

# Generador de tráfico sintético para la evaluación del rendimiento de cachés

F.J. González-Cañete, Raúl Jiménez-Jiménez, E. Casilar

Departamento de Tecnología Electrónica,

Universidad de Málaga

ETSI de Telecomunicación, Campus de Teatinos, 29071, Málaga.

fgc@uma.es, ruljimenez@yahoo.es, ecasilar@uma.es

**Resumen-** En este artículo se presenta un generador de tráfico telemático para la evaluación del rendimiento de cachés tanto en redes cableadas como redes inalámbricas con acceso a Internet. Para ello, el trabajo se basa en los estudios previos sobre las características del tráfico Web para implementar un generador de tráfico con dichas características. El generador se configura para tener en cuenta parámetros como el número de documentos referenciados una única vez, la popularidad, la localidad temporal, la correlación entre tamaño de documentos y su frecuencia de acceso, el modelado del tamaño de los documentos y el tipo de los mismos. La aplicación ha sido validada para comprobar que las características del tráfico generado coinciden con los valores ideales.

**Palabras Clave-** generador de tráfico, caché

## I. INTRODUCCIÓN

En los últimos años se ha producido un considerable aumento de la popularidad de las páginas Web, que se han introducido en todos los ámbitos de nuestra sociedad. Muchas razones explican este incremento, entre las que destacamos: la disponibilidad de interfaces gráficas de usuario para navegar, la existencia de editores y herramientas de soporte para la creación y publicación de documentos Web; una tendencia de investigadores, instituciones educacionales y públicas, y organizaciones comerciales para usar la Web como diseminador de información y de enlace con sus clientes. Este crecimiento ha llevado consigo la aparición de problemas tales como: grandes congestiones en la red, un bajo ancho de banda, altas latencias al momento de extraer un documento, sobrecarga en servidores y accesos simultáneos masivos a un servidor, entre otros.

Por otro lado, también ha habido un gran auge de las redes inalámbricas móviles. Estas redes se caracterizan por tener unas importantes restricciones en lo referente a ancho de banda y capacidad de los terminales. Por lo tanto, cuando este tipo de redes intentan acceder a través de pasarelas hacia Internet, se produce un problema de calidad de servicio aún mayor que en las redes cableadas debido a las restricciones anteriormente comentadas.

Muchos han sido los intentos para paliar estos efectos, pero sin duda, una de las soluciones más utilizadas es el uso de cachés, cuya misión es la de almacenar aquellos documentos que han sido solicitados por los usuarios con mayor frecuencia. Cuando se produce la petición de un documento ya consultado, éste será servido directamente por la caché disminuyendo, por tanto, el tiempo de servicio, la sobrecarga de servidores y la sobrecarga de la red. Estas cachés pueden estar localizadas tanto en los clientes

(navegadores Web) como en servidores *proxy*. En el caso de las cachés en los navegadores, el tráfico generado es nulo o muy pequeño, ya que, cuando el usuario solicita un documento que ya tiene almacenado en la cachés, éste es servido directamente por ella o, como mucho, se realizará una consulta al servidor para comprobar que el documento sigue siendo válido. Por otro lado, los servidores *proxy* son cachés intermedias que se sitúan entre los usuarios y los servidores de forma que almacenan aquellos documentos que son más frecuentemente accedidos por una comunidad de usuarios. Al encontrarse los *proxies* situados más cerca de los usuarios que los propios servidores Web, el tiempo de respuesta se reduce, así como la carga en los servidores Web.

Por tanto, el uso de cachés es una buena solución para agilizar la atención de peticiones a documentos Web tanto en redes cableadas como móviles, pero el estudio de estas cachés necesita de tráfico real en la red. Aunque existen muestras de tráfico real de carácter público del tráfico Web que soportan determinados *proxies* de Internet [1], no es así para redes inalámbricas. Es aquí donde surge la necesidad de la creación de un generador de tráfico sintético que sirva para evaluar esquemas de caché tanto en redes cableadas como inalámbricas. La ventaja de usar un generador de tráfico sintético radica en la posibilidad de ir modificando las diferentes características del tráfico para así analizar el comportamiento de las cachés.

Actualmente, existen diferentes simuladores de tráfico sintético como ProWGen [2], TrGen [3] y GenSyn [4]. Estos simuladores son capaces de crear muestras de tráfico sintético a partir de la introducción de una serie de parámetros o bien mediante la introducción de muestras de tráfico real, que son analizadas y posteriormente se simula su comportamiento. Sin embargo, solo ProWGen es capaz de generar tráfico sintético para evaluar el rendimiento de cachés, ya que es el único que genera tráfico a nivel de documentos Web. Desafortunadamente, ProWGen no es capaz de distinguir el tipo de los documentos que generan sus muestras. Por lo tanto, el objetivo del presente trabajo es la creación de un generador de muestras de tráfico telemático a nivel de documentos Web que cumpla todas las características que se consideran necesarias para dicho tipo de tráfico. El generador a implementar será capaz, además, de diferenciar los tipos de los documentos.

El resto de este artículo está estructurado como sigue: en el capítulo II se analizan los fundamentos matemáticos de las características del tráfico a generar y se especifica la forma

de implementar dichas características; en el capítulo III se valida el generador implementado comprobando que las muestras sintéticas generadas coinciden con lo esperado; en el capítulo IV se describe el entorno de usuario de la aplicación generada; finalmente en el capítulo V se comentan las principales conclusiones del presente trabajo.

## II. IMPLEMENTACIÓN

Se ha creado una herramienta sobre el entorno MATLAB que tiene como objetivo la generación de muestras de tráfico para su posterior procesado en simuladores de cachés. El resultado de la aplicación es la creación de ficheros donde se especifican estas muestras de tráfico que serán las que utilicen posteriormente dichos simuladores.

La implementación de la aplicación se divide en dos partes, la primera es la generación en sí de las muestras de tráfico a partir de una serie de parámetros, y la segunda es la interfaz gráfica con el usuario donde se interactúa en la elección de los parámetros deseados.

