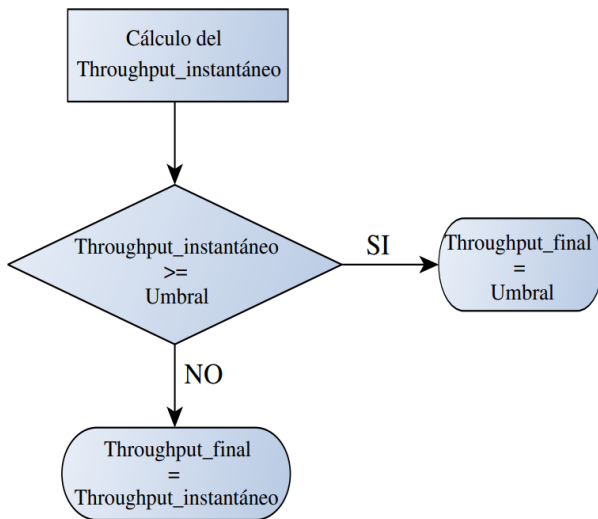


Fig. 6. Algoritmo para la técnica de alisado.

la más adecuada para este tipo de tráfico y el tamaño de los paquetes se convierte en un parámetro que no se debe alterar, como se ha explicado en las secciones II-C y III-A. El tiempo entre paquetes se ha identificado como el principal parámetro susceptible de ser modificado, para así disminuir el impacto del tráfico a ráfagas que llega al *buffer* de los dispositivos de acceso. Por otro lado, a la hora de alisar se debe tener en cuenta una serie de umbrales de alisado (por ejemplo, de 2 Mbps a 5 Mbps). Dichos umbrales deben estar siempre por encima del *throughput* medio alcanzado por la aplicación para no impactar negativamente su funcionamiento.

2) Conformado de Tráfico:

a) *Algoritmos de alisado*: Se ha implementado el algoritmo de bajo costo computacional mostrado en la Fig. 6, donde se calcula el *throughput* instantáneo de cada paquete utilizando el tamaño del paquete y el tiempo entre paquetes. Luego, se compara con el umbral de alisado seleccionado (que varía de 2 a 5 Mbps). Si el *throughput* instantáneo excede el umbral seleccionado, el paquete se retrasa; de lo contrario, el tiempo de transmisión del paquete no se modifica. De esta forma, es posible obtener un flujo con una tasa de transmisión menor que el umbral seleccionado, optimizando así el rendimiento de la transmisión. Independientemente del nivel de alisado aplicado, la nueva traza alisada que se ha obtenido tiene una duración de 18 minutos, al igual que la traza original.

Para los niveles de alisado se han seleccionado los siguientes valores: 2, 2.5, 3, 3.5, 4, 4.5 y 5 Mbps. No se deben utilizar niveles de alisado inferiores a 1.84 Mbps, ya que la traza alisada aumentaría su duración. De forma similar, si se utilizan umbrales mayores a 5 Mbps, las trazas alisadas resultantes no varían significativamente con respecto a la de 5 Mbps. Debido a que el algoritmo debe ejecutarse para cada nivel de alisado, esto permite simular que existen siete algoritmos diferentes, y por lo tanto, validar la metodología con múltiples algoritmos.

Para los niveles de alisado con valores cercanos al *throughput* promedio exigido por la aplicación (≈ 1.84 Mbps), los picos más altos de *throughput* (Fig. 5), que son generados por

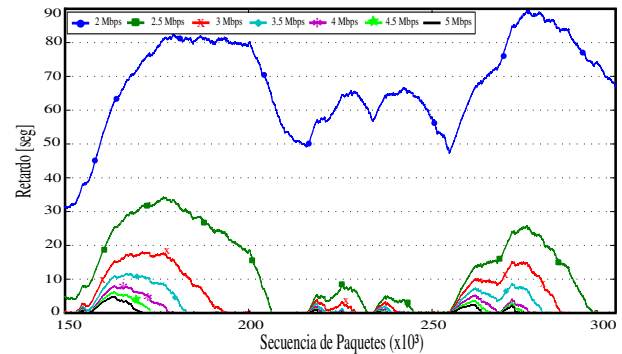


Fig. 7. Retardo introducido para cada nivel de alisado.

los paquetes que llegan a ráfagas, se reducen considerablemente debido a que se añade un retardo en algunos de dichos paquetes. Esto ayuda a evitar la congestión en los *buffers* y, por lo tanto, la pérdida de paquetes.

