

HERRAMIENTA PARA AUTOMATIZACIÓN DE MEDIDAS DE TIEMPO REAL EXTREMO A EXTREMO

Luis A. Casadesus, Julián Fernández-Navajas, José Ruiz-Mas, José M^a Saldaña, José I. Aznar,
Eduardo Viruete Navarro
{luis.casadesus, navajas, jruiz, jsaldana, jiaznar, evirute}@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- In this paper we present a tool able to perform systematic end-to-end measurements in IP networks. The tool allows monitoring QoS parameters like bandwidth, One Way Delay (OWD), Round Trip Time (RTT), delay variation (jitter) and packet loss rate. UDP packet bursts of different sizes and inter-packet times can be used. The tool consists of a process that sends a UDP stream and another one that receives it. Multiple streams can be sent simultaneously to emulate several communications. The sending process is responsible for managing the start of each communication using a signaling protocol to exchange information with the receiving one. The receiver is responsible for saving the arrival time and size of the packets on a file. Finally, an analysis process allows to obtain objective and subjective QoS parameters, like ITU R-Factor and MOS. The tool allows different VoIP traffic models and codecs.

I. INTRODUCCIÓN

El rápido crecimiento de Internet y el incremento de la potencia de las computadoras han hecho que cada día aumente el número de aplicaciones que utilizan la red, como por ejemplo los nuevos servicios multimedia en tiempo real, entre los que destacan la Voz sobre IP (*Voice over IP*, VoIP), la videoconferencia, la televisión sobre IP o la telemedicina. Estas nuevas aplicaciones en tiempo real necesitan un comportamiento fiable respecto al retardo para brindar un servicio de buena calidad.

Por ello, para un proveedor de servicios es importante monitorizar la red para conocer de antemano si podrá soportar el lanzamiento de un nuevo servicio. Existen herramientas que sirven para caracterizar diversos parámetros de una red: retardo, variación del retardo (*jitter*), ancho de banda máximo, ancho de banda disponible y tasa de pérdidas [1].

La monitorización puede llevarse a cabo mediante medidas de parámetros extremo a extremo (*end-to-end*, E2E) o centralizadas. Las primeras se basan en medidas en los extremos de la red, mientras que las segundas utilizan información obtenida dentro de la propia red para cuantificar los parámetros de Calidad de Servicio (*Quality of Service*, QoS). En caso de no tener control sobre la red, las medidas a utilizar deben ser E2E.

Estas medidas pueden ser pasivas, que se obtienen observando el tráfico, o activas [2], que introducen paquetes en la red y comprueban el servicio obtenido.

Para determinar cómo experimentan los usuarios la calidad de un servicio de VoIP a partir de los parámetros de QoS es frecuente utilizar el Modelo E, definido en la ITU-T Rec. G.107 [3]. Se trata de un modelo de cálculo que resulta útil para la planificación de transmisiones en red, asegurando la satisfacción de los usuarios en la transmisión extremo a extremo. El resultado básico del modelo es una cuantificación escalar de la calidad de transmisión, el Factor R, una medida de la calidad de transmisión de voz que va desde 0 en el peor caso hasta 100 en el mejor. El Factor R determina el MOS (*Mean Opinion Score*) que varía de 1 a 5 (1 mala, 2 pobre, 3 regular, 4 buena y 5 excelente).

El objetivo de este trabajo es presentar una herramienta orientada al estudio sistemático de comunicaciones de tráfico multimedia en tiempo real E2E, buscando la automatización de tareas. Esta herramienta permite la estimación de las prestaciones de Internet y sus rutas E2E, usando UDP para la realización de medidas que permiten estimaciones de parámetros de QoS. Esta herramienta nos permitirá concretamente estudiar la calidad para servicios de VoIP al obtener el Factor R y el MOS dependiendo del modelo de tráfico y la codificación utilizada en la comunicación.

