0054300



Trab ITC 2012 P5.

REPUBLICA BOLIVARIANA DE VENEZUELA UNIVERSIDAD CATOLICA ANDRES BELLO ESCUELA DE TELECOMUNICACION

SISTEMA DE INFORMACIÓN ACADÉMICO Y CIENTÍFICO PARA EL ANÁLISIS Y LA SIMULACION DE TRAFICO CAÓTICO EN REDES DE PAQUETES DE NUEVA GENERACIÓN

Trabajo presentado por MSc/Esp. Franklin Planchart para optar al escalafón de profesor agregado.

Indice	
1 Objetivo.	2
2. Importancia de caracterizar el tráfico.	2
3 Caracterización y generación de tráfico autosimilar	16
4 Relación matemática de los métodos de cálculo de H con los algoritmos de	
cómputo	19
4.1Método Rango Rescalado R/S	19
4.2 Método Varianza/Tiempo	21
4.3 Método Transformada Wavelet	21
5 Software de Análisis y simulación de tráfico autosimilar	30
6 Estructura del software con interfaz gráfica.	45
7 Validación del software	68
7.1 Validación con vectores sintéticos FGNIFOURIER:	69
7.2 Validación con Vectores sintéticos FGNDWv2	70
7.3 Validación mediante cálculos en archivos provenientes de Bellcore Morristo	wn
Research and Engineering http://ita.ee.lbl.gov/html/contrib/BC.html	73
7.4 Validación mediante un vector acumulado	74
7.5 Prueba del archivo fgn8.dat, cargado directamente por Darryl Veitch	75
8. Conclusiones	77
9Recomendaciones	77
910 Programa de actividades	78
911 Bibliografía	70

MSc./Esp. Franklin Planchart

CANTV. Centro de estudios de Telecomunicaciones. Ministerio del Poder popular de Ciencia, Tecnología y Telecomunicaciones. República Bolivariana de Venezuela

RESUMEN.

El proceso de paquetización en las redes ha facilitado el proceso de integración de los servicios. En esta misma medida, las características del tráfico han tenido un importante giro a un comportamiento caótico. Este caos se refleja en la repetición de ráfagas de información en todas las escalas de tiempo, lo que encaja en la teoría de fractales y autosimilaridad en las redes. Protocolos como el IP requieren de la activación de mecanismos que adviertan al protocolo de usuario, TCP y SCTP, para que los mismos activen los mecanismos de recuperación y disminución de las ráfagas. Los fenómenos de sincronización global (estancamiento de todos los usuarios por congestión en la red) están muy relacionados a esta autosimilaridad del tráfico, dado que tradicionalmente los protocolos de Transporte están aislados de la red y la red está aislada del usuario a nivel de las capas de comunicación.

Por ello, es necesario determinar la situación de la red. Caracterizar el tráfico. Identificarlo. Dicha identificación está basada en la determinación del parámetro de Hurst, para lo cual existen diversos métodos. Los trabajos de tres científicos que desarrollaron módulos en Matlab, fueron analizados en este desarrollo. Ellos son: Reinaldo Scappini, Darryl Veitch y Hae-Duck Joshua Jeong. Dichos módulos calculan el parámetro H con diferentes métodos matemáticos en el dominio del tiempo y la frecuencia, al igual que generan tráfico autosimilar en la misma forma.

El software desarrollado en este trabajo está basado en interfaz gráfica, toma dichos módulos y los une en una sola aplicación, mediante un riguroso análisis de integración de las bases de datos, de los gráficos y adaptación de diversos módulos para facilitar la automatización.

Un componente del software está orientado al sector educativo de Ingeniería de Telecomunicaciones o Computación del último semestre o año y otra parte está orientada a la investigación de campo de Científicos e Ingenieros de la Red.

SUMMARY.

The process of packetized networks has facilitated the process of integration of services. To this extent, the characteristics of traffic have had a major twist to a chaotic behavior. This chaos is reflected in repetition of burst of information at all scales of time, which fits in the theory of fractals and self similarities in networks. Protocols such as IP require activation mechanisms that alert to the user, using TCP or SCTP protocol for example, that activate the mechanisms of recovery and decrease of the burst. The phenomena of global synchronization, that brings as a consequence the stagnation of all users by network congestion, are closely related to this selfsimilarities of the traffic, given that transport protocols are isolated from the network.

It is therefore necessary to determine the status of the network. Characterize and identify it. This identification is based on the determination of the Hurst parameters, for which there are several methods. The work of three scientists who developed modules in Matlab were analyzed in this development. They are Reinaldo Scappini, Darryl Veitch y Hae-Duck Joshua Jeong. These modules calculate the H parameter with different mathematical methods in the domain of time and frequency, similar to stochastically self similar traffic generated in the same way.

The software developed in this work is based on graphical interface. It takes these modules and binds together them in a single application, through a rigorous analysis of integration of the databases, graphics and adaptation of many modules to facilitate automation.

A component of the software is geared to the education sector of telecommunications engineering or computing in the last period and another part is oriented to research field of scientist and engineers from the network.

1.- Objetivo.

El presente tiene como finalidad entregar un avance para la validación del software mencionado en el asunto, desarrollado por el MSc. en Ingeniería Eléctrica Franklin Planchart, como herramienta para ser utilizada en la investigación del estudio del Doctorado individualizado en la UCV y ser implementada a nivel académico en las universidades cuyo sistema curricular tenga incluido el tópico de Ingeniería de Tráfico en la República Bolivariana de Venezuela.

2. Importancia de caracterizar el tráfico.

Este estudio partió del análisis del caos. Este último se define como la repetición de una irregularidad en diferentes escalas de tiempo. Esa repetición ha conllevado a los fractales. Se dice entonces que ese suceso de la irregularidad a diferentes escalas representa la autosimilaridad estocástica. Un parámetro que determina o explica dicho comportamiento es el parámetro de Hurst, dado que el mismo afecta directamente el cálculo del 'Tiempo' de ocupación del servidor cuando se atiende un paquete y a los sistemas de almacenamiento para los paquetes que llegan a los nodos, Routers o Switches.

Sin duda alguna los modelos de tráfico tradicionales basados en Poisson bajo el esquema actual de paquetización de los servicios, no se adaptan ni soportan lo que en realidad ha estado ocurriendo en las redes debido al proceso de integración de todos los servicios, dado que la tendencia es el caos en las redes de paquetes. La tendencia es la autosimilaridad estocástica.

En el proceso Poisson, los eventos son totalmente independientes, por lo que se concluye que son procesos sin memoria. La siguiente es la función de probabilidad de ocurrencia de un evento "x", en este caso, la llegada de una determinada cantidad de paquetes y λ es el promedio de paquetes por segundo.

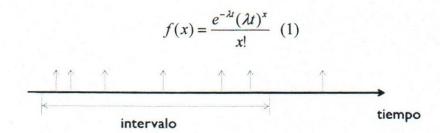


Figura 1. Probabilidad de cantidad de paquetes entre tiempos de llegadas

Igual importancia tiene el expresar los tiempos entre llegadas de los paquetes. En un proceso Poisson, la probabilidad de tiempos entre llegadas de paquetes viene dada por:

$$f(x) = \lambda e^{-\lambda x} (2)$$
$$0 \le x \le \infty$$

Donde x es el tiempo de ocurrencia y f(x) la probabilidad de que ese tiempo ocurra.

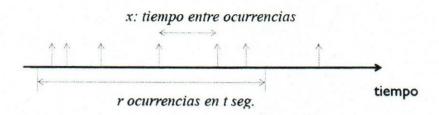


Figura 2. Probabilidad de tiempos de ocurrencia entre llegadas de los paquetes

De esta forma, el trabajo aquí desarrollado, orienta todos los algoritmos a realizar los cálculos en esos dos tópicos. Se hace la excepción de que los paquetes fueron expresados en bytes. Entonces los cálculos se refieren a cantidad de bytes y tiempo entre llegadas de los paquetes.

Internalicemos la importancia de esta aseveración. Si llamamos lambda_i al promedio de paquetes entrante y lambda_o al promedio de paquetes salientes, se puede detallar el fenómeno en la tabla 1, donde se observa que a pesar de tener el mismo promedio de atención a los paquetes, que es la salida, los requerimientos de capacidad de memoria para los paquetes se incrementan en forma alarmante, debido a que son impactados en la medida que el parámetro Hurst crece o tiende a la unidad, lo que es un indicador del tráfico caótico.

La nomenclatura utilizada en la primera fila es la anotación de Kendall. En este caso MM1 es un sistema con régimen de entrada exponencial y atención de los paquetes exponencial con un solo servidor, o en el caso de telecomunicaciones, se habla de un solo troncal o línea de conexión. La anotación EHM1, se refiere a un sistema con régimen de entrada de paquetes caótico y despacho o atención de esos paquetes en régimen exponencial con un solo servidor. Las áreas encerradas son un ejemplo de visualización. En este caso, manteniendo un promedio de 500 paquetes por segundo de entrada, siendo atendidos a 510 paquetes por segundo, en tráfico MM1, y un tamaño promedio de 65.535 bytes por paquete, que es la máxima longitud de un paquete en IPv4, hay una necesidad de memorias de almacén de 50 paquetes máximo, mientras que el extremo de un tráfico caótico o autosimilar la necesidad de almacenamiento se proyectó a un máximo de 2.573 paquetes máximo. Es obvio que si la red se dimensiona bajo un modelo Poisson, la red colapsará o degradará el servicio, manifestándose en abruptas caídas de la calidad de servicio por instrucciones del protocolo TCP a nivel del usuario, o en algunos casos, los protocolos de red, como Frame Relay, ATM y MPLS aplicarán los respectivos mecanismos de control de congestión implícita o explícita según sea el caso. Es decir, el problema se traslada desde la red a las aplicaciones. El protocolo UDP ni siquiera activa algún mecanismo de protección ante la congestión sino que simplemente, entrega lo que llega, y es la aplicación la que sufrirá las degradaciones. Aunque claro está, el UDP es utilizado para mensajes cortos y no tanto para servicios, ejemplo, consultas a un DNS (Servidor de Nombres de Dominio), y para transmitir servicios isocrónicos-síncronos como la voz y el video en tiempo real.