Existen dos posibles enfoques para sintetizar muestras de tráfico [5]. La primera de ellas consiste en muestrear o permutar muestras de tráfico reales reordenando las peticiones de modo que se genere una nueva carga de tráfico diferente a partir de la carga de tráfico original. A este enfoque se le denomina *basado en traza real*. La segunda aproximación, denominada *aproximación analítica*, usa modelos matemáticos para modelar las características más interesantes del tráfico, y utiliza la generación de números aleatorios para producir tráfico que estadísticamente coincide con dichos modelos. Dado que la aproximación analítica ofrece la flexibilidad de modificar los parámetros de los modelos matemáticos para regular las características del tráfico a generar, éste ha sido el método empleado.

El generador de tráfico incorpora cinco características de tráfico seleccionadas, las cuales han sido identificadas como importantes en estudios previos de muestras de tráfico en servidores [2] [5] [6] [7]: los documentos referenciados una sola vez, la popularidad de los documentos, la distribución del tamaño de los documentos, la correlación entre el tamaño de los documentos y su popularidad y, finalmente, la localidad temporal.

### A. Documentos referenciados una sola vez

Muchos estudios de tráfico Web en servidores y proxy han mostrado que gran parte de las peticiones hacia un servidor o un proxy son realizadas una sola vez, independientemente de la duración del acceso [8] [9]. A este tipo de documentos se les denomina *one-timers*. Es evidente, que no tiene ningún beneficio almacenar en caché este tipo de documentos puesto que nunca más van a ser requeridos. De hecho, sería deseable la existencia de algoritmos ubicados en las cachés para discriminar a estos documentos de modo que no inunden la caché reduciendo así su efectividad.

El enfoque utilizado para modelar las referencias a documentos *one-timers* es determinar cuántos de los documentos distintos en la traza de tráfico a crear deberían ser *one-timers*. Utilizando el porcentaje de *one-timers* como un parámetro, se permite al usuario especificar el valor deseado. Una vez especificado, se puede hallar con facilidad el número de *one-timers*, con lo que las referencias a estos documentos se fijan a uno.

### B. Popularidad

Una característica común en el tráfico Web es la irregular distribución de las referencias a los distintos documentos [10] [11]. En muchos casos se aplica la ley de Zipf [12] para modelar la popularidad de los documentos [13] [14] [15]. La ley de Zipf expresa una relación exponencial entre la popularidad  $P$  (número de veces que se repite la petición de dicho documento) y su ranking  $r$  (que se extrae de la ordenación de los documentos según su popularidad). Esta relación se expresa en la ecuación (1):

$$P = \frac{c}{r^\alpha} \quad (1)$$

donde  $c$  es una constante y  $\alpha$  es un valor entre cero y uno.

Algunos investigadores han encontrado que el valor de  $\alpha$  es cercano a la unidad [13] [15], precisamente siguiendo la ley de Zipf. Otros autores [2] [4] [14] han encontrado que el valor de  $\alpha$  es menor que la unidad, y que la distribución puede ser descrita “como la de Zipf” con el valor de  $\alpha$  variando dependiendo del tráfico. Este comportamiento típico puede describirse como una línea recta de pendiente negativa  $\alpha$  si representamos en unos ejes logarítmicos  $P$  frente a  $r$ . Este ajuste lineal suele ser casi perfecto para la parte principal del cuerpo de la distribución. Sin embargo, suele desajustarse tanto para los elementos más populares como para los menos populares (debido a los *one-timers*) [6].

Para determinar la popularidad de los restantes documentos, el primer paso es calcular la constante de proporcionalidad  $c$  de la fórmula de Zipf en la ecuación (1) usando para ello los parámetros conocidos como el número de referencias ( $N$ ), número de documentos distintos ( $a$ ), y el número de *one-timers* ( $b$ ). Puesto que todos los *one-timers* tienen un número de referencias igual a uno, el ranking  $r$  del primer *one-timer* es  $a-b+1$ , y el ranking del último *one-timer* es  $a$ , la constante  $c$  puede ser calculada usando el *one-timer* situado en el centro del rango de los *one-timers*,  $r=a-(b/2)$  y con popularidad  $P=1$ . Sustituyendo estos valores en la fórmula de Zipf, obtenemos la ecuación (2).

$$1 = \frac{c}{(a-b/2)^\alpha} \quad (2)$$

Despejando  $c$  de la ecuación (2) obtenemos la ecuación (3).

$$c = (a-b/2)^\alpha \quad (3)$$

Una vez que se ha calculado el valor de  $c$ , la popularidad de cada uno de los documentos que faltan por hallar (aquellos que ocupan el ranking desde uno hasta  $a-b$ ), puede ser calculada fácilmente directamente de la fórmula de Zipf.

Desafortunadamente, después de determinar la popularidad de cada uno de los documentos como se ha indicado anteriormente, la suma de las popularidades de todos los documentos no siempre coincide exactamente con  $N$  (que es el número total de peticiones a generar). Hay dos razones principales para ello: el cálculo de la constante  $c$  no es exacto, y que todas las popularidades han sido redondeadas al entero más próximo. Para resolver este problema, se ha añadido un paso adicional para ajustar más la popularidad de cada documento. Para ello se ha escalado la

popularidad de los no *one-timers* (aquellos que ocupan los puestos de popularidad del uno al  $a-b$ ), manteniendo en todo momento la pendiente  $\alpha$ . El factor de escala  $s$  se determina usando la ecuación (4).

$$s = (N - b) / \sum_{i=1}^{a-b} P_i \quad (4)$$

Siguiendo con estos cálculos, el número total de peticiones es muy cercano a  $N$  (típicamente dentro del 1-5% y casi siempre dentro del 10%).