Se ha organizado el artículo de la siguiente forma: La sección II presenta trabajos relacionados. La sección III expone la arquitectura del sistema y su funcionamiento. En la sección IV se describe el entorno en el que se ha implementado y probado el sistema. La última sección detalla las conclusiones de este trabajo.

II. TRABAJOS RELACIONADOS

Para cuantificar las exigencias de QoS de los servicios en tiempo real es necesario disponer de herramientas fiables [4], que pueden ser clasificadas en dos grandes grupos: E2E y centralizadas. Las primeras se basan en la obtención de datos desde los extremos de la red, considerándola como una caja negra. Por el contrario, las segundas utilizan información obtenida de la propia red, como son las estadísticas de los *router*. Se trata de dos puntos de vista

complementarios: el del usuario final que no controla la red, y el del operador que posee toda la información.

En la actualidad, los usuarios finales de Internet disponen de los populares *test* de velocidad del acceso [5, 6] para la realización de medidas contra un servidor geográficamente cercano. Sin embargo, estos *test* no son del todo útiles para servicios en tiempo real porque la información multimedia en muchos casos fluye entre usuarios, no a través de un servidor central. Por ello, si lo que se pretende es la realización de medidas E2E para estimar los parámetros de QoS en los enlaces entre usuarios, son más ventajosas soluciones como la arquitectura de medidas propuesta en el proyecto AQUILA [7] o en [8] donde se utiliza el retardo en un sentido (*One Way Delay, OWD*) para estimar el ancho de banda disponible en enlaces de Internet.

Muchos de estos estudios se basan en insertar tráfico en un extremo de la red y analizar los resultados en el otro extremo. Para generar tráfico en la red existen numerosas herramientas útiles como D-ITG [9], TG [10] y MGEN [11] que hemos probado, pero no se adaptan exactamente a los requisitos de nuestro estudio, como son el poder funcionar de forma desatendida, automática, con precisión de milisegundos y enviando varias comunicaciones simultáneas.

III. ARQUITECTURA DEL SISTEMA

El desarrollo de esta herramienta nos permitirá calcular parámetros objetivos y subjetivos de QoS. Para ello se propone una arquitectura emisor-receptor que nos permita el estudio sistemático de redes de forma automática. Se trata de una herramienta capaz de realizar comunicaciones E2E entre usuarios mediante el envío y recepción de varias secuencias de ráfagas de tráfico UDP. Este método ya se ha tratado en estudios realizados por nuestro grupo [12, 13]. Ahora se utilizará para hacer un estudio que nos permita determinar la calidad de una o varias comunicaciones de VoIP. Esto nos permitirá hacer un seguimiento de parámetros de QoS, que nos indicarán si en la red se ha efectuado una provisión adecuada de los recursos para garantizar, con una alta probabilidad, el funcionamiento de los servicios.

Mediante el análisis del tráfico recibido obtendremos los parámetros de retardo, pérdidas y *jitter*, con los cuales podemos calcular el Factor R que determina el MOS para cada comunicación. Esto nos permite tanto estimar la calidad para un determinado número de llamadas como cuántas llamadas nos permitiría un enlace manteniendo una determinada calidad. Para hacer el cálculo del Factor R se han utilizado las expresiones analíticas obtenidas en [14].

Para la realización de estas medidas se crearán *test* descritos por el administrador del sistema. Un *test* se describe a partir de la caracterización de uno o varios flujos simultáneos. Cada flujo se describe a partir de un conjunto de parámetros: origen, destino, número de ráfagas a enviar; tiempo entre ráfagas; número de paquetes por ráfaga; tamaño y tiempo entre paquetes dentro de la ráfaga, si el tráfico es de ida y vuelta (*rtt*) o solo ida (*one way*) y la codificación. Cada flujo emula una comunicación de tiempo real, o un flujo de tráfico interferente. Dependiendo del servicio a emular se utilizan unos u otros parámetros: para tráfico interferente no se utilizaría la codificación o para tráfico de voz no se mandarían varias ráfagas.

