

# Desempeño de tráfico tipo streaming en una red de datos simulada

Luciano Iglesias<sup>1</sup>, Luis Marrone<sup>2</sup>

Instituto de Investigación en Informática LIDI (III-LIDI)<sup>1</sup>,  
Laboratorio de Investigación en Nuevas Tecnologías Informáticas (LINTI)<sup>2</sup>,  
Facultad de Informática,  
Universidad Nacional de La Plata  
{li,lmarrone}@info.unlp.edu.ar

**Resumen** Este artículo presenta un estudio e investigación realizados, en vinculación al área de redes de datos, en particular referido al desempeño en la transferencia de información. Describe el proceso sobre cómo es dicha transferencia sobre una red simulada por software. Realiza una comparación de los desempeños alcanzados para distintas velocidades de transferencia de datos tipo streaming. Se analizan los resultados, en base a una serie de parámetros considerados como son el throughput, el delay (retardo), el packet loss (pérdida de paquetes) y el jitter (variación del retardo). Se concluye con una reflexión sobre qué parámetros afectan en mayor o menor medida las transferencias realizadas.

**Key words:** CBR, throughput, delay, packet loss, jitter, P2P, tráfico autosimilar, streaming, TCP

## 1. Introducción

Este trabajo está enmarcado en el contexto del desarrollo de una Tesis de Maestría en Redes de Datos, de la Facultad de Informática de la UNLP. En dicha tesis se analizan las herramientas para poder determinar cuál o cuáles son los mejores recursos a utilizar en una red P2P<sup>1</sup>, en función de su conectividad. Se tiene como objetivo llegar a armar un ranking que ordene los nodos (que tienen el recurso de streaming deseado), de acuerdo al nivel de conectividad que se tiene con cada uno de estos y al tipo de recurso (voz, música, video, video en alta definición, etc.). Al mismo tiempo, para llegar a esta meta es necesario analizar cuáles son los parámetros que influyen en el proceso transferencia de datos, y de qué manera lo hacen.

---

<sup>1</sup> Las redes P2P o peer-to-peer son sistemas distribuidos consistentes de nodos de borde interconectados capaces de autoorganizarse dentro de topologías de redes con el propósito de compartir recursos tales como contenidos, ciclos de CPU, almacenaje y/o ancho de banda, capaces de adaptarse a las fallas y acomodarse a poblaciones transitorias de nodos mientras mantienen una aceptable conectividad y rendimiento, sin requerir intermediación o soporte de un servidor o autoridad global centralizada.

La velocidad de transferencia sobre un enlace de datos está, principalmente, determinada por las características de dicho enlace y las capacidades de cómputo de los sistemas intermedios, que llevan a cabo la comunicación. El throughput, el delay, el packet loss y el jitter son parámetros que cuantifican dicha transferencia [1].

El tráfico de datos es la secuencia de movimientos de elementos de datos a través de diversos dispositivos físicos. El típico elemento de datos es una secuencia continua de bits que conforman un paquete de datos. Cuando estos paquetes pasan a través de un dispositivo físico, se produce habitualmente una demora provocada por la recepción (generalmente acompañada del encolamiento del paquete), luego por el procesamiento (donde el paquete es analizado), y por último, por el despacho (donde finalmente el paquete abandona el dispositivo). Todo este procedimiento causa demoras. Generalmente, los arribos se encolan sucesivamente como entradas de los clientes y allí, experimentan una posible espera, para luego ser servidos antes de que se produzca su despacho exitoso por la salida. Para poder construir un modelo del arribo y posterior servicio se utilizan variables aleatorias. La naturaleza estadística de los arribos puede ser expresada de diferentes formas. Por ejemplo, si los sucesivos tiempos entre arribos (interarrival times – IATs) son independientes, una especificación de las condiciones iniciales (que determine el instante de tiempo en el cual la operación de encolamiento comienza), en conjunto con la Función de Densidad de Probabilidad (probability density function – pdf) de los IATs, son suficientes para describir completamente la naturaleza de los arribos. La variable aleatoria con distribución de Pareto para los IATs permite armar uno de estos modelos. Esta variable aleatoria exhibe algunas características importantes, según los valores de los parámetros de su pdf. Su varianza puede ser finita o infinita. Las variables aleatorias con varianza infinita encuentran aplicación en la caracterización de tráfico de datos a ráfagas [2].

