

Dimensionamiento de redes inalámbricas de banda ancha utilizando NS-3 mediante síntesis de tráfico autosimilar

Rabinovich, Daniel
Universidad Blas Pascal

Osimani, César
Universidad Blas Pascal

Olocco, Gino
Universidad Blas Pascal

Abstract

En el dimensionamiento de una red inalámbrica de banda ancha frecuentemente se tiene la dificultad de no poder reproducir repetidamente una situación real del tráfico que la red debe atender. Esto complica la evaluación de las consecuencias de cambios en los parámetros de configuración más significativos. El presente trabajo propone utilizar el entorno de simulación de redes NS-3 para crear perfiles de usuarios que generen tráfico que responda a un entorno real, y luego utilizarlo para el dimensionamiento de redes inalámbricas de banda ancha. Considerando que el tráfico de paquetes no limitado responde a una distribución autosimilar es posible dimensionar los enlaces críticos de la red analizando la autosimilaridad del tráfico simulado mediante la utilización de logscale diagrams.

Palabras clave

autosimilaridad, captura, logscale diagram, NS-3, redes, simulación, tráfico

Introducción

Para el dimensionamiento de una red es de gran utilidad poder contar con un perfil de tráfico específico que se pueda repetir a voluntad para realizar el análisis del impacto que producen los cambios en los parámetros de red. El tráfico real a servir se genera en determinados momentos del día, semana y/o mes, por lo que actuar solamente en esos momentos sería una forma poco eficiente de diseñar la red. Poder generarlo en un entorno real supone disponer de un gran número de recursos operando en forma simultánea, lo que resulta casi imposible en la práctica. Surge entonces la necesidad de tener un tráfico que se comporte como el real, pero que esté disponible en todo momento para el diseñador. Ese tráfico debe primero caracterizarse, y luego generarse en forma artificial. El objetivo principal es demostrar que es posible utilizar tráfico sintético, generado con el simulador NS-3 [1], para dimensionar redes inalámbricas o mixtas, determinando qué tan autosimilar es el mismo

frente a la modificación de los principales parámetros de red.

Estudio del tráfico de la red

La finalidad del estudio de tráfico de la red es el dimensionamiento apropiado de los dispositivos de la misma. Simplificadamente, esto es hablar del tiempo de servicio y de la longitud de las colas (figura 1). Para ello es necesario estimar el comportamiento del tráfico, y dado que es un fenómeno aleatorio, sólo puede expresarse convenientemente por los valores estadísticos correspondientes a la distribución identificada, tales como promedio, varianza, función de distribución acumulativa complementaria (CCDF), función de autocorrelación (ACF), etc.

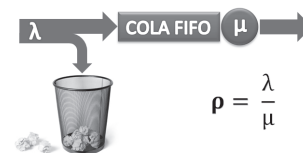


Fig. 1: Versión simplificada de cualquier dispositivo de red: μ es la velocidad de servicio y λ es la velocidad de arribo de paquetes.

Lo primero que se realizó fue estudiar el tráfico de la red de la UBP (Universidad Blas Pascal). A tal fin se acordó con las autoridades del centro de cómputos realizar la captura allí mismo, ya que existe una división lógica lo suficientemente restringida por las configuraciones de enturamiento como para que las capturas correspondieran en su gran mayoría a tráfico de los APs (Access Points) o del módem del ISP (Internet Service Provider). La red está compuesta tal y como se muestra en la figura 2. Como una excepción se nos informó que en el acceso cableado al AP de la cantina había conectado un consumidor de Internet no inalámbrico.

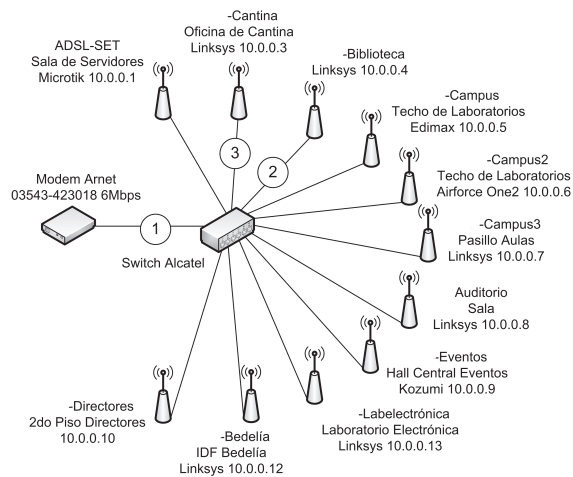


Fig. 2: Topología de la red inalámbrica de la UBP.

Captura de tráfico

Se llevaron a cabo capturas de paquetes utilizando el software Wireshark [2]. Asumiendo que esta red no tiene conectadas estaciones cableadas, las capturas básicamente están constituidas por paquetes que han sido o serán transportados inalámbricamente mediante WiFi. Con el objetivo de que la cantidad y tamaño de los archivos de las capturas tuvieran valores manipulables se realizó un análisis de capturas preliminares del cual surgió que una longitud de 79 bytes es suficiente para almacenar hasta los encabezados más largos de cualquier frame. La tabla 1 resume los datos de las muestras capturadas.

Tabla 1: Información sobre capturas realizadas.