Para facilitar la repetibilidad, cada flujo se podrá reiterar cada cierto tiempo entre una fecha y hora inicial y final. Esta

información se almacena en un documento XML que leerá el emisor para llevar a cabo el envío del tráfico. Para describir los mensajes de control que se intercambian entre los extremos se utilizará también XML, que permite empaquetar de forma sencilla y extensible los mensajes que se intercambian en el sistema utilizando un estándar del W3C [15].

Para generar tráfico equivalente al de diferentes servicios, la información de los intervalos de transmisión de paquetes y el tamaño se deben poder leer de un fichero aparte, especificado en la descripción del flujo. Para ello es necesario tener almacenadas las trazas del tráfico de ese servicio, lo que nos permite emular comunicaciones cuando los tiempos y tamaños de paquete no son constantes y nos garantiza poder repetir las pruebas en las mismas condiciones.

Por cada flujo disponemos del instante de envío y recepción de cada paquete, así como el orden de llegada. A partir de estos datos se hace el análisis correspondiente para obtener los parámetros objetivos (retardo, *jitter*, y tasa de pérdidas) y subjetivos (Factor R, MOS).

A. Módulos del sistema

El sistema consta de cuatro módulos. En la Fig. 1 se muestra la ubicación de los módulos del sistema atendiendo dos flujos de distinto tipo.

- El *emisor* es el encargado de leer el fichero de descripción del *test*, iniciar la comunicación con el otro extremo involucrado y enviar el tráfico.

- El *receptor* se ocupa de recibir el tráfico, extraer la información de cada paquete (identificador del flujo, número de serie del paquete y hora de envío), agregar la hora de llegada y el tamaño y escribir estos datos en un fichero.

- El *reenviador* se encarga de reencaminar los paquetes al origen del tráfico en el caso en que el emisor y el receptor estén en el mismo ordenador.

- El *analizador* toma los registros del tráfico y realiza los cálculos correspondientes para obtener los parámetros objetivos y subjetivos, que se escriben en un fichero XML para posteriores análisis.

Como se ha visto, dentro de la arquitectura tenemos cuatro módulos. En cada comunicación intervienen dos extremos, el local y el remoto. Si el tráfico es *rtt*, en el extremo local se encontrarán el emisor y receptor mientras el extremo remoto reenvía el tráfico; si es *one way*, en local está el emisor y en remoto el receptor. El módulo de análisis puede ubicarse en cualquiera de los extremos o en una instancia aparte.

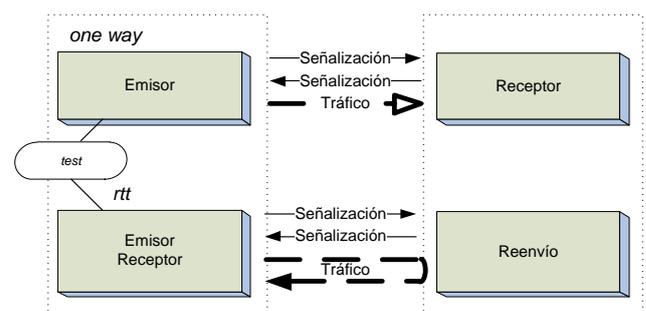


Fig. 1. Módulos del sistema

Como se pueden enviar varios flujos simultáneamente y pueden ser de dos tipos, una instancia del extremo local puede enviar y recibir al mismo tiempo, del mismo modo que el extremo remoto puede recibir y reenviar simultáneamente.

B. Descripción de un test

Como hemos dicho antes, un *test* consiste en una o varias comunicaciones. Cada comunicación consiste en una secuencia de ráfagas de paquetes UDP de tamaño y tiempo variables desde el emisor al receptor, donde se registran y almacenan los paquetes recibidos. Presentamos el formato del *test* en la Fig. 2.