Estos aspectos teóricos se exponen aquí con el fin de dar contexto a los experimentos que luego se describen en este trabajo, y que retomarán este marco. A continuación se hace una breve revisión de algunos enfoques que justifican la utilización de este mecanismo para modelizar el tráfico.

## 2. Internet y el tráfico autosimilar

A partir de 1993, aparecieron varios estudios en la literatura que documentaban que el patrón del tráfico de datos se puede modelar de manera correcta por un proceso autosimilar, en una amplia variedad de situaciones relacionadas con redes de datos del mundo real. Existen muchos ejemplos de tráfico que se adecuan mejor a un modelo autosimilar que a uno de Poisson (que venía siendo utilizado), entre los que se encuentra el tráfico Ethernet, el HTTP, TCP sobre enlaces WAN, la parte interactiva de una conexión TELNET, la etapa de transferencia de datos sobre una conexión FTP. A partir de todos estos acontecimientos es que cobra sentido armar un escenario de simulación con tráfico autosimilar para trazar un paralelismo con lo que pasa hoy en Internet [3].

El fenómeno de la autosimilaridad tiene un profundo impacto en el desempeño de las redes de datos. Estudios basados en la teoría tradicional de colas (que utilizan la distribución de probabilidad de Poisson) muestran que el crecimiento brusco del delay en un sistema, comienza recién cuando se alcanza una utilización del 80 %, mientras que capturas de tráfico Ethernet e ISDN muestran que ese crecimiento brusco empieza antes, entre el 50 y el 60 %. Diferentes investigaciones revelan que esos problemas de desempeño se deben a la autosimilaridad [3]. Uno de los descubrimientos más importantes que se realizó sobre Ethernet, es que a medida que la carga aumenta, el grado de autosimilaridad aumenta. Este resultado es sin duda muy importante, ya que es justamente con mucha carga, donde las cuestiones relacionadas al desempeño toman un papel relevante. A partir de estos resultados, se empezó a abandonar la teoría de que la agregación de tráfico de datos generado a partir de la multiplexación un gran número de fuentes independientes, resulta en un proceso de Poisson.

La existencia de tráfico autosimilar en los backbones, producto de la agregación de flujos de múltiples fuentes, incide en el desempeño de los routers, ya que estos perciben que los paquetes llegan de ráfagas. Dichas ráfagas degradan el desempeño porque deben procesar mucha información de golpe, lo que agrega demoras al procesamiento de los paquetes y termina incrementando el delay, o debido a este incremento se produce el llenado de las colas en los routers lo que produce el descarte de paquetes y su consecuente degradación en los flujos individuales. Todo este fenómeno tiene un impacto directo en el tamaño de los buffers, con los que los routers deben contar, como para poder hacer frente a los requerimientos del tráfico en situaciones de alta carga. No es la intención de este trabajo determinar cuál es el tamaño que esos buffers deben tener [4], pero sí ver cómo influye la existencia del tráfico autosimilar en un flujo individual entre dos hosts en los bordes de la red.

### **3. Metodología para el desarrollo de la investigación**

Luego de abordar el estudio teórico de la temática planteada, se diseñó un trabajo experimental con el fin de analizar los diferentes parámetros que intervienen en la transmisión de información en una red de datos, y de esta manera, generar una estrategia adecuada para el armado del ranking de nodos, que se mencionó inicialmente en este artículo.

Para el análisis empírico de las transferencias se utilizó un simulador de redes de datos discreto llamado Network Simulator [5] en su versión 2, que se nombrará a partir de ahora como ns2. El tráfico que se usó para la simulación es del tipo CBR (Constant Bit Rate), ya que es acorde al tipo de flujo que generan las aplicaciones de streaming.

En lo que resta de este artículo se describirá: cómo se diseñó y planificó el escenario de simulación y cómo se llevaron a cabo las mismas. Se presentarán los resultados que se obtuvieron de manera gráfica para poder verlos comparativamente. Finalmente, se realizará una interpretación de dichos resultados.