### C. Distribución del tamaño de los documentos

Estudios previos de tráfico Web [6] [9] han demostrado que la distribución de los tamaños de los documentos de tráfico Web es de cola pesada. Este tipo de distribuciones implica que relativamente pocos documentos de tamaño muy grande acumulan un porcentaje elevado del volumen total de datos del tráfico Web. Por tanto, la distribución del tamaño de los documentos afecta significativamente en el diseño de estrategias de caché. Almacenando en caché sólo documentos pequeños se puede reducir el número de peticiones enviadas a los servidores de origen, y puede convertirse en una tasa elevada de aciertos de documentos en caché, pero también se traduce en un bajo número de tasa de aciertos por byte. En el otro extremo, almacenando en caché documentos de gran tamaño provoca una elevada tasa de acierto por byte a costa de la tasa de aciertos por documentos.

Para una evaluación efectiva de la gestión de las estrategias en caché, la distribución de cola pesada de los tamaños de documentos debe ser incorporada en la generación de muestras de tráfico, en particular, la “pesadez” de la cola debe ser ajustable de modo que su impacto en las cachés pueda ser evaluado.

En la aplicación desarrollada, el modelado de la distribución del tamaño de los documentos se ha dividido en tres partes:

1. Se modela la cola de la distribución usando una distribución de Pareto.
2. Se modela el cuerpo de la distribución usando una distribución logarítmica normal.
3. Se unen ambas distribuciones.

La distribución de Pareto de doble exponencial es un ejemplo de distribución de cola pesada y ha sido la utilizada para modelar la cola de la distribución del tamaño de los documentos. Su función densidad de probabilidad se muestra en la ecuación (5) y su función de distribución acumulada en la ecuación (6).

$$p(x) = \alpha \cdot k^\alpha \cdot x^{-\alpha-1} \quad \alpha, k > 0, x \geq k \quad (5)$$

$$F(x) = P(X \leq x) = 1 - (k/x)^\alpha \quad (6)$$

El parámetro  $\alpha$ , llamado índice de cola, determina la “pesadez” de la cola de la distribución. La distribución tiene varianza infinita y si  $\alpha \leq 1$ , entonces la distribución tiene media infinita. Esto implica que pequeños valores de  $\alpha$  representan colas pesadas (la mayoría del volumen estaría presente en la cola de la distribución).  $k$  es un parámetro que determina dónde empieza la cola de la distribución al

representa el valor más pequeño posible que puede tomar la variable aleatoria de la distribución de cola pesada.

Los parámetros  $\alpha$  y  $k$  que caracterizan la distribución del tamaño de los documentos pueden ser determinados usando una representación logarítmica de la distribución complementaria [4] [9] [15]. Si trabajamos con la función de distribución acumulada complementaria, se puede comprobar que se verifica la ecuación (7).

$$\frac{d \log \bar{F}(x)}{d \log x} = -\alpha, \quad \alpha > k \quad (7)$$

La pendiente  $\alpha$  puede ser estimada al exhibir un comportamiento lineal. Este método es empleado para estimar  $\alpha$  en los estudios de tráfico sintético puesto que  $\alpha$  y  $k$  son proporcionados como parámetros en el proceso de generación de muestras.

Incorporar el modelo de cola pesada en la aplicación comienza distinguiendo entre los documentos en dos grupos: aquéllos que se encuentran en el cuerpo de la distribución y aquéllos que se encuentran en la cola. El porcentaje de documentos en la cola es especificado como un parámetro de entrada del generador de tráfico.

Para generar el tamaño de los documentos se despeja en la distribución de la cola de Pareto en función de los parámetros de entrada. El objetivo es obtener, para un número aleatorio uniforme  $y$  entre (0,1), el correspondiente valor de  $x$ . Despejando en la ecuación (6) obtenemos la ecuación (8).

$$x = k / (1 - y)^{\frac{1}{\alpha}} \quad (8)$$

Si  $y$  es una variable aleatoria uniforme (0,1) entonces  $1-y$  es también un valor aleatorio uniforme (0,1). Por tanto, la ecuación (8) quedaría como se muestra en la ecuación (9).

$$x = k / (y')^{\frac{1}{\alpha}} \quad \text{con } y' = (1 - y) \quad (9)$$

Así, una vez que  $\alpha$  y  $k$  han sido especificados en la aplicación, los valores de cola pesada pueden ser generados usando la ecuación (9).

Para modelar el cuerpo de la distribución del tamaño de los documentos se usa una distribución logarítmica normal [16]. La distribución logarítmica normal tiene la propiedad que si  $X \approx N(\mu, \sigma^2)$ , es decir,  $X$  está normalmente distribuida con media  $\mu$  y varianza  $\sigma^2$ , entonces  $e^X$  tiene una distribución logarítmica normal con parámetros  $\mu$  y  $\sigma^2$ , denotado por  $LN(\mu, \sigma^2)$ . La función de densidad de probabilidad de una distribución logarítmica normal se muestra en la ecuación (10).

$$p(x) = \frac{1}{x \sigma \sqrt{2\pi}} e^{\frac{-(\ln x - \mu)^2}{2\sigma^2}} \quad (10)$$

Aunque la distribución logarítmica normal no tiene una forma cerrada y, por tanto, no tiene una función de distribución acumulada, hay una aproximación sencilla que explota la propiedad anterior de la distribución logarítmica normal para generar variables logarítmicas normales. Así, para generar una variable logarítmica normal es suficiente

con generar una variable aleatoria  $x$  normal  $N(\mu, \sigma^2)$ , y entonces devolver  $e^x$  como una variable logarítmica normal. Sin embargo,  $\mu$  y  $\sigma^2$  son la media y la varianza de una distribución normal, por lo que se debe, a partir de dichos valores, obtener la media y la varianza de una distribución logarítmica normal. Esos valores son (11) y (12):

$$\mu_l = e^{\mu + \sigma^2/2} \quad (11)$$

$$\sigma_l^2 = e^{2\mu + \sigma^2} (e^{\sigma^2} - 1) \quad (12)$$

De este modo, si se quiere generar una variable aleatoria logarítmica normal con un determinado  $\mu_l$  y  $\sigma_l^2$ , primero se deben resolver las ecuaciones (13) y (14).

$$\mu = \ln \left( \frac{\mu_l^2}{\sqrt{\sigma_l^2 + \mu_l^2}} \right) \quad (13)$$

$$\sigma^2 = \ln \left[ (\sigma_l^2 + \mu_l^2) / \mu_l^2 \right] \quad (14)$$

Una vez que se tienen estos valores, se puede generar una variable aleatoria normal  $N(\mu, \sigma^2)$ , con valores provenientes de una distribución logarítmica normal, y con esa variable aleatoria devolver  $e^x$ , con lo que ya se habrá generado una variable aleatoria logarítmica normal de media  $\mu_l$  y varianza  $\sigma_l^2$ .