Punto	1	2	3
Muestra	T1111019	T1111103	T1111118
Inicio	19/10/11 18:38:30	03/11/11 16:03:26	18/11/11 17:11:05
Fin	26/10/11 11:19:31	18/11/11 17:08:35	06/12/11 16:49:37
Horas	185	362	432
Frames	84.910.247	13.760.335	8.385.194
Bytes	55,6e09	8,4e09	4,5e09

Las figuras 3 y 4 corresponden a la muestra T1111019, en paquetes y en bytes respectivamente, donde cada barra representa una hora de tráfico. Si bien la figura 4 muestra una clara preeminencia del download sobre el upload, prácticamente en una relación 8:1, en la figura 3, que expresa el tráfico en paquetes, la relación es apenas 1,3:1. Esto responde a la característica de acuse de recibo del protocolo TCP, para el cual se utilizan paquetes sin carga útil con la bandera ACK seteada en '1'. Según la figura 4, la máxima cantidad de bytes recibidos desde Internet

durante la hora pico es de 13,1e08; esto permite calcular una velocidad, promediada en una hora, de 364kBps o 2,91Mbps. También se aprecia, en las figuras 3 y 4, que fuera de los horarios de actividad académica, Sábados, Domingos, y días de semana entre las 23 hs. y la 9 hs. el tráfico cae a niveles muy bajos.

Análisis de autosimilaridad

Un objeto autosimilar [3] es exactamente, aproximadamente o estadísticamente similar a una o más partes de sí mismo. En la figura 5 se muestran algunos ejemplos. En el tráfico de datos, este fenómeno es también descrito como "invariancia de escala". [4]

La autosimilaridad se cuantifica con el coeficiente de Hurst, H . Sólo existe autosimilaridad si H varía entre 0,5 y 1, siendo 1 el más alto grado. Una de las estimaciones del coeficiente de Hurst más confiable y rápida de calcular [5], se realiza mediante el análisis de los gráficos LD (Logscale Diagrams). Si los LDs presentan una traza predominantemente recta en la mayoría de las octavas, con una pendiente entre 0 y 1, se trata de tráfico autosimilar. En la figura 6 se muestran los LDs de tráfico autosimilar de Internet correspondiente a dos capturas, en un horario sin saturación y en otro con saturación debido a la congestión [6]. En la gráfica del vínculo saturado se aprecia un aumento de la energía para las primeras octavas, en comparación con la gráfica del vínculo no saturado. Esto indica una variación importante en el tráfico medido en unidades pequeñas de tiempo, relacionado con la superposición de ráfagas de transmisión y el accionar de los mecanismos de control de congestión de TCP. [7]

El coeficiente de Hurst H está vinculado con la pendiente β mediante (1).

$$H = \frac{1 + \beta}{2} \quad (1)$$

Crovella y Bestavros [8] fueron unos de los primeros en buscar posibles causas, en los niveles más altos del sistema, para que el tráfico de la WWW sea autosimilar. Entre ellas enumeran la distribución del tamaño de los archivos, efectos del *caching*, preferencias de los usuarios, el efecto del tiempo de pensamiento de los usuarios, y superposición de muchas transferencias de archivos en una LAN (Local Area Network). Ellos explican la distribución autosimilar del tráfico de red en términos de los tiempos de transmisión individual, basados en el mecanismo descrito en [9].

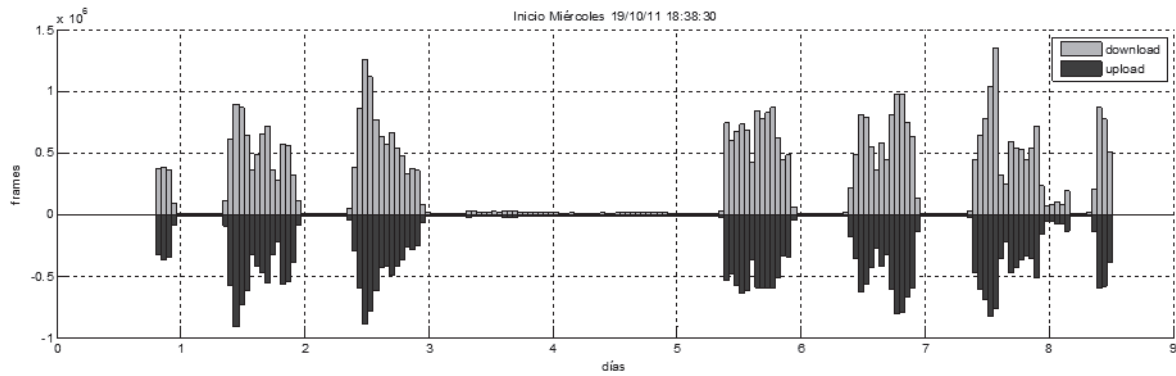


Fig. 3: Paquetes por hora para la muestra T111019.