## 4. Escenario de simulación

La generación del core de la red (un grafo de 200 nodos) se llevó a cabo usando un software que de manera aleatoria dispuso los nodos y los enlaces entre ellos, de forma que el grafo quede totalmente conexo. Los routers (nodos) tuvieron todos un grado mayor o igual a uno lo que garantizó que todos los nodos estuvieran conectados con al menos uno de los otros nodos. Los enlaces (aristas) tuvieron asignados aleatoriamente la velocidad de conexión full-duplex entre 3 y 4 Mbps, y el delay menor a 3 milisegundos. Por otro lado, se conectaron dos host (nodos con un único vínculo a la nube de routers), que fueron los encargados de realizar las transferencias de streaming, uno como emisor y el otro como receptor. Estos hosts se conectaron vía un enlace con una velocidad de conexión full-duplex de 2 Mbps y un delay de 10 milisegundos. El tráfico entre el emisor y el receptor fue encaminado a través de siete saltos, de entre los routers de la red.

## 5. Generación sintética de tráfico autosimilar

El tráfico autosimilar fue modelizado mediante la agregación de fuentes ON-OFF que utilizan la distribución de Pareto, mencionada en la introducción de este trabajo. Los paquetes son enviados a velocidad fija durante los períodos ON, y no se envían paquetes durante los períodos de OFF. Ambos períodos son tomados de dos variables aleatorias independientes con distribución de Pareto, teniendo un tamaño de paquete fijo. Estas fuentes pueden ser usadas para generar tráfico agregado que exhiba Dependencia de Largo Alcance (Long Range Dependency - LRD).

En la simulación sobre ns2 se utilizó un tiempo de ráfaga inicial (ON) de  $\frac{1}{2}$  segundo, un tiempo ocioso inicial (OFF) de  $\frac{1}{2}$  segundo, una velocidad constante de transferencia de 200 Kbps, un tamaño de paquete fijo de 210 bytes, y se pasó como parámetro de forma de la distribución de Pareto 1,5 [6] (lo que hace que tenga una varianza infinita).

## 6. Ejecución de la simulación

Una vez creado el grafo se procedió a iniciar, en cada nodo, una fuente de transmisión de datos con distribución de Pareto comenzando en un instante aleatorio. De esta forma, se obtiene un escenario de tráfico autosimilar sobre el que se cursan los datos a medir. Para poder analizar los factores involucrados en el tiempo de transferencia, en la investigación, se llevaron a cabo una serie de simulaciones con ns2. Se eligieron ciertos escenarios que a continuación se describen.

Para desarrollar el análisis empírico a través de la simulación se armaron tres categorías de tráfico, según la velocidad a la que se inyectaron los datos en la comunicación. Las tres categorías usaron un tráfico del tipo CBR (Constant Bit Rate), a diferentes velocidades. Una funcionó a 400Kbps, otra a 1Mbps y la otra a 1,5Mbps. El propósito de tener las diferentes categorías es ver qué ocurre

con aplicaciones que requieren esas velocidades de transferencia para funcionar de manera correcta. Dicha simulación se llevó a cabo para distintos volúmenes de datos comenzando con 1KByte hasta 600KBytes iterando con un incremento de 1KByte por simulación, para cada una de las tres velocidades que fueron mencionadas antes. Por cada simulación se generó un archivo con la traza y se calculan las métricas que se mencionan a continuación.

## 7. Métricas o parámetros considerados en la simulación

A continuación se listan las métricas utilizadas y la forma en que se calcularon para el flujo CBR:

Throughput:

$$T = \frac{\text{successful\_data}}{\text{transmission\_time}} \quad . \quad (1)$$

Delay Promedio extremo-a-extremo:

$$D = (T_d - T_s); D_p = (1/pkts) \sum_{i=1}^{pkts} D_i \quad . \quad (2)$$

donde,  $T_d$  es el instante de recepción del paquete en el destino y  $T_s$  es el instante de emisión del paquete en el origen. Si  $T_d$  no existe, porque el paquete no llega a destino, D no se puede calcular.

Jitter Máximo:

$$J = |D_{i+1} - D_i|; J_m = \text{MAX}(J_i) \quad i \in [1, pkts] \quad . \quad (3)$$

donde,  $D_i$  es el delay del i-ésimo paquete y  $D_{i+1}$  el delay del siguiente. Tanto  $D_i$  como  $D_{i+1}$  deben ser calculables para que J se pueda calcular.