Tabla 1. Tamaño del buffer acorde al valor de H. Fuente: [12]

A STREET, NAME OF THE PARTY OF					
63336	MANAGEMENT OF THE PARTY OF THE	The second secon	The second section of the second section is	AND THE PARTY OF T	
					THE RESERVE OF THE PERSON NAMED IN COLUMN TWO IS NOT THE PERSON NAMED IN COLUMN TWO IS NOT THE PERSON NAMED IN
		The same of the sa			EHM1
	THE RESERVE OF THE PERSON NAMED IN COLUMN 1				H=0.8691
50,1492	64,3928	61,6123	103,3284	287,4396	2572,9000
17.	10,0000	00,0040	111,0104	139,3901	1692,6000
				54,1879	178,9612
			15,2693	31,0827	209,3018
		7,4996	12,9169	24,8921	37,2316
	5,2018	6,1899	11,1019	16,7464	24,7449
	4,4683	4.8879	6,3597	11,2600	31.7442
	3,6374	4,3407	6,9113	7.9215	28,2351
2,9686	3,3388	3,7879	4.7479		98.1219
2,5843	2,9751	3,3259	4,7976		9,5666
2,3947	2,6473	2,8920	3.4678		128,2639
2,2042	2,4035	2.6670	3.0876		6,5934
2,0211	2,2214	2.4052			9.1586
1,8373	2.0660	2.2328			7.1829
1,7148	1.8626		The state of the s		6.6455
1,5775	1.7680				6.2956
1,4985	1.6959				7.5994
1,4276	1,6015				4,0779
1,3538					3,4356
1,2865					5,0995
1.2203		THE RESERVE THE PERSON NAMED IN COLUMN TWO IS NOT THE PERSON NAMED IN COLUMN TWO IS NAMED			2.5858
-					The state of the s
				THE RESERVE THE PERSON NAMED IN COLUMN TWO IS NOT THE OWNER.	2,8688 2,7206
		THE RESERVE THE PERSON NAMED IN COLUMN 2 IS NOT THE OWNER.	1,4000	2,1073	2,7206
1,0668	1,1768	1.2255	1,3705	1,7565	2,2576
	11,1740 7,0964 5,4754 4,6788 3,8262 3,3542 2,9686 2,5843 2,3947 2,2042 2,0211 1,8373 1,7148 1,5775 1,4985 1,4276 1,3538 1,2865 1,2203 1,1662 1,1144	MM1 EHM1 Q medio simt H=0.5695 50,1492 64,3928 17,006 11.1740 10,7365 7,0964 8,3229 5,4754 6,0817 4,6788 5,2018 3,8262 4,4683 3,3542 3,6374 2,9686 3,3388 2,5843 2,9751 2,3947 2,6473 2,2042 2,4035 2,0211 2,2214 1,8373 2,0660 1,7148 1,8626 1,5775 1,7680 1,4985 1,6959 1,4276 1,6015 1,3538 1,5126 1,2865 1,3955 1,2203 1,3378 1,1662 1,3028 1,1144 1,2285	65536 MM1 EHM1 EHM1 Q medio simt H=0.5695 H=0.6129 50,1492 64,3928 61,6123 11,1740 10,7365 12,8026 7,0964 8,3229 12,1713 5,4754 6,0817 7,4996 4,6788 5,2018 6,1899 3,8262 4,4683 4,8879 3,3542 3,6374 4,3407 2,9686 3,3388 3,7879 2,5843 2,9751 3,3259 2,3947 2,6473 2,8920 2,2042 2,4035 2,6670 2,0211 2,2214 2,4052 1,8373 2,0660 2,2328 1,7148 1,8626 2,0687 1,5775 1,7680 1,8954 1,4985 1,6959 1,7859 1,4276 1,6015 1,6912 1,3538 1,5126 1,5871 1,2865 1,3955 1,4714 <td>MM1 EHM1 EHM1<!--</td--><td>MM1 EHM1 EHM1<!--</td--></td></td>	MM1 EHM1 EHM1 </td <td>MM1 EHM1 EHM1<!--</td--></td>	MM1 EHM1 EHM1 </td

Observe que igualmente a pesar de casi duplicar la velocidad de despacho, ejemplo 990 paquetes por segundo, en régimen MM1, la necesidad de posiciones para almacenar el paquete es 1, y en el extremo caótico, se puede decir que hay una necesidad para almacenar 3 paquetes.

En la misma forma se pueden realizar comparaciones en sistemas en los cuales se realizaron o implantaron esquemas para poder tratar estas situaciones mediante el manejo dinámico de memorias. Es decir, la longitud de las memorias de almacenamiento cambia y se adaptan, aprovechando la característica de autosimilaridad o de semejanza estadística en las diferentes escalas. Ello quiere decir, que se puede predecir lo que va a ocurrir, y en función de ello, adaptar la memoria, de tal forma que no sea muy grande en necesidades pequeñas, y que sea grande cuando venga la carga pesada. Refiérase a las figuras 3 y 4.

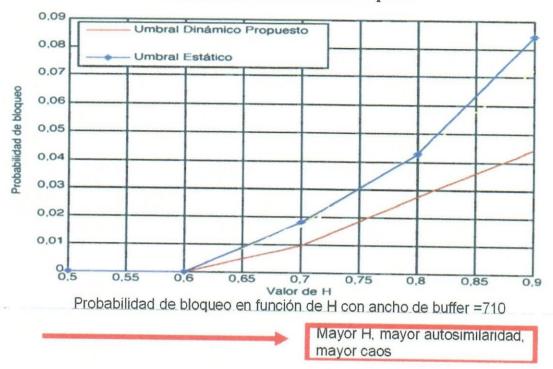


Figura 3. Probabilidad de Bloqueo acorde al valor H y al tipo de umbral implementado. Fuente: [13]

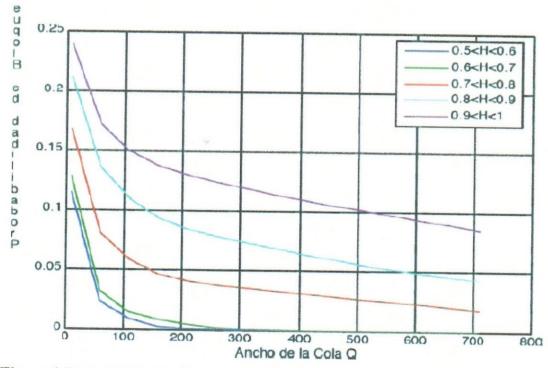


Figura 4. Probabilidad de Bloqueo acorde al valor del parámetro H. Fuente [13]

En este caso, la probabilidad de bloqueo disminuye cuando se introduce un esquema de umbral dinámico en el tamaño de las memorias de buffer para el almacén de los paquetes

que van llegando, y la cola que se origina, acorde al régimen de despacho y el ritmo del mismo.

Las fórmulas de Little para sistemas M/M/1 y M/D/1 en lo que respecta a los elementos en espera en los buffer más el elemento en el servidor son respectivamente:

$$r = \frac{\rho}{(1 - \rho)}$$
 (3)

$$r = \frac{\rho^2}{2(1-\rho)} + \rho$$
 (4)

Donde ρ es el tiempo de atención o despacho del paquete por parte del servidor o troncal. En un sistema con tráfico autosimilar con dependencia larga en la correlación temporal, el tamaño de la cola viene dado por:

$$r = \frac{\rho^{\frac{1}{2(1-H)}}}{(1-\rho)^{\frac{H}{(1-H)}}}$$
 (5)

Siendo H el parámetro de Hurst. En la misma forma, en los esquemas tipo M/M/n, se pueden aplicar las fórmulas de Little para calcular el porcentaje de utilización de los servidores y la capacidad de las memorias de entrada.

Tabla 2. Muestras de cálculos utilizando las fórmulas 3,4 y 5

P	ERES1	MOI	EHM1	EHM1(log)	R(EHM1/MM1)	Valor de H	9	MAN	MD1	EHMI	EHM1(log)	R(EHM1/MM1)	Valor de
0,60	1.50	1.05	1.508+00	1,768-01	1,00E+00		0.60	1,50	1,05	1.096-01	1,045+00	7.26E+00	
0.70	2.33	1,52	2.336-00	3,688-01	1,00€+00		0.70	2,33	1,52	5.085+01	1.70E+00	2 17E+01	1
0.80	4.00	2.40	4,00E+00	6,02E-01	1,00E-00	_	0.80	4.00	2.40	3.58€ +02	2.555-00	8 94E -01	
0.90	9,00	4.95	9.005-00	9,548-01	1.00£-00	1	0.90	9.00	4,95	7.688+03	3.896-00	8 548 +02	1
0.95	19,00	9,97	1.906-01	1,285-00	1,00E+00	0,50	0.95	19.00	9.97	1,415-05	5.15E-00	7.416-03	0.80
0.97	32,33	16,65	3,238-01	1.51E+00	1.00E+00	1	0.97	32 33	16.65	1.148-06	6.068+00	3548-04	
0.96	49.00	24,99	4,90E-01	1.69E-00	1.00€+00	_	0.98	49.00	24.99	5.948-06	6.775-00	1,215-01	1
0,99	99.00	50.00	9.90E+01	2.00E+00	1.00E+00	_	0.99	99.00	50.00	9.758-07	7 996 -00	9.858 -05	1
1,00	999.00	500,00	9,99€+02	3,00E+00	1,00E+00	_	1,00	999.00	500,00	9.986-11	1.205-01	9.996 -08	
0,60	1.50	1.05	2.09€+00	3.20E-01	1.396+00		0.60	1.50	1.05	2.97E+02	2.47E-00	1 98E -02	1
0,70	2.33	1,52	3,90€+00	5,91E-01	1,67E+00] -	0.70	2.33	1.52	8 545-03	3 935 -00	3 666 -03	1
0.80	4,00	2.40	8.468+00	9.27E-01	2.116+00] -	0.80	4.00	2.40	6 405 -05	5.81E-00	1 605 -05	1
0,90	9.00	4,95	2.778+01	1,448-00	3,088+00	-00 0 60 -00 -00	0.90	9.00	4.95	5.90E+08	8.77E+00	6.568 +07	0.90
0.95	19,00	9,97	8,396+01	1,925+00	4,42E+00		0.95	19.00	9.97	3.965-11	1,165-01	2.096+10	
0,97	32.33	16,65	1,856-02	2.275-00	5.738+00		0.97	32.33	16.65	4 36E+13	1.365 -01	1.356 +12	1
89,0	49.00	24.99	3.458-02	2.54E-00	7.04E+00		0.98	49.00	24.99	1.771-15	1.525-01	3.606 +13	1
0,99	99.00	50,00	9,888+02	2.99€+00	9.97£+00		0.99	99.00	50.00	9.51E-17	1.806 -01	9.615-15	
1.00	999.00	500,00	3.168+04	4.50E+00	3.166+01		1.00	999,00	500,00	9,958-26	2,706-01	9.966 -23	
0.60	1,50	1,05	3,625+00	5.59E-01	2.41E+00		0.60	1.50	1.05	2.01E+28	2.835-01	1.346+28	+
0,70	2.33	1.52	9 168+00	9,62E-01	3.936+00	1 -	0.70	233	1.52	1.055-44	4 40E-01	4.495-43	
0.80	4,00	2.40	2.958+01	1.47E-00	7.37E+00		0.80	4.00	2.40	2 258+64	6.44E-01	5.638-63	1
0,90	9.00	4,95	1.818+02	2,26E-00	2,016+01	1 -	0.90	9.00	4.95	5.152.96	9.67E-01	5.738+95	
0.95	19.00	9.97	9,976+02	3.00€+00	5.25€+01	0.76	0.95	19.00	9.97	4.888 - 127	1.285-02	2 575-126	0.99
0.97	32.33	16.65	3,406+03	3.53€+00	1,05€+02		0.97	32.33	16.65	1.27E-150	1506-02	3.936-143	
0,98	49,00	24,99	8,918-03	3.95€+00	1,826-02	1 -	0.96	49.00	24.99	5.75E+167	1,685-02	1.175+166	
0.99	99.00	50.00	4.56E-04	4.668+00	4.61E+02	1 -	0.99	99.00	50.00	6.05E+197	1.985 +02	6 115-195	-
1.00	999.00	500.00	9.985+06	7,00E+00	9.996-03	_	1.00	999.00	500.00	9.51E+296	2.97E-02	9.525+293	-