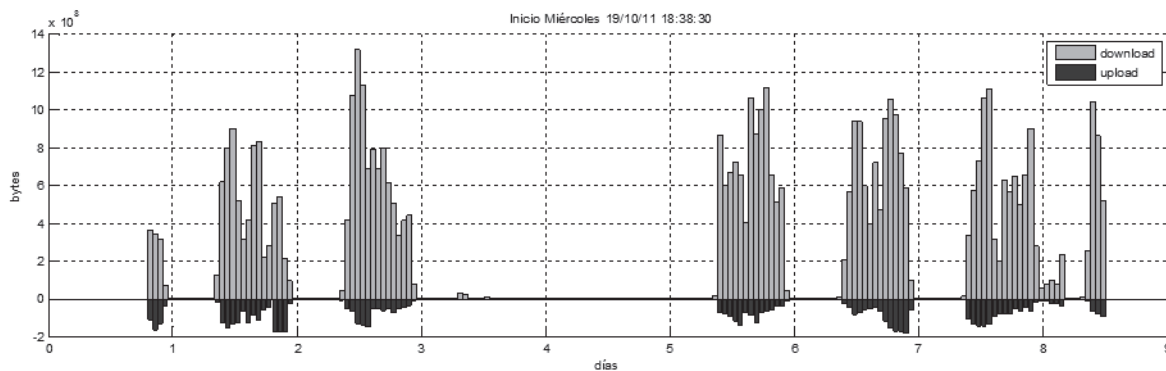


Fig. 4: Bytes por hora para la muestra T111019.

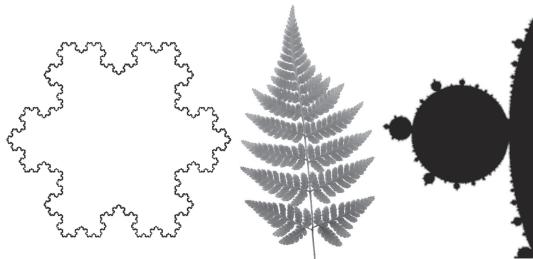


Fig. 5: Objetos autosimilares. De izquierda a derecha: curva de Koch, helecho, secuencia de Mandelbrot.

Dado que lo se busca es encontrar un perfil de usuario promedio para el dimensionamiento de redes, se realizó un extenso análisis de autosimilaridad sobre el comportamiento de aquellos en la captura T111019, para determinar la porción de tráfico óptima a utilizar como control.

En la figura 7 se muestran los LDs correspondientes a tráfico del momento pico de consumo según la figura 4, y a tráfico de otro momento no pico. Estas muestras fueron nombradas, considerando día y hora dentro de la captura, como T111019-2-11 y T111019-1-14, respectivamente. La elección de la muestra T111019-1-14 quedará evidenciada inmediatamente.

Se han aproximado las trazas con una recta para determinar su pendiente y, con dicho valor, se calculó un coeficiente de Hurst de 0,75 según (1), indicando un grado de autosimilaridad aceptable. Nótese la semejanza entre ambas trazas de la figura 7 y, a su vez, con la gráfica no congestionada de la figura 6.

Como puede apreciarse en la figura 7, los LDs de ambas muestras presentan un crecimiento homogéneo. Se decidió entonces utilizar como control para la simulación la muestra T111019-1-14, debido a que presenta un comportamiento tan autosimilar como el de la muestra T111019-2-11 y la magnitud de la misma es considerablemente menor, lo que simplificará las simulaciones en cuestiones de tiempo.

Análisis estadístico del tráfico capturado

Wireshark se ha empleado para realizar las capturas y luego se utilizó para generar estadísticas. Para ello se aplicó un filtro a toda la muestra T111019-1-14 limitando el análisis a los frames que van o vienen hacia o desde la nube de Internet (punto 1 en la figura 2).

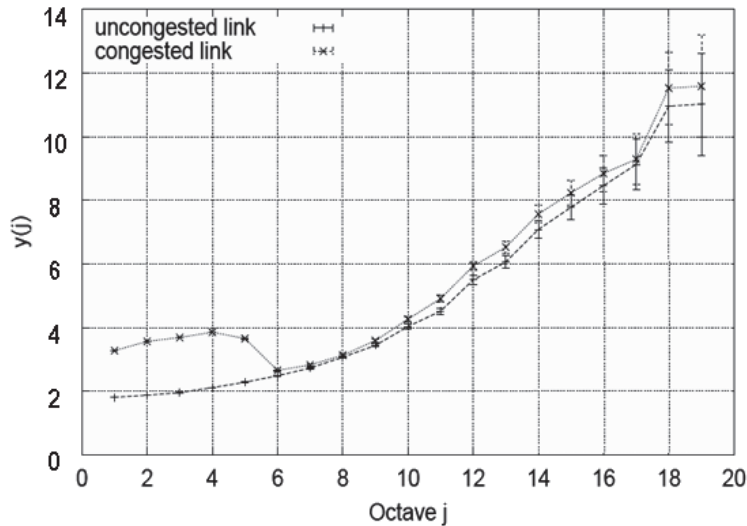


Fig. 6: LDs mostrando efectos de la congestión en el tráfico.