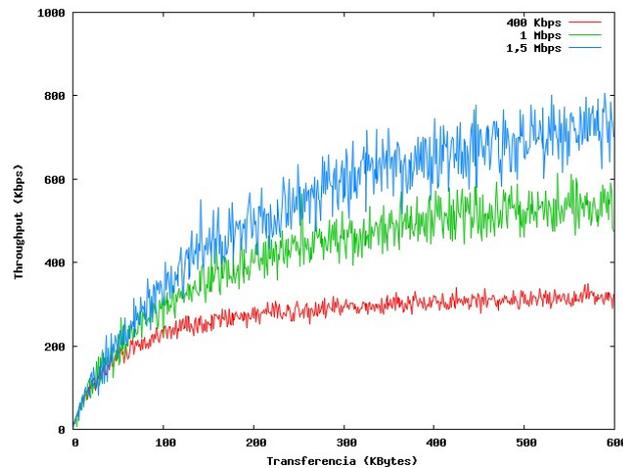
Packet Loss:

$$L = \left[ \left( \frac{\text{drop\_pkts}}{\text{sent\_pkts}} \right) 100 \right] \% \quad . \quad (4)$$

A partir de la ejecución de la simulación planteada, se abordó el análisis de resultados con el fin de considerar cómo impactan los diferentes parámetros tenidos en cuenta, y cómo esta información resulta relevante a los fines planteados de realizar el ranking de nodos que poseen el recurso deseado. A continuación se detallan los resultados, organizándolos a partir de las métricas consideradas.

## 8. Resultados de la simulación

A partir de la traza generada por la ejecución de las sucesivas simulaciones, se calcularon las métricas antes mencionadas.



**Figura 1.** El eje de las  $x$  denota el *tamaño de la transferencia*, que va de 1 a 600 Kbytes, mientras que el eje de las  $y$  determina el *throughput* de cada una de las transferencias. Cada color indica una velocidad de transmisión distinta y tiene su referencia en la parte superior derecha de la figura.

### 8.1. Throughput

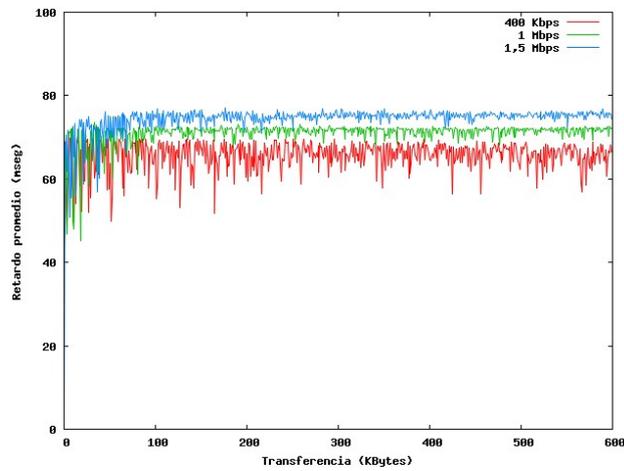
La Fig. 1, permite observar el throughput. Ninguno de los 3 casos alcanza un nivel satisfactorio, ya que está siempre muy por debajo del desempeño teórico para el enlace. Para el caso de las transferencias a 400 Kbps, parecería estabilizarse en torno a los 350 Kbps, lo que resulta en el rendimiento más aceptable. Para los casos de 1 y 1,5 Mbps, promedia en la mitad de lo pretendido, lo que es aún peor que en el caso de 400 Kbps. Se puede vislumbrar que el nivel de packet loss tiene que ser significativo, porque los valores reales del throughput están muy lejos de los deseados. Para lograr una calidad aceptable de streaming por estos canales se requiere un buffering importante en el receptor, meticulosamente calculado, y que demandaría una espera inicial prolongada.

