

CONTROLADOR BASADO EN REDES NEURONALES HOPFIELD PARA UN CONMUTADOR ATM CROSSBAR

Ricardo Lent, Peter Yamakawa

Facultad de Ingeniería Eléctrica y Electrónica, Sección de Post-Grado
Universidad Nacional de Ingeniería

RESUMEN

Este artículo describe un modelo perfeccionado de la red neuronal Hopfield propuesta por Marrakchi y Troudet (1989) [1], adaptada para emplearla como controlador en un conmutador ATM crossbar con buffers FIFO en los puertos de entrada. Utilizando modelos estadísticamente auto-similares para estimar el tráfico entrante al sistema demostramos que el controlador propuesto incrementa notablemente el rendimiento general del conmutador.

ABSTRACT

This article describes an enhanced version of a Hopfield neural network originally proposed by Marrakchi and Troudet (1989) for large crossbar switches [1]. We have adapted this network to control the internal flow of cells in a modern ATM crossbar switch with FIFO buffers at the input ports. The proposed model, under Self-Similar input traffic, improves the general performance of the switch.

INTRODUCCION

La estructura crossbar para conmutadores de Modo de Transferencia Asíncrono (ATM) con buffers FIFO en los puertos de entrada es una arquitectura sencilla de implementar [2]. Consta de N puertos de entrada y N puertos de salida unidos por enlaces independientes, los que son elegidos lógicamente mediante elementos de conmutación que permiten el paso de células desde un puerto de entrada hasta uno de salida (Fig. 1). El estado de los elementos de conmutación es manejado por un controlador interno usando información previamente generada y contenida en una tabla de rutas, junto con la información que viaja en la cabecera de cada célula [3].

Una vez establecido un circuito interno para transportar una célula desde un puerto de entrada a uno de salida, el acceso a dicho puerto es bloqueado a los demás puertos de entrada. Dado que la probabilidad de que llegue más de una célula en diferentes puertos de entrada destinadas a un mismo puerto de salida es alta, se utilizan

buffers para retener las células hasta que puedan ser procesadas cuando se liberen los recursos internos del conmutador. Existe un fenómeno, conocido como *bloqueo por la primera célula*, asociado a esta arquitectura de conmutación, que provoca un alto degradamiento en el flujo interno de células para medianas y altas cargas de tráfico. El flujo aproximado máximo reportado anteriormente [4,5] es de 65%.

La razón de este problema radica en la política de uso de los buffers FIFO; cuando existe más de una célula en cola para transmitirse en un buffer de entrada, solamente la primera es elegible para su entrega interna. Las demás deben esperar en la cola hasta alcanzar la primera posición en el buffer. Hecho que ocurre a pesar de que existan puertos de salida libres a los que estén destinados. El impacto en el flujo interno de éste fenómeno puede apreciarse en la Fig. 2 para un conmutador crossbar simétrico de cuatro puertos.