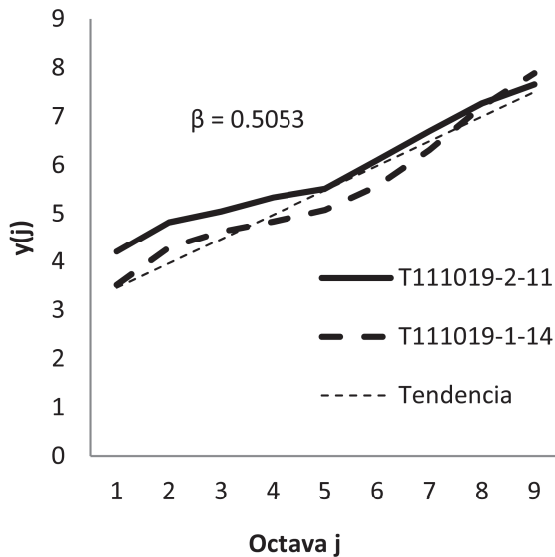


Fig. 7: Análisis de autosimilaridad sobre las muestras T111019-2-11 y T111019-1-14.

El efecto en la cantidad de bytes totales es pequeño, ya que se excluyen algunos frames ICMP, protocolos de enrutamiento, etc. La muestra en observación disminuye de 432.601.823 a 429.734.690 bytes, esto es -0,66 %. El filtro es *eth.addr==00:0c:42:32:e3:b0*, correspondiente a la dirección de capa 2 del módem xDSL que brinda la conexión de 6Mbps. Los paquetes que viajan hacia o desde Internet tendrán como destino u origen tal dirección de capa 2, respectivamente.

La figura 8 muestra, siempre haciendo referencia a todos los bytes que construyen los frames, que el 99,94 % del tráfico es IPv4, y está conformado en un 98,45 % por

el protocolo TCP y en un 1.2 % por el protocolo UDP. El 0.35 % restante está compuesto por paquetes ARP, ICMP y LLC.

La figura 9 muestra la composición del tráfico de download. El protocolo HTTP, que es el que mayoritariamente se emplea para navegar por la web, constituye el 95,57 % del tráfico total de download. La tabla 2 muestra valores estadísticos obtenidos con Wireshark de la muestra T111019-1-14.

El tráfico de redes, y en especial el tráfico de Internet, es un fenómeno complejo y cambiante; un campo peligroso para arriesgar simplificaciones. Por otro lado si se pretende generar tráfico simulado para estimar el comportamiento de una red, sea esta simulada o real, instalar muchos generadores de tráfico de varios tipos resultaría en una cantidad excesiva de parámetros de ajuste. Esta situación oscurecería la claridad de las relaciones causa-efecto que se quieren establecer. Dado que el tráfico HTTP representa aproximadamente un 96 % del tráfico total de download (figura 9), constituye la materia prima más importante para lo que se pretende simular, por lo que se determinó desarrollar aplicaciones en los nodos para generar tal tipo de tráfico, de manera que el mismo tenga un comportamiento autosimilar, característica a analizar que normalmente se da en tráfico real de Internet.

Generación de tráfico autosimilar

El tráfico autosimilar puede ser construido multiplexando una gran cantidad de fuentes individuales SI-NO (ON-OFF sources) [8] [9] [10], que tienen intervalos de SI transmisión y de NO transmisión con distri-

Display filter: eth.addr==00:0c:42:32:e3:b0										
Protocol	% Packets	Packets	% Bytes	Bytes	Mbit/s	End	Packets	End	Bytes	End
Frame	100.00 %	565152	100.00 %	429734690	1.905	0	0	0	0.000	
Ethernet	100.00 %	565152	100.00 %	429734690	1.905	0	0	0	0.000	
Internet Protocol Version 4	99.20 %	560636	99.94 %	429463172	1.904	0	0	0	0.000	
Transmission Control Protocol	91.97 %	519744	98.45 %	423069149	1.876	206326	12801319		0.057	
User Datagram Protocol	6.90 %	39010	1.20 %	5175850	0.023	2	120		0.000	

Fig. 8: Estadística de protocolos generada por Wireshark, sobre la muestra T111019-1-14.

Display filter: eth.src==00:0c:42:32:e3:b0										
Protocol	% Packets	Packets	% Bytes	Bytes	Mbit/s	End	Packets	End	Bytes	End
Frame	100.00 %	322538	100.00 %	371685136	1.648	0	0	0	0.000	
Ethernet	100.00 %	322538	100.00 %	371685136	1.648	0	0	0	0.000	
Internet Protocol Version 4	99.02 %	319366	99.95 %	371494258	1.647	0	0	0	0.000	
Transmission Control Protocol	94.22 %	303892	99.41 %	369490896	1.638	44620	2777918		0.012	
Hypertext Transfer Protocol	75.55 %	243678	95.57 %	355230217	1.575	596	42801		0.000	

Fig. 9: Composición del tráfico TCP download sobre la muestra T111019-1-14.

Tabla 2: Resumen de los resultados estadísticos del segmento de la muestra T111019-1-14.

