

# Evaluación de la Calidad Subjetiva de Juegos Online según el Dispositivo de Acceso

José M<sup>a</sup> Saldaña, Julián Fernández-Navajas, José Ruiz-Mas, Luis Casadesus

E-mail: {jsaldana, navajas, jruiz, luis.casadesus}@unizar.es

Grupo de Tecnologías de las Comunicaciones – Instituto de Investigación en Ingeniería de Aragón

Dpt. IEC. Centro Politécnico Superior Universidad de Zaragoza

Edif. Ada Byron, 50018, Zaragoza

**Abstract-** A study of the effect of access router buffer on subjective quality for a First Person Shooter (FPS) online game is presented in this work. Since mid or low-end routers are frequently used in access networks, their parameters and buffer policies have to be considered in order to warrant an acceptable quality for the users of real-time multimedia services. Subjective quality estimators depending on network impairments have been developed for different services. Some studies have shown that the key parameters for FPS games are delay and jitter. So the influence of the buffer size on these parameters has been studied, taking into account the bandwidth limit of the access network. A subjective quality estimator based on delay and jitter is used, showing a surprising “valley” behavior when bandwidth limit is reached, in which quality grows with the amount of background traffic. This is mainly caused by the jitter reduction when the queue is full.

## I. INTRODUCCIÓN

Internet fue diseñada como una red que entrega la información sin garantizar un valor máximo para el retardo. Con el paso de los años, la red ha crecido y también se utiliza para proporcionar servicios con requerimientos estrictos de tiempo real. Uno de los servicios de este tipo que más rápidamente ha crecido es el denominado VoIP (*Voice over IP*, voz sobre IP), que actualmente está reemplazando muchos sistemas de telefonía tradicionales. Pero este cambio se debe hacer de manera que los usuarios no noten una disminución en la calidad del servicio.

Por ello, se han propuesto estimadores de calidad que, una vez identificados los problemas de la red, cuantifican su efecto en la calidad percibida por los usuarios. Uno de ellos es el *E-Model* [1] de la ITU que, a partir del retardo, las pérdidas de paquetes y otros parámetros como por ejemplo el *codec* utilizado, obtiene una estimación de la calidad, sin necesidad de realizar encuestas a los usuarios cada vez que se pone en marcha un servicio.

Otro de los servicios de tiempo real de más éxito son los juegos *online*. En este artículo estudiaremos un género de juegos con unos requerimientos de tiempo real muy estrictos: los FPS (*First Person Shooter*, tirador en primera persona), que crean un escenario virtual compartido por unas decenas de usuarios que deben cumplir una misión o acabar con los enemigos. Utilizan el protocolo UDP, por ser más adecuado para su gran interactividad, teniendo en cuenta que el jugador tiene un control directo sobre el personaje que maneja.

Algunos estudios [2], [3] han buscado desarrollar estimadores de calidad similares al *E-Model*, pero adaptados a este tipo de juegos. Como veremos, en primer lugar se deben identificar los parámetros de la red con mayor influencia en la calidad percibida, para después combinarlos

adecuadamente y obtener así una fórmula del MOS (*Mean Opinion Score*, puntuación de la opinión media). Como veremos, este tipo de juegos consideran como parámetros clave el retardo y el *jitter* (variación del retardo), en lugar del retardo y las pérdidas de paquetes, como ocurría para VoIP.

Este hecho tiene muchas implicaciones, teniendo en cuenta el escenario en el que se suelen utilizar este tipo de aplicaciones: los jugadores se conectan mediante redes de acceso con un ancho de banda limitado y *router* de gama media o baja. Estos *router* tienen comportamientos diferentes en función de la implementación de su *buffer* o cola de salida. En este artículo estudiaremos el efecto de este dispositivo en los estimadores de calidad basados en el retardo y el *jitter*.

La siguiente sección del artículo discute los trabajos relacionados. La sección III presenta las pruebas y resultados, y el artículo se cierra con las conclusiones.