b) *Estimación de retardo*: Para cada algoritmo utilizado, se obtiene el retardo añadido a cada paquete de la traza. Para mostrar el peor caso (el mayor retardo introducido), la Fig. 7 presenta un zoom del retardo añadido a cada paquete para cada uno de los niveles de alisado (series de 2 a 5 Mbps). Para el nivel más alto de alisado (2 Mbps), se introducen retardos intolerables por SopCast y para cualquier otro servicio en tiempo real (≈ 90 segundos). Sin embargo, a medida que se disminuye el umbral de alisado, también disminuyen el valor y la frecuencia de los retardos añadidos. Se puede observar una cantidad significativa de paquetes que no cambian su tiempo de envío. De la Fig. 7 se deduce que los retardos añadidos ocurren cuando los paquetes llegan en ráfagas.

c) *Selección de conformado de tráfico*: Para cada nivel de alisado, se obtienen los retardos añadidos a cada paquete con el objetivo de determinar el umbral de alisado apropiado a aplicar a este tipo de flujo (Fig. 7). Como ejemplo, se puede mencionar que al 80% de los paquetes, de la traza alisada a 3 Mbps, se les ha añadido un retardo inferior a los 20 segundos, el cual es un valor tolerado por SopCast de acuerdo con [7]. Esta es la razón por la cual se ha seleccionado la traza alisada a 3 Mbps para las pruebas que se presentan en la Sección IV. Estos valores de retardo se han seleccionado en función del comportamiento del tráfico, para controlar la pérdida de paquetes causada por la llegada del tráfico a ráfagas y para mejorar la calidad de la comunicación.

IV. PRUEBAS Y RESULTADOS

En esta sección se presentan dos pruebas diferentes para analizar la pérdida de *bytes* y de paquetes, y también el *throughput* logrado por diferentes aplicaciones cuando comparten un mismo enlace de subida. Inicialmente, se hace un análisis de la pérdida de *bytes* y de paquetes por separado, aunque son dos parámetros que están muy relacionados. Por un lado, analizar la pérdida de *bytes* ayuda a determinar si el *throughput* de la aplicación aumenta o disminuye. Por otro lado, es necesario determinar la naturaleza de los paquetes perdidos (si son grandes o pequeños), ya que el funcionamiento

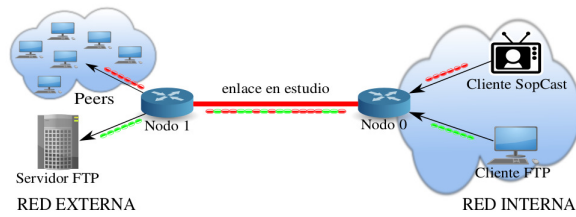


Fig. 8. Escenario de pruebas.

de la aplicación P2P-TV se vería afectado en función de qué paquetes se pierden (Sección II-B).

En la Prueba 1, los parámetros de QoS se analizan cuando la aplicación P2P-TV tiene acceso a toda la capacidad del enlace de subida. Por otro lado, en la Prueba 2 se estudia un entorno más realista, donde SopCast comparte el enlace de subida con un tráfico de fondo, el cual consiste en un servicio FTP que utiliza TCP-SACK (una de las variantes más utilizadas del protocolo TCP [22]) como protocolo de transporte. Para dicha prueba se analizan la tasa de pérdida y el *throughput* alcanzado por ambas aplicaciones.

En ambas pruebas, se seleccionaron siete valores para la capacidad de enlace ascendente (2, 2.5, 3, 3.5, 4, 4.5 y 5 Mbps), y para cada uno de ellos, a la traza de SopCast se le aplica ocho niveles de alisado (2, 2.5, 3, 3.5, 4, 4.5, 5 Mbps y sin alisar), por lo que tenemos 56 situaciones diferentes para cada una de las pruebas. Los resultados muestran el promedio de 100 simulaciones para cada situación. La traza de 18 minutos seleccionada en la Sección III-B1 se ha utilizado en ambas pruebas. Todos los datos podrían ser comparables entre diferentes tecnologías de acceso (por ejemplo, *Ethernet*, *ADSL* y *cable módem*) utilizando capacidades de enlace similares, debido a que el análisis de los flujos se realiza a nivel de IP.

A. Entorno de las Pruebas

La Fig. 8 muestra el escenario utilizado para las Pruebas 1 y 2, donde se ha analizado el comportamiento de los parámetros de QoS. Para la selección e implementación del escenario de pruebas se han considerado puntos sensibles y críticos de las redes de acceso, principalmente los entornos domésticos los cuales presentan limitaciones. En este caso el enlace de subida es el más restrictivo y puede ser compartido por el tráfico de diversas aplicaciones.