De la tabla 2 se concluye que el porcentaje de utilización es la relación entre el flujo entrante y el saliente al nodo. Una aproximación a la unidad indica que el sistema tiende a la inestabilidad. Por supuesto, en régimen entrante de Poisson, las necesidades de buffer se disparan. Pero debe observarse que en un régimen caótico, la necesidad de buffer es

inmensamente superior. En este orden de ideas es necesario tener una claridad en lo que es la salida de un troncal. La velocidad de salida (velocidad del reloj de sincronismo) no necesariamente tiene una relación directa con la cantidad de paquetes o tramas colocadas en el troncal. En una red ATM, se puede decir que la cantidad de celdas transmitidas se obtiene al dividir la velocidad en transmisión entre 53 bytes, que es el tamaño de la celda, y es bien sabido que las celdas van unidas y sincronizadas con la cabecera HEC. En un sistema que procesa tramas y paquetes no necesariamente es así, aunque se hace un esfuerzo para que las tramas vayan unas unidas a las otras, sincronizadas con una palabra de 6 bytes y un delimitador, en el caso de Ethernet. En el caso de IP, que es nivel 3, la sincronización se obtiene por información del tamaño del paquete. En conclusión, es necesario detallar la cantidad promedio de paquetes o tramas en el medio de transmisión. Al hacerlo, es posible que el porcentaje de utilización de los procesadores sea mayor al estimado. Lo otro es que en servicios como la voz, se trata que el paquete sea pequeño, dado que se utiliza norma de codificación G.723 o G.729. En la misma forma se utilizan técnicas para comprimir la cabecera del IP.

La otra situación es que el tamaño de los paquetes promedio en el caso de TCP, hasta la década de los 90, mantenía un promedio exponencial. Por tanto, los procesadores fueron diseñados para esos promedios. Hoy existe una posible tendencia a que no se cumpla dicha regla, por lo que el manejo del sistema de colas, en lo que respecta a servicios elásticos e inelásticos pueda estar afectando los nodos en una forma o que exista una tendencia a ello.

Por tanto, las cifras de porcentaje de utilización de los procesadores de salida a nivel binario o de velocidad en el reloj del nodo, o porcentaje de utilización de los troncales pudiesen no reflejar lo que en realidad está sucediendo en la red. Sobretodo si estamos en un ambiente que tiende al caos o a la autosimilaridad estocástica. Un aumento de buffers, tampoco es una condición de solución a los conflictos en la red, dado que al aumentar los buffers en forma estática, se pudiesen incrementar en forma descontrolada los retardos en la red, incrementando peligrosamente los efectos de jitter y wander en la transmisión de los paquetes al igual que los retardos que activan los mecanismos de inicio lento con crecimiento de ventana exponencial en protocolos como el TCP y SCTP.

En las figuras 5 y 6 se observa el impacto en los buffers, acorde al porcentaje de utilización del servidor que está conectado al troncal. La primera es un nodo de conmutación diseñado en ambiente exponencial tipo Poisson. El segundo es el caso de una central telefónica clásica, basada en ocupación de ranuras de tiempo. A nivel de poder comparar, la ocupación de buffer representa el logaritmo en base 10.

La figura 6 es un nodo en ambiente autosimilar con los buffers calculados, representados en forma de logaritmo en base 10. Observe la comparación con la superficie plana, la cual representa el tráfico tipo M/M/1 y M/D/1. El plano que se levanta representa el tráfico EH/M/1. Estas gráficas son la representación de la tabla 2.

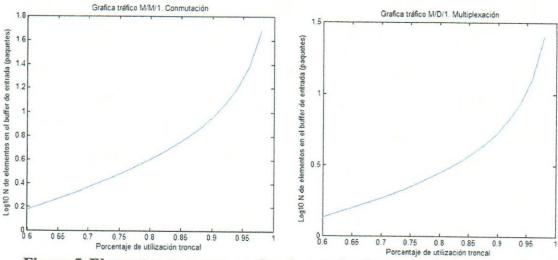


Figura 5. Elementos en espera en el nodo acorde al porcentaje de utilización del procesador

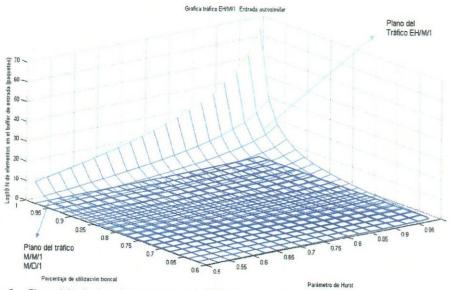


Figura 6. Cantidad de elementos residentes en el nodo acorde al porcentaje de utilización en el procesador y el parámetro de Hurst.

Las figuras 7 y 8 expresan la misma información en diagramas de barras. En la figura 8 se esboza por cada por cada muestreo en porcentaje de utilización 20 muestras de parámetros H, creciendo desde 0.5 hasta 0.98.

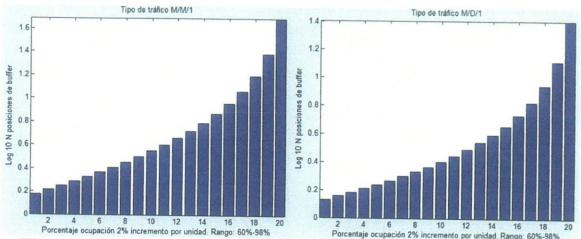


Figura 7. Representación de un nodo de conmutación de paquetes y una central telefónica con entrada exponencial.

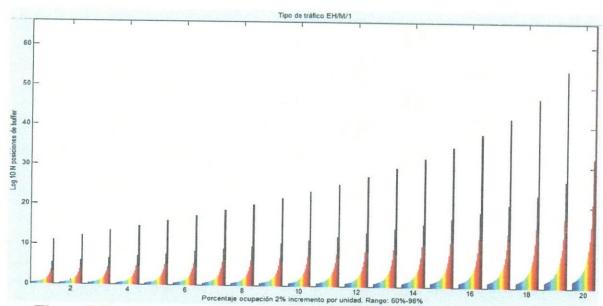


Figura 8. Representación de un nodo de conmutación de paquetes con entrada autosimilar y atención exponencial. Cantidad de 20 muestras con parámetro de Hurst entre 0.5 y 0.98 por cada muestra de porcentaje de utilización.

Adicionalmente, en los sistemas clásicos, el concepto de linealidad es aplicable y utilizando superposisición, la resolución en sistemas de redes se logra mediante sumatorias de los flujos en los nodos. Esta es una gran ventaja dada la sistematización de ecuaciones, cálculos y software existentes. Por los momentos, en los sistemas de tráfico caótico la superposición no se puede aplicar.

En este sentido, para analizar los impactos a nivel de laboratorios, se puede simular tráfico caótico con larga dependencia o fuerte correlación temporal tomando fuentes de Poisson (figura 9). Apagando y encendiendo las fuentes de Poisson aleatoriamente mediante distribución de Pareto (figura 10), se obtiene un tráfico autosimilar. Entonces, para estos casos, se pudiese llegar a una estrategia futura de aplicar las ecuaciones tradicionales

realizando superposición de las fuentes a instantes diferentes. Por supuesto, el cómputo es muy elevado. Es temprano para decir entonces que los esquemas de Poisson no tendrán un impacto en el futuro en el análisis de tráfico autosimilar. Un posible proyecto es realizar generación mediante fuentes de Poisson, y utilizando múltiples computadores para los cálculos de buffer y porcentaje de utilización de los servidores.

Así, siendo la función densidad y la función de distribución de Pareto como se expresa a continuación:

$$f(x) = \alpha b^{\alpha} x^{-(\alpha+1)}$$

$$F(x) = \Pr[X \le x] = 1 - \left(\frac{b}{x}\right)^{\alpha} \quad (6)$$

Con a>0

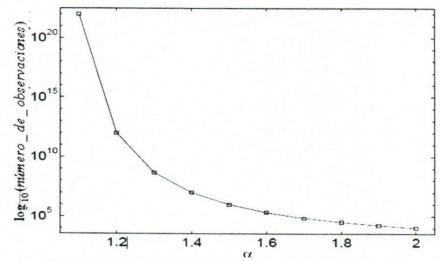


Figura 9. Curva de función de distribución de Pareto

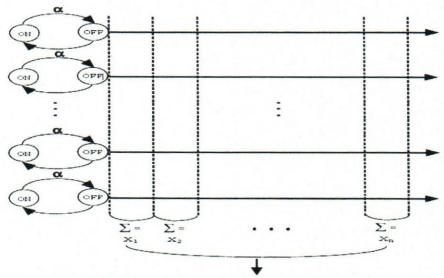


Figura 10. Generación de Tráfico Autosimilar a través de fuentes de Poisson y tiempos de encendido-apagado con distribución Pareto. Fuente: [10]

En el diagrama de la figura 9, el valor de α viene dado por:

$$H = \frac{3 - \alpha}{2} \ (7)$$

Por lo que $1 < \alpha < 2$, para que el tráfico sea autosimilar.

De esta manera, siendo las fuentes de Poisson, implica que mediante superposición se pudiesen aplicar las ecuaciones de la teoría clásica de tráfico, y las ecuaciones de análisis matricial de Jackson. La figura 11 es un ejemplo de aplicación del teorema de Jackson en redes con tráfico caracterizado como clásico basado en Poisson.

Pobl. 1 Pobl. 1 Pobl. 2 Se dispone de servidores con tasa individual de servicio
$$\mu = 12$$
. Determinar en cada nodo el número mínimo de servidores de forma que la red de S.E. presente estado estacionario. Calcular las demoras medias en todos los S.E. de la red.
$$\begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} = \begin{bmatrix} 10 \\ 5 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 1/2 & 0 & 0 \\ 1/2 & 1 & 2/3 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & 0 \\ -1/2 & 1 & 0 \\ -1/2 & -1 & 1/3 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} = \begin{bmatrix} 10 \\ 5 \\ 0 \end{bmatrix}$$

$$\lambda_1 = 10$$

$$\lambda_2 = 5 + 1 / 2 \lambda_1$$

$$\lambda_3 = 1 / 2 \lambda_1 + \lambda_2 + 2 / 3 \lambda_3$$

$$\lambda_3 = 45$$

Figura 11. Aplicación del Teorema de Jackson para cálculo de tráfico en redesbasadas en distribución de probabilidades de Poisson

El análisis a través de la teoría clásica de tráfico se extiende hasta la automatización en el cálculo de troncales en centrales telefónicas, a través de la determinación la de probabilidad de tener disponibilidad de circuitos, en este caso, la tradicional ranura de tiempo (time slot, en inglés) de los E1 (primera jerarquía PDH, Jerarquía Digital Plesiócrona de transmisión). En la figura 12 se resume el fundamento de cálculo en centrales basadas en establecimiento de circuitos de ranuras de tiempo.