### 8.2. Delay

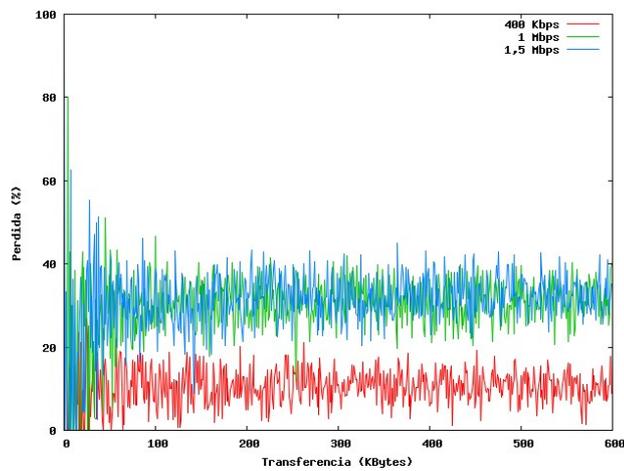
Como se puede observar, a través de la Fig. 2, el delay tiene un máximo relativamente constante en todas las categorías, y tiene algunos picos hacia abajo que son especialmente significativos en la conexión de 400 Kbps. Esto permite deducir que la espera en los routers es, usualmente, la misma para cada uno de los paquetes, y que tiene ciertos momentos más aliviados en la transferencia de 400 Kbps, que no aparecen con las otras dos velocidades de 1 y 1,5 Mbps.

### 8.3. Packet Loss

La Fig. 3, muestra que la pérdida de paquetes es muy alta en todos los casos, y empeora a medida que aumenta la velocidad de transferencia. Esto



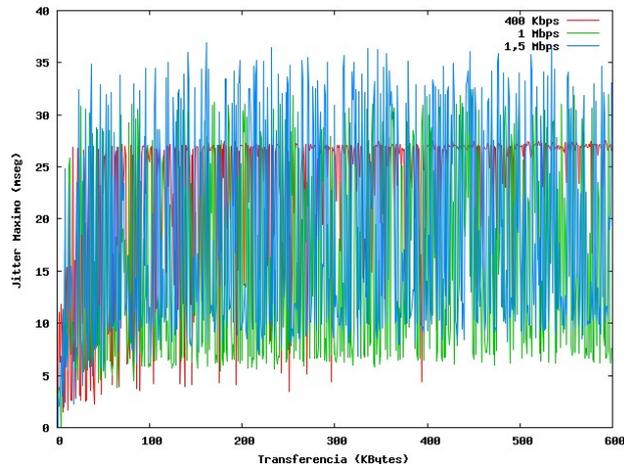
**Figura 2.** El eje de las  $x$  denota el *tamaño de la transferencia*, que va de 1 a 600 Kbytes, mientras que el eje de las  $y$  determina el *delay promedio* de cada una de las transferencias. Cada color indica una velocidad de transmisión distinta y tiene su referencia en la parte superior derecha de la figura.



**Figura 3.** El eje de las  $x$  denota el *tamaño de la transferencia*, que va de 1 a 600 Kbytes, mientras que el eje de las  $y$  determina el *porcentaje de packet loss* de cada una de las transferencias. Cada color indica una velocidad de transmisión distinta y tiene su referencia en la parte superior derecha de la figura.

es coincidente con la merma producida en el throughput. Este es uno de los principales enemigos del streaming, y requiere de un protocolo que sea resistente a las pérdidas, que implemente mecanismos de retransmisión o de recuperación de estas situaciones.

#### 8.4. Jitter



**Figura 4.** El eje de las  $x$  denota el *tamaño de la transferencia*, que va de 1 a 600 Kbytes, mientras que el eje de las  $y$  determina el *jitter máximo* de cada una de las transferencias. Cada color indica una velocidad de transmisión distinta y tiene su referencia en la parte superior derecha de la figura.

El jitter, como ya se mencionó, se define como la variación del retardo, por lo que un delay relativamente constante refleja un jitter, también, relativamente constante y bajo. La figura 4 muestra cómo se comportan los peores casos, y siempre están por debajo de los 40 ms, lo que es un resultado bueno para una aplicación de tipo streaming.

Este análisis acerca de cómo se comportan las diferentes métricas en el escenario simulado, con las consideraciones enunciadas, permite arribar a las primeras conclusiones de la investigación, que no son más que un punto inicial para continuar el camino hacia el objetivo final planteado. En el siguiente apartado, se enuncian algunas de estas conclusiones preliminares.