Parámetro	Valor	Unidad
Duración de la captura	1800	segundos
Direcciones IPv4 locales	52	direcciones
Download TCP (incluye frame Ethernet)	369.503.478	bytes
Upload TCP (incluye frame Ethernet)	53.578.253	bytes
Download TCP	304.025	paquetes
Upload TCP	215.852	paquetes
Direcciones IPv4 locales HTTP	36	direcciones
Download TCP HTTP (incluye frame Ethernet)	356.522.802	bytes
Upload TCP HTTP (incluye frame Ethernet)	17.606.904	bytes
Download TCP HTTP	264.394	paquetes
Upload TCP HTTP	162.260	paquetes
Solicitud promedio GET	12.000	bytes
Intervalo promedio entre GETs	19	segundos

bución *heavy-tailed*. Este mecanismo puede corresponder a una red de estaciones de trabajo, cada una de las cuales puede estar silenciosa o transfiriendo datos a una velocidad constante. Cabe aclarar que la expresión *heavy-tailed*, no tiene un equivalente en español de uso generalizado. La misma se refiere a la caída lenta de la curva de la función de distribución acumulativa complementaria CCDF (Complementary Cumulative Distribution Function). La distribución estadística *heavy-tailed* es, por antonomasia, la distribución Pareto. En [8], los autores recogieron evidencia mostrando que el tamaño de los archivos transferidos, y por lógica también el tiempo para transferirlos, tienen una distribución *heavy-tailed*, esto es el tiempo de SI. También evidenciaron que el tiempo de NO es *heavy-tailed* aunque menos “heavy”, esto es que la pendiente α de la CCDF sea mayor en valor absoluto. En [9] se establece la importante relación (dada por (2)) entre el coeficiente de Hurst, que cuantifica el fenómeno, y la pendiente α de la CCDF.

$$H = \frac{(3 - \min(\alpha_{SI}, \alpha_{NO}))}{2} \quad (2)$$

Tabla 3: Parámetros de la distribución Pareto.

Parámetro	Definición
Escala	$b > 0$
Forma	$1 < \alpha \leq 2$
Soporte	$x \in [b; +)$
PDF	$f(x) = \frac{\alpha b^\alpha}{x^{\alpha+1}}$
CDF	$F(x) = P[X \leq x] = 1 - \left(\frac{b}{x}\right)^\alpha$
CCDF	$\overline{F}(x) = P[X > x] = \left(\frac{b}{x}\right)^\alpha$
Valor medio	$E(x) = \frac{b\alpha}{\alpha-1}$
Varianza	∞

Entonces, el desarrollo de una aplicación cliente que solicite datos a una aplicación servidor, repetidamente, permitiría obtener un tráfico con una autosimilaridad especificada por su coeficiente de Hurst, bajo las siguientes condiciones:

- La cantidad de datos debe ser un valor sorteado de una distribución Pareto (tiempo de SI).

- Antes de realizar la próxima petición debe transcurrir un tiempo también tomado de una distribución Pareto (tiempo de NO).
- Los α correspondientes a ambas distribuciones Pareto deben satisfacer la ecuación de la CCDF de la tabla 3.

NS-3 - Creación del modelo

Como consecuencia de las consideraciones anteriores se optó por desarrollar una aplicación que simule una red con usuarios HTTP, recreando de esta forma las características reales del tráfico oportunamente capturado (muestra T111019-1-14-HTTP).

A tal efecto se desarrolló un módulo en C++, como una pieza más, integrante del sistema NS-3. El mismo está compuesto de una aplicación servidor que atiende en el puerto 80, que se corresponde con un objeto de la clase `HttpServerHelper`, y se instala en uno o varios nodos de la red. Además, el módulo está integrado por aplicaciones cliente, objetos de la clase `HttpClientHelper`, que se instalan en la cantidad de nodos peticionantes que sea necesaria. Tanto `HttpServerHelper` como `HttpClientServer` son clases personalizadas orientadas a cumplir con el objetivo principal de este trabajo, pero heredan los atributos y el comportamiento de la clase *Application* de NS-3 [11].

Cada cliente requiere a su servidor la transferencia, mediante un comando GET [12], de una cantidad de bytes sorteada de una distribución Pareto (tiempo de SI, definido por el atributo `ParetoTamanoPaquete`). Una vez enviado el paquete, el cliente deja transcurrir un tiempo, también sorteado de una distribución Pareto (tiempo de NO definido por el atributo `IntervaloPareto`), realizando nuevas iteraciones hasta agotar un máximo prefijado de requerimientos (atributo `MaxPackets`) o hasta que el servidor detenga su actividad, lo que ocurra primero. Muchos clientes instalados en sus respectivos nodos realizan estos pedidos concurrentemente, generando de esta forma tráfico con distribución autosimilar. Esta estructura permite simular muchos navegadores de Internet (aplicaciones cliente), peticionando a uno o más servidores (aplicaciones servidor) a través de un módem, donde la conexión xDSL al ISP es de una capacidad limitada. Esta topología recrea situaciones de redes locales con acceso a Internet, donde la demanda en horas pico podría saturar el enlace.

La aplicación cliente, al comenzar, inicia una sesión TCP con el servidor y requiere,

con el comando GET del protocolo HTTP, el envío de una determinada cantidad de bytes. Por ejemplo, un cliente que envía en un paquete la cadena `GET /18188.html HTTP/1.1` solicita al servidor el envío de 18.188 bytes, los que serán enviados por el mismo en el menor tiempo posible, conforme la red lo permita, dividiéndolos en la menor cantidad de paquetes según el MTU (Maximum Transfer Unit) de la red. El servidor responde al requerimiento anterior de la siguiente manera: `HTTP/1.1 200 OK INICIO DE DATOS XXXXXXXXXXXXX ...` siguen las Xs hasta completar los 18.188 bytes requeridos por el cliente ... `XXXXXXXX – FIN DE DATOS`.