En la red interna, un cliente P2P-TV y un cliente FTP comparten el mismo enlace de acceso a Internet, que es el foco de este análisis. El otro extremo está compuesto por una red externa, con cientos de *peers* P2P-TV dispersos por Internet, y un servidor FTP. La capacidad del enlace de subida varía entre 2 y 5 Mbps, se ha tenido en cuenta el análisis preliminar del *throughput* medio alcanzado por la aplicación P2P-TV seleccionada. No se han considerado capacidades de enlace inferiores a 1.84 Mbps, para evitar el deterioro del rendimiento de la aplicación. Las pruebas se repetirán para diferentes capacidades del enlace y para cada nivel de alisado de SopCast, además, se obtendrá la pérdida de paquetes y de *bytes* y el *throughput* logrado para la traza de SopCast sin alisar.

En muchas tecnologías de redes de acceso, el enlace de subida es el enlace más restrictivo con una capacidad significativamente menor en comparación con el enlace de bajada; por lo tanto, es el lugar donde el estudio de este fenómeno puede resultar más interesante. En el Nodo 0, ciertos parámetros de red son modificados: se implementa un *buffer* de 50 paquetes con una política de gestión *drop-tail* (estos valores son ampliamente utilizados en dispositivos de acceso comercial [12]), y la capacidad del enlace se varía, como se mencionó anteriormente. Además, el enlace de bajada está dimensionado con una capacidad de enlace de 100 Mbps y un *buffer* de 500 paquetes, para evitar pérdidas en este enlace. Este estudio se centra en el comportamiento del enlace entre el Nodo 0 y el Nodo 1, analizándose algunos parámetros de QoS, como la pérdida de *bytes* y paquetes, el retardo y el *throughput* alcanzado por cada una de las aplicaciones que comparten el enlace.

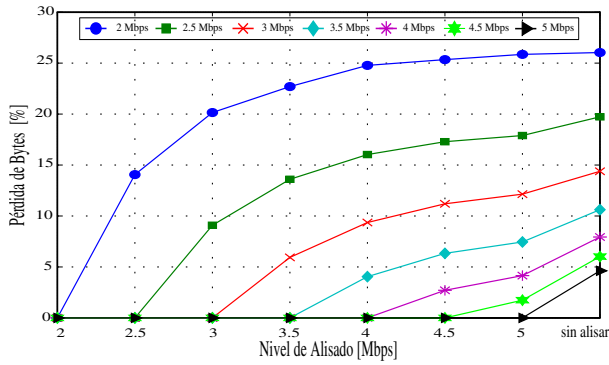
La topología de red propuesta en la Fig. 8 se ha implementado usando *Network Simulator (NS)*. Las trazas reales de SopCast se han capturado y procesado previamente, como se explicó en la Sección III, y luego se generan como flujos UDP en *NS*, utilizando los mismos tamaños de paquetes y tiempos entre paquetes de la traza original. Para el flujo de datos FTP (Prueba 2), se emplearon agentes TCP-SACK y FTP, utilizando las implementaciones estándares de *NS* para dichos protocolos. Como ocurre en un escenario real, los flujos no se inician simultáneamente, por lo que la simulación tiene un período inicial en el cual los flujos comienzan aleatoriamente.

B. Prueba 1: Análisis de QoS de P2P-TV sin Tráfico de Fondo

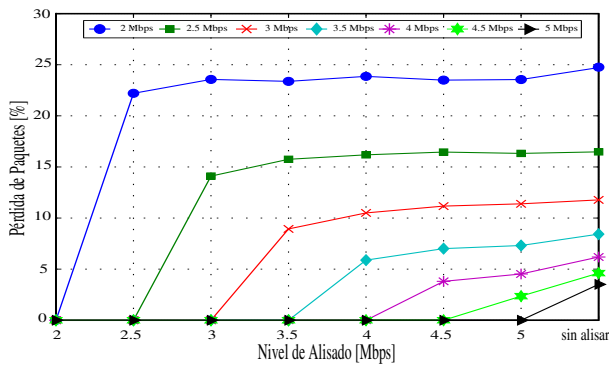
La Fig. 9a y 9b representan el porcentaje de pérdida de *bytes* y de paquetes para la aplicación P2P-TV, respectivamente. En el eje X se muestran los niveles de alisado (de 2 Mbps a 5 Mbps) utilizados en las pruebas; también se incluyen los resultados para la traza sin alisar. Cada serie representa una capacidad de enlace diferente. Como es de esperar, en todos los casos la pérdida de *bytes* y paquetes disminuye a medida que la capacidad del enlace es mayor, debido a que el enlace presenta una capacidad suficiente para soportar las ráfagas generadas por la aplicación P2P-TV, y por lo tanto, la pérdida de paquetes en el *buffer* es menor.