## 9. Conclusiones

A partir de las simulaciones realizadas, se puede observar considerando los gráficos de las figuras presentadas, cómo influye el tráfico autosimilar agregado

en cada una de las métricas calculadas, siendo el packet loss y, por consiguiente, el Throughput los dos más afectados negativamente. Este análisis es válido debido a la amplia variedad de protocolos que tienen en el tráfico que generan un fuerte componente autosimilar estadístico, cómo ya se mencionaron anteriormente diversos ejemplos. Sin embargo, esto no invalida otra diversidad de análisis que puedan hacerse utilizando la teoría de colas tradicional que puede utilizarse para modelar otras tantas situaciones donde el tráfico se adecua mejor a dicho modelo.

Este trabajo se enfoca en ver cómo es afectado un flujo de streaming por el resto del tráfico. A diferencia del tráfico TCP, que se autoregula gracias a sus mecanismos para evitar la congestión, el rate de inserción en el streaming es CBR. Además, se enmarca en la influencia que tiene el tráfico autosimilar que circula por la red. La elección de llevar a cabo la simulación en una red cuyo tráfico es autosimilar, se debe al gran volumen de tráfico autosimilar que hay en Internet. El tráfico P2P, así como la World Wide Web (HTTP), tiene un comportamiento autosimilar. A partir de esto resulta de interés, modelar el tráfico de Internet (o por lo menos una parte grande de ella) como autosimilar.

A la luz de los resultados, se puede ver que existe una relación entre el throughput  $t$  tan bajo que se puede observar en todos los casos, y un packet loss alto. En los casos de 1 y 1,5 Mbps el descarte de paquetes oscila entre el 20% y el 45%. Estos dos parámetros muestran que hay un alto grado de congestión en la red.

Los tres casos podrían no ser los ideales para realizar una transferencia de tipo streaming, ya que requerirían de un alto grado de buffering para amortiguar las malas condiciones del enlace. Al principio se planteó el objetivo de armar un ranking de nodos ordenado según la conveniencia en utilizarlos para hacerse del servicio de streaming deseado, pero como por ahora se realizó la simulación con una única fuente sólo podemos adelantar, que en esta etapa de la investigación todavía falta verificar otras fuentes para poder armar el ranking y elegir.

Como se mencionó con anterioridad, estos primeros resultados y conclusiones, constituyen elementos centrales para continuar el camino de la investigación planteada.

## Referencias

1. Al Morton, Vilho Raisanen, y Glenn Grotefeld, Network performance measurement with periodic streams, Noviembre 2002, <http://tools.ietf.org/html/rfc3432>
2. G.R. Dattatreya, Performance Analysis of Queuing and Computer Networks, 1o ed. (Chapman and Hall/CRC, 2008).
3. William Stallings, High-speed networks: TCP/IP and ATM design principles (Prentice Hall New Jersey, 1998).
4. Guido Appenzeller, Isaac Keslassy, y Nick McKeown, Sizing Router Buffers, IN PROCEEDINGS OF ACM SIGCOMM (2004): 281–292.
5. The Network Simulator - ns-2 <http://www.isi.edu/nsnam/ns/>
6. William Stallings, High-Speed Networks and Internets: Performance and Quality of Service, 2o ed. (Prentice Hall, 2002).
7. Karanjit S. Siyan, Inside TCP/IP, 3o ed. (New Riders, 1996).
8. Dunigan, Tom, Optimizing bulk transfers over high latency/bandwidth nets <http://www.csm.ornl.gov/~dunigan/netperf/bulk.html> – Jan 2003.
9. Malli, Mohammad, Barakat, Chadi, Application-Aware Model for Peer Selection. Thème COM - Systèmes communicants Projet Planète Rapport de recherche n 5587 - May 2005.
10. Teng, Fei, Shu, Tao, Lixin, Gao, Guerin, R., Zhi-li, Zhang, Light-Weight Overlay Path Selection in a Peer-to-Peer Environment. Dept. Elec. & Comput. Eng., Univ. of Massachusetts, Boston, MA.
11. Wei, Li, Shanzhi, Chen, Tao, Yu, UTAPS: An Underlying Topology-Aware Peer Selection Algorithm in BitTorrent, Advanced Information Networking and Applications, 2008. AINA 2008. 22nd International Conference on Volume , Issue , 25-28 March 2008 Page(s):539 – 545.