Puesto que se modela tanto la cola de la distribución como el cuerpo de la misma de forma separada, se deben tener en cuenta ciertas restricciones necesarias a la hora de unir ambas distribuciones. Primero, los valores que se encuentren en el cuerpo de la distribución no pueden encontrarse también en la cola. Segundo, el caso contrario tampoco debe producirse. Tercero, cuando se represente la función de distribución acumulada, debe producirse una transición suave desde el cuerpo hasta la cola.

Un primer paso para lograr estos objetivos sería limitar superiormente el tamaño de los documentos generados para el cuerpo de la distribución. Es decir, si se genera un valor mayor que  $k$ , se descarta y se genera un nuevo valor. Los efectos colaterales de esta restricción son:

1. Despues de la generación de las variables logarítmicas normales, la media y la desviación típica de los valores resultantes son menores que los especificados en los valores de entrada.
2. La distribución del tamaño de los documentos resultantes puede presentar una obvia discontinuidad donde se unen ambas distribuciones.

Otra posibilidad sería eliminar la restricción de tamaño superior, resolviendo estos dos problemas, pero a cambio, se incrementaría el porcentaje de documentos en la cola de la distribución. En la aplicación se ha optado por la segunda opción, no obstante, en el código del mismo se deja abierta la posibilidad de la primera opción.

#### D. Correlación entre el tamaño de los documentos y su popularidad

Numerosos estudios de tráfico Web en *proxies* muestran que muchos de los documentos transferidos en la Web tienen un tamaño pequeño [5] [6] [14]. Una cuestión natural que se plantea al respecto es si existe alguna correlación estadística

entre la frecuencia de acceso a un determinado documento y su tamaño. Algunos estudios [15] [17] han mostrado que hay una muy pequeña correlación entre su frecuencia de acceso y su tamaño, aunque esta cuestión es todavía tema de debate.

En la aplicación desarrollada, para otorgarla de mayor flexibilidad, se ofrece la posibilidad de que el tráfico generado tenga correlación positiva, negativa o cero entre la popularidad de los documentos y su tamaño. Una correlación positiva implica que los documentos de mayor tamaño tienen mayor popularidad y correlación negativa implica que los documentos de menor tamaño tengan más probabilidad de ser requeridos (mayor popularidad). Mientras que correlación cero no otorga ninguna correlación entre la popularidad de los documentos y su tamaño. El hecho de permitir la existencia o no de correlación radica en la posibilidad de explorar distintos algoritmos de caché según cada una de estas opciones.

Modelar e incorporar estas características de correlación dentro de la generación de muestras de tráfico se realiza en tres etapas:

1. Generar un conjunto de popularidades de documentos usando la aproximación comentada en la sección B.
2. Generar un conjunto de tamaños de documentos usando la aproximación de la sección C.
3. Usar una técnica de mapeo para introducir correlación positiva, negativa o cero entre la popularidad de los documentos y su tamaño.

El algoritmo a seguir es el siguiente:

1. Se genera una lista  $P$  de popularidades para  $n$  documentos distintos usando la aproximación explicada en la sección B. Se ordena la lista  $P$  en orden ascendente.
2. Se genera una lista  $S$  de tamaños de documentos para  $n$  documentos distintos usando la aproximación explicada en la sección C. Se ordena la lista  $S$  en orden ascendente.
3. Se calculan los valores de la función de distribución acumulada para los elementos de la lista  $P$ . Se construye una nueva lista  $P_{nueva}$ , con valores de la forma  $(v_i, p_i)$ , donde  $v_i$  es un valor único de popularidad de la lista  $P$  y  $p_i$  representa su probabilidad acumulada.
4. Se calculan los valores de la función de distribución acumulada para los elementos de la lista  $S$ . Se construye una nueva lista  $S_{nueva}$ , con valores de la forma  $(v_i, p_i)$ , donde  $v_i$  es un valor único de tamaño de documento de la lista  $S$  y  $p_i$  representa su probabilidad acumulada.
  - a) Se repiten los siguientes pasos  $n$  veces para generar una nueva lista  $L$  que represente la popularidad y el tamaño de los documentos distintos. La entrada  $j$  ( $1 \leq j \leq n$ ) en  $L$  es de la forma  $(f_j, s_j)$ , donde  $f_j$  representa la popularidad y  $s_j$  el tamaño del documento. La entrada  $(f_j, s_j)$  es determinada de la siguiente forma: Se genera un número aleatorio  $r_j$  dentro una distribución aleatoria uniforme de rango  $(0, 1)$ .
  - b) Se busca en la lista ordenada  $P_{nueva}$  el primer elemento  $i$  que satisface  $r_j \leq p_i$ . Si  $p_i = r_j$  ó  $i = 1$ , entonces se hace  $f_j = v_i$ . En otro caso, se interpola linealmente usando la ecuación (15).

$$f_j = v_{i-1} + \frac{(r_1 - p_{i-1}) \times (v_i - v_{i-1})}{(p_i - p_{i-1})} \quad (15)$$

