

Medida y análisis del tráfico multimedia en redes móviles de cuarta generación

Pablo Serrano, Carlos J. Bernardos, Ignacio Soto, José I. Moreno
Departamento de Ingeniería Telemática. Universidad Carlos III de Madrid
Av. Universidad, 30, Edif. Torres Quevedo.
28911 Leganés (Madrid)
Teléfono: 91 624 8878 Fax: 91 624 8749
E-mail: {pablo, cjb, isoto, jmoreno}@it.uc3m.es

***Abstract.** Nowadays, quality of service (QoS) is a major concern in network development, existing several approaches: IntServ, DiffServ... However, design of efficient provisioning mechanisms must take into account a proper traffic characterization, not only quantitative (in terms of bandwidth, burst sizes, ...) but also qualitative (degradation in performance due to losses, jitter, ...). In this paper we present a characterization of multimedia traffic in a prototype of a fourth-generation network (4G) – developed and implemented inside the Moby Dick project-, in order to set a knowledge base to be used both for the development and the evaluation of QoS provisioning mechanisms.*

1 Introducción

Puede decirse que con la alta penetración de la telefonía móvil por un lado, y el uso de Internet por otro, se está implantando una nueva “cultura de uso” de las telecomunicaciones, siendo su ejemplo más representativo el servicio de mensajes cortos (SMS). Debido al indiscutible éxito de este último, recientemente se viene gestando una nueva ola de aplicaciones para los terminales móviles, que bien puede ser vista como resultado de la sinergia entre estos dos mundos tradicionalmente tan separados –la conmutación de circuitos por un lado, la de paquetes por el otro–.

Las características más representativas de esta “nueva ola” de servicios –valgan como ejemplo los servicios de localización de grupos o mensajes multimedia– son las siguientes:

Empleo del “always on”, esto es, hacer uso y facilitar la disponibilidad perpetua al usuario, evitando o minimizando el proceso de acceso al servicio.

Personalización del servicio, bien sea mediante consulta de las preferencias del usuario, bien por el empleo de información obtenida en tiempo real (por ejemplo, la posición geográfica)

Nuevos modelos de facturación: frente al tradicional cobro por tiempo, los nuevos servicios permiten nuevas formas de pago, ya sea por mensajes recibidos, por publicidad introducida...

Simplicidad y sencillez del sistema final: característica muy importante, pues los servicios terminan implementándose sobre terminales de recursos escasos y manejo poco cómodo.

Por otra parte, existe un notable esfuerzo de investigación en el campo de los sistemas “All-IPv6”, esto es: arquitecturas que empleen IPv6 para transportar toda la información de usuario, de control y de gestión. Este esfuerzo se ve motivado e impulsado por la aparición de estos nuevos servicios para redes móviles, dado que se consigue una implantación de nuevas aplicaciones más rápida y eficaz, pudiéndose reutilizar desarrollos en entornos claramente diferenciados (surgiendo redes que integran muy diferentes tecnologías: WLAN, Bluetooth, GPRS...).

Debido a la juventud de este campo de investigación, varias son las iniciativas que existen para potenciar el desarrollo de sistemas “All-IPv6”. De éstas, cabe mencionar el proyecto Moby Dick [1], que pretende proporcionar un mecanismo de gestión eficaz de la macro y micro movilidad, con un sistema de AAC (Authentication, Authorization, Accounting, and Charging) integrado, y asegurando la QoS (Quality of Service) proporcionada a los usuarios; todo ello resultando en una arquitectura de red móvil de cuarta generación (4G).

Enmarcado en dicho proyecto se encuentran las tareas realizadas: proporcionar una caracterización adecuada del tráfico a ser cursado por la red, con objeto de facilitar el desarrollo de un mecanismo eficiente de provisión de QoS [2]; para ello fue preciso emplear aplicaciones IPv6 sobre una plataforma piloto de red 4G [3].