Los *test* se definen por uno o varios flujos descritos por los siguientes atributos:

- Identificador.
- Dirección IP del emisor.
- Puerto del emisor.
- Dirección IP del *host* remoto.
- Puerto del *host* remoto para recibir el flujo.
- Puerto de control del *host* remoto.
- Tiempo de espera para repetir el flujo.
- Fecha y hora de inicio.
- Fecha y hora de fin.
- Número de ráfagas a lanzar.
- Número de paquetes por ráfaga.
- Tamaño de cada paquete.
- Tiempo entre ráfagas.
- Tiempo entre paquetes.
- Tipo de tráfico (*rtt* o *one way*).
- *Codec* (Codificación del flujo. P.ej: G729a, G711).
- Fichero (Nombre del fichero de donde se leen tamaños y tiempos si no son constantes).

Es importante tener en cuenta que los tiempos para la repetición del flujo y el tiempo entre ráfagas se calculan desde el final de un flujo o una ráfaga hasta el inicio de un nuevo flujo de la siguiente ráfaga. Sin embargo, el tiempo entre paquetes se calcula desde el inicio de envío del paquete hasta el siguiente inicio.

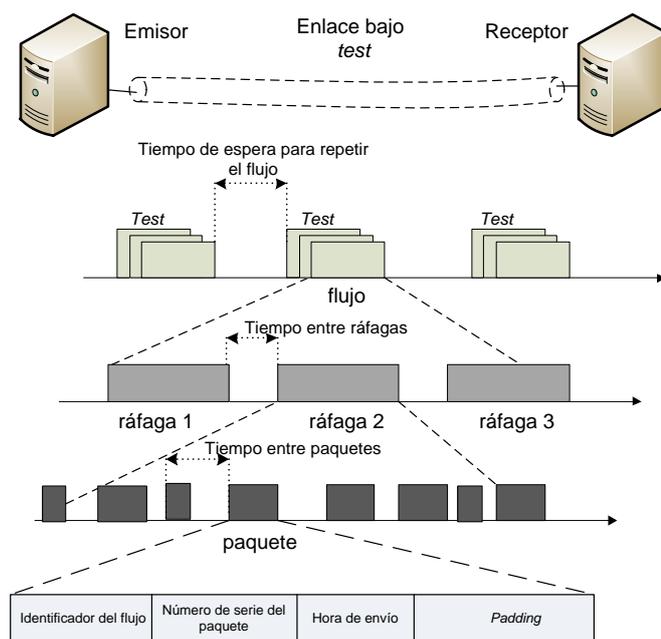


Fig. 2. Formato del *test* y estructura de un paquete

El campo de datos de los paquetes UDP tendrá el formato que aparece en la Fig. 2. La longitud mínima del paquete a enviar corresponde al tamaño que permita incluir esos campos. El campo *padding* contiene bits aleatorios para completar el tamaño definido por el usuario si es necesario.

C. Funcionamiento general

En este apartado describiremos el orden de ejecución de la aplicación. En primer lugar se debe ejecutar en el extremo remoto, que va a recibir o reenviar el tráfico. Posteriormente se inicia el extremo local, donde se encuentra el emisor. En ambos se indica la dirección IP y el puerto usado para señalización de control. En el caso del extremo local debe existir además el fichero XML con la descripción del *test*.

El emisor inicialmente lee del fichero la descripción de los diferentes flujos a enviar durante el test. A continuación comprueba la hora de inicio de cada flujo. Cuando es la hora de comenzar un flujo, el emisor envía un mensaje de inicio al otro extremo, indicando el puerto y el tipo de tráfico. El extremo remoto envía entonces un mensaje de *ACK* para que comience el flujo, terminando así la fase de señalización y comenzando el envío del flujo.

Si el tráfico es *rtt*, el emisor se configura también como receptor y el otro extremo solamente devuelve el tráfico. Si es *one way*, el emisor sólo envía el tráfico mientras que el otro extremo actúa como receptor. En la Fig. 3. se muestra un diagrama de mensajes.