El o los objetos de la clase `HttpServerHelper` se instalan en nodos diferentes y se configura en cada uno el tiempo de comienzo y finalización del servicio. Los objetos de la clase `HttpClientHelper` se instalan en otros nodos apuntando a la dirección IP del servidor, al cual realizarán las peticiones. Es posible definir tiempos de inicio y finalización y además tres atributos a cada objeto de la clase `HttpClientHelper`:

- `MaxPackets`, es la cantidad máxima de GETs que puede enviar un objeto `HttpClientHelper` por sesión.
- `ParetoTamanoPaquete`, es la cantidad promedio, de una distribución Pareto, de bytes solicitados con el comando GET.
- `IntervaloPareto`, es el tiempo promedio, de una distribución Pareto, entre el momento en el que el cliente termina de enviar una petición hasta que envía un nuevo GET.

Se focalizó la simulación sobre los elementos críticos del dimensionamiento de una red de usuarios que acceden a Internet a través de una conexión xDSL que oficia como cuello de botella. El tráfico download (HTTP) simulado debe satisfacer, en relación a la muestra T111019-1-14-HTTP, los siguientes requisitos:

- Cantidad de bytes.
- Cantidad de paquetes o frames.
- Coeficiente de Hurst.

En la figura 10, los nodos superiores representan a servidores distribuidos en Internet, a los cuales se accede por medio del ISP. Para conectarse al ISP se utiliza el módem

xDSL. Dentro de la red local se tiene un switch, el centro de la estrella, que conecta los distintos APs de la arquitectura. Vinculados con cada AP existen nodos donde está instalada la aplicación cliente. Algunas de estas estaciones son móviles y otras fijas, buscando simular tanto el comportamiento de usuarios que utilizan Internet a través de dispositivos móviles como de aquellos que se encuentran sentados en un lugar fijo utilizando un dispositivo con conectividad inalámbrica. Nótese la similitud entre la figura 10 y 2.

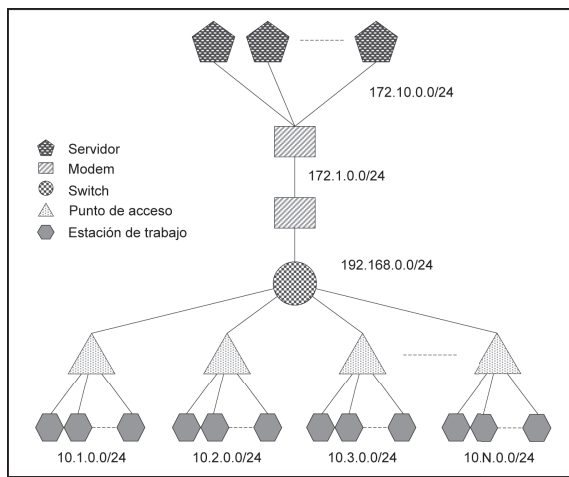


Fig. 10: Topología genérica de la red simulada con NS-3.

Ajuste del modelo

Una vez definido el modelo mostrado en la figura 10 se procedió a ajustarlo mediante un proceso iterativo hasta obtener resultados aceptables. En la tabla 4 se pueden ver los resultados de las simulaciones realizadas más significativas, donde las celdas rellenas indican que el valor ha cambiado con respecto a la simulación anterior.

En la figura 11 se muestran los LDs correspondientes a las distintas iteraciones de ajuste, comparadas con el LD correspondiente al tráfico HTTP de la muestra de control T111019-1-14. De la observación de la figura 11 y los resultados de la tabla 4 surge que, básicamente, la iteración 4 es la que más se asemeja a la captura T111019-1-14. Sin embargo, lo simulado difiere en cierta medida de lo capturado. Se entiende que la diferencia no es significativa como para invalidar los resultados.

Utilización del modelo para dimensionamiento de la red

Las simulaciones anteriores son un es-

fuerzo para que el simulador reproduzca una realidad registrada en la captura T111019-1-14-HTTP. En una red Ethernet cableada de 100Mbps, con conexiones WiFi 802.11g/n de throughput superior a 22Mbps, donde todas las estaciones acceden o intentan acceder a Internet, la conexión xDSL, de sólo 6Mbps es, sin duda, el cuello de botella que establece el nivel de satisfacción de los usuarios en las estaciones de trabajo, cableadas o inalámbricas. La siguiente serie de simulaciones intenta mostrar el efecto provocado sobre el tráfico, por un aumento primero y luego disminuciones del ancho de banda de la conexión a Internet. Los demás parámetros no se modificaron y corresponden a los de la iteración 4 de la tabla 4.