Figura 12. Modelo para aplicación del Erlang B en el cálculo de tráfico

Donde:

$$a[1 - Er_1(m, a)] \quad (8)$$

En la ecuación 8, a es el tráfico ofrecido en Erlang, m es el número de circuitos y $Er_1(m,a)$ es la probabilidad de "no disponibilidad" para un tráfico ofrecido a, y m circuitos o ranuras de tiempo en el troncal o servidor. El basamento matemático proviene de funciones estadísticas binomiales, las cuales en ciertas condiciones tienden a ser exponencial. Para el

caso de centrales, es orientada a sistemas de correlación temporal cero (no hay memoria de correlación ni hay memoria física de buffer).

En la tabla 3, se esquematiza un ejemplo de cálculo para un requerimiento de 1.2 Erlangs, y una probabilidad de bloqueo de 4%. La solución es aproximada y da como resultado 4 circuitos o ranuras de tiempo.

Tabla 3. Extracto de tabla ErlangB

a/m	1	2	3	4
0,8	0,444444	0,150943	0,038694	0,007679
1,0	0,500000	0,200000	0,062500	0,015385
1,2	0,545455	0,246575	0,089776	0,026226
1,4	0,583333	0,289941	0,119180	0,040043

En general, si la tasa de llamadas es λ por unidad de tiempo y la duración media de una Llamada es h entonces el tráfico A en Erlangs es:

$$A=\lambda.h(9)$$

Una excelente forma de practicar cálculos con Erlang B es a través de la máquina de Wikilman. Algo básica, pero permite aprehender el concepto. Ver figura 13.

http://personal.telefonica.terra.es/web/vr/erlang/cerlangb.htm

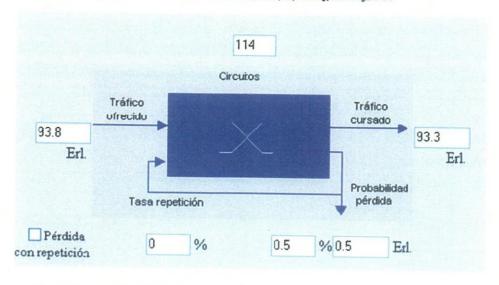


Figura 13. Máquina de Wikilman para practicar cálculos basados en Erlang B

En la teoría de colas, el Erlang B es de aplicación básica en el dimensionado de centrales telefónicas. Sin embargo, las centrales se caracterizan por ser un sistema sin almacenamiento (excepto la señalización en paquetes). Por ser un servicio isocrónico-sincrónico (el retardo en red debe ser constante y las muestras de voz son colocados en sistemas PCM que van hacia la red PDH o SDH), la memoria disponible es apenas para recibir el byte de la muestra PCM y colocarla en el troncal en forma inmediata.

Sin embargo, en la misma forma que a través de las formulas de Little se llega a un sistema de ecuaciones para N servidores, tipo M/M/N, a través del Erlang B, que es la ecuación 8, se pueden obtener las ecuaciones para calcular la memoria para elementos en espera y el tiempo promedio de espera en sistemas con memoria.

$$Er_2(m,a) = \frac{Er_1(m,a)}{1 - \frac{a}{m} + \frac{a}{m}Er_1(m,a)}$$
(10)

$$T_{w} = \frac{1}{\mu(m-a)} Er_{2}(m,a)$$
 (11)

Siendo μ el número de circuitos servidores para las muestras entrantes. Tw es el promedio de tiempo de espera. E_{r2} se refiere ahora al cálculo de Erlang C. Es decir, sistemas con memoria y sistemas de colas.

$$w = \frac{a}{m-a} Er_2(m,a)$$
 (12)

Donde w es el promedio de muestras de información en espera (palabras de 8 bits), o se puede trasladar al caso de paquetes, pero siempre en la consideración de que los servidores son circuitos de 64 Kbps. Sin embargo, se pudiese decir que la palabra aumenta en tamaño, y el circuito será nx64 Kbps, según la ingeniería del proyecto. Inclusive podría ser una división de 64 Kbps, ejemplo, servidores de 9.6 Kbps o 19.2 Kbps, velocidades típicas en equipos terminales como los Puntos de Ventas y Cajeros Automáticos.

Según todos estos conceptos de tráficos, y visualizando la diversidad de aspectos de este campo de las telecomunicaciones, la Unión Internacional de Telecomunicaciones mantiene o sugiere normas y estándares en el área de Ingeniería de Tráfico [16]. Dichas normas conllevan al desarrollo de modelos de tráfico para simulación y análisis de red (figura 14).

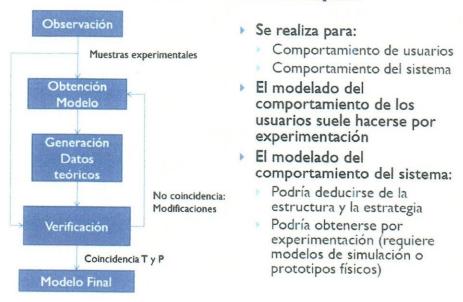


Figura 14. Modelado de un sistema de análisis de tráfico según ITU-T. Fuente: [16]

Dichos modelajes, mediciones y monitoreos lograrán impactos positivos en el diseño, mantenimiento y operación de la red, tal cual reflejan algunos ejemplos de actividades continuas en las figuras 15 y 16.

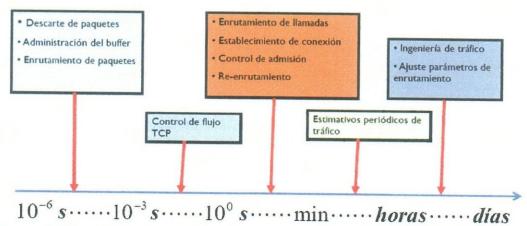


Figura 15. Escalas de tiempo para las tareas de operación y gestión de las redes. Fuente [16]

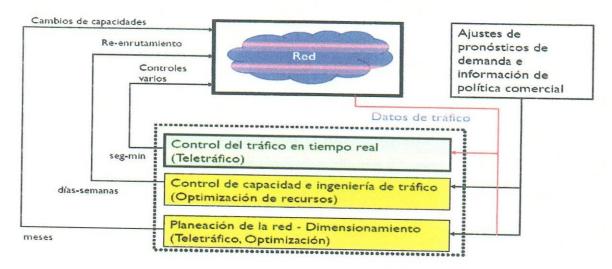


Figura 16. Posicionamiento del análisis de tráfico en la planificación, la ingeniería y la operación de la red. Fuente: [16]

3.- Caracterización y generación de tráfico autosimilar

Antes de describir el software diseñado se presenta una información resumida de las características principales de los tráficos y los métodos para generarlo.

La figura 17 permite ubicar el tráfico en el contexto no autosimilar y autosimilar. Una vez ubicado en ese contexto debemos indicar que el tráfico a analizar es autosimilar discreto en sentido amplio, y no en sentido estricto, dado que las fórmulas matemáticas pasan a ser una aproximación de las realidades estadísticas en cada una de las escalas temporales. Las figuras 17 y 18, permiten visualizar esa ubicación en el contexto de la autosimilaridad.

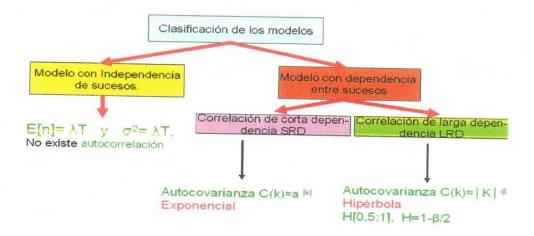


Figura 17. Caracterización de los modelos de tráfico en Redes de paquetes

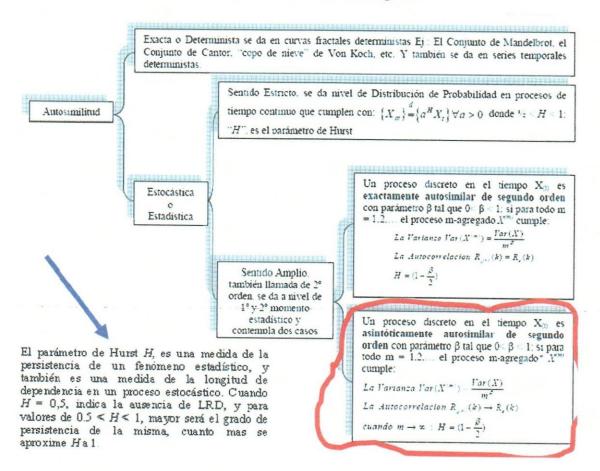
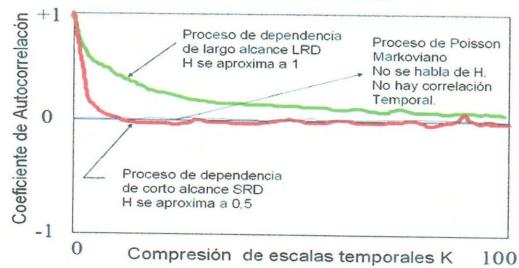


Figura 18. Ubicación en el contexto de la autosimilitud en sentido amplio. Fuente: [7]

Llegado este punto, una interpretación matemática que sintetiza la semejanza de las estadísticas referidas a la varianza y la autocorrelación, es el factor de autocorrelación, el cual consiste en un cálculo de la autocovarianza (la misma muestra es comparada una con otra a diferentes escalas) dividida entre las desviaciones estándar de las escalas comparadas. Ese cálculo da un gráfico normalizado en el eje y. Ver Figura 19.

Función de Autocorrelación



Coeficiente de correlación -Es una función de autocorrelación normalizada. p(t1.t2)=C(t1.t2)/(σ1 σ2).

Figura 19. Factor de autocorrelación

$$\rho = \frac{\left[(X_{(t)} - \mu)(X_{(t+\tau)} - \mu) \right]}{\sigma_{(t)}\sigma_{(t+\tau)}} = \frac{C_{(\tau)}}{\sigma_{(t)}\sigma_{(t+\tau)}} = \frac{Auto \cos arianza}{\sigma_{(t)}\sigma_{(t+\tau)}} \Rightarrow \rho = \frac{R_{(\tau)} - \mu^2}{\sigma_{(t)}\sigma_{(t+\tau)}}$$
(13)

Una función de autocorrelación en un proceso autosimilar se expresa en la siguiente ecuación:

$$\rho_k = \frac{1}{2} \left[(k+1)^{2H} - 2k^{2H} + (k-1)^{2H} \right]$$
 (14)

Esta es la base del desarrollo de ecuaciones para analizar el tráfico. Las mismas están centradas en el dominio del tiempo. Como las funciones en dicho dominio tienen una forma de expresarse en el dominio de la frecuencia, los desarrollos más exactos para determinar el parámetro H se han realizado en éste último. Así, las dos herramientas básicas han sido la Transformada de Fourier y la Transformada Wavelet o teoría de ondículas.