El extremo configurado como receptor se encarga de registrar en un fichero los paquetes recibidos. Por cada paquete guarda los siguientes datos: identificador del flujo, número de serie del paquete, tiempo de emisión, tiempo de recepción y tamaño del paquete.

A partir de la información guardada en el fichero, se pueden calcular el retardo, la variación del retardo (*jitter*), y la tasa de pérdidas. Para los flujos que se corresponden con comunicaciones en los que hayamos indicado la codificación, se puede calcular el Factor R.

IV. IMPLEMENTACIÓN Y PRUEBAS

El sistema ha sido desarrollado utilizando lenguaje de programación C. Cada comunicación que se establece es un hilo de ejecución en envío y recepción. El hilo que genera se encarga de poner el sello de tiempo en el envío y el que recibe extrae la información del paquete, pone el sello de tiempo en recepción y escribe en un fichero. En el caso en que se mida el *rtt*, el extremo remoto se encarga de configurar reglas *iptables* para reenviar los paquetes.

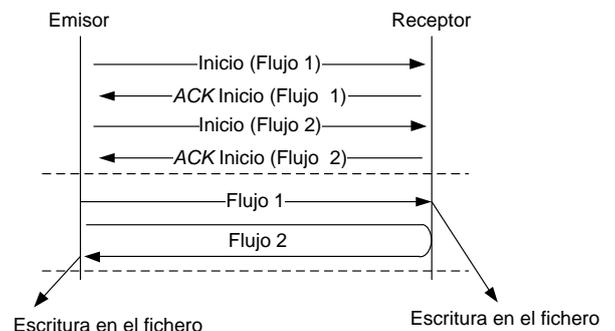


Fig. 3. Señalización y envío

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

REFERENCIAS

- [1] S. M. Abdule, W. T. Chee, A. Mustaf, A. H. Abdalla, "Evaluation of an Efficient Approach to provide QoS to VoIP" Proceedings 2nd Information and Communication Technology Seminar, ICTS 2006.
- [2] T. Anjali, C. Scoglio, "TEMB: A Tool for End-to-End Measurement of Available Bandwidth in IP Networks," Proceedings of IEEE ELMAR 2003, Junio 2003.
- [3] The E-model, a computational model for use in transmission planning, ITU-T Recommendation G.107. Abril. 2009.
- [4] E. Viruete, J. Fernández-Navajas, E. Macián, I. Martínez, J. García, J. Ruiz, "Análisis de métodos de estimación de la capacidad de accesos a Internet para aplicaciones en tiempo real". Actas VI Jornadas de Ingeniería Telemática (JITEL), pp. 465-472, Septiembre 2007.
- [5] <http://www.isposure.co.nz/index.htm>, última visita 5/4/2011.
- [6] <http://www.speedtest.net/>, última visita 5/4/2011.
- [7] F. Strohmeier, H. Dörken, B. Hechenleitner, "AQUILA Distributed QoS Measurement", Proc. COMOCOM8, pp. 177-185, 2001.
- [8] H. Ningning, P. Steenkiste, "Evaluation and characterization of available bandwidth probing techniques" *Selected Areas in Communications, IEEE Journal*, vol.21 n.6, pp.879-894, Ago. 2003.
- [9] A. Botta, A. Dainotti, A. Pescapè, "Multi-protocol and multi-platform traffic generation and measurement", INFOCOM 2007 DEMO Session, May 2007, Anchorage (Alaska, USA)
- [10] <http://www.postel.org/tg/tg.htm>, última visita 5/4/2011.
- [11] <http://cs.itd.nrl.navy.mil/work/mgen/index.php>, última visita 5/4/2011.
- [12] E. A. Viruete, J. Fernández-Navajas, E. Macián, I. Martínez, R. del Hoyo, J. García, Analysis of Capacity Estimation Methods for Real-Time Applications in Internet Accesses, V Annual IEEE Consumer Communications and Networking Conference (CCNC), Enero 2008.
- [13] J.M. Saldaña, J. Fernández-Navajas, J. Ruiz-Mas, E. Viruete, L. Casadesus, J.I. Aznar. *Sistema Multiagente de Medidas Activas E2E*. Actas del XXIV Simposium Nacional de la Unión Científica Internacional de Radio (URSI 2009), pp. 247-248, ISBN 978-84-8102-550-7. Santander (Spain). Sep. 2009.
- [14] R.G. Cole, J.H. Rosenbluth. "Voice over IP performance monitoring". SIGCOMM Comput. Commun. Rev. 31, 2 (Apr. 2001), pp. 9-24.
- [15] T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler, F. Yergeau. Extensible Markup Language (XML) 1.0 (Fifth Edition) W3C Recommendation 26 Nov. 2008. <http://www.w3.org/TR/xml/>, última visita 30/3/2011.