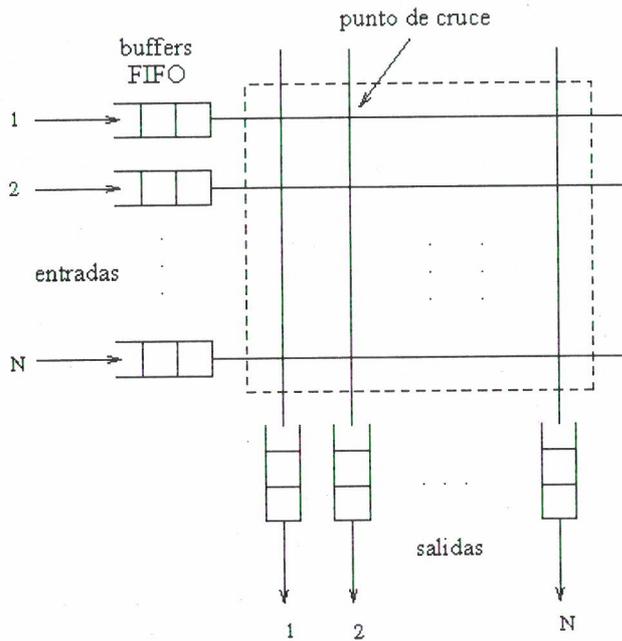


Fig. 1: Estructura crossbar de conmutación ATM

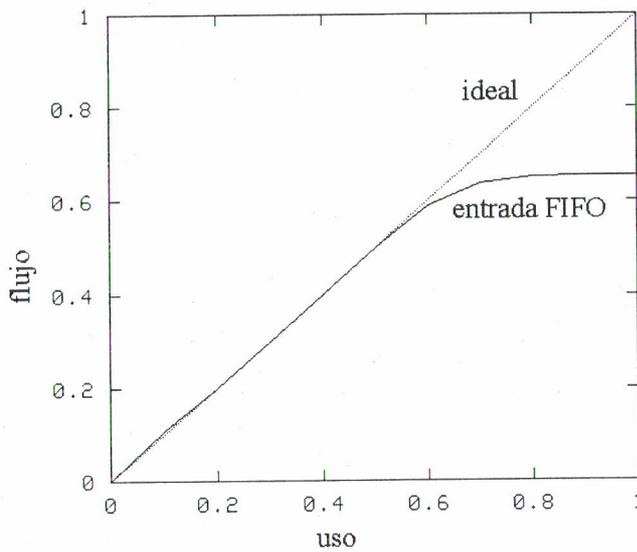


Fig. 2: Rendimiento del conmutador crossbar con buffers FIFO en los puertos de entrada

EI CONTROLADOR

Para solucionar el problema del *bloqueo por la primera célula*, podemos colocar en cada puerto de entrada al conmutador N buffers FIFO diferentes, correspondientes a los N puertos destino y construir un controlador que realice la

selección óptima de los buffers que pueden entregar sus células. Asumimos, sin perder generalidad, que el conmutador tiene igual número de puertos de entrada y de salida.

Definimos una matriz $N \times N$ cuyos elementos $R = r_{ij}$ representan el estado de cada buffer en todo momento; si existe al menos una célula esperando en un puerto de entrada i destinada a un puerto de salida j , el elemento correspondiente se activa $r_{ij}=1$, de otro modo se desactiva $r_{ij}=0$.

Así mismo, representamos el estado de los elementos de conmutación interna usando una matriz $C = c_{ij}$. Una configuración válida de ésta matriz es una matriz de permutación de la matriz identidad que representa los elementos de conmutación que se cierran ($c_{ij}=1$) o se abren ($c_{ij}=0$) permitiendo o no la entrega interna de células entre los puertos correspondientes.

La red neuronal Hopfield propuesta consta de $N \times N$ neuronas v_{ij} de valores continuos, implementadas usando amplificadores operaciones, cada una con voltaje de entrada u_i , voltaje de salida v_i y función de activación $f(u_i)$ (Fig. 3). Cada neurona controla el estado de un elemento de conmutación interno. Si una neurona en particular está activa (con su salida cercana a "1"), el punto de cruce correspondiente debe cerrarse permitiendo el paso de una célula. Por otro lado, si está desactivada (con su salida cercana a "0"), el punto correspondiente debe abrirse evitando el paso de células.

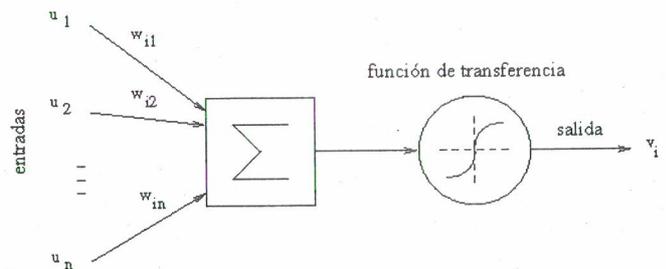


Fig. 3: Modelo de una neurona en la red Hopfield

Los pesos asociados a las neuronas son implementados usando conductancias w_{ij} que conectan la salida de la j -ésima neurona con la entrada de la i -ésima neurona (Fig. 4). La red

neuronal es simétrica en cuanto $w_{ij}=w_{ji}$, pero sin conexión entre la salida y entrada de una misma neurona $w_{ii}=0$.