Una adecuada caracterización de los efectos de la pérdida de QoS en el rendimiento de las aplicaciones permite, además, evaluar adecuadamente el comportamiento de la red desde el punto de vista del usuario (por ejemplo, en situaciones de trasposos –

Tabla 1 Tamaño de las medidas efectuadas

Tipo de Flujo	Sentido	Duración (seg)	Pqts	Bytes
Audio Stream ¹		60,77	3.009	1.185.548
Quake 2 ²	Cliente (C) - Servidor (S)	389,38	48.433	4.655.617
	S - C	389,26	23.352	4.105.000
VoIP ³	A-B	114,37	2.841	303.987
	B-A	113,72	3.696	395.472
Video Stream ⁴		137,60	23.860	32.848.140
Tetrix ⁵	C-S	253,59	1.940	160.311
	S-C	253,55	1.938	197.140

Notas: 1) Se empleó el RAT, con codificación "Linear 16"
 2) Partida de 6 jugadores, "todos contra todos"
 3) Conversación mediante el RAT, con el codificador

GSM

4) Se empleó la aplicación VideoLAN y un trailer de 2 minutos de duración

5) Partida de 6 jugadores

handover-, se puede comprobar si éste resulta muy perturbador en la provisión del servicio).

El resto del presente artículo se organiza como sigue: en la sección 2 se definen los objetivos del análisis, la sección 3 presenta las medidas realizadas, analizándose las mismas en la sección 4. La sección 5 analiza la tolerancia de las aplicaciones a pérdidas de QoS, concluyéndose el artículo en la sección 6.

2 Objetivos del análisis

Como se ha indicado, el objetivo del presente análisis es realizar una caracterización "adecuada" del tráfico multimedia más frecuente en redes móviles de cuarta generación. Por "adecuada" entendemos "aquella que permita realizar reservas pertinentes de QoS, siendo posible predecir la degradación de servicio causada ante recortes en los recursos disponibles". Este análisis debe ser realizado en dos fases:

En primer lugar, es preciso caracterizar *cuantitativamente* los flujos multimedia, en términos de ancho de banda y tamaño de paquetes, no sólo para medir las propiedades del flujo sino también para tener valores de partida de cara a la restricción de los mismos.

En segundo lugar, es preciso valorar *cualitativamente* el impacto de la pérdida de QoS en los diferentes flujos: esto es, se trata de analizar la tolerancia de las aplicaciones frente a la disminución de los requisitos de calidad exigidos para un comportamiento óptimo (evaluado éste de forma subjetiva por varios usuarios).

2.1 Entorno de pruebas

A nuestro juicio, las aplicaciones de mayor importancia en las redes 4G pueden clasificarse en cinco grupos:

Telefonía: aplicación imprescindible en redes móviles, que en caso concreto de redes 4G será una heredera de los actuales sistemas de VoIP.

Streaming: tanto de audio como de video, posibilitado por el ancho de banda disponible y las capacidades de los terminales.

Juegos interactivos: de competición entre distintos usuarios finales.

Mensajería instantánea: derivada de los actuales SMS, con nuevas capacidades multimedia (MMS).

Aplicaciones "tradicionales": de uso común en Internet, como puede ser el ftp, e-mail, web.

En el presente artículo nos centraremos en los tres primeros grupos, siendo las aplicaciones escogidas las siguientes:

RAT [4] (Robust Audio Tool), tanto para VoIP como para streaming de audio, basándose en un modelo de comunicación punto a punto.

VideoLAN [5], aplicación que sigue un modelo cliente (C)-servidor (S), empleada para caracterizar el streaming de video.

Quake 2 [6] y **Tetrix** [7], representando dos modelos diferentes de juego interactivo en red, que al igual que el VideoLAN implementan un modelo cliente-servidor. Si bien el primero exige ciertas capacidades gráficas y de manejo no disponibles en los terminales móviles de hoy en día, se puede considerar paradigmático en su campo de aplicación.

Esta selección tuvo como requisitos los siguientes: representatividad de la aplicación, existencia de licencia restrictiva y uso de IPv6. Resulta adecuado indicar que todas las aplicaciones emplean UDP como protocolo de transporte, salvo el Tetrix –que utiliza TCP -.