La siguiente gráfica (20) es indicativa de un esquema que muestra los métodos para determinar el parámetro H. Los expresados en el dominio del tiempo permiten calcular aproximadamente el parámetro H, mientras que los del dominio de la frecuencia permiten determinarlo exactamente.

En lo referente a la generación de un tráfico autosimilar, igualmente existe la posibilidad de hacerlo en los dos dominios. Los basados en el dominio del tiempo encuentran como máximo exponente a la posibilidad de modular fuentes no autosimilares, cuyas características están orientadas a un tráfico de Poisson Markoviano, con apagados y

encendidos de la fuente cuya probabilidad de tiempo de apagado y encendido corresponden a funciones del tipo de cola pesada, lo cual simula ráfagas [11]. En el dominio de la frecuencia, se puede generar el tráfico a través de la transformada de Fourier de la autocorrelación, con Periodogramas (minimización de la expresión de transformada de Fourier de la autocorrelación entre la serie de Fourier de la señal) y a través de la Transformada Wavelet.

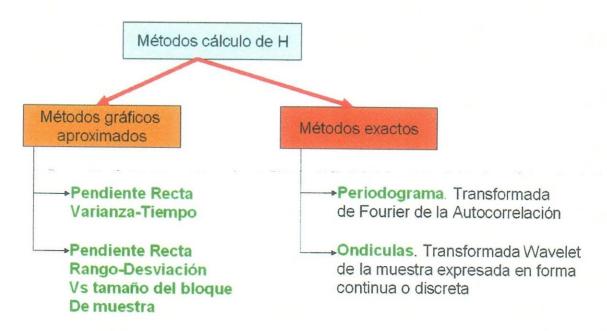


Figura 20. Métodos para determinar el parámetro de Hurst

4.- Relación matemática de los métodos de cálculo de H con los algoritmos de cómputo Los diversos métodos para el cálculo del parámetro de Hurst tienen demostraciones matemáticas que han posibilitado llegar a expresiones que han permitido realizar dimensionamientos correctos de las redes, a pesar de que todavía no existe un tratado de Ingeniería de tráfico como el heredado del sistema tradicional, que abrió todo un esquema hacia lo referente al Erlang B y al Erlang C.

4.1.-Método Rango Rescalado R/S

Hacia los años 50 y 60 Harold Edwin Hurst (1 Enero 1880 -7 Diciembre 1978) realizó y publicó novedosos estudios relacionados a las crecidas y sequías de algunos ríos. A través de sus estudios descubrió una relación que se observa inclusive en los anillos de los troncos de los árboles:

$$\frac{R}{S} \approx \left(\frac{N}{2}\right)^H$$
 (15)

Siendo N la cantidad de anillos o huellas dejadas en la sequía y crecida de los ríos. En el caso de redes de datos N es la cantidad de muestras. R es el Rango de la muestra y S es la desviación estándar de la muestra. Para el caso de estudio, la muestra comenzará en un

máximo, y se irá dividiendo por un factor. En el caso de los algoritmos desarrollados por Reinaldo Scappini [7], la facilidad de los algoritmos permite seleccionar el tamaño del factor con el cual se dividirá. El algoritmo protege que dicho factor permita cuando mínimo calcular una muestra. Es decir, el denominador con el cual se dividirá la muestra no puede ser menor a la unidad. La última muestra cuando se divida, no puede igualmente ser menor a la unidad. Como una tabla o gráfico expresa más que mil palabras, refiérase a la tabla 4. En el gráfico se escogió un factor de división igual a 2. El software fue modificado para que solo dividiera en un factor de 2 para mayor simplicidad en el diseño de interfaces gráficas.

Tabla 4. Cálculo del tamaño de muestras en el método R/S

	71	m_0 = Cantidad total de datos Muestra inicial	
	m_1	 	$m_1 = \frac{m_0}{2}$
m_2			$m_2 = \frac{m_1}{2}$
m_3			$m_3 = \frac{m_2}{2}$
m_4			$m_4 = \frac{m_3}{2}$

El rango vendría dado por la siguiente expresión, y N será el tamaño de la muestra definida como m.

$$R(m) = \left[\max \sum_{i=1}^{m} X_i - \left[\overline{X_m} \right] - \min \sum_{i=1}^{m} X_i - \left[\overline{X_m} \right] \right]$$
 (16)

R(S) es la desviación estándar del bloque.

La ecuación queda expresada en la forma siguiente:

$$\log\left(\frac{R(m)}{R(s)}\right) \approx H\log(N) - H\log(2) \quad (17)$$

Por cada una de las iteraciones obtenemos una relación para graficar. El método se abrevia como R/S, y la pendiente H es el parámetro de Hurst.

En el método explicado, el promedio se toma con respecto a la muestra m. Una versión mejorada de este método consiste en tomar el promedio de todas las muestras m y se les vuelve a promediar. Es decir, será el promedio de los promedios. Como el rango y la

desviación estándar se calculan referente a dicho promedio, el cálculo será más refinado y tendrá un mayor mejor margen de confianza. Dicho método se denomina Rango Reescalado Promedio.

4.2.- Método Varianza/Tiempo

La base matemática de dicho método es a través de la generación de varias representaciones temporales de las muestras, mediante técnica de agrupamiento. Es como si se construyeran "clases" estadísticas en el plano temporal. En este caso "m" representa la escala temporal.

$$X_k^{(m)} = \frac{1}{m} \sum_{i=km-(m-1)}^{km} x_i$$
 (18)

$$Var(X^{(m)}) \approx \frac{Var(X)}{m^{\beta}}$$
 (19)

Donde β está relacionado con H mediante la ecuación:

$$H = 1 - \frac{\beta}{2} \tag{20}$$

Aplicando logaritmo a la ecuación 19:

$$\log Var(X^m) = \log Var(X) - \beta \log(m)$$
 (21)

De esta forma, la pendiente permite calcular el parámetro H.

4.3.- Método Transformada Wavelet

Los métodos exactos analizados en este trabajo, y expresados en los módulos de software respectivos, están basados en los Wavelets u ondículas.

Todo efecto en el tiempo tiene otro en el dominio de la frecuencia. Basado en este principio, es abundante la literatura que ha establecido relaciones importantes en ambos dominios, donde aparece el parámetro H.

La evolución en el concepto de transformadas llevó a una fase de análisis multiresolución (MRA) a través de ondículas. La razón de ello, es que la Transformada de Fourier no expresa en que momento ocurren determinadas frecuencias. Un método intermedio antes de llegar a los análisis con ondículas, fue la Transformada Corta de Fourier (Gabor), en la cual se establece una especie de barrido a la señal s(t) con una ventana angosta en el tiempo, y se calcula la Transformada de Fourier en esa ventana. Luego la ventana se va desplazando, por lo que la función de Integral es aplicada en el rango del tamaño de la ventana.

Las limitaciones en cuanto al ancho de la ventana, lo cual implica grandes sacrificios en baja o alta frecuencia, según el ancho elegido¹, orientó a la implementación de una

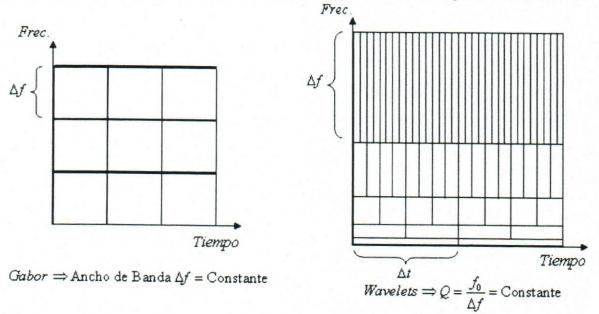


Figura 21. Tamaño de las ventanas en la Transformada Corta de Fourier y en la Transformada Wavelet

metodología de ventana variable, en amplitud y ancho de la ventana. Estas son las ondículas o Wavelets. En la tabla 5 se detallan los parámetros del Wavelet u ondícula, basados en compresión y desplazamiento del Wavelet.

Tabla 5. Parámetros del Wavelet

Traslación	Cambio de escala	Traslación y cambio de escala
ψ(t – b)	$\frac{1}{\sqrt{a}} \psi \left(\frac{t}{a}\right)$	$\frac{1}{\sqrt{a}} \psi \left(\frac{t-b}{a} \right)$

Existe todo un tratado en diversas documentaciones de la forma como se llega al análisis multiresolución. Colocaremos extractos de estas ecuaciones.

esto significa que el análisis se hace con un ancho de banda constante.

Existe un límite inferior para los parámetros Δt y Δf , de forma tal que siempre se verifica la relación $\Delta t \Delta f \geq \frac{\pi}{4}$. Entonces si se escoge un Δt pequeño, esto es la mínima distancia en tiempo que pueden estar dos impulsos para tener una buena resolución en tiempo, forzosamente tendrá un Δf mas grande que disminuirá la resolución en frecuencias, y viceversa, teniendo a Δf como la mínima distancia medida en Hz. a la que pueden estar dos componentes frecuenciales. Se concluye entonces, que existe un compromiso entre tiempo y frecuencia y una vez elegido el tamaño de la ventana, éste es el mismo para todas las frecuencias,

La filosofía de la teoría de ondículas es realizar el barrido a una señal varias veces, así como se puede observar en la figura 22. De esta forma, el valor de los coeficientes Wavelets cambia en cada barrido. La ondícula se va modificando a cada barrido en lo que respecta al ancho de la misma. De esta forma las altas frecuencias serán representadas con factores de compresión bajos y las bajas frecuencias con factores de compresión altos. Ello es debido a la forma como se representa el Wavelet (ver Tabla 5). A continuación se representa la ecuación matemática de la transformada Wavelet:

$$S_{W}(a,\tau) = \int_{-\infty}^{\infty} s(t)\psi(t-\tau)dt \ (22)$$

$$S_W(a, \tau) = \langle s, \psi_{a, \tau} \rangle; a \in \Re, \tau \in \Re$$
 (23)

Donde S_W son los coeficientes Wavelets. a es el factor de compresión o dilatamiento del Wavelet y τ es el desplazamiento del mismo.