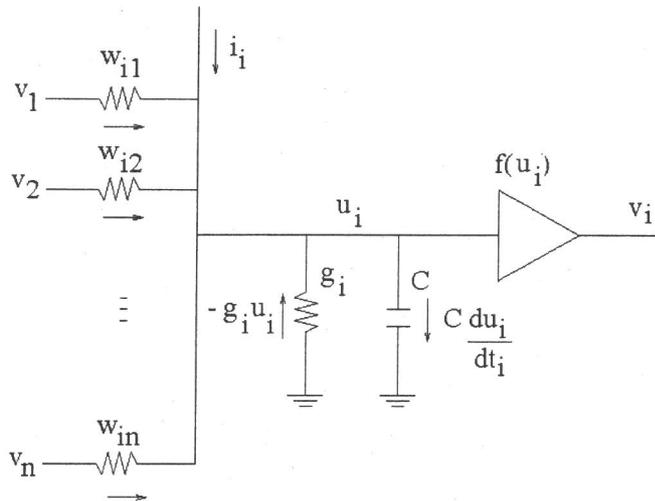


Fig. 4: Diagrama esquemático de una neurona en la red Hopfield

Además cada neurona presenta una conductancia g_i con tierra producto de las absorciones parásitas de corriente en el amplificador y una conductancia C responsable de los estados transitorios en el modelo.

En esta red, la función de energía tiene la siguiente expresión de Lyapunov [3]:

$$E(v) = -\frac{1}{2}v^T Wv - i^T \int_0^{v_i} f_i^{-1}(z) dz \quad (1)$$

donde f_i^{-1} es la inversa de la función de activación f_i .

EL MODELO

Las redes neuronales Hopfield son sistemas dinámicos que evolucionan en el tiempo en forma discreta o continua. Puede demostrarse que la energía computacional de la red decrece continuamente hasta alcanzar un punto de equilibrio o mínimo estable en su espacio de estado [6].

Aprovechando esta particularidad, establecemos una función objetivo (o función de penalización), la que se hace equivalente a la función Lyapunov de energía de la red neuronal Hopfield (1). Determinando una función objetivo adecuada para

nuestro problema lograríamos la optimización del flujo interno en el conmutador.

Marrakchi y Troudet [1] propusieron una función de energía compuesta por dos términos dirigidos a penalizar la presencia de más de una neurona activa por cada columna y fila del conmutador. Esta función se muestra a continuación:

$$E = E_1 + E_2 \quad (2)$$

$$E_1 = A \sum_x \sum_i \sum_{j,i \neq j} v_{xi} v_{xj} \quad (3)$$

$$E_2 = B \sum_i \sum_x \sum_{y,x \neq y} v_{xi} v_{yi} \quad (4)$$

El primer término alcanza un mínimo cuando al menos una neurona está activada en cada fila. La potencia de la conexión entre las neuronas en la misma fila es denotada por A . Similarmente, el segundo término alcanza un mínimo cuando al menos una neurona está activada en cada columna. La potencia de la conexión entre las neuronas, en este caso en la misma columna, es denotada por B . Luego la energía computacional de la red será mínima cuando al menos una neurona esté activa por cada fila y columna en la matriz C (Fig. 5).

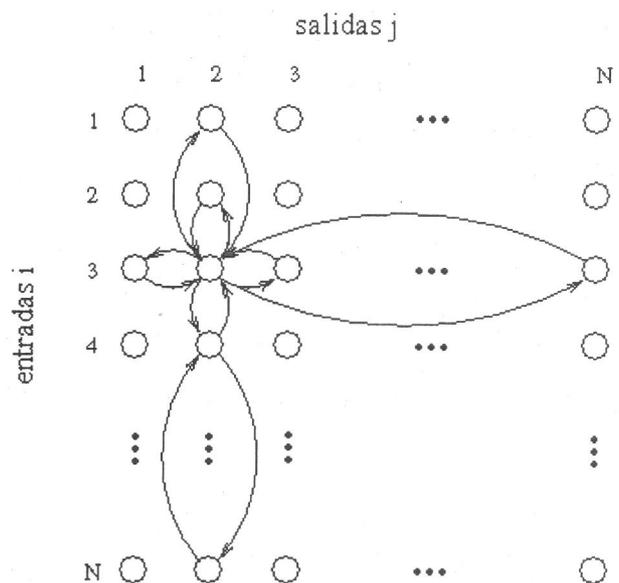


Fig. 5: Conexiones inhibitorias en la red neuronal Hopfield

Sin embargo, la presencia de filas o columnas con todas las neuronas desactivadas corresponde también a un mínimo aceptable en este modelo. Este estado no es conveniente dado que representa la no conmutación interna en el sistema.