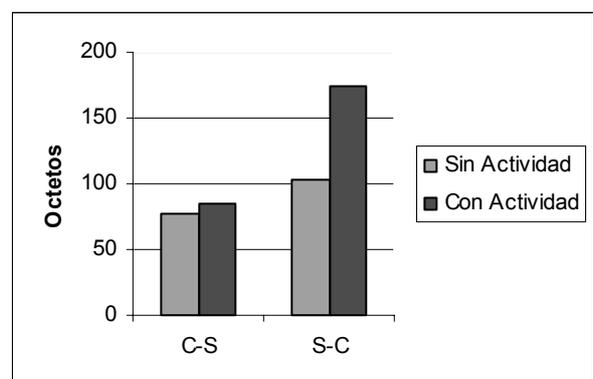


Figura 1 Octetos por paquete generados en función de la actividad de los participantes en una partida de Quake 2 de seis jugadores

La plataforma de pruebas utilizada para realizar las medidas presentadas en este artículo aparece en la Fig. 2. Dicho escenario de pruebas forma parte de la plataforma de pruebas del proyecto Moby Dick, reflejando la estructura de un posible piloto de red 4G. Está formada íntegramente por máquinas con S.O. RedHat-7.2 Linux-2.4.16. Hubo que añadir, no obstante, un elemento exógeno: un nodo intermedio que simulase variaciones en la QoS recibida por la aplicación. Dichas variaciones de consiguieron por medio de la utilización del emulador NISTNET [10]. NISTNET permite a una máquina Linux, configurada como router, emular una amplia variedad de condiciones de red. Este emulador soporta sólo conexiones IPv4, por lo que fue necesario establecer un túnel IPv6-en-IPv4 para poder emplear dicho software. ESCORPION.IPV6 y LARVA10.IPV6 eran los extremos del túnel y el emulador NISTNET se ejecutaba en PULGA.IPV6. Tal y como se aprecia en la Fig. 2, todo paquete IPv6 entre ESCORPION.IPV6 y LARVA10.IPV6 va a través del túnel, pudiendo variarse las condiciones de la conexión (retardo, jitter, pérdidas, etc) entre ambas máquinas mediante NISTNET. Estas variaciones no afectan a las características de la conexión entre otras dos máquinas cualesquiera del escenario.

3 Realización de las medidas

Las medidas fueron tomadas una vez iniciada la aplicación –para evitar efectos de inicio de sesión-. Se realizó una captura de tamaño suficientemente representativo (más de dos mil paquetes por flujo y sentido) mediante el Ethereal [8], para pasar posteriormente a analizar los datos con el Tcpstat [9].

Dos fueron las variables que se midieron: el tamaño del paquete (a nivel IP) y la tasa de inyección de paquetes (muestreada ésta cada medio segundo). Los resultados básicos de la captura se muestran en la Tabla 1.

En primer lugar se realizaron las medidas, para luego emplear el emulador NistNet poder modificar las condiciones de funcionamiento de la aplicación. Partiendo de una caracterización cuantitativa sobreestimada, se fue disminuyendo la QoS recibida en cuatro dimensiones: (de forma independiente) retardo, pérdidas de paquete, ancho de banda disponible, variación del retardo.

4 Características del tráfico

Antes de comentar los resultados obtenidos, es preciso señalar que –por lo general- el tráfico generado por las aplicaciones escogidas depende de la realización particular que se haya efectuado. Esto es: el streaming no genera una tasa de tráfico constante o predecible a priori, sino que depende del fichero en cuestión que se esté transmitiendo. Otro ejemplo más significativo es una conversación de VoIP que implemente detección de silencios: el tráfico generado se comportará a ráfagas. Y en el caso de los juegos interactivos este comportamiento es aún más drástico, dado que los datos generados dependen de la actuación del jugador –las situaciones pasivas (jugadores estáticos) generan menos tráfico que las activas-. Este último hecho se ilustra en la Fig. 1.