## II. TRABAJOS RELACIONADOS

En los últimos años se han publicado muchos estudios sobre la problemática del dimensionado del *buffer* [4]. El problema se ha abordado principalmente para *router* del núcleo de la red, y para flujos TCP. Como resumen, podemos decir que la tradicional regla de utilizar el producto del retardo por el ancho de banda fue rebatida por el denominado *Stanford model* [5], que proponía el uso de *buffer* más pequeños. Posteriormente [6] se han propuesto tamaños aún menores (denominados *tiny buffer*) con una capacidad de varias decenas de paquetes, que resultan suficientes para obtener una utilización del 80 o 90% en el enlace. El trabajo que presentamos se centra en los problemas que experimenta el tráfico de tiempo real cuando comparte el enlace con otros tráficos. Se considerarán *router* de gama media o baja, por ser los más utilizados en nuestro escenario. Se probarán diferentes tamaños del *buffer* presentando comparativas para mostrar la influencia de este parámetro en los servicios de tiempo real.

El problema de cómo medir la calidad subjetiva en juegos *online* se ha tratado en varios trabajos. En [7] se consideró solamente el retardo, y se obtuvo una expresión para el MOS a partir de este único parámetro. Otro estudio [8] se basó en encuestas a jugadores reales en un entorno de red que añadía retardos y pérdidas controlados; se comprobó que los resultados eran distintos según el juego utilizado: mientras que *Halo* dejaba de funcionar a partir del 4% de pérdidas, *Quake III* funcionaba bien, y los jugadores no notaban un descenso de la calidad hasta que las pérdidas llegaban al 35%. La causa de este sorprendente resultado es que algunos

juegos incorporan métodos para ocultar a los jugadores estas limitaciones de la red. Por otro lado, en [9] se evaluaron cuatro juegos. Sólo se consideraron el retardo y el *jitter*, y los resultados mostraron diferentes niveles de calidad subjetiva para cada juego.

El primer modelo de MOS para un FPS (*Quake IV*) se presentó en [2]. Este juego, como su predecesor, incorpora un algoritmo para ocultar al usuario las pérdidas de paquetes. Por eso, se utilizaron el retardo y el *jitter* como parámetros de red a considerar en el modelo. Se obtuvo una fórmula polinómica para el MOS, utilizando regresión multidimensional.

En trabajos anteriores nuestro grupo estudió la influencia del tamaño del *buffer* para VoIP [10], y los resultados mostraron que el MOS tiene un comportamiento monótono decreciente, según aumenta el tráfico de fondo. En ese caso, la influencia del *jitter* no se tuvo en cuenta directamente, dado que las aplicaciones de voz incluyen un *buffer de de-jitter*, que añade un retardo y descarta los paquetes que llegan tarde para ser reproducidos. En el presente estudio, dado que los juegos no son normalmente una aplicación abierta, no podremos saber en qué casos se usa un *buffer de de-jitter*.

### III. PRUEBAS Y RESULTADOS

#### A. Escenarios de interés

El escenario que estudiaremos puede verse en la Fig. 1: un número de máquinas están conectadas al servidor a través del mismo *router*. Este escenario puede encontrarse, por ejemplo, en un *cibercafé*, donde los juegos *online* son una de las actividades más comunes. Este escenario se estudió con mayor profundidad en [11]. La conexión a Internet se comparte entre varios servicios, algunos de ellos de tiempo real. El tráfico total ofrecido al *router* puede llegar a exceder la capacidad del enlace; en esos momentos, el *router* descartará paquetes y añadirá retardo, dependiendo de su tamaño y de las políticas de gestión de la cola que incluya.

#### B. Metodología de las pruebas

Para las pruebas se ha elegido un juego FPS, por ser una de las aplicaciones con mayores requisitos de tiempo real. Se ha seleccionado *Quake IV*, para el que existe un modelo de calidad [2] que obtiene un MOS a partir del retardo y el *jitter*. En primer lugar, se necesitan trazas de tráfico del juego, que se han obtenido del proyecto CAIA [12]. Las trazas se capturaron en partidas reales, realizadas en condiciones controladas, y contienen 5.000 paquetes por usuario. La Fig. 2 muestra los histogramas del tamaño de paquete y del tiempo entre paquetes. Puede observarse que el tamaño medio a nivel IP es de 79,5 bytes, y que el juego genera 64 paquetes por segundo. Por tanto, el ancho de banda es de unos 40,7 kbps por jugador.