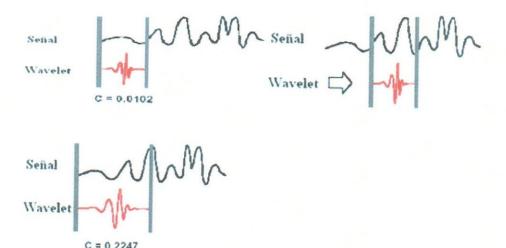


Figura 22. Proceso de convolución de las ondículas o Wavelets con la señal

El análisis multiresolución con Wavelets nos lleva a una ecuación, en la cual la recuperación de la s(t) se hace nuevamente multiplicando la señal por los Wavelets, pero el proceso lleva a separar la ecuación en un filtro para la baja frecuencia y otro para la alta. Se puede decir, que en un primer barrido, la alta frecuencia sale con claridad, pero la baja frecuencia debe pasar por un nuevo proceso de multiplicación por los Wavelets.

$$s(t) = \sum_{k} a_{s}(j,k)\phi_{j,k}(t) + \sum_{j=1}^{j} d_{s}(j,k)\psi_{j,k}(t)$$
 (24)

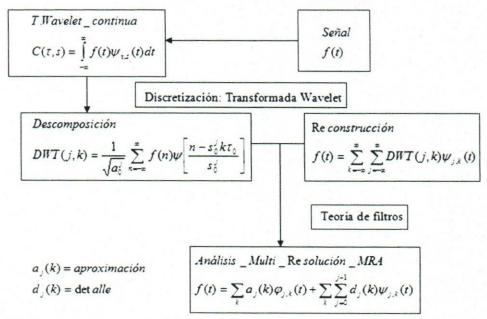


Figura 23. Proceso e Transformación Wavelet y recuperación de la señal mediante MRA

Siendo la primera expresión un filtro pasa bajo y el segundo un pasa alto en la ecuación 24. Igual se concluye que la primera parte es una aproximación y la segunda un detalle. Los términos $\phi_{j,k}(t)$ y $\psi_{j,k}(t)$ son entonces los coeficientes Wavelets . El primero para las aproximaciones y el segundo constituye los detalles. Igualmente, la figura 23 representa en forma matemática el proceso completo de transformación y recuperación de la señal a través de la transformada Wavelet. En manera gráfica (figuras 24 y 25) se puede diagramar lo explicado.

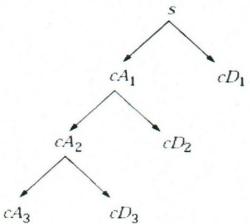


Figura 24. Procesos de filtros paso bajo y paso alto en Transformada Wavelet

Para este caso, el resultado de la Transformada inversa Wavelet, implica hacer pasar la señal por filtros, o explicado en términos matemáticos, multiplicar por los coeficientes y realizar la siguiente operación:

$$s(t) = cA_3 + cD_3 + cD_2 + cD_1$$
 (25)

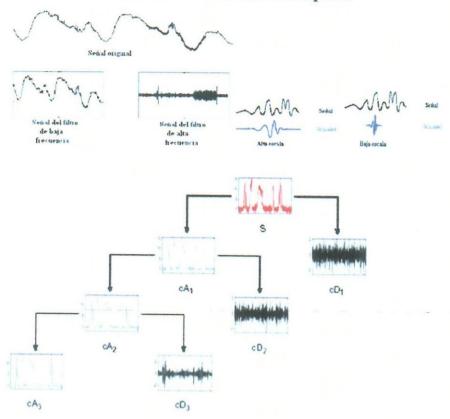
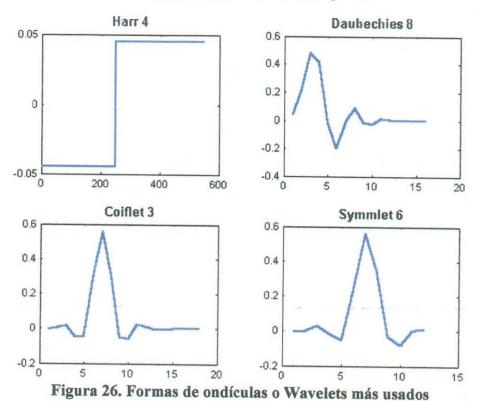


Figura 25. Esquema visual del proceso recuperación de la señal

Para calcular las coeficientes, existen diversos tipos de ondículas para obtener un Wavelet madre. A continuación se mencionan algunos Wavelets u ondículas. Ver figura 26.

- -Haar
- -Daubechies
- -Coiflet
- -Symmlet



Los Wavelet Haar trabajan en base a dos coeficientes, generándose bajo el siguiente criterio matemático:

$$\phi_0(i) \begin{cases} \frac{1}{\sqrt{2}} \to i = 0, 1 \\ 0 \to Otro \end{cases}$$
 (26)

$$\psi_0(i) \begin{cases} \frac{1}{\sqrt{2}} \to i = 0 \\ \frac{-1}{\sqrt{2}} \to i = 1 \\ 0 \to Otro \end{cases}$$
 (27)

Los Wavelets Daubechies tienen más coeficientes y son utilizados con mucha frecuencia para determinar parámetros H y para generar tráfico con características de autosimilaridad. Dado que es el método utilizado en los módulos de software de este trabajo, tanto en la generación de tráfico, como en el monitoreo, se describe a continuación el método Daubechies(4). Con dicha explicación se puede hacer la comparación con Daubechies(6) que es el método utilizado en el módulo de determinación de parámetro H de Darryl Veitch

(LDestimate.m, con su variante en este trabajo denominado LDestimatev2.m). La tabla 6 expresa las razones de utilizar Wavelets Daubechies para el caso de señales discretas.

Tabla 6. Comparación ondículas Haar y Daubechies. Fuente: [7]

	Valores	promedio:	s de H estima	ados		Valores p	romedios	de H estima	dos
Método	.6	.6 .7			Método	.8		.9	
	Ĥ	$\Delta H(\%)$	Н	$\Delta H(\%)$		Ĥ	$\Delta H(\%)$	Ĥ	$\Delta H(\%)$
Haar	.6073 (.580, .635)	+1.220	.7141 (.687, .742)	+2.018	Haar	.8217 (.794, .849)	+2.712	.9299 (.902, .958)	+3.327
Daub(2)	.6019 (.574, .630)	+0.323	.6984 (.671, .726)	-0.229	Daub(2)	.7943 (.767, .822)	-0.709	.8898 (.862, .917)	<u></u> 1.137
Daub(4)	.6026 (.575, .630)	+0.433	.7039 (.676, .731)	+0.554	Daub(4)	.8055 (.778, .833)	+0.684	.9074 (.880, .935)	+0.821
Daub(8)	.6026 (.575, .630)	+0.430	.7031 (.676, .731)	+0.445	Daub(8)	.8039 (.776, .831)	+0.486	.9049 (.877, .932)	+0.545
Daub(16)	.6013 (.574, .629)	+0.214	.6987 (.671, .726)	-0.185	Daub(16)	.7962 (.769, .824)	-0.474	.8938 (.866, .921)	-0.694

La generación de muestras de tráfico autosimilar se implementó mediante módulos de software FGNIFOURIER.m y FGNDWv2.m. El primero genera muestras a través de la transformada de Fourier y el segundo a través de transformada Wavelet[10]. En la bibliografía especificada se indican en detalle los algoritmos de generación de tráfico autosimilar.

La matriz a continuación representa los coeficientes Wavelets Daubechies(4).

$$\begin{bmatrix} C_0 & C_1 & C_2 & C_3 & 0 & 0 & 0 & \dots \\ C_3 & -C_2 & C_1 & -C_0 & 0 & 0 & 0 & \dots \\ & & C_0 & C_1 & C_2 & C_3 & 0 & \dots \\ & & & C_3 & -C_2 & C_1 & -C_0 & 0 & \dots \\ & & & & & & & & \\ & & & & & & \\$$

La transformada inversa Wavelet deberá ser procesado por la transpuesta de esta matriz.

Luego de un proceso de despeje de ecuaciones, los valores de los coeficientes Wavelets Daubechies quedan definidos así:

$$\begin{cases} C_0 = (1 + \frac{\sqrt{3}}{4}) * \sqrt{2} \\ C_1 = (3 + \frac{\sqrt{3}}{4}) * \sqrt{2} \\ C_2 = (3 - \frac{\sqrt{3}}{4}) * \sqrt{2} \\ C_3 = (1 - \frac{\sqrt{3}}{4}) * \sqrt{2} \end{cases}$$
(29)

El algoritmo para la transformada Wavelet a señales discretas se aplica mediante el siguiente procedimiento:

Si el vector de datos indicado continuación es parte un vector de datos (múltiplo de 4), donde cada muestra representa el número de paquetes o el tiempo entre arribos:

$$\begin{bmatrix} y_0 & y_1 & y_2 & y_3 & y_4 & y_5 & y_6 & y_7 & y_8 & y_9 & y_{10} & y_{11} & y_{12} & y_{13} & y_{14} & y_{15} \end{bmatrix}$$
 (30)

El vector se multiplica por la matriz, dando como resultado:

$$\begin{bmatrix} s_0 & d_0 & s_1 & d_1 & s_2 & d_2 & s_3 & d_3 & s_4 & d_4 & s_5 & d_5 & s_6 & d_6 & s_7 & d_7 \end{bmatrix}$$
 (31)

Donde s está referido a aproximación y d a detalle.

El vector es permutado a fin de que las aproximaciones y los detalles queden en grupos

$$\begin{bmatrix} s_0 & s_1 & s_2 & s_3 & s_4 & s_5 & s_6 & s_7 & d_0 & d_1 & d_2 & d_3 & d_4 & d_5 & d_6 & d_7 \end{bmatrix}$$
 (32)

El proceso se repite para el grupo de aproximación, quedando como resultado:

$$\begin{bmatrix} S_0 & D_0 & S_1 & D_1 & S_2 & D_2 & S_3 & D_3 & d_0 & d_1 & d_2 & d_3 & d_4 & d_5 & d_6 & d_7 \end{bmatrix} (33)$$

Se realiza nuevamente el proceso de reagrupar quedando como sigue:

$$\begin{bmatrix} S_0 & S_1 & S_2 & S_3 & D_0 & D_1 & D_2 & D_3 & d_0 & d_1 & d_2 & d_3 & d_4 & d_5 & d_6 & d_7 \end{bmatrix}$$
 (34)

En este caso, dado que los coeficientes son cuatro, el proceso debe parar.

La figura 27 resume la forma de recuperar la señal.

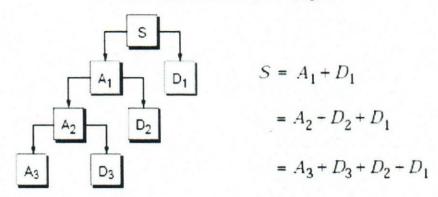


Figura 27. Visualización de la aplicación de MRA

La reconstrucción de la señal es exactamente lo opuesto, y se debe multiplicar por la transpuesta o inversa de la matriz de coeficientes Wavelets.

Obtenidos los coeficientes de detalle Wavelet, la Varianza de los mismos guarda relación con el parámetro Hurst. Ello queda expresado en la siguiente ecuación:

$$\log_2(E[d_x(j,k)]) = j\gamma + \log_2(C)$$
 (35)

Proveniente de la ecuación:

$$E[d_x(j,k)] \propto 2^{j\gamma}$$
 (36)

Donde se tiene que:

$$\gamma = 2H - 1$$

$$\gamma = 2H + 1 \tag{37}$$

La primera expresión de la ecuación 37 es para un proceso autosimilar con larga dependencia, y la segunda es para un proceso autosimilar con incrementos estacionarios independientes. La ecuación 35 y 36 son la base del algoritmo planteado y desarrollado en los módulos de determinación del parámetro de Hurst y la generación de tráfico autosimilar LRD (dependencia de largo alcance).

5.- Software de Análisis y simulación de tráfico autosimilar.

El software implementado tiene un alto componente académico y un componente de investigación científica. Las interfaces son orientadas a objetos, y existen algunas pequeñas implementaciones no orientadas a objeto y están relacionadas a crear facilidad de validación del software.