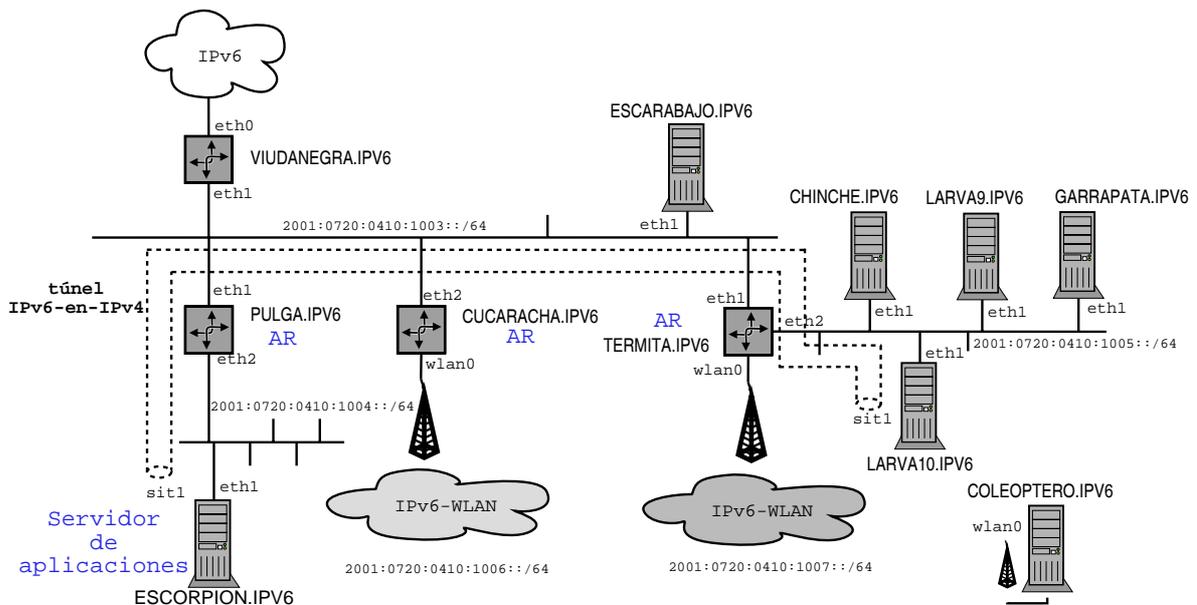


Figura 2. Escenario de pruebas 4G empleado

Tabla 2. Tamaño en octetos de los paquetes IP generados por las aplicaciones

Tipo de Flujo	Sentido	Medio	Máx	Mín	Desv	Desv / Media
Audio Stream		380,00	380	380	0,00	0,00
Quake 2	C-S	82,12	110	73	3,73	0,05
	S-C	161,79	1.051	90	48,12	0,30
VoIP	A-B	93,00	93	93	0,00	0,00
	B-A	93,00	93	93	0,00	0,00
Video Stream		1.363,93	1.364	424	6,09	0,00
Tetris	C-S	68,63	330	46	36,12	0,53
	S-C	87,72	329	60	60,45	0,68

Hechas las anteriores consideraciones, resultan bastante claros los resultados obtenidos -mostrados en las Tablas 2 y 3- donde se han procesado mediante Tcpstat las capturas realizadas con el Ethereal. Se indica, para cada flujo y sentido (cliente-servidor y servidor-cliente para los juegos, sólo este último para el streaming, punto a punto para VoIP), tanto el tamaño medio de los paquetes como la tasa media de inyección de éstos a la red. Además, dado que resulta interesante analizar la variación de estos valores, se presentan las cotas superior e inferior, así como la desviación típica de los mismos -y su valor normalizado con la media, que resulta mucho más representativo-. Se han resaltado los valores de este último cociente superiores a 0.3, que indican una “variabilidad apreciable” del flujo.