Estudiaremos el enlace más restrictivo, que en este caso es el de subida, dado que muchas tecnologías de acceso son asimétricas, con un ancho de banda mucho menor en el *uplink* que en el *downlink*. En primer lugar, se ha separado el tráfico cliente-a-servidor, que se combinará para distintos números de usuarios, como ya se hizo en [11].

En este estudio se considerará a 20 jugadores compartiendo un acceso con 2 Mbps o 3 Mbps de subida, y un *buffer drop-tail* FIFO (Fig. 1) con cuatro posibles tamaños: 10, 20, 50 y 100 kB, que se pueden considerar *tiny buffer* [4]. Se han realizado simulaciones en Matlab, introduciendo las

trazas de tráfico del juego, junto con un tráfico de fondo, en el *buffer*. El tráfico de fondo varía desde 0 hasta el ancho de banda de la conexión. Sigue una distribución de Poisson para el tiempo entre paquetes, con los siguientes tamaños: el 50% son de 40 bytes a nivel IP, el 10% son de 576, y el restante 40% son de 1.500 bytes [13]. Para cada punto de las gráficas se han simulado 400 segundos de tráfico. El retardo se mide como RTT (*Round-Trip-Time*, retardo de ida y vuelta), y se compone del tiempo en el *buffer* y de un tiempo adicional de transmisión, con 30 ms de media, y varianza 5, que puede ser un retardo típico en un escenario intra-regional [14]. El retardo máximo aceptable por los jugadores se suele considerar en torno a 200 o 225 ms [8].

#### C. Resultados de las pruebas

La Fig. 3 a) muestra el RTT para la conexión de 2Mbps. Por claridad, se han representado las gráficas de 2 y 3 Mbps utilizando la misma escala en el eje horizontal. Puede observarse que, al alcanzar el límite de ancho de banda, cuando el tráfico de fondo es de 1200 kbps, el retardo crece proporcionalmente al tamaño del *buffer*. Vemos que el *buffer* de 100 kB no se debe usar si el tráfico está por encima del límite, pues añade un gran retardo. El *buffer* de 50 kB estaría justo en el límite de retardo aceptable (220 ms).

La Fig. 3 b) muestra un comportamiento similar, creciendo cuando se alcanza el límite. Pero en este caso, el hecho de tener un mayor ancho de banda hace que los valores de retardo mejoren. Por ejemplo, el *buffer* de 50 kB produce retardos en torno a 150 ms.

El modelo utilizado [2] mide el *jitter* como la desviación estándar del retardo. La Fig. 4 a) muestra su valor para la conexión de 2 Mbps. Puede verse que con el *buffer* más pequeño se obtiene un buen comportamiento pero, según aumenta el tamaño, aparece un pico que corresponde al límite del ancho de banda. La causa es que, al saturarse el enlace, crece la ocupación del *buffer*. Pero por encima del límite, el *buffer* estará siempre lleno, lo que hace que todos los paquetes tengan el mismo retardo y, por tanto, el *jitter* se reduzca considerablemente. Si comparamos las Fig. 4 a) y 4 b), podemos ver que el pico para el *buffer* de 100 kB se ha reducido desde 100 hasta 70 ms, y esta disminución se observa también para el resto de tamaños de *buffer*. Esto sugiere la existencia de una relación entre el límite de ancho

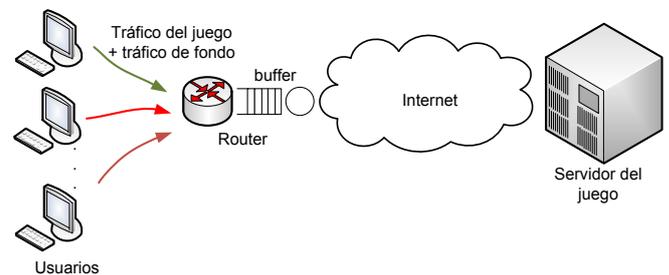


Fig. 1. Escenario de las pruebas

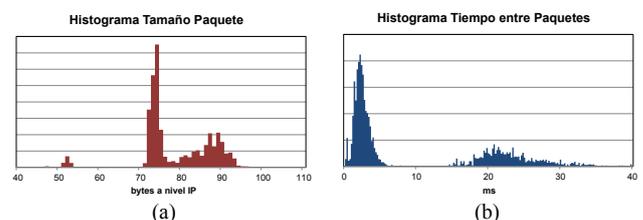


Fig. 2. Histogramas de *Quake IV*: a) tamaño de paquete; b) tiempo entre paquetes