Facilidades del software.

El software fue desarrollado en MATLAB, versión 7.9.0.

Las facilidades del software permiten:

- Generar tráfico autosimilar a través de Transformada Inversa de Fourier con Ruido Gaussiano
- Modulo externo .m para generar tráfico autosimilar a través de la transformada inversa Wavelet.
- Determinación del parámetro H de un tráfico usando los métodos temporales y método de transformada Wavelet
- Módulo para análisis científico o de ingeniería de diversas muestras de tráfico

Tabla 7. Módulos .m desarrollados por Reinaldo Scappini. Universidad de la Plata.(*)
Indica una nueva versión

Aplicaciones desarrolladas por Rica	rdo Scappini. Universidad Nacional de la Plata
Módulo Matlab.m	Aplicación
CargaVectores.m	Carga de Vectores y generación de procesos sintéticos equivalentes para la muestra número 1.
Carga Vectores2.m(*) (versión de Carga Vectores,m)	Carga de Vectores y generación de procesos sintéticos equivalentes para la muestra número 2.
SerieTEscalada.m	Graficador de comportamiento
SerieTiempoEscaladoComparada.m	Graficador de comportamiento para dos procesos
RangoEscalado.m	Estimación de H, por el método gráfico de Rango Reescalado
RangoEscaladoPromedio.m	Idem anterior para dos procesos
VarianzaTiempo.m	Estimación de H, por el método gráfico de Varianza vs Tiempo
Varianzav2.m(*) (versión de VarianzaTiempo)	Igual anterior, cambios fundamentales para adaptar al modo objeto, cuando se llame al módulo.Compatibilidad
VarianzaTiempoComparada.m	Idem anterior para dos procesos
VarianzaW.m	Estimación H, por el método regresión de Varianza coeficientes Wavelets vs Octavas
VarianzaEv2.m(*) (versión de VarianzaW.m)	Modificación al anterior para adaptarlo al modo gráfico
FBM.m	Estudio gráfico del comportamiento de autocorrelación de dos procesos
FGNIFOURIER.m	Generación de secuencias autosimilares sintéticas con exponente H determinado, con transformada de Fourier

Todos los módulos fueron modificados para adaptarlos al modo objeto. Así, las entradas a funciones se dan desde los objetos. Igualmente, gran parte de los cambios están orientados a poder manipular los gráficos generados por la función, dado que el despliegue de gráficos desde los módulos .m se manejan en forma diferente a los módulos .fig en Matlab. Como

las funciones en .m buscan graficar y plasmar las informaciones resultantes en el estado presente en la versión desarrollada por Scappini, fue necesario realizar cambios en las funciones para almacenar los valores de interés y poder realizar manipulación externa y por supuesto, poder almacenar y generar archivos históricos. Pero los cambios no alteran el fundamento algorítmico desarrollado por el autor que implementó los módulos.m básicos.

A continuación se explican los módulos .m en Matlab.

• function []=CargaVectores(nombre_archivo). Nombre del programa: CargaVectores.m (CargaVectores modificado por Franklin Planchart para almacenar los datos obtenidos en las muestras)

Primero, antes de extraer los vectores de las muestras, las mismas deben ser almacenadas. En el proceso se debe utilizar un software o un equipo analizador de protocolos de comunicación que almacene muestras, a diferentes días, horas, meses, o períodos, a los cuales se les hará especies de "biopsias" tomadas en micromuestras de un millón (un millón de paquetes, un millón de tiempos de llegada). Estas serán las entradas al programa CargaVectores.m y CargaVectores2.m

A continuación se detalla el procedimiento para extraer muestras a formato .txt o .dat para ser tratadas por los mencionados módulos. Se coloca textualmente como lo explica Ricardo Scappini en su tesis [7]:

"Se utiliza el software Wireshark, que es un conocido analizador de protocolos basado en tepdump, que permite manipular trazas que utilicen formato compatible con el tepdump y además cuenta con una utilidad de línea de comandos llamada Tshark, que resulta particularmente apropiada para esta tarea como se muestra a continuación:

- · Se toma el primer millón de tramas de la traza y se lo convierte en un archivo con la extensión .pcap. La sintaxis del comando es: Tshark –r [nombre de la traza] –c 1000000 –w [nombre.pcap].
- · Si se quiere trabajar con los tiempos entre arribos, creamos el archivo correspondiente, de la siguiente manera: Tshark —r [nombre.pcap] —e frame.time_delta —T fields >nombre.txt. Esto lo que hace es, leer el archivo .pcap que se creo con el millón de trazas, y lo filtra mediante el contenido del campo timestamp, del que a su vez establece la diferencia con la lectura anterior creando el valor tiempo entre arribos para cada trama y luego guarda el archivo en formato ASCII.
- · Del mismo modo si se desea trabajar con la longitud en bytes de la trama, se utiliza el campo frame.len para el filtrado de la siguiente forma: Tshark —r [nombre.pcap] —e frame.len —T fields > nombre.txt. Obteniendo de esta forma la salida en formato ASCII que es la más cómoda para poder utilizarla con el Matlab.
- · Se aclara que es posible crear archivos con un número arbitrario de campos, como podría ser tiempos entre arribos y longitud de la trama, sencillamente repitiendo en la misma línea –e [nombre de campol], -e[nombre de campo2], se recomienda la consulta de la documentación de Tshark como así también la del Wireshark para ver la importante capacidad de filtrado y análisis de protocolos que tiene este software y que puede ser de

mucha utilidad en trabajos donde se requiere la utilización de archivos con formato .pcap" [7].

Los archivos .TXT utilizados en el software desarrollado tienen el último formato. Es decir, el comando utilizado fue: Tshark -r [nombre.pcap] -e frame.time_delta -e frame.len -T fields > nombre.txt..

En algunos casos los archivos cuando se bajan de la Internet, no están en .txt. Entonces lo que se hace es renombrar el archivo. Se debe tener sumo cuidado de que los archivos estén ordenados en esa forma. Es decir, la primera columna está referida al tiempo absoluto, y la segunda está referida a la longitud del paquete en Bytes. El software se encargará de extraerlos y hacer el tratamiento para generar de allí cuatro archivos: el original referido a tiempo y longitud de bytes, y otros dos, los cuales son muestras exponenciales de tiempo o bytes, generados por un método de regresión. Es decir, con los datos originales se hace un forzamiento para regenerar tráfico clásico. Este es el método de regresión.

Para dar un ejemplo, utilicemos el archivo BC-pAug89.txt, el cual se encuentra, o debe estar disponible en el mismo directorio en donde se encuentren los programas.

```
dato='BC-pAug89.txt';
datos=load(strread(dato, '%c'));
```

Ahora, en la variable *datos* se ha almacenado la matriz de información que se encuentra en BC-pAug89.txt. A continuación se coloca un extracto de dicha información, la cual se observa al escribir el nombre de la variable:

```
>> datos(30:40,1:2)

ans =

1.0e+005 *

7.7403     0.0006
7.7403     0.0006
7.7403     0.0006
7.7403     0.0006
7.7403     0.0006
7.7403     0.0006
7.7403     0.0006
7.7403     0.0006
7.7403     0.0006
7.7403     0.0006
7.7403     0.0006
7.7403     0.0008
7.7403     0.0008
7.7403     0.0008
```

La primera columna son los tiempos de llegada y la segunda, como se explicó, es la longitud en bytes. Todos los datos están multiplicados por 10⁵, como se expresa en el formato de salida.

Forma de extraer datos de las muestras: datos=load(strread(dato, '%c'));

En donde la variable 'dato' es el nombre del archivo en .TXT.

Se extrae la respectiva columna de tamaño byte y tiempo de llegadas de los paquetes. El software llevará el tiempo de llegada a tiempos entre llegadas.

vec=datos(:,2); % Vector longitud en bytes de los de los paquetes ethernet vet=datos(:,1); % Vector conteniendo los tiempos en que arribaron los paquetes de los paquetes. El software calcula los tiempos entre llegadas.

De dicha información se hace un recálculo y se generan archivos no autofractales. Ello implica que se almacena un vector de tamaño de paquetes y otro de tiempos entre llegadas del tipo autosimilar, y un vector de tamaño de paquetes y tiempo entre llegadas no autosimilar (tráfico de Poisson generado por regresión y calculando un Lambda). Así del primer archivo seleccionado .TXT, se almacenan cuatro vectores con un millón de posiciones cada vector:

save vecs.txt vec –ascii % Tamaño paquetes autosimilar save tieas.txt tiea –ascii % Tiempo entre llegadas autosimilar save xpoiss.txt xpois –ascii % Tamaño paquetes Poisson save tieaps.txt tieap –ascii % Tiempo entre llegadas Poisson

 function []=CargaVectores(nombre_archivo). Nombre del programa: CargaVectores2.m

Este programa ejecuta la misma función del anterior, pero almacena los datos en otros vectores:

save vecs2.txt vec –ascii % Tamaño paquetes autosimilar save tieas2.txt tiea –ascii % Tiempo entre llegadas autosimilar save xpoiss2.txt xpois –ascii % Tamaño paquetes Poisson save tieaps2.txt tieap –ascii % Tiempo entre llegadas Poisson

• function [FGNFourier]=FGNIFOURIER(n,H). Nombre programa: FGNIFOURIER.m

Este programa recibe como entrada el número de muestras y el parámetro de Hurst. A partir del espectro de frecuencia del ruido y la forma como se relaciona con el parámetro H, se procede a calcular una serie de números con la Transformada Inversa de Fourier. Luego se procede a tomar la parte real. Los números generados son almacenados en la forma siguiente, a través de los códigos en el programa:

FGNFourier=real(ifft(zFGN)); save x.txt FGNFourier -ascii; %se guardan los números generados en x.txt save n.txt n -ascii; % se almacenan los números de muestras y el save H.txt H -ascii; % parámetro H

• function [d]=SerieTEscalada(xf,ut,fe,cs,nombre). Nombre Programa: SerieTEscalada.m

El llamado a esta función, tiene 5 parámetros de entrada. El programa fija algunas de estas. Así, en xf, se colocan las muestras provenientes del respectivo vector sintético o archivo seleccionado en la muestra número 1. El parámetro ut se fija en .001, lo cual indica que se hace una simulación de que cada muestra se recolecta en .001seg. Es solo una estrategia para calcular series temporales. En lo sucesivo, cada escala se multiplica por diez, dando como resultado una nueva escala acumulativa de eventos (tiempos o paquetes), hasta llegar a muestras recolectadas cada segundo. Para ello, el parámetro fe se fijó en 10. El siguiente parámetro, cs indica cuantas series se desea ver. Aquí se fijó dicho parámetro a 4. El software dará un mensaje cuando con la cantidad de datos del vector, no se pueda construir todas las escalas fijadas. En el parámetro nombre se coloca el nombre del archivo.

function[]=SerieTEscaladaComparada(direccion1,vector1,direccion2,vector2). Nombre programa: SerieTEsacaladaComparada.m

Este programa llama al programa SerieTescalada.m dos veces, a fin de ejecutar con un vector y luego el otro con el que se desea comparar. Se le transfieren entonces el nombre y las muestras del vector, las cuales pueden ser tamaño de los paquetes o tiempos entre llegadas de los paquetes. Esta carga está automatizada en el menú en modo objeto. Las variables direccion1 y direccion2 son los nombres de las muestras y vector1, vector2, es decir, las muestras seleccionadas.