A la vista de las tablas, lo primero que se aprecia es que el flujo de streaming apenas varía. Por su parte,

la conversación de VoIP se comporta de acuerdo a lo esperado: la carga de los paquetes es siempre la misma, pero su tasa de inyección varía entre dos extremos (0 pps y 52 pps, este último valor correspondiente al estándar GSM).

El Quake 2 no modifica substancialmente su velocidad de envío de paquetes, sino que modifica la carga de los mismos, debido al empleo de un esquema diferencial para la codificación de la información. No obstante, en ocasiones debe transmitir un paquete de un tamaño considerable, lo cual se explica debido a la necesidad de resincronizar el sistema [11]. Por último, el juego de Tetris modifica notablemente tanto la tasa de inyección de paquetes como el tamaño de los mismos, hecho bastante lógico si se tiene en cuenta el cambio de filosofía que plantea frente al Quake 2: en aquél se establece una conexión TCP, siendo fundamental que cada comando llegue al destino; en

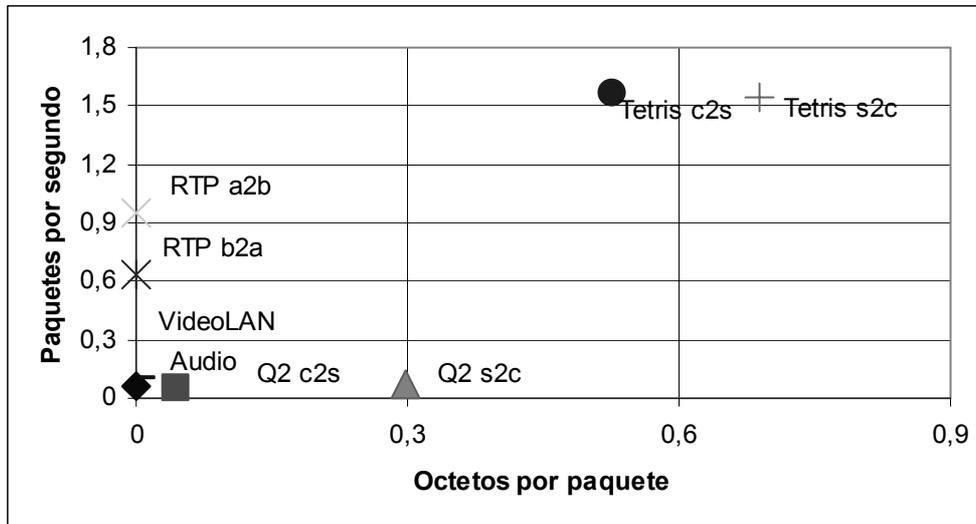


Figura 3. Componentes de variabilidad de las aplicaciones

Tabla 3. Paquetes por segundo generados por las aplicaciones

Tipo de Flujo	Sentido	Medio	Máx	Mín	Desv	Desv / Media
Audio Stream		49,33	52	28	2,86	0,06
Quake 2	C-S	124,35	144	94	6,53	0,05
	S-C	59,95	72	36	4,32	0,07
VoIP	A-B	24,81	52	0	23,52	0,95
	B-A	32,42	52	0	20,67	0,64
Video Stream		172,90	196	18	18,47	0,11
Tetris	C-S	7,64	98	0	11,97	1,57
	S-C	7,63	88	0	11,82	1,55

Tabla 4. Valores de QoS aceptables para las distintas aplicaciones

Aplicación	Pérdidas Paquete	Retardo/Sentido (ms)	Jitter/Sentido (ms)	BW L3 (kbps)
Audio Stream	2 %	∞	100	144
Quake 2	15 %	100	150	24
VoIP	10 %	150	50	30
Tetris	20 %	∞	∞	1

Nota 1: El programa empleado para simular pérdidas en la QoS ofertada a las aplicaciones (NISTNet) no permitió manejar flujos de gran ancho de banda, por lo que fue preciso prescindir de la aplicación de streaming de video.

Nota 2: Los valores marcados como ∞ indican un valor del parámetro superior a 500 ms.