- c) En función de la correlación:
- Si se desea correlación positiva, se usa  $r_1$  para buscar en la lista  $S_{nueva}$  exactamente del mismo modo que se ha hecho en el paso b) para buscar el valor del tamaño de documento  $s_j$ .
  - Si se desea correlación negativa, se usa  $1-r_1$  para buscar en la lista  $S_{nueva}$  exactamente del mismo modo que se ha hecho en el paso b) para buscar el valor del tamaño de documento  $s_j$ .
  - Si se desea que no haya ningún tipo de correlación, se genera otro número aleatorio  $r_2$  dentro de una distribución aleatoria uniforme de rango  $(0, 1)$  y se emplea para buscar en la lista  $S_{nueva}$  exactamente del mismo modo que se ha hecho en el paso b) para buscar el valor del tamaño de documento  $s_j$ .
- d) Se añade el par de valores  $(f_j, s_j)$  a la lista  $L$  como la popularidad y el tamaño del documento único  $j$ .
- La lista  $L$  ahora representa una nueva lista de  $n$  documentos distintos, los cuales tienen su propia popularidad y tamaño de documento con el deseado valor de correlación introducido.
  - Se normaliza para ajustar la popularidad de los documentos al número total de peticiones a generar. Pero se va a seguir manteniendo el número de *one-timers* como aquél que se ha introducido como parámetro, eludiendo el hecho de que al ser elegidos aleatoriamente según su función de distribución acumulada saldrían muchos menos *one-timers*, ya que suponen un porcentaje menor sobre la función de distribución acumulada de popularidad.

#### E. Localidad temporal

La localidad temporal hace referencia a la tendencia de documentos referenciados en el pasado a volver a ser nuevamente referenciados en el futuro. El número de veces que cada documento debería aparecer en las muestras de tráfico (la popularidad) ya se conoce. Sin embargo, dado que una referencia a un documento es generada en un instante de tiempo  $t_o$ , no está claro cuándo se va a producir la próxima referencia a dicho documento. Es por esto, que la presencia de la localidad temporal en la generación de tráfico tiene un efecto muy importante para las pruebas con cachés.

La aproximación utilizada para el modelado de la localidad temporal está basada en el modelo de pila finita LRU (*Least Recently Used*). Una pila LRU es una lista de todos los documentos ordenados según hayan sido referenciados recientemente [13], esto es, el último que haya sido referenciado estará en primer lugar de la pila y el que fue referenciado hace más tiempo estará en la última posición. La pila es actualizada dinámicamente cada vez que se procesa una referencia. En muchos casos, esta actualización implica el tener que añadir un nuevo elemento en la cima de la pila empujando el resto hacia abajo, en otros casos, implica extraer un elemento existente en el interior de la pila y trasladarla a la cima de la misma, desplazando al resto de los elementos hacia abajo.

Una pila LRU de tamaño finito es una pila LRU que sólo puede almacenar un número  $m$  de documentos. Experimentos realizados por Mahanti [6] sugieren que  $m=1000$  es un número adecuado para capturar la presencia de localidad temporal en cargas de tráfico Web. No obstante, para la realización de todo tipo de pruebas en cachés, se permite que el tamaño de la pila sea especificado por el usuario como un parámetro.

El aspecto más importante en una pila LRU es que cada posición en la pila tiene asociada una probabilidad de referencia. Las probabilidades son asociadas a la posición de la pila y no a los documentos. Las probabilidades de las posiciones de la pila pueden ser proporcionadas por modelos u obtenidas de analizar muestras de tráfico reales.

Por ejemplo, supongamos que las popularidades de los distintos documentos en la carga de tráfico están representadas por la ecuación (16):

$$D = \{x_1, x_2, \dots, x_n\} \quad x_1 \geq x_2 \geq \dots \geq x_n \quad (16)$$

Entonces, las probabilidades  $a_i$  pueden ser calculadas para cada documento  $i$  usando la ecuación (17):

$$a_i = x_i / \sum_{j=1}^n x_j \quad \text{para } i = 1, 2, \dots, n \quad (17)$$

Hay dos formas posibles de utilizar esas probabilidades: estática y dinámica. En la aproximación estática, si la pila finita tiene un tamaño fijo  $m$  de modo que  $m \leq n$ , la probabilidad acumulada  $y_i$ , se calcula como se muestra en (18):

$$y_i = \sum_{j=1}^i a_j \quad (18)$$

La probabilidad acumulada calculada para cada posición de la pila es asignada al comienzo de la generación de muestras y no puede ser cambiada durante dicho proceso. Esta técnica genera una localidad temporal estadística homogénea para todos los documentos en la traza.

En la aproximación dinámica, cada vez que la pila LRU es modificada (ya sea al mover un documento desde otra posición de la pila a la cima de la misma o bien por traer un nuevo documento a la pila), las probabilidades acumuladas para cada posición de la pila son recalculadas usando los valores  $a_i$  de los documentos que actualmente ocupan cada posición de la pila. Esta aproximación puede modelar propiedades de localidad temporal heterogéneas. Por ejemplo, si en el instante temporal  $t_o$ , las dos primeras posiciones de la pila contienen los documentos con identificadores 1 y 2 respectivamente, entonces sus estados pueden ser representados como  $St_o = \{a_1, a_2\}$ , donde  $a_1$  es la probabilidad de referenciar el documento 1 y  $a_2$  es la probabilidad de referenciar el documento 2. Por lo tanto, la probabilidad acumulada de referenciar esas dos posiciones de la pila son  $a_1$  y  $a_1 + a_2$ , respectivamente. Asumiendo que en el instante  $t_1$ , el documento en la posición dos de la pila es movido a la posición uno, esta acción provocará que haya que recalcular las probabilidades acumuladas asociadas a cada posición de la pila. En ese caso, los estados de la pila se convierten en  $St_1 = \{a_2, a_1\}$ . Así, las probabilidades

acumuladas para referenciar estas dos posiciones de la pila son ahora  $a_2$  y  $a_2+a_1$ , respectivamente.

El proceso de generación de referencias comienza con la pila LRU vacía. Si se elige la aproximación estática, cada posición de la pila tiene asociada una probabilidad acumulada de referenciar a esa posición. En el caso de la aproximación dinámica, las probabilidades están sin inicializar. El proceso de generación de referencias comienza generando un número aleatorio  $x_i$  de una distribución aleatoria uniforme de rango (0,1). Después se debe comprobar si ese documento seleccionado está en la pila o no (si  $x_i \leq y_i$ ). Pueden producirse tres circunstancias:

- a) La pila está vacía.
- b) La pila no está vacía pero el próximo documento a referenciar no está en la pila.
- c) La pila no está vacía y el próximo documento a referenciar ya está en la pila.