Por ejemplo, en la Fig. 9a, para la traza sin alisar y una capacidad de enlace de 2 Mbps, la pérdida de *bytes* es aproximadamente del 26%, mientras que para un enlace de 5 Mbps se reduce a casi al 5%. Además, aunque los umbrales de alisado disminuyen, también lo hace la tasa de pérdida de *bytes*. Cuando el nivel de alisado es igual o inferior a la capacidad del enlace, la tasa de pérdida se reduce al 0%. Esto ocurre porque la traza alisada no presenta picos de *throughput* por encima del umbral de alisado utilizado, y por lo tanto, la probabilidad de pérdida causada por el desbordamiento del *buffer* se anula. Como ejemplo, para una capacidad de enlace de 3.5 Mbps, la traza sin alisar tiene aproximadamente un 10% de pérdida de *bytes* y a medida que los umbrales de alisado disminuyen, la pérdida de paquetes desaparece. Este patrón es el mismo para todas las series representadas.

La Fig. 9b muestra el porcentaje de pérdida de paquetes con un comportamiento similar al que se muestra en la Fig. 9a. La



(a) Bytes



(b) Paquetes

Fig. 9. Pérdida de paquetes P2P-TV a) bytes b) paquetes.

pérdida de paquetes disminuye a medida que los umbrales de alisado disminuyen. Cuando los niveles de alisado son iguales o inferiores a la capacidad del enlace, no hay pérdida de paquetes.

Al mismo tiempo, se puede observar que el porcentaje de pérdida de paquetes es ligeramente menor que el porcentaje de pérdida de bytes en todos los casos: en la Fig. 9b se puede ver que la pérdida de paquetes aumenta abruptamente cuando los niveles de alisado están cerca de la capacidad del enlace. Sin embargo, hay un punto en el que el porcentaje de pérdida de paquetes varía muy poco con el aumento de los umbrales de alisado. Además, en la Fig. 9a se puede observar que el porcentaje de pérdida de bytes aumenta sin cambios abruptos, por lo que se puede concluir que los paquetes pequeños son más susceptibles de ser descartados para niveles de alisado cercanos a la capacidad del enlace; sin embargo, para niveles más altos de alisado, los paquetes grandes son más susceptibles de ser descartados. Además, se puede observar que no hay pérdida de paquetes para niveles de alisado inferiores o iguales a la capacidad de enlace utilizada, pero se agregan valores muy altos de retardo (como se explica en la Sección III-B2), que no pueden tolerar los usuarios de aplicaciones P2P-TV u otros servicios en tiempo real.

La Fig. 10 muestra el *throughput* alcanzado por la aplicación P2P-TV usando un enlace ascendente de 3 Mbps. En el eje X, se muestran los niveles de alisado utilizados y la traza sin alisar; en el eje Y, se presenta el *throughput* (en Mbps) alcanzado y la capacidad del enlace que queda disponible. Los resultados

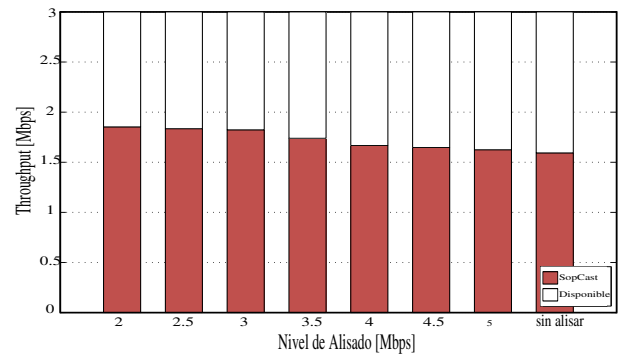


Fig. 10. *Throughput* alcanzado por P2P-TV para una capacidad del enlace de 3 Mbps.

muestran que los recursos de red podrían optimizarse si se aplican técnicas de alisado de tráfico, logrando así un mayor *throughput* de la aplicación bajo las mismas condiciones de red.