de banda y el máximo tamaño de *buffer* tolerado: a mayor ancho de banda, se podrá utilizar un *buffer* de más tamaño.

La Fig. 5 muestra las pérdidas para cada tamaño de paquete, cuando se usa el *buffer* de 10 kB. Puede observarse que los paquetes pequeños disfrutan de una clara ventaja al utilizarse un *buffer drop-tail*. También nos damos cuenta de que, para un tamaño de *buffer* fijo, las pérdidas son menores a mayor ancho de banda.

Presentaremos finalmente los resultados de calidad percibida. La fórmula propuesta en [2] calcula en primer lugar una variable  $x$  que da cuenta de los parámetros de la red, y que depende del RTT (*ping\_average*) y del *jitter*:

$$x = 0,104 * ping\_average + jitter\_average \quad (1)$$

La fórmula del MOS se obtiene a partir de  $x$ :

$$MOS = -0,00000587x^3 + 0,00139x^2 - 0,114x + 4,37 \quad (2)$$

Como se ha visto en la Sección II, las pérdidas no se consideran significativas si están por debajo de un umbral, dado que este juego incluye un método muy efectivo para ocultar las pérdidas al jugador. Esto también ocurría en *Quake III*, que sorprendentemente funcionaba bien incluso con tasas de pérdida próximas al 35% [8]. En el trabajo que estamos presentando, las pérdidas del juego están siempre por debajo de ese umbral (Fig. 5).

Las Fig. 6 a) y b) presentan el MOS. Aunque algunos estudios, utilizando el mismo criterio que en VoIP, consideran el MOS aceptable a partir de 3,5, otros afirman que un valor de 3 puede ser suficiente, y que los jugadores empiezan a abandonar el servidor cuando el valor llega a 2 [9]. Las gráficas muestran un comportamiento sorprendente, en comparación con el MOS de otros servicios como VoIP, en los que el descenso es monótono decreciente con el aumento del tráfico de fondo. Observamos que, cuando el tráfico de fondo es pequeño, el resultado es el que se podría esperar: a mayor tráfico, peor calidad. Según nos acercamos al límite del ancho de banda, la calidad empeora, debido al

aumento del *jitter*. Pero sorprendentemente, cuando el tráfico ofrecido sobrepasa el límite, la calidad vuelve a crecer, consiguiendo mejores valores. Este comportamiento anómalo, en “valle”, está causado por la reducción del *jitter* mostrada en la Fig. 4.

Si consideramos aceptable el MOS en torno a 3, en ambas figuras podemos observar que sólo el *buffer* más pequeño (10 kB) consigue resultados aceptables para cualquier valor del tráfico de fondo. El resto de *buffer* mantienen la calidad hasta que se llega al 90% del ancho de banda. Pero para el *buffer* de 20 kB con 3 Mbps, si el tráfico ofrecido crece por encima del límite, se vuelven a obtener valores aceptables de MOS.

Así pues, podemos decir que, por un lado, los resultados del retardo no son sorprendentes: según se reduce el *buffer* y aumenta el ancho de banda, el *buffer* puede vaciarse más rápido. Pero el *jitter* presenta un pico, y después vuelve a reducirse porque el *buffer* está siempre lleno, lo que causa el comportamiento anómalo del MOS. Aunque los resultados se han obtenido para un modelo de calidad concreto, el comportamiento será similar para otros modelos de calidad que se basen en el retardo y el *jitter*.