En los casos a) y b), se selecciona un documento al azar del conjunto de documentos distintos que quedan aún por referenciar. Se genera una referencia para el documento seleccionado y el contador de referencias que le faltan por generar (es decir, su popularidad) es decrementada en una unidad. Si no quedan más referencias por generar de este documento seleccionado, entonces el documento es eliminado del conjunto de documentos distintos que quedan por referenciar. En caso contrario, el documento es movido hasta la cima de la pila, desplazando a los otros documentos hacia abajo en la pila si es el caso b). En el caso c), se busca desde el comienzo de la pila al mayor elemento de la misma que verifique  $x_i \leq y_i$ . Una vez encontrado se generará una referencia para ese documento y se decrementará en una unidad el contador de referencias que le quedan por generar (popularidad). Si ya no le quedan más referencias por generar, el documento es eliminado de la pila y todos los documentos situados debajo de él suben una posición en la pila. En otro caso, el documento es movido a la cima de la pila, desplazando el resto de elementos, si los hubiera, una posición hacia abajo.

En el modelo de pila dinámica LRU, cada uno de estos casos provocará nuevos cálculos de las probabilidades acumuladas asociadas con esas posiciones de la pila.

#### F. Tipos de documentos

Los documentos existentes en la Web se clasifican en función de su tipo en: aplicaciones, audio, imágenes, mensajes, texto y video. La aplicación desarrollada permite seleccionar el porcentaje de cada tipo de documentos que se desean generar. Además, permite la generación de muestras de tráfico para cada uno de los tipos de documentos especificando todos los parámetros anteriormente mencionados y posteriormente mezclar las muestras generadas para generar una muestra conjunta.

### III. VALIDACIÓN

La aplicación desarrollada requiere de doce parámetros de entrada para generar muestras de tráfico. Según diversos estudios de las características del tráfico Web [5][6][8], los parámetros para la generación de tráfico más usuales son los representados en la Tabla 1. Además, hay que especificar el número total de peticiones que se desea generar y la

correlación existente entre el tamaño de los documentos y su frecuencia de acceso.

Se ha generado una muestra de tráfico con las características de tráfico presentes en la Tabla 1, que es la comentada en el resto del epígrafe. Además, se han generado múltiples muestras modificando cada uno de los parámetros por separado y se ha comprobado que, para todos los casos, las características de la muestra generada coinciden con las esperadas.

Parámetro	Valor
Porcentaje de documentos distintos	30%
Porcentaje de <i>one-timers</i>	70%
Pendiente de Zipf	0.75
Índice de cola-pesada	1.2
Comienzo de la cola en bytes, $k$	10000
Porcentaje de documentos en la cola pesada	20%
Media de la distribución logarítmica normal ( $\mu$ )	7000
Varianza de la distribución logarítmica normal ( $\sigma$ )	11000
Modo de la pila para la localidad temporal	Dinámico
Tamaño de pila para la localidad temporal	1000
Correlación entre tamaño y popularidad	0

Tabla 1. Parámetros usuales para la generación de tráfico

En la gráfica mostrada en la Fig. 1 se puede verificar que las muestras de tráfico generadas tienen un comportamiento “como el de Zipf”, esto es, al trazar en ejes logarítmicos, la popularidad de los documentos frente a su ranking producido al ordenar dicha popularidad, se observa como sigue una distribución lineal con una pendiente igual a  $-0.76045$ , mientras el caso ideal sería  $-0.75$ . Mencionar que tanto la muestra generada como la muestra ideal están casi superpuestas y es sólo en los valores extremos donde se puede apreciar una mínima diferencia.

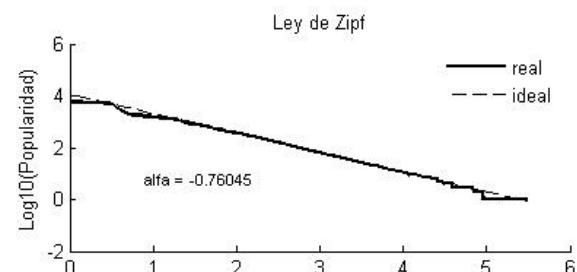


Fig. 1. Análisis del comportamiento de la popularidad

En la Fig. 2 se aprecia el comportamiento de la correlación entre el tamaño de los documentos y su frecuencia de acceso. En ella se observa como no hay ningún tamaño de documento que predomine sobre el resto, es decir, todos los tamaños de documento tienen el mismo peso. El valor de correlación obtenido es de 0.00029 pero el resultado que se expone es 0, puesto que sólo se distingue entre -1, 0 y 1. Por tanto, el resultado coincide con el valor de correlación de entrada.

La Fig. 3 representa la función de distribución acumulada de los tamaños de los documentos en el cuerpo de la distribución, es decir, si el valor de  $k$  es de 10.000 bytes, Es decir, cómo se distribuyen los tamaños de los documentos

para valores menores que  $k$ . Es quizás, la característica que más se aleja al valor ideal. Se explica en el amplio rango de valores aleatorios que se pueden generar y en los errores de aproximación de los modelos matemáticos utilizados tanto para la generación de muestras como para su posterior análisis. Puesto que para la generación de muestras se debe llegar a un compromiso entre todas las variables de entrada, se ha considerado que sea en el tamaño de los documentos donde recaiga el mayor margen de error. No obstante, si que se puede apreciar que el comportamiento de las distribuciones de tamaño de las muestras son bastantes similares.