Para permitir mayor flujo interno de células en la estructura de conmutación, introducimos un tercer término en la ecuación 2 para asegurarnos de que exactamente una neurona esté activada por cada fila y columna del conmutador (con lo que garantizamos que exactamente N neuronas estén activadas y siempre exista conmutación). Esto lo logramos restando el número total de neuronas activadas del número de puertos N del conmutador:

$$E = E_1 + E_2 + E_3 \quad (5)$$

$$E_3 = C \sum_x \sum_i (v_{xi} - N)^2 \quad (6)$$

Esta nueva función de energía tendrá un valor mínimo de cero para todas las matrices que contengan exactamente un "1" por columna y por fila. Cualquier otra combinación de elementos tendrán valores más altos de energía que serán penalizados.

La minimización de la ecuación 5 favorece la configuración adecuada de los elementos de conmutación para la entrega interna de células.

RENDIMIENTO DEL MODELO

Investigamos el rendimiento de nuestro modelo usando la técnica de simulación discreta de eventos [7,8]. Para la simulación de la red neuronal, los valores de los pesos se colocaron en $A=3$, $B=3$ y $C=2$. Estos valores experimentalmente aseguran una adecuada y rápida convergencia de la red neuronal a uno de sus estados estables.

Asumimos el tráfico de entrada a cada puerto del conmutador como un proceso estadísticamente autosimilar. Estos modelos estadísticos representan con gran precisión las características del flujo real de células ATM [9,10,11]. Los datos

del tráfico entrante fueron generados usando el método de la transformada rápida de Fourier [12]. En la Fig. 6 se muestra el rendimiento del flujo interno respecto al nivel del tráfico entrante en un conmutador ATM crossbar 4 x 4 conteniendo buffers FIFO con capacidad de retener hasta cuatro células por puerto de entrada. Utilizamos este tamaño de conmutador en particular para facilitar la simulación del sistema. Conmutadores de mayor tamaño deberían tener un comportamiento relativamente inferior dado que al existir mayor número de puertos, la competencia por los recursos internos es mayor.

Los resultados se comparan con el rendimiento de un conmutador ATM de similares características controlado por la red neuronal Hopfield propuesta. Se aprecia un sustancial incremento en el flujo de células en niveles medios y altos de tráfico con el uso del controlador interno. Esta mejora también se aprecia en una menor probabilidad de pérdida de células (Fig. 7).

Por otro lado, la clasificación de células en los puertos de entrada para permitir el uso del controlador neuronal, produce una mejora en el tiempo promedio de espera de las células en los buffers (Fig. 8).

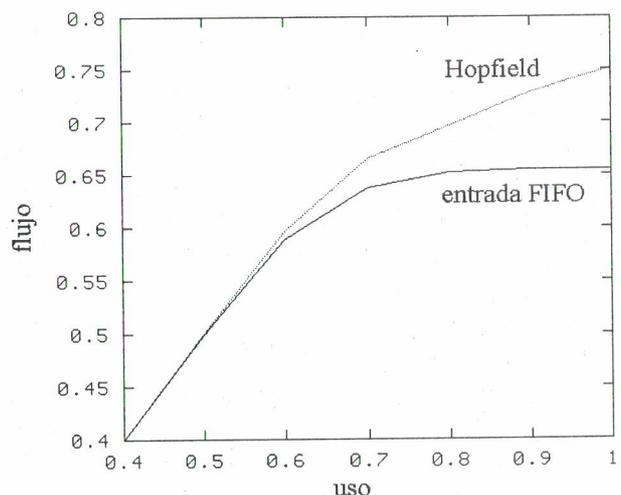


Fig. 6: Rendimiento del flujo en un conmutador ATM crossbar con 4 puertos de entrada y 4 puertos de salida.

Intuitivamente, puede pensarse que un incremento en la capacidad de los buffers en los puertos de

entrada de un conmutador puede favorecer la retención de un mayor número de células para su procesamiento posterior según la disponibilidad de los recursos internos del sistema.

Esta idea podría traducirse en una reducción en la probabilidad de pérdida de células y en un eventual incremento en el flujo interno del conmutador.