Como resumen de los resultados, se puede destacar la importancia de integrar todos los parámetros de calidad en un valor de MOS. No podemos simplificar el problema estudiando por separado cada parámetro de red. En el caso de la aplicación estudiada son el retardo y el *jitter*, que se ven afectados por el ancho de banda, como era de esperar. Pero los resultados muestran también una gran influencia del tamaño del *buffer*. Se ha visto que un ancho de banda mayor permite *buffer* más grandes. Una aplicación que funciona bien en una red local puede experimentar problemas al atravesar una red de acceso para interactuar con un servidor situado en Internet. Por eso, el tamaño del *buffer* es un parámetro de primera importancia, que debe ser diseñado teniendo en cuenta todas estas relaciones, especialmente en el caso de *router* para redes comerciales de acceso.

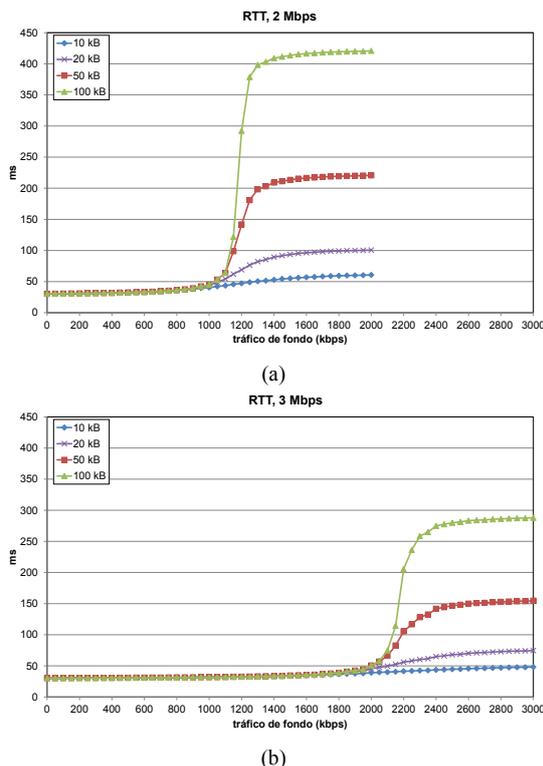


Fig. 3. Retardo en función del tráfico de fondo a) 2Mbps; b) 3 Mbps

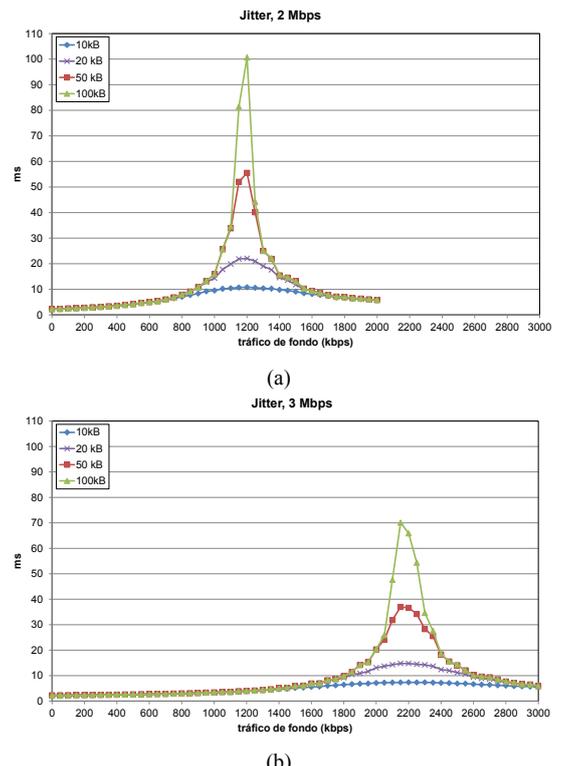
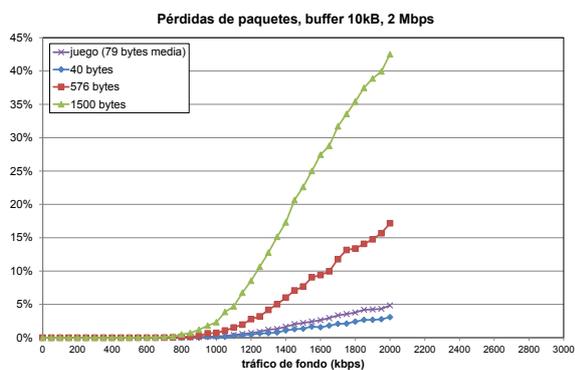
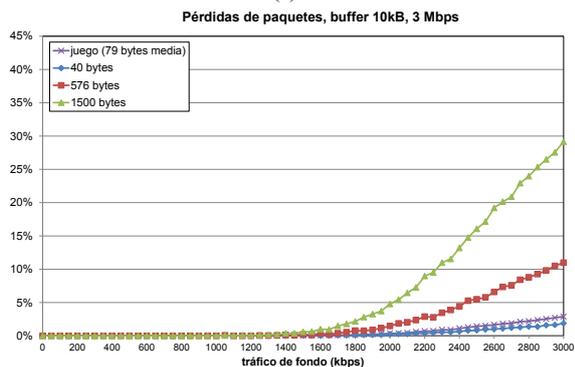