Aunque el enlace tiene suficiente capacidad, la aplicación no puede alcanzar el *throughput* necesario (1.84 Mbps) porque el tráfico a ráfagas satura el *buffer* de acceso y causa la pérdida de paquetes. Como se esperaba, si el umbral de alisado disminuye, la ocupación del enlace alcanzada por la aplicación aumenta. Además, el *throughput* logrado por la traza sin alisar es de ≈ 1.55 Mbps; sin embargo, si se utiliza un umbral de alisado de 3 Mbps, se puede alcanzar ≈ 1.84 Mbps. Se puede concluir que en este caso las técnicas de alisado mejoran la utilización del enlace.

C. Prueba 2: Análisis de QoS para P2P-TV con Tráfico de Fondo

En esta prueba, se analiza cómo influye un tráfico de fondo (el cual no se alisa) en la aplicación P2P-TV (original y alisada) cuando comparten el enlace de subida. El tráfico de fondo seleccionado es un servicio FTP que utiliza el protocolo TCP-SACK para el transporte. En la Fig. 11a y 11b se muestran las tasas de pérdida de bytes y de paquetes cuando un cliente P2P-TV y un servicio FTP comparten el mismo enlace de subida. De forma similar a la Fig. 9, cada serie corresponde a diferentes capacidades de enlace. Como se esperaba, la tasa de pérdida disminuye mientras más se alisa la traza. En este escenario, el servicio P2P-TV siempre presenta algunos niveles de pérdida, a diferencia de lo que se muestra en la Fig. 9a.

Como ejemplo se puede mencionar, que para una capacidad de enlace de 3.5 Mbps en la Fig. 11a, y utilizando la traza sin alisar, se obtiene una tasa de pérdida de bytes cercana al 20%, que podría reducirse a un 8% cuando se utilizan un nivel de alisado de 3.5 Mbps y menos del 5% para un nivel de alisado de 2 Mbps. Los niveles de pérdida de bytes en SopCast, independientemente del nivel de alisado y la capacidad de enlace utilizada, se deben a la combinación de flujos FTP y P2P-TV que llenan el *buffer* en el Nodo 0. Por lo tanto, la probabilidad de pérdida de bytes aumenta ligeramente cuando se comparan datos de la Fig. 11a y 9a.

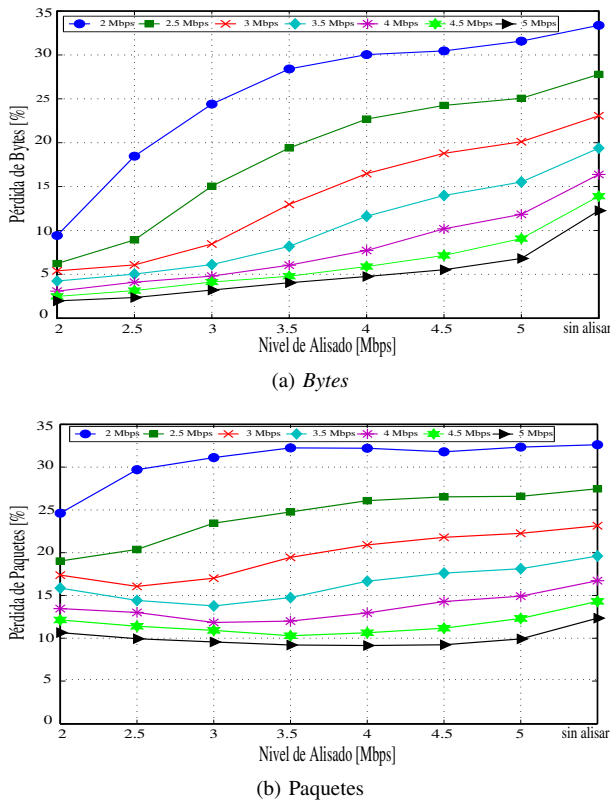


Fig. 11. Pérdida de P2P-TV al compartir el enlace con un servicio FTP a) bytes b) paquetes.

La Fig. 11b muestra el porcentaje de pérdida de paquetes para el flujo P2P-TV. La pérdida de paquetes disminuye a medida que aumenta el nivel de alisado, como sucedió en la Fig. 9b. La diferencia es que, en este caso, aparece un ligero aumento en la pérdida de paquetes para niveles más altos de alisado, lo que podría indicar que los paquetes pequeños tienen una mayor probabilidad de pérdida.