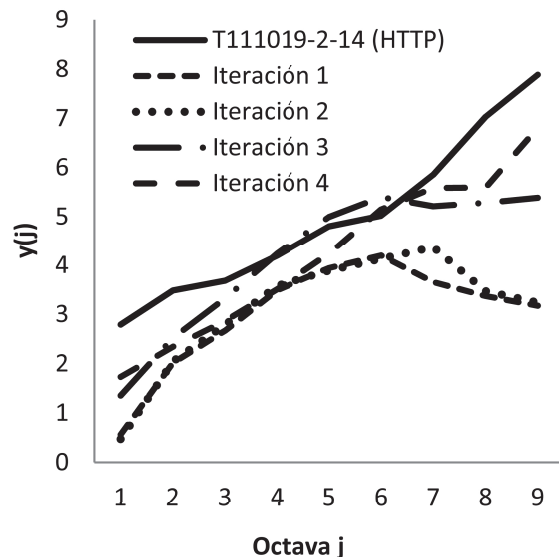


Fig. 11: Análisis de autosimilaridad para las distintas iteraciones de ajuste del modelo.

En la tabla 5 se resumen los resultados de las nuevas simulaciones, incluyendo la correspondiente a la iteración 4. Es importante destacar que hasta 1,75Mbps no hay una reducción apreciable en los bytes del download Internet y tampoco en la característica autosimilar de la captura. Para 1,5Mbps y 1Mbps ya se aprecia una notable reducción en la cantidad de bytes y la autosimilaridad disminuye. En la figura 12 se muestran los LDs correspondientes a la tabla 5.

Un dato importante es que la característica autosimilar del tráfico de Internet sólo subsiste si no existe regulación. En este caso tal regulación depende del cuello de botella de la conexión a Internet. Si se observa con detalle la figura 12 se puede apreciar que el tráfico comienza a ser aceptablemente auto-

Tabla 4: Resultados de simulaciones en NS-3, buscando un comportamiento estadístico parecido al de la captura T111019-1-14-HTTP.

Sim. N°	N° Est.	N° Serv.	Tam. GET [B]	Int. GET [s]	Pareto α	GETs/ Ráfaga	DL [MB]	Mb/s	kB/paq	Tiempo [s]
1	36	1	12.000	19	1,25	1	99	0,443	1,25	1800
2	36	1	12.000	19	1,5	1	98	0,436	1,29	1800
3	36	2	12.000	9,5/19	1,5	1	272	1,215	1,29	1800
4	36	2	12.000	9,5/19	1,5	2	409	1,825	1,27	1800
T111019-1-14-HTTP							356,5	1,581	1,35	1800

similar a partir de un ancho de banda superior a 1.75 Mbps, y el throughput promedio es cercano al de la muestra T111019-1-14-HTTP. Este resultado da un indicio de que conociendo la topología de la red a construir, la cantidad de usuarios y el perfil de consumo de éstos, es posible simular el escenario en cuestión y comenzar a aumentar el ancho de banda desde capacidades arbitrariamente pequeñas hasta encontrar un comportamiento aceptablemente autosimilar.

La tabla 5 y la figura 12 indican que un ancho de banda de 1,75Mbps es aceptable para el escenario de la red de UBP si se considera que la red se comportará en todo momento como lo hace durante la muestra T111019-1-14. Por otra parte, es posible definir una capacidad óptima de 3Mbps, ya que para anchos de banda superiores el comportamiento de los usuarios permanece constante. Esto es, con una capacidad de 3Mbps los usuarios se sienten libres al navegar por Internet.

Lo anterior no sorprende, ya que el throughput promedio en la muestra T111019-1-14-HTTP es de 1,581Mbps y, tal como se predijo, una capacidad de acceso a Internet de 1,75Mbps sería aceptable mientras que una de 3Mbps sería óptima.

Conclusiones y futuros trabajos

Las observaciones previas permiten proponer las siguientes reglas generales relativas a la simulación, con NS-3, de entornos reales con múltiples estaciones de trabajo y un único acceso a Internet, utilizando las clases `HttpServerHelper` y `HttpClientHelper`:

1. Construir una topología de red que se asemeje lo más posible a la red a desplegar.
2. Considerar la cantidad de estaciones simuladas igual a la cantidad real que se pretende instalar.

3. Ajustar todos los parámetros configurables según el perfil de usuario que corresponda al escenario en estudio.

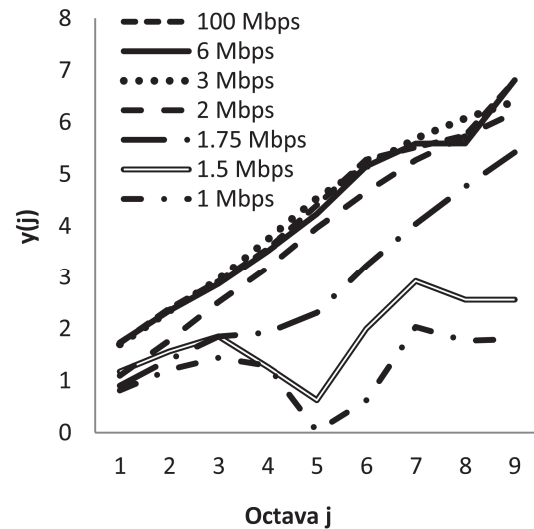


Fig. 12: Análisis de autosimilaridad sobre las variaciones de ancho de banda.

Tabla 5: Resultados de simulaciones en NS-3 variando la capacidad de la conexión a Internet.