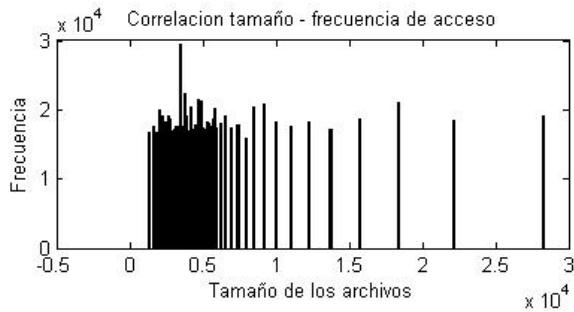


Fig. 2. Correlación entre el tamaño y la frecuencia de acceso

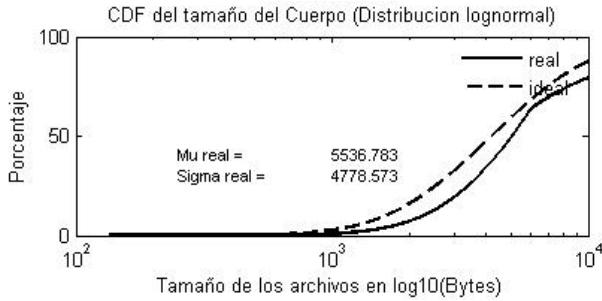


Fig. 3. Análisis de la función de distribución acumulada

En la Fig. 4, se muestra la función de densidad de probabilidad de los tamaños de los documentos tanto en el cuerpo como en la cola de la distribución. Esta gráfica nos proporciona una idea muy intuitiva del tamaño de los documentos.

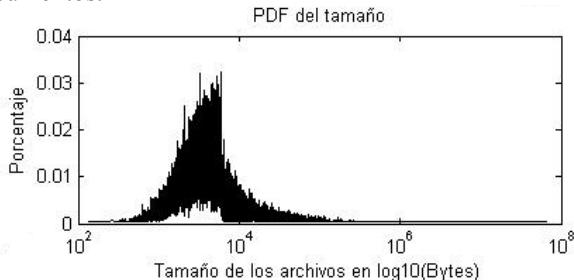


Fig. 4. Función de densidad de probabilidad de los tamaños

En la Fig. 5 se muestra la pendiente de la cola de Pareto, tanto para la muestra obtenida como para la muestra ideal, que se pasa como parámetro de entrada. El valor de inicio del comportamiento de cola de Pareto lo marca el valor de  $k$ . Se puede observar que ambos conjuntos de muestras son prácticamente coincidentes.

La Fig. 6 representa la localidad temporal presente en las referencias a los documentos. Es decir, cuando se produce una referencia a un documento es relativamente más probable que en un instante corto de tiempo se vuelva a producir una

nueva referencia a dicho documento. Éste fenómeno queda patente en la mencionada gráfica.

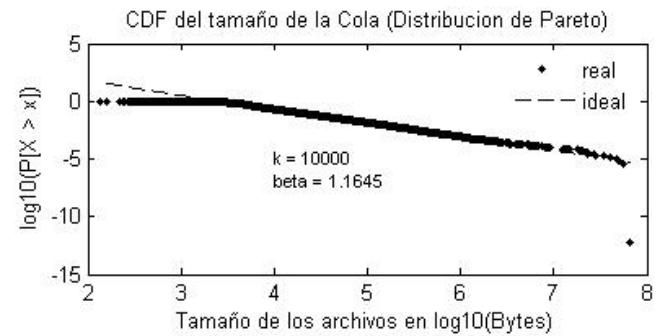


Fig. 5. Análisis de la pendiente de cola de Pareto

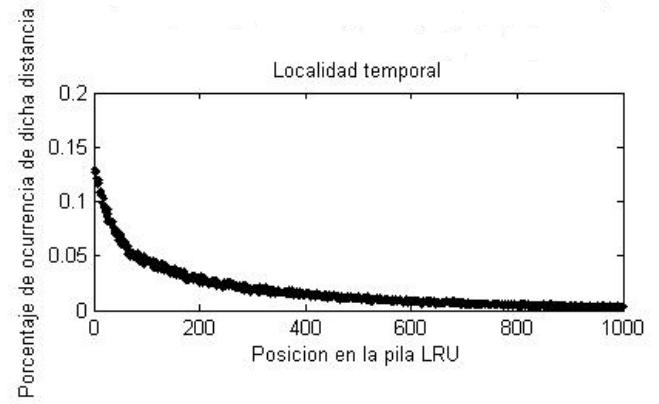


Fig. 6. Análisis de la localidad temporal en modo dinámico

#### IV. EL ENTORNO DE USUARIO

A la herramienta de generación de tráfico desarrollada se le ha provisto de un entorno gráfico para facilitar la configuración de los parámetros de generación del tráfico. La ventana principal de la aplicación puede verse en la Fig. 7.

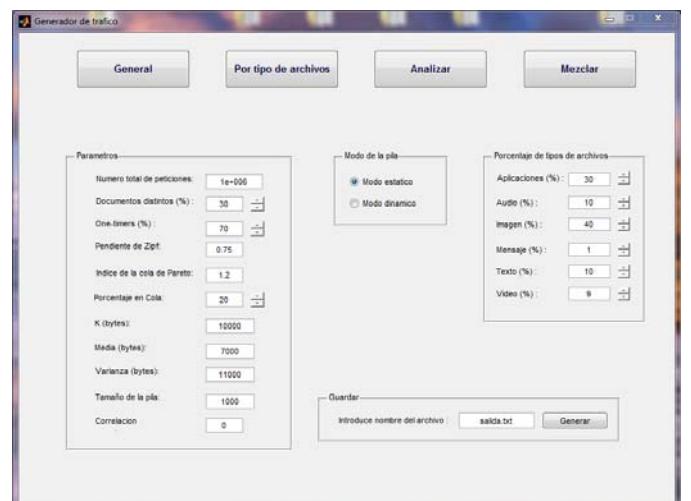


Fig. 7. Entorno de usuario del generador de tráfico

La aplicación se ha dividido en cuatro secciones a las que se accede mediante los botones situados en la parte superior de la ventana principal:

- General: En esta sección se introducen todos los parámetros necesarios para la generación de las muestras de tráfico sintético comentadas en apartados

- previos. Además se especifica el nombre del archivo en el que se generará la muestra.
- Por tipos de archivos: Esta sección es muy similar a la anterior, la principal diferencia es que antes de introducir los parámetros para la generación de las muestras de tráfico, se deberá indicar de qué tipo de documentos serán las muestras que se van a generar. Una vez elegido el tipo de documento, se introducirán los parámetros necesarios y el nombre del archivo a generar.
  - Analizar: Esta sección permite verificar el correcto funcionamiento de la aplicación, ya que muestra información relativa a las muestras que contiene el archivo seleccionado, como la Ley de Zipf, la función de probabilidad acumulada del cuerpo de la distribución de tamaño, la función de probabilidad acumulada de la cola de la distribución de tamaño, la correlación entre el tamaño de los documentos y su frecuencia de aparición, la función de probabilidad de los tamaños de los documentos y la localidad temporal de las repeticiones de las peticiones de los documentos comparadas con sus respectivas gráficas ideales.
  - Mezclar: Esta sección permite mezclar muestras generadas en distintos archivos. A priori, se realiza una mezcla aleatoria uniforme, dejando abierta la posibilidad de incluir cualquier otro tipo de distribución para dicha mezcla.