éste, es más importante la caducidad de los mensajes, primando la llegada de información reciente (pues no es excesivamente grave la pérdida de un paquete; el sistema añade redundancia para subsanar esto).

Con objeto de ilustrar los resultados obtenidos, en la Fig. 3 se muestran las componentes de variabilidad de los flujos. Cuatro son pues los tipos de comportamiento observados:

Variabilidad en tasa y carga: el Tetrinet, que emplea TCP como capa de transporte.

Variabilidad sólo en tasa: la aplicación de VoIP, por el esquema de detección de silencios

Variabilidad sólo en carga por paquete: el Quake 2, por el uso de codificación diferencial.

Variabilidad mínima: los flujos de streaming.

De forma ortogonal, resulta interesante resaltar por otro lado el efecto de añadir las cabeceras, dado el – por lo general- reducido tamaño de los paquetes. Esto

se ilustra en la Fig. 4, donde aparece la tasa de tráfico a lo largo de la torre de protocolos; se trata de una cuestión no despreciable para los mecanismos de control de flujo (entre ellos, reserva de QoS): en función del nivel en que se realice la función de control, será preciso emplear un valor u otro (relacionados por la eficiencia de empaquetado).

5 Tolerancia de las aplicaciones

En la Tabla 4 aparecen los umbrales de funcionamiento obtenidos para cada aplicación considerada, estimados aquéllos de forma independiente por cada columna (por ejemplo: el aumento del retardo se analizó con el resto de parámetros no limitados). Se trata, indudablemente, de una situación de funcionamiento ideal, dado que en las redes IP suelen venir de la mano estas “diferentes dimensiones” (pérdidas, retardo, jitter) de la QoS. Por ello, si bien en la Tabla 4 se muestra que con el 80% de paquetes entregados una conversación VoIP se puede cursar aceptablemente, esta pérdida no debe ocurrir a ráfagas: otros autores [12] analizan el tiempo máximo de pérdida aceptable, estimado en 60 ms.

Otra hecho a tener en cuenta es que las aplicaciones pueden ser más o menos tolerantes a pérdidas de paquetes, pero no tanto a disminuciones de ancho de banda. El motivo es claro: las primeras descartan paquetes de forma puntual, espaciadas, mientras que las segundas lo hacen a ráfagas. En una “situación 4G”, ello implicaría que pérdidas en el enlace radio (interferencias, traspasos) podrían ser menos importantes que una situación de congestión puntual en el núcleo de red de cara al funcionamiento percibido. Y por ello mismo, la fiabilidad en el nivel de enlace no es imprescindible para determinadas aplicaciones, pudiendo ser incluso perjudicial.

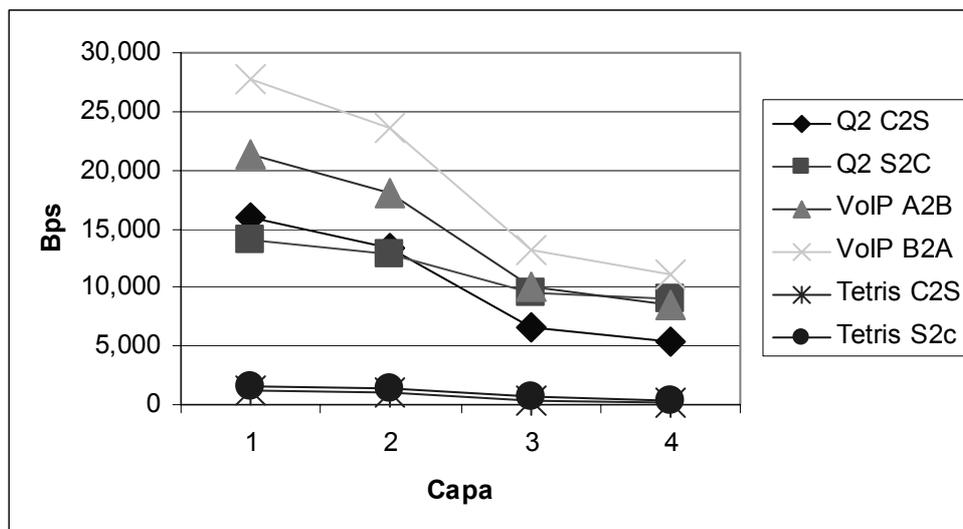


Figura 4. Variación del ancho de banda generado al atravesar la torre de protocolos