Sim. N°	BW [Mbps]	DL [MB]	Mb/s	kB/paq
4	6	409	1,825	1,27
5	100	404	1,801	1,27
6	3	399	1,780	1,27
8	1.75	387	1,725	1,27
9	1.5	341	1,520	1,29
10	1	227	1,014	1,25
T111019-1-14-HTTP		356,5	1,581	1,35

Es posible plantear una serie de pasos prácticos para que los resultados de una simulación en NS-3, prevean con éxito el comportamiento en una red real con iguales relaciones cliente-servidor y topología:

- Construir en NS-3 la topología de red, incluyendo lo siguiente:
 - Crear todos los nodos necesarios considerando:
 - Nodos que representen a los servidores.
 - Nodo que represente al ISP.
 - Nodo que represente al módem con conexión a Internet.
 - Nodos que representen elementos de red tales como switches, routers y hubs.
 - Nodos que representen estaciones de trabajo cableadas.
 - Nodos que representen los APs para redes WiFi.
 - Nodos que representen estaciones WiFi.
 - Configurar los dispositivos de red:
 - Crear dispositivos de red: Placas de red, placas WiFi, etc.
 - Asociarlos a los canales correspondientes: Point To Point, CSMA, WiFi, etc.
 - Instalarlos en los nodos: estaciones, switches, routers, etc.
- Instalar la pila de protocolos TCP/IP en cada nodo.
- Asignar una dirección IP a cada dispositivo de red.
- Configurar uno o varios servicios de enrutamiento de paquetes, o tablas de enrutamiento, en los nodos.
- Instalar las aplicaciones en los nodos, en este caso:
 - Un objeto de la clase `HttpServerHelper` en cada servidor.
 - Objetos de la clase `HttpClientHelper` en cada estación de trabajo.
- Ajustar los parámetros de la topología y los atributos de las aplicaciones que generan tráfico conforme a las conclusiones arribadas anteriormente.

Habiendo realizado las configuraciones anteriores, se puede comenzar a simular, y luego analizar la autosimilaridad del tráfico, mediante el cálculo del coeficiente de Hurst a partir de la pendiente estimada con los LDs. Así, será posible determinar si la capacidad elegida para la iteración N es suficiente para servir la cantidad de tráfico que los usuarios de la red proponen.

Debido a que, por ejemplo, existen diferencias entre los perfiles de usuario en el campus de una universidad y los perfiles de usuario en un ambiente laboral, queda pendiente, como futuro trabajo, la definición de perfiles de usuario estándares y específicos (laborales, educacionales, de ocio, etc.) de manera de poder contar con ellos para simular y dimensionar cualquier escenario.

Agradecimientos

A los Ingenieros Héctor C. Riso y Víctor H. Frisón por colaborar en la captura del tráfico, al Ing. Roberto R. Rossi por su colaboración en la redacción, a la UBP y al MinCyT de la Provincia de Córdoba.

Referencias

- [1] <http://www.nsnam.org/overview/what-is-ns-3/>
- [2] <http://www.wireshark.org/about.html>
- [3] Mandelbrot B.B. *The Fractal Geometry of Nature*. W.H. Freeman & Co., 1982.
- [4] Leland W.E., et. al. *On the Self-similar Nature of Ethernet Traffic (Extended Version)*. IEEE/ACM Transactions On Networking, Vol. 2, No. 1, pp. 1-15, 1994.
- [5] Jeong, H.J. *Modelling of self-similar teletraffic for simulation*. PhD Thesis, University of Canterbury, New Zeland, 2002.
- [6] Uhlig S., Bonaventure O. y Ravier C. *3D-LD: a Graphical Wavelet-based Method for Analyzing Scaling Processes*. Proc. of the 15th Specialist Seminar, 2002.
- [7] Wisitpongphan N., Peha J. M. *Effect of TCP on Self-Similarity of Network Traffic*. Proceedings of 12th IEEE International Conference on Computer Communications and Networks (ICCCN), 2003.
- [8] Crovella M.E. y Bestavros A. *Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes*. IEEE/ACM Transactions On Networking, Vol. 5, No. 6, 1997.
- [9] Willinger W., et al. *Selfsimilarity through high-variability: Statistical analysis of Ethernet LAN traffic at the source level*. IEEE/ACM Transactions on Networking, Vol. 5, pp. 71-86, 1997.

[10] Rabinovich D. *Desarrollo de un modelo generador de tráfico autosimilar para la evaluación de los algoritmos de asignación de ancho de banda en WiMAX*. Tesis de Maestría en Redes de Datos, Facultad de Computación, Universidad Nacional de la Plata, 2011.

[11] <http://www.nsnam.org/documentation/>

[12] Fielding R., et. al. *Hypertext Transfer Protocol* -- *HTTP/1.1*. IETF RFC, 1999.

Datos de contacto

Daniel Rabinovich. Universidad Blas Pascal. Donato Álvarez 380, Córdoba, Argentina. CP: 5147. E-mail: rabinovichdaniel@gmail.com

César Osimani. Universidad Blas Pascal. Donato Álvarez 380, Córdoba, Argentina. CP: 5147. E-mail: cesarosimani@gmail.com

Gino Olocco. Universidad Blas Pascal. Donato Álvarez 380, Córdoba, Argentina. CP: 5147. E-mail: gino.olocco@gmail.com