Una vez que se han introducido todos los parámetros y se ejecuta la generación de muestras, la aplicación genera un archivo con el nombre indicado por el usuario. Este archivo consta de una lista en la que cada fila hace referencia a un documento. Cada fila del fichero tiene tres campos: el primer campo es el identificador de archivo, especificado mediante un número para facilitar su procesamiento; el segundo campo indica el tamaño del archivo en bytes; y el tercer campo indica el tipo de archivo generado: aplicación, audio, imágenes, mensajes, texto y video.

## V. CONCLUSIONES

El presente artículo describe una aplicación desarrollada en Matlab para la generación de muestras sintéticas de tráfico para la evaluación del rendimiento de redes cableadas o inalámbricas con caché. La aplicación permite configurar los parámetros típicos a tener en cuenta en una caché Web, como son los documentos referenciados una única vez (*one-timers*), la popularidad de los documentos, la distribución del tamaño de los documentos, la correlación entre el tamaño de los documentos y su popularidad, la localidad temporal y los tipos de los documentos. Para cada una de las mencionadas características se ha especificado la forma en que han sido implementadas en la aplicación, comentando además los fundamentos matemáticos de las mismas.

La aplicación ha sido validada generando muestras de tráfico con diferentes características y comprobando que los valores generados coinciden con los ideales.

Finalmente, se ha procurado crear un entorno de usuario que permite, de forma sencilla e intuitiva, la generación de las muestras sintéticas de tráfico.

## AGRADECIMIENTOS

Este trabajo ha sido parcialmente financiado mediante el proyecto TEC2009-13763-C02-01

## REFERENCIAS

- [1] Proyecto IRCaché: <http://www.ircache.net/>
- [2] M. Busari, C. Williamson, "ProWGen: A Synthetic Workload Generation Tool for the Simulation Evaluation of Proxy Caches", Computer Networks, pp. 779-794. Jun. 2002.
- [3] F.J. Ridruejo, A. González, J. Miguel-Alonso. "TrGen: a Traffic Generation System for Interconnection Network Simulators", International Conference on Parallel Processing, 2005. 1st. Int. Workshop on Performance Evaluation of Networks for Parallel, Cluster and Grid Computing Systems (PEN-PCGCS'05). ICPP 2005 Workshops. 14-17 Junio 2005.
- [4] Poul E. Heegaard, "GenSyn - a Java based generator of synthetic Internet traffic linking user behaviour models to real network protocols", Presentation at ITC Specialist Seminar on IP Traffic Measurement, Modeling and Management, Monterey (EEUU), Septiembre 2000.
- [5] G. Abdulla, E. Fox, M. Abrams, y S. Williams, "WWW Proxy Traffic Characterization with Application to Caching" Technical Report TR-97-03, Computer Science Department, Virginia Tech., Marzo 1997.
- [6] A. Mahanti y C. Williamson, "Web Proxy Workload Characterization" Technical Report, Department of Computer Science, University of Saskatchewan, Feb. 1999.
- [7] P. Badford y M. Crovella, "Generating Representative Web Workloads for Network and Server Performance Evaluation", 1998 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems, pp.151-160, Julio 1998.
- [8] A. Mahanti, Web Proxy Workload Characterization and Modelling, M.Sc. Thesis, Department of Computer Science, University of Saskatchewan, Septiembre 1999.
- [9] M. Arlitt and C. Williamson, "Internet Web Servers: Workload Characterization and Performance Implications", IEEE/ACM Transaction on Networking, vol. 5, no 5, pp. 631-645, Oct. 1997.
- [10] M. Arlitt, L. Cherkasova, J. Dilley, R. Friedrich, y Tai Jin, "Evaluating Content Management Techniques for Web Proxy Caches", 2nd Workshop on Internet Server Performance, Atlanta, Georgia, Mayo 1999.
- [11] C. Roadknight, I. Marshall y D. Vearer, "File Popularity Characterization", 2nd Workshop on Internet Server Performance (WISP 99), Atlanta, Georgia, Mayo 1999.
- [12] Referencias a la ley de Zipf: <http://www.nslj-genetics.org/wli/zipf/>
- [13] V. Almeida, A. Bestavros, M. Crovella, y A. Oliveira, "Characterizing Reference Locality in the WWW", in Proceedings of the 1996 International Conference on Parallel and Distributed Information Systems (PDIS 96), pp. 92-103, Dic. 1996.
- [14] L. Breslau, P. Cao, L. Fan, G. Phillips, y S. Shenker, "Web Caching and Zipf-like Distributions : Evidence and Implications", IEEE Infocom '99 Conference, New York, NY, Marzo 1999.
- [15] M. Crovella and A. Bestavros, "Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes", IEEE/ACM Transactions on Networking, vol. 5, no.6, pp.835-846, Dic. 1997.
- [16] A. Law y W. Kelton, "Simulation Modeling and Analysis", Segunda Edición, Ed. Mc-Graw-Hill, 1991
- [17] A. Mahanti, C. Williamson, y D. Eager, "Traffic Analysis of a Web Proxy Caching Hierarchy", IEEE Network, vol. 14, no. 3, pp. 16-23, Mayo/Junio 2000.