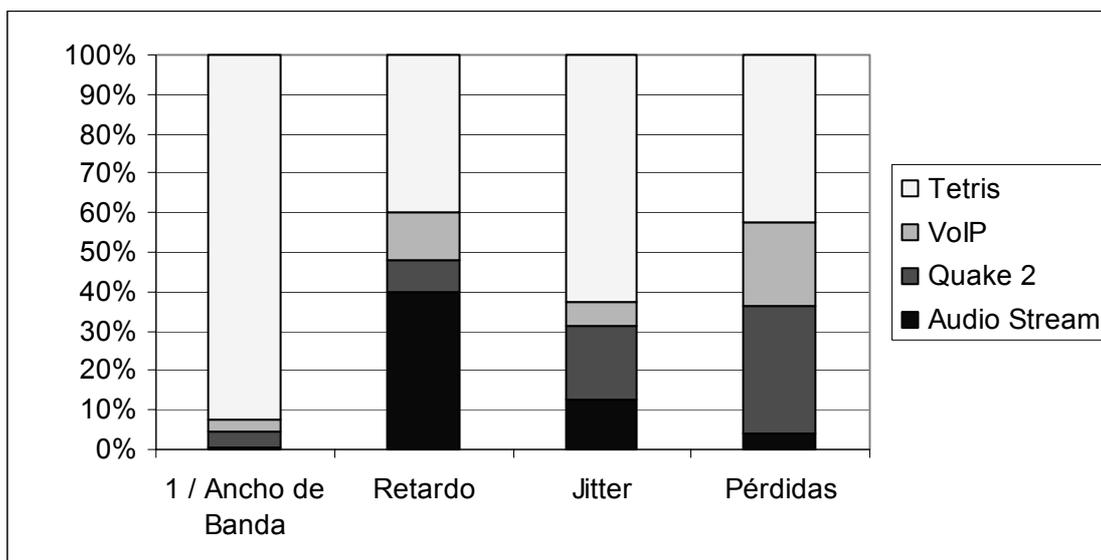


Figura 5. Comparación de requisitos de QoS de las diferentes aplicaciones

Por todo lo anterior, el valor numérico del ancho de banda que aparece en la Tabla 4 es notablemente mayor que el ilustrado en la Fig. 4 (que representa únicamente un valor medio, sin tener en cuenta la existencia de ráfagas).

En la Fig. 5 se comparan las cuatro aplicaciones, desde el punto de vista de sus requisitos de QoS (un mayor tamaño de una columna implica mayor tolerancia o menor necesidad de una determinada variable). Las principales implicaciones son:

Ancho de banda: el Tetrinet es el que menos necesita; el Quake 2 y VoIP se encuentran más o menos igualados, mientras que las necesidades del streaming son muy elevadas.

Retardo: en este caso, es el Quake 2 el más exigente, aunque la aplicación de VoIP tiene unos requisitos similares; streaming y Tetrinet comparten puesto, debido al límite de 500 ms impuesto.

Jitter: nuevamente, el Tetrinet resulta el más tolerante, dado que un jugador puede proseguir su partida un tiempo sin necesidad de mantener comunicación con el servidor. El Quake 2 y el streaming aparecen a continuación –aunque este último depende del tamaño del buffer que emplee–, mientras que VoIP tiene los requisitos más estrictos.

Pérdidas: el flujo de streaming resulta el más sensible a pérdidas en los paquetes, siguiéndole la aplicación de VoIP. Quake 2 y Tetrinet presentan una tolerancia a pérdidas bastante notable.