Para conseguir regularidad en los tiempos de envío de los paquetes, calculamos el tiempo en que hay que enviar el próximo paquete y dormimos el hilo el tiempo que resulta de restar este tiempo del tiempo tomado en ese momento. Esto también nos permite que no se acumulen los retardos.

Para obtener el *OWD* es importante una correcta sincronización de los relojes de los extremos. Una posibilidad para obtener una referencia de sincronización de alta precisión es utilizar sistemas GPS, que pueden dar una precisión de microsegundos, pero esa solución resulta costosa. Otra solución es utilizar NTP (*Network Time Protocol*), pero esta no es una solución tan precisa ya que, según las características de la red, puede obtener una precisión de milisegundos.

En un entorno real de medidas de telefonía, al no ser NTP una solución precisa, y resultar costosos los equipos GPS, al tratarse de un servicio interactivo, se suele usar la aproximación de estimar el *OWD* como la mitad del *rtt*.

Se realizaron pruebas utilizando un testbed simple. Utilizamos dos mini-pc iguales, con un procesador I3-370M equipados con Linux 2.6.38, y conectados con un cable cruzado a sus interfaces de red PCI *Express Gigabit Ethernet*. Con este *testbed* se pudo enviar y recibir de manera fiable hasta 80.000 paquetes por segundo y se lograron alcanzar 460 Mbps. Se probaron además hasta 60 comunicaciones concurrentes a partir del modelo de tráfico del códec de voz G.729a, funcionando de manera correcta. El escenario de pruebas se muestra en la Fig. 4.

V. CONCLUSIONES

Se ha presentado una herramienta para monitorizar comunicaciones E2E en Internet midiendo parámetros de QoS, que permite realizar el seguimiento de estos parámetros a lo largo del tiempo.

Hemos conseguido una herramienta que funciona de forma desatendida, automática, con precisión de milisegundos y que puede enviar varias comunicaciones simultáneamente en tiempo real, poniendo atención en generalizar la descripción del test para que funcione con servicios de diferente naturaleza.

En el caso de comunicaciones telefónicas, la herramienta utiliza el *modelo E* para analizar los resultados y generar un Factor R estimado para tráfico VoIP, directamente relacionado con la calidad percibida por un usuario (MOS), utilizando codificación G.729a y G.711, y permitiendo así planificar la implementación de servicios de voz sobre IP en diferentes enlaces.

En estos momentos se realiza el reenvío de los paquetes utilizando reglas *iptables*. Pero sería interesante poder medir también el *owd* de un flujo, además del *rtt*. Esto requeriría capturar el paquete en el destino, y luego reenviarlo, y nos permitiría no tener que repetir el experimento cuando se quieren obtener ambos tiempos para un mismo flujo.

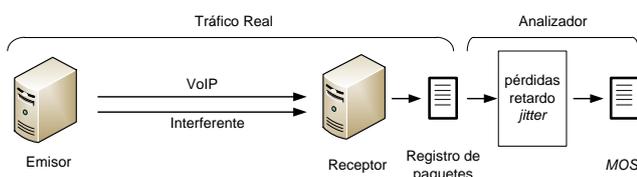


Fig. 4. Escenario de pruebas.