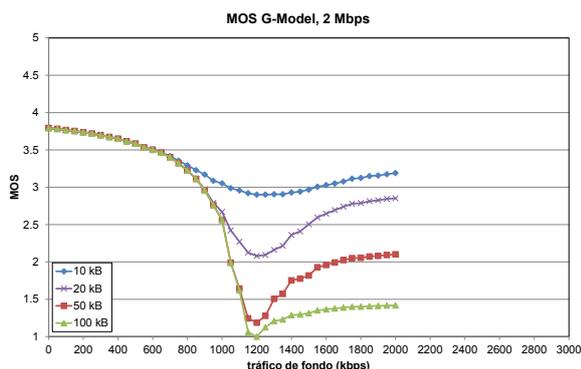
Fig. 4. Jitter en función del tráfico de fondo a) 2Mbps; b) 3 Mbps



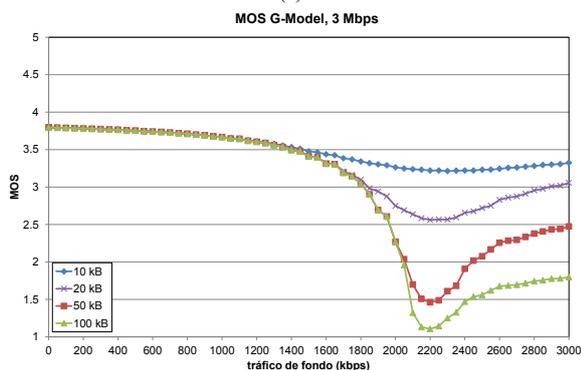
(a)



(b)

Fig. 5. Pérdidas para cada tráfico para el *buffer* de 10kB

(a)



(b)

Fig. 6. MOS en función del tráfico de fondo a) 2Mbps; b) 3 Mbps

#### IV. CONCLUSIONES

En este trabajo se ha estudiado el efecto del dispositivo de acceso a Internet en la calidad percibida para ciertos juegos FPS *online*. Los parámetros de red principales a considerar son el retardo y el *jitter*, mientras que las pérdidas no influyen si están por debajo de cierto umbral.

Las simulaciones realizadas han obtenido el retardo y el *jitter* producidos por el *buffer* de un *router* de acceso, en función del tráfico de fondo. Se han añadido también retardo y *jitter* correspondientes al tiempo de transmisión en la red. Se han combinado trazas reales de un juego para obtener el tráfico de 20 jugadores que comparten el mismo acceso. Los resultados muestran un pico del *jitter*, que produce un “valle” en la gráfica del MOS, produciendo una región anómala, en la que el MOS crece al aumentar el tráfico de fondo.

Los *buffer* pequeños presentan mejores resultados que los grandes. Sería interesante que los desarrolladores de juegos identificasen el comportamiento del *buffer* del *router* para así adaptar el tráfico, teniendo en cuenta la relación entre el tamaño del *buffer* y el ancho de banda. En trabajos futuros se podrá estudiar esta relación cuando se utilizan políticas diferentes a *drop-tail* FIFO en el *buffer*, mostrando los beneficios de priorizar el tráfico de tiempo real.