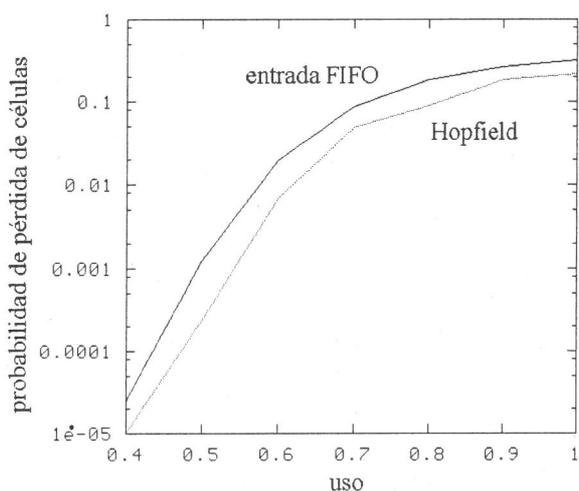


Fig. 7: Probabilidad de pérdida de células en un conmutador ATM crossbar 4x4

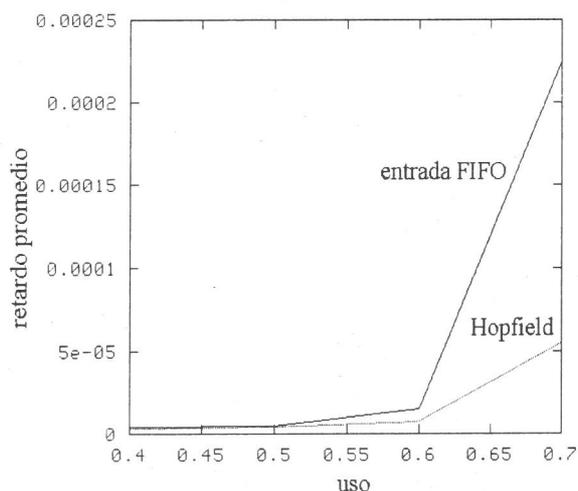


Fig. 8: Retardo promedio de las células en un conmutador ATM crossbar 4x4.

Puede apreciarse, en la Fig. 9 y Fig. 10, que el incremento en la capacidad de almacenamiento del conmutador con buffers FIFO únicos por

puerto de entrada no contribuye a elevar sustancialmente el rendimiento general del sistema.

El flujo en este caso particular, alcanza rápidamente un valor no superable de 65%. Además, el incremento en la capacidad de los buffers eleva dramáticamente el tiempo promedio de espera de las células (Fig. 11).

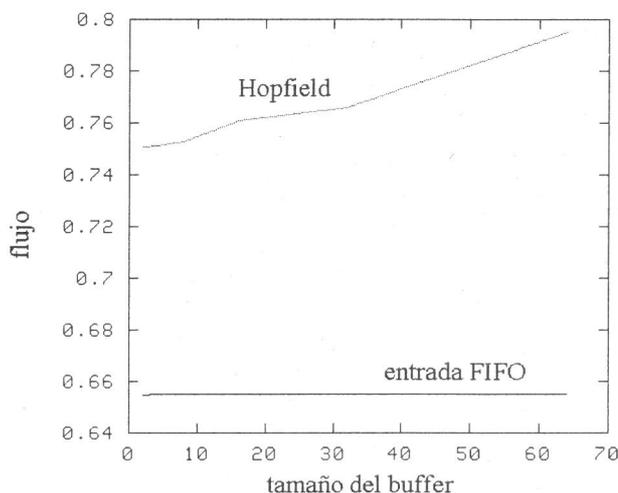


Fig. 9: Flujo interno en el conmutador ATM en función al tamaño del buffer.

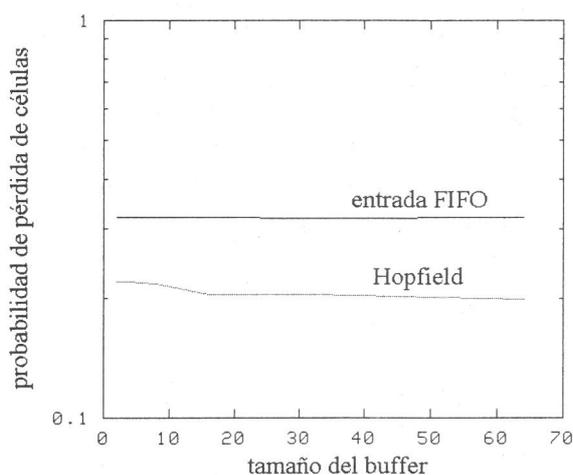


Fig. 10: Efecto en la probabilidad de pérdida de células al incrementar la capacidad de los buffers.