6 Conclusiones

En este artículo se han analizado y presentado algunas de las aplicaciones más importantes en redes móviles 4G: herramientas de streaming de audio y

video, aplicaciones de telefonía y juegos interactivos. La medición del flujo generado, así como el análisis de la tolerancia a una disminución de la QoS recibida, ha proporcionado una serie de consideraciones ciertamente válidas para el diseño eficiente de mecanismos de provisión de QoS:

- En primer lugar, las redes 4G deberán tratar con flujos de muy diversa índole, tanto por la arquitectura que suponen (modelo cliente – servidor, o bien punto a punto) como por los requisitos de ancho de banda. Además, estos requisitos se han caracterizado según la tasa de inyección de paquetes y según el tamaño de los paquetes. Estas dos características del tráfico generado por las aplicaciones condicionan el diseño del mecanismo de provisión de QoS (i.e.: para el ajuste de los parámetros de un token bucket).
- Por otra parte, con determinadas aplicaciones (Quake 2, VoIP) las cabeceras que añaden los protocolos de niveles inferiores (principalmente IPv6) conllevan un aumento del ancho de banda necesario muy significativo. Por ello puede que sea interesante estudiar la implementación de mecanismos de compresión de aquéllas.
- Un análisis cualitativo como el realizado permite establecer criterios para el diseño de mecanismos de QoS: por ejemplo, se ha visto hasta qué punto un flujo de streaming no es muy tolerante a excesivas pérdidas, pero sí más al jitter, que una conversación VoIP (lo que resultaría, en un entorno DiffServ, en un marcado diferente). En otra ejemplo, una partida de Quake 2 no exige ancho de banda, pero sí una cota de retardo poco elevada, resultando el Tetrinet mucho más tolerante.

Con este trabajo, además, se da pie a la evaluación desde el punto de vista del usuario de las prestaciones de la red, y no en términos de milisegundos o bits por segundo. De esta forma, en situaciones de pérdidas de QoS (por congestión, por traspaso –handover-) se puede evaluar el comportamiento del sistema en términos subjetivos (por ej.: se escucha *mal*), pudiéndose comparar algoritmos con el criterio más relevante (*mean opinion scores*). Es necesario dar un mayor empuje a este tipo de análisis, sin menospreciar las valoraciones técnicas de los sistemas.

Agradecimientos

Este trabajo ha sido financiado parcialmente por la Comisión Europea a través del proyecto "Moby Dick: Mobility and Differentiated Services in a Future IP Network" (IST-2000-25394).

Referencias

- [1] Mobility and Differentiated Services in a Future IP Network, Moby Dick Project, <http://www.ist-mobydick.org/>
- [2] Hans Einsiedler et al, "The Moby Dick Project: A Mobile Heterogeneous ALL-IP Architecture", Advanced Technologies, Applications and Market Strategies for 3G ATAMS 2001, Kraków, Poland, June 17-20, 2001
- [3] Victor Marques et al, "An architecture supporting end-to-end QoS with user mobility for systems beyond 3rd generation", IST Mobile & Wireless Telecommunications Summit 2002, Thessaloniki, Greece, June 17-19, 2002
- [4] Robust Audio Tool, <http://www-mice.cs.ucl.ac.uk/multimedia/software/rat/>
- [5] VideoLAN project, <http://www.videolan.org/>
- [6] Quake 2 for IPv6, <http://www.viagenie.qc.ca/en/ipv6/quake2/ipv6-quake2.shtml>
- [7] Tetrinet (multiplayer tetris game), <http://www.tetrinet.org>
- [8] Ethereal Network Analyzer, <http://www.ethereal.com/>
- [9] Tcpstat homepage, <http://www.frenchfries.net/paul/tcpstat/>
- [10] NISTNet Project, <http://snad.ncsl.nist.gov/itg/nistnet/>
- [11] Unofficial Quake Network Protocol, <http://www.gamers.org/dEngine/quake/QDP/qnp.html>
- [12] Mansour J. Karam, Fouad A. Tobagi, "Analysis of the Delay and Jitter of Voice Traffic Over the Internet", IEEE Infocom 2001