#### AGRADECIMIENTOS

Este trabajo ha sido financiado parcialmente por el proyecto CPUFLIPI (MICINN TIN2010-17298), el proyecto MBACToIP, de la Agencia Aragón I+D e Ibercaja Obra Social, y el Proyecto Cátedra Telefónica, Univ. Zaragoza.

#### REFERENCIAS

- [1] “The E-model, a computational model for use in transmission planning”, ITU-T Recommendation G.107, 2005.
- [2] A. F. Wattimena, R. E. Kooij, J. M. van Vugt, O. K. Ahmed, “Predicting the perceived quality of a first person shooter: the *Quake IV G-model*”. In Proc. 5th SIGCOMM workshop Network and system support for games (NetGames '06), ACM, New York, NY, USA, 2006.
- [3] M. Ries, P. Svoboda, M. Rupp, “Empirical study of subjective quality for Massive Multiplayer Games”, Systems, Signals and Image Processing, 2008. IWSSIP 2008. 15th International Conference on, pp.181-184, 25-28 June 2008.
- [4] A. Vishwanath, V. Sivaraman, and M. Thottan. “Perspectives on router buffer sizing: recent results and open problems”, SIGCOMM Comput. Commun. Rev. 39, 2 (March 2009), 34-39.
- [5] G. Appenzeller, I. Keslassy, and N. McKeown. “Sizing router buffers”, In SIGCOMM '04, pages 281–292, New York, USA, 2004. ACM Press.
- [6] M. Enachescu, Y. Ganjali, A. Goel, N. McKeown, T. Roughgarden. “Part III: routers with very small buffers”. SIGCOMM Comput. Commun. Rev. 35, 3, pp 83-90, Jul 2005.
- [7] C. Schaefer, T. Enderes, H. Ritter, M. Zitterbart. “Subjective quality assessment for multiplayer real-time games”. In Proc 1st workshop on Network and system support for games (NetGames '02). ACM, New York, NY, USA, 74-78. 2002.
- [8] S. Zander, G. Armitage, “Empirically Measuring the QoS Sensitivity of Interactive Online Game Players”. In Proc. Australian Telecommunications Networks & Applications Conference (ATNAC 2004), Sydney, Australia, Dic. 2004.
- [9] M. Dick, O. Wellnitz, L. Wolf. “Analysis of factors affecting players' performance and perception in multiplayer games”. In Proc. 4th ACM SIGCOMM workshop on Network and system support for games (NetGames '05). ACM, New York, NY, USA, 1-7, 2005.
- [10] J. Saldana, J. Murillo, J. Fernandez-Navajas, J. Ruiz-Mas, E. Viruete Navarro, J. I. Aznar, “QoS and Admission Probability Study for a SIP-Based Central Managed IP Telephony System,” New Technologies, Mobility and Security (NTMS), 2011 4th IFIP Int. Conf., Feb. 2011.
- [11] J. Saldana, J. Murillo, J. Fernandez-Navajas, J. Ruiz-Mas, J. I. Aznar, E. Viruete, “Bandwidth Efficiency Improvement for Online Games by the use of Tunneling, Compressing and Multiplexing Techniques,” Proc. Int. Symp. Performance Evaluation Computer Telecom. Systems SPECTS 2011, pp.227-234, The Hague, Netherlands, Jun. 2011.
- [12] L. Stewart, P. Branch, *Quake4*, Map: q4dm1, 7players, 20Jul2006. Centre for Advanced Internet Architectures SONG Database, [http://caia.swin.edu.au/sitcr/song/files/quake4\\_310706\\_1\\_q4dm1\\_7\\_fragment.tar.gz](http://caia.swin.edu.au/sitcr/song/files/quake4_310706_1_q4dm1_7_fragment.tar.gz)
- [13] Cooperative Association for Internet Data Analysis, “NASA Ames Internet Exchange Packet Length Distributions”.
- [14] AT&T Global IP Network, <http://ipnetwork.bgtmo.ip.att.net/pws>