Las Fig. 12a y 12b muestran el comportamiento del *throughput* alcanzado por los flujos P2P-TV y FTP (respectivamente) para cada nivel de alisado de SopCast, cuando ambos servicios comparten el enlace de subida. En el eje X se representa los diferentes niveles de alisado aplicados a SopCast así como también la traza de SopCast sin alisar. Analizando el caso del enlace de 2 Mbps, SopCast alcanza 1.65 Mbps (Fig. 12a) que es un *throughput* mayor que el de la traza sin alisar, que solo alcanza 1.2 Mbps, es decir, la utilización de recursos de red para este tráfico se mejora en aproximadamente 450 Kbps por cada usuario, lo que representa más de un 35%. Por otro lado, el tráfico de fondo FTP (Fig. 12b) disminuye su *throughput* en 400 Kbps (cerca de un 33%) cuando comparte el enlace con la traza de SopCast alisada a 2 Mbps. Por lo tanto, se podría decir que el FTP ha perdido el *throughput* que ha sido ganado por el servicio P2P-TV.

Al igual que en la en la Prueba 1, se ha seleccionado el enlace de 3 Mbps para representar el *throughput* logrado por P2P-TV y el servicio FTP, ya que presentan un comportamiento similar para todas las capacidades de enlace

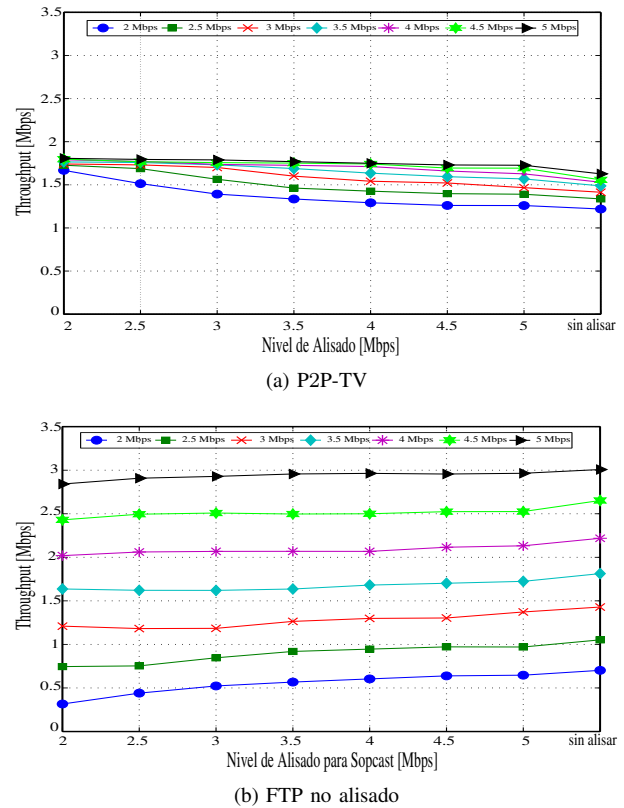


Fig. 12. *Throughput* alcanzado por cada servicio para cada nivel de alisado de SopCast a) P2P-TV b) FTP no alisado

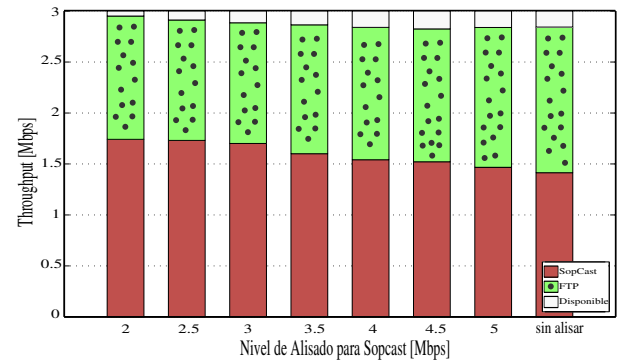


Fig. 13. *Throughput* alcanzado por el tráfico de SopCast (alisado) y el por el FTP (no alisado), para cada nivel de alisado de SopCast en un enlace con capacidad de 3 Mbps.

utilizadas. La Fig. 13 muestra la distribución del *throughput* logrado por cada uno de los tráficos para cada nivel de alisado de la aplicación P2P-TV.

Con el objetivo de determinar los niveles de alisado más apropiados para aplicar al flujo P2P-TV (UDP), las pruebas se repiten para todos los niveles de alisado aplicados a la traza de SopCast. En teoría, la aplicación P2P-TV debería lograr su máximo *throughput* y el resto de la capacidad disponible debería ser utilizado por el servicio FTP. Sin embargo, los resultados representados en la Fig. 13 muestran que SopCast no alcanza el mismo *throughput* en todos los casos, y el FTP no ocupa toda la capacidad del enlace disponible.