Con el uso del controlador basado en redes neuronales Hopfield obtenemos un incremento notorio en el rendimiento general del conmutador al elevar la capacidad de los buffers.

Hemos obtenido valores para el flujo interno en la simulación hasta de 84.8%; flujo 19.3% superior al modelo original.

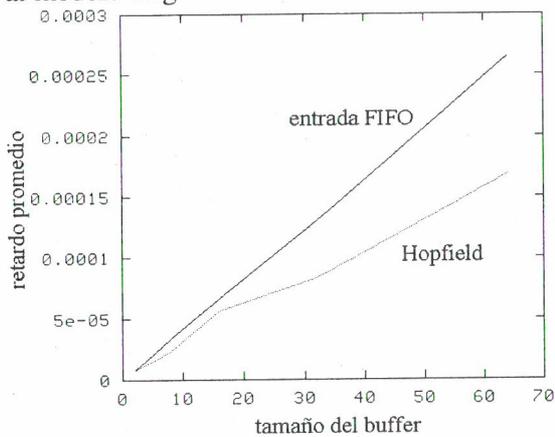


Fig. 11: Efecto en el retardo promedio de las células al incrementar la capacidad de los buffers.

CONCLUSIONES

En este estudio hemos presentado un modelo mejorado de la red neuronal Hopfield propuesta en [1], adaptado para emplearlo en un conmutador ATM crossbar con buffers FIFO en los puertos de entrada.

Nuestro modelo evita la inhibición de la conmutación interna para determinados estados de la red neuronal, caso no contemplado por el modelo original, lo que nos permite lograr un mayor flujo interno de células en el conmutador.

El modelo fue examinado en simulación usando procesos auto-similares para emular el tráfico ATM de entrada al conmutador, comprobándose que el controlador propuesto eleva efectivamente el flujo de transferencia interno de células respecto al modelo con buffers FIFO en los puertos de entrada hasta en un 22.8% bajo cargas severas de tráfico. El rendimiento del flujo para cargas bajas y medias de tráfico alcanza el 100% de eficiencia.

Además se comprobó el rendimiento superior del modelo propuesto respecto a la probabilidad de pérdida de células y al retardo promedio. En general las células sufren un retardo 35% menor que en el esquema con buffers FIFO únicos por puerto de entrada.

REFERENCIAS

1. **A. Marrakchi / T. Troudet**, A Neural Net Arbitrator for Large Crossbar Packet-Switches. IEEE Transactions on Circuits and Systems, p1039-1041, Jul 1989.
2. **Ellen E. Witte**, A Quantitative Comparison of Architectures for ATM Switching Systems. Oct 1991
3. **Raif O. Onvural**, Asynchronous Transfer Mode Networks: Performance Issues. Artech House 1994.
4. **Michael G. Hluchyj / Mark J. Karol**, Queuing in High-Performance Packet Switching. IEEE JSAC, p1587-1597, Dec 1998.
5. **Surya Kiran Pappu**, Performance Analysis of Input and Output Queuing Strategies of an ATM Switch Under Self-Similar Traffic. Clemson University M.Sc. thesis, Aug 1996.
6. **Jacek M. Zurada**, Introduction to Artificial Neural Systems. ISBN 0-314-93391-3.
7. **José Antão Beltrão / Jacques Philippe Sauve / William Ferreira Giozza**, Redes Locales de Computadoras. Protocolos de Alto Nivel y Evaluación de Prestaciones, ISBN 0-07-450140-2.
8. **Victor S. Frost / William Wood Larue Jr. / K. Sam Shanmugan**, Efficient Techniques for the Simulation of Computer Communications Networks, IEEE JSAC. Jan 1988.
9. **Will E. Leland**, On the Self-Similar Nature of Ethernet Traffic (Extended Version), IEEE/ACM Transactions on Networking. Feb 1994.
10. **Nicolas D. Georganas**, Self-Similar (Fractal) Traffic in ATM Networks.
11. **Ricardo Lent / Peter Yamakawa**, Naturaleza Fractal del tráfico Internet. Tecnia, Julio 1998. Vol8 Nro 1.
12. **Vern Paxson**, Fast Approximation of Self-Similar Network Traffic. Lawrence Berkeley Laboratory, University of California, Apr 1995.