También se puede ver que cuando el nivel de alisado es menor o igual que la capacidad del enlace, el *throughput* de la aplicación P2P-TV mejora. Por ejemplo, en el caso de la traza sin alisar, SopCast alcanza aproximadamente 1.45 Mbps y el FTP cerca de 1.40 Mbps, dejando inutilizado 0.15 Mbps. Por otro lado, si se considera el nivel de alisado más alto (2 Mbps), el *throughput* alcanzado por P2P-TV es de 1.75 Mbps, el mayor valor de *throughput* obtenido por SopCast en esta prueba.

Como se ha estudiado en [22], en función de la implementación del protocolo TCP, se conseguirá una mayor o menor ocupación de la capacidad del enlace. Los valores de *throughput* alcanzados por el servicio FTP podrían ser diferentes según la variante de TCP que se utilice (SACK, FACK, Reno, New Reno, Vegas, entre otras), todo depende de si la implementación de TCP es más o menos drástica a la hora de dejar de transmitir tráfico cuando detecta congestión. Otras implementaciones como New Reno y SACK utilizan la información de pérdida de paquetes para reducir su tasa de envío. Por otro lado, Vegas utiliza el aumento del Round-Trip Time (RTT) para prevenir la pérdida de paquetes. Se ha dejado para futuros estudios el análisis del *throughput* en este mismo escenario cuando se utilizan diferentes variantes de implementación del protocolo TCP.

V. CONCLUSIONES

En este documento se ha presentado una metodología que permite conformar el tráfico de aplicaciones en tiempo real, utilizando una técnica de alisado con el fin de restringir el *throughput* instantáneo exigido por la aplicación a niveles razonables, con el objetivo de prevenir la pérdida de paquetes y así lograr un aumento de la QoS. Las pruebas han demostrado que niveles más altos de alisado de tráfico pueden reducir los niveles de pérdida y aumentar el *throughput*. Como contrapartida, se introducen ciertos niveles de retardo en algunos paquetes que llegan en ráfagas al *buffer*. Estos niveles de retardo son tolerados por la aplicación utilizada; para estos casos se recomienda establecer un balance entre el retardo añadido y el *throughput* alcanzado. En trabajos futuros, se estudiará el comportamiento de TCP según la variante empleada, centrándose en su agresividad y cómo esto influye en los parámetros de QoS de las aplicaciones P2P-TV.

AGRADECIMIENTOS

Este trabajo ha sido financiado por DGA/CeNITEQ, Ministerio de Economía y Competitividad TIN2015-64770-R, Fondo Europeo de Desarrollo Regional, Universidad de Zaragoza y Banco Santander.

REFERENCES

- [1] J. P. U. Duque and N. G. Gomez, "Throughput analysis of p2p video streaming on single-hop wireless networks," *IEEE Latin America Transactions*, vol. 13, no. 11, pp. 3684–3689, Nov 2015.
- [2] M. Sanna and E. Izquierdo, "Live scalable video streaming on peer-to-peer overlays with network coding," *IEEE Latin America Transactions*, vol. 11, no. 3, pp. 962–968, May 2013.

- [3] S. Tang, Y. Lu, J. M. Hernández, F. A. Kuipers, and P. V. Mieghem, "Topology dynamics in a P2PTV network," in *Networking*, ser. Lecture Notes in Computer Science, vol. 5550. Springer, 2009, pp. 326–337.
- [4] M. Cha, P. Rodriguez, S. Moon, and J. Crowcroft, "On next-generation telco-managed p2p tv architectures," in *IPTPS'08 Proceedings of the 7th international conference on Peer-to-peer systems*, 2011.
- [5] Y. Liu, Y. Guo, and C. Liang, "A survey on peer-to-peer video streaming systems," *Peer-to-Peer Networking and Applications*, vol. 1, no. 1, pp. 18–28, Mar 2008.
- [6] U. R. Krieger and R. Schwesinger, "Analysis and quality assessment of peer-to-peer iptv systems," in *IEEE International Symposium on Consumer Electronics*, April 2008, pp. 1–4.
- [7] G. Marfia, A. Sentivelli, S. Tewari, M. Gerla, and L. Kleinrock, "Will IPTV ride the peer-to-peer stream?" *IEEE Communications Magazine Special Issue on Peer-to-Peer Streaming*, June 2007.
- [8] J. Saldana, J. Fernández-Navajas, J. Ruiz-Mas, and L. Casadesus, "On-line games traffic multiplexing: Analysis and effect in access networks," *KSII Transactions on Internet & Information Systems*, vol. 6, 2012.
- [9] L. Sequeira, J. Navajas, and J. Saldana, "The effect of the buffer size in qos for multimedia and bursty traffic: When an upgrade becomes a downgrade," *KSII Transactions on Internet and Information Systems*, vol. 8, no. 2, pp. 3159–3176, 09 2014.
- [10] W. Rutherford, L. Jorgenson, M. Siegert, P. Van Epp, and L. Liu, "16 000–64 000 b pmtu experiments with simulation: The case for super jumbo frames at supercomputing05," *Optical Switching and Networking*, vol. 4, no. 2, pp. 121–130, 2007.
- [11] R. Stanojevi and R. Shorten, "Trading link utilization for queuing delays: An adaptive approach," *Computer Communications*, vol. 33, no. 9, pp. 1108 – 1121, 2010.
- [12] L. Sequeira, J. Fernández-Navajas, J. Saldana, J. R. Gállego, and M. Canales, "Describing the access network by means of router buffer modelling: a new methodology," *The Scientific World Journal*, vol. 2014, 2014.
- [13] F. Chang, W.-c. Feng, W.-c. Feng, and J. Walpole, "Provisioning on-line games: A traffic analysis of a busy counter-strike server," *Computer Communication Review*, vol. 32, p. 18, 01 2002.
- [14] A. Borges, P. Gomes, J. Nacif, R. Mantini, J. M. Almeida, and S. Campos, "Characterizing sopcast client behavior," *Computer Communications*, vol. 35, no. 8, pp. 1004 – 1016, 2012.
- [15] T. Silverston and O. Fourmaux, "Measuring p2p iptv systems," in *Proceedings of NOSSDAV*, vol. 7, 2007, p. 2.
- [16] I. Quintana, J. Ruíz-Mas, J. Navajas, L. Casadesus, L. Sequeira, and J. Saldana, "Influencia del buffer del router en la distribución de vídeo p2p-tv," 09 2012.
- [17] B. Fallica, Y. Lu, F. Kuipers, R. Kooij, and P. Van Mieghem, "On the quality of experience of sopcast," in *International Conference on Next Generation Mobile Applications, Services, and Technologies*, 2008.
- [18] A. Vakili and J.-C. GréGoire, "Qoe management for video conferencing applications," *Comput. Netw.*, vol. 57, no. 7, pp. 1726–1738, May 2013.
- [19] D. Bonfiglio, M. Mellia, M. Meo, N. Ritacca, and D. Rossi, "Tracking down skype traffic," in *IEEE INFOCOM 2008 - The 27th Conference on Computer Communications*, April 2008, pp. 261–265.
- [20] M. Amiri, H. A. Osman, and S. Shirmohammadi, "Datacenter traffic shaping for delay reduction in cloud gaming," in *2016 IEEE International Symposium on Multimedia (ISM)*, Dec 2016, pp. 569–574.
- [21] E. Alessandria, M. Gallo, E. Leonardi, M. Mellia, and M. Meo, "P2p-tv systems under adverse network conditions: A measurement study," in *IEEE INFOCOM*, April 2009, pp. 100–108.
- [22] J. Saldana, M. Suznjevic, L. Sequeira, J. Fernandez-Navajas, M. Matijasevic, and J. Ruiz-Mas, "The effect of tcp variants on the coexistence of mmorpg and best-effort traffic," in *International Conference on Computer Communications and Networks (ICCCN)*, July 2012, pp. 1–5.



Idelkys Quintana received her M.Sc. in Telecommunications from Central University of Las Villas in 2010. She is currently a Ph.D. candidate at University of Zaragoza. Her research interests include network optimization and QoS/QoE in multimedia services.



Luis Sequeira received his Ph.D. (Cum Laude) in Information Technologies at University of Zaragoza in 2015. Currently, he is a Research Associate at Kings College London working on QoS, SDN and orchestration of network services.



Julian Fernandez received his Ph.D. in Telecommunications from University of Zaragoza in 2000. He is currently an Associate Professor at University of Zaragoza. His research activity lies in wireless networks, distributed multimedia system and QoS/QoE.



Jose Ruiz received his Ph.D. degree in Telecommunications from University of Zaragoza in 2001. He is currently an Associate Professor at University of Zaragoza. His research activity lies in wireless networks, distributed multimedia system and QoS/QoE.



Jose Saldana received his Ph.D. in Information Technologies in 2011 from University of Zaragoza. He is currently a Senior Researcher at University of Zaragoza. His research interests focus on QoS, traffic optimization and management in Wireless LANs.