• function []=Varianzav2(vector,nombre,esc). Nombre programa Varianzav2.m

Calcula el parámetro H mediante la automatización de la siguiente ecuación:

$$\log Var(X^m) = \log Var(X) - \beta \log(m) \quad (38)$$

m se refiere a la escala y β está relacionado con el factor H, con lo que la pendiente de la recta, calculada con el método de cuadrados mínimos refleja el parámetro H a través de β .

El software original tiene la particularidad de introducir como parámetro la Base del número de muestras, la cual se incrementa con un exponente que crece en forma geométrica para la siguiente suma de muestras. Ejemplo, si se elige como base el 2, la primera muestra será 2, la segunda 4, y así sucesivamente hasta cubrir todas las muestras, que para este caso se ha estandarizado a un millón de muestras. Si la base es 3, la primera muestra acumula los tres primeros datos, la segunda 9 y así sucesivamente.

Tal facilidad fue anulada en esta versión, fijando dicho parámetro en 2. Así se transfiere a través del parámetro 'esc' el número 2 en el módulo.

function []=VarianzaTiempoComparado(direccion1,vector1,direccion2,vector2). Nombre programa: VarianzaTiempoComparado.m

Este programa llama al programa Varianzav2.m dos veces para dos vectores. Los parámetros especifican los nombres de las muestras y los vectores con las muestras. Se pueden comparar el vector sintético con la muestra 1 e igualmente la muestra 1 se puede comparar consigo mismo en lo que respecta a los vectores autosimilares y los de Poisson. El software orientado a objeto ha explotado dicha posibilidad de cruces comparativos, claro usando la versatilidad de éste interesante módulo.

function []=RangoRescalado(vector,nombre). Nombre programa: RangoRescalado.m

Módulo para calcular el valor del parámetro de Hurst, por el método del Rango Reescalado R/S, también llamado estadístico R/S. Se basa en el método de aproximación por mínimos cuadrados y tiene como epicentro el cálculo del Rango de la serie y la desviación estándar de la misma.

$$\log\left(\frac{R(m)}{R(s)}\right) \approx H\log(N) - H\log(2)$$
 (39)

La pendiente de la recta determinará H en forma aproximada. Tiene como parámetros de entrada el nombre de la muestra y el vector contentivo de las muestras.

En este caso, el promedio básico para realizar los cálculos está basado en el promedio del primer bloque.

• function []=RangoRescaladoPromedio(vector,nombre). Nombre programa: RangoRescaladoPromedio.m

Es una mejora a la versión anterior. Es el mismo principio en lo que respecta a la ecuación, pero los cálculos de Rango y Desviación Estándar están referidos al promedio calculado en cada uno de los bloques constituidos en le aglomeración de las muestras.

• function []=VarianzaEv2(vector,identif). Nombre programa: VarianzaEv2.m

Es un módulo que permite diagramar el logaritmo en base dos de la Varianza de los coeficientes Wavelets contra las octavas, siendo las octavas los logaritmos en base dos de las compresiones del Wavelet u ondícula. Ello está basado en la siguiente expresión:

$$\log_2(Var[d_x(j,k)])\alpha j\gamma$$
 (40)

Donde Y está relacionado con el parámetro de Hurst según la expresión:

$$H = \frac{1-\gamma}{2}$$
 (41).

El parámetro k está referido a los desplazamientos del Wavelet.

Así, a través de la recta se obtiene H en forma indirecta, con el método de pendiente, y aplicando método de mínimos cuadrados para graficar la recta.

• function []=FBM(vector,nombre,hurst,lag). Nombre programa FBM.m

Este módulo grafica el comportamiento de la correlación de un proceso de FGN teórico y la de un vector que representa un proceso FGN sintético tomado de trazas reales.

Los parámetros de entrada están referidos a las muestras (vector), nombre de la muestra, parámetro de Hurst y la el máximo número de escala (lag). Con dicho se número se calcula rápidamente un vector de autocorrelación sintética con el parámetro H indicado, y se compara con la muestra seleccionada. La siguiente expresión es la que permite el cálculo de los números de la autocorrelación sintética:

$$R(i) = .5(((i+1)^{2H}) + ((i-1)^{2H}) - (2i^{2H}))$$
 (42)

El valor de i se va iterando hasta llegar al número especificado en lag.

Tabla 8. Módulos desarrollados por Darryl Veitch. LDestimatev2.m es una modificación al original LDestimate.m

Aplicaciones desarrolladas por Darryl Veitch					
Módulos .m	Aplicación				
LDestimate.m	Determina el valor de alfa acorde a las condiciones establecidas en el programa.				
LDestimatev2.m(*)	Automatiza el cálculo de H mediante iteración automática de J1,J2, número de desvanecimientos.				
Newchoosej1.m	Selecciona el mejor jl, para un jl específico.				
Regrescomp.m	Determina rangos j1-j2 de confianza, acorde al valor Q.				
initDWT discrete.m	MRA para calcular coeficientes Wavelets				
wtspec.m	Efectúa la descomposición mediante Wavelets Daubechies				

function [alphaest,cfCest,cfest,Cest,Q,j1opt,yj,varj] =
 LDestimate(data,regu,j1,j2,discrete_init,calcj1,printout). Nombre programa:
 LDestimate.m

Este programa permite el cálculo de H mediante los coeficientes Wavelets, según las siguientes ecuaciones:

$$\log_2(E[d_r(j,k)]) = j\gamma + \log_2(C)$$
 (43)

Se le conoce como diagrama Logaritmo-Escala, y dado que V está relacionado con H, significa que se puede calcular H aplicando una regresión. Pero a diferencia de las anteriores, esta regresión es iterativa, no es única, y calcula la eficiencia de dicha regresión en un segmento dado, con un valor de compresión y escalamiento sucesivo. La decisión no se hace con el mayor factor de calidad, sino que se considera que la diferencia de las compresiones j1 y j2 son significativas. En este caso mayores a 4 (asumido por el autor), lo cual se manifiesta en un cambio de versión denominado LDestimatev2.m.

Primero se calculan todos los coeficientes y acorde a la escala de compresión J. Dicho cálculo da una gráfica en la que no se puede calcular H, pues según la zona de compresión, el valor de H va pasando por varios cambios. Así, el software determina un factor de calidad en determinadas zonas de la compresión, para poder aplicar una curva de regresión en una determinada zona, y la cual entonces constituye una recta. Nuevamente, la pendiente juega un factor importante. Igualmente, el cálculo arroja cuatro valores de H y la lógica dirá cual es el que se seleccionará. Ejemplo si tres valores de H son negativos, o son superiores a uno(1) se toma el positivo menor a uno(1).

Ello está expresado en las ecuaciones de decisión siguiente, utilizadas en el módulo:

```
HLRD = (alphaest + 1)/2 (44);

H = (3-alphaest)/2 (45);

h = (alphaest - 1)/2 (46);

D = (5 - alphaest)/2 (47);
```

Siendo la siguiente la salida tipo texto original del software:

```
fprintf('Goodness of fit statistic Q (probability of data assuming regression valid): \%6.5f \ n',Q)
fprintf('Scaling parameters are: alpha (LRD) H (LRD rewrite) H=h (ss,Holder) D (frac dim, if alpha in (1,3)) \n')
fprintf(' Estimates: \%4.3f (\%4.3f) \%4.3f \\%4.3f\n',alphaest,HLRD,H,D)
fprintf(' CI"s: [\%4.3f, \%4.3f] [\%4.3f, \%4.3f]
```

La ecuación H = (3-alphaest)/2, fue introducida, acorde a pruebas de validación. En este caso es una modificación al software que queda documentada. Ello es debido a que en el proceso de automatización y generación de matrices por cada escala de compresión y cada selección de desvanecimiento del wavelet, las relaciones del factor de calidad con respecto al escalamiento y el H generado, llevaban a decisiones totalmente erróneas en el sistema de cómputo. La fórmula introducida es posible que en otras versiones de LDEstimate haya estado presente y se le haya borrado por error. De todas formas el módulo está en período de prueba durante el año 2012.

Los parámetros de entrada son: data, se refiere al vector con las muestras, j1 y j2 son los escalamientos. El cálculo se realiza entre los valores j1 y j2 seleccionados y eso dará una recta por regresión, la cual puede ser o no la solución al cálculo de H. Esta iteración es

manual y las salidas gráficas permiten seleccionar el próximo j1, j2. Los otros 3 parámetros se recomienda estén en 1, dado que los dos primeros habilitan la ejecución de módulos de software internos en LDestimate, y el último habilita la salida gráfica. Solamente en LDestimatev2 se anula la gráfica para poder realizar cientos de veces el software para buscar puntos óptimos en las matrices generadas.

function

[alphaest,cfCest,Cest,Q,j1opt,yj,varj,HLRD1,Hc,Qc,auxindice,intervalo,j 1dif,j2dif] = LDestimatev2(data,regu,j1,j2,discrete_init,calcj1,printout). Nombre programa: LDestimatev2.m

El cambio a este módulo reside en que las salidas de la función se obtienen por cada iteración de j1,j2 y número de desvanecencias para generar matrices de parámetro de Hurst, métricas de factor de calidad, matrices de j1 y j2, matrices de las diferencias entre j1 y j2, y el índice que resultó del cálculo del parámetro. A través de la determinación de auxindice (diferencia entre j1 y j2 mayor a 4, un H positivo, un Q alto, no así el mayor en forma estricta) se elige el parámetro H. Este módulo fue automatizado por el autor, aunque la versión original tiene los mecanismos para habilitar las iteraciones, las mismas no fueron utilizadas, sino que se creo un método propio.

Los módulos desarrollados por Hae-Duck Joshua de la Universidad de Canterbury[10], para generar un tráfico autosimilar basado en wavelets, utiliza módulos distribuidos según la siguiente estructura o árbol:



Figura 24. Estructura de módulos desarrollados por Hae-Duck Joshua

Tabla 9. Aplicaciones desarrolladas por Hae-Duck Joshua Jeong

Aplicaciones desarrolladas por Hae-Duck Joshua Jeong					
Módulos	Aplicación				
FGNDW.m	Genera un tráfico autosimilar mediante la transformada inversa Wavelet				
FGNDWSpectrum	Crea n frecuencia y calcula el espectro de frecuencia de la potencia.				
Daub	Retorna los coeficientes Wavelets de Deubechies				

• function [] = FGNDWv2(). Nombre programa: FGNDWv2.m

Este programa no se utiliza directamente en el software diseñado pero se utiliza para generar un vector a través de coeficientes Wavelets u ondículas. Con la transformada inversa Wavelet se generan las muestras de tráfico autosimilar. Esta versión solo introduce cambios en la versión original FGNDW.m en las entradas de parámetros de la función. Las mismas se solicitan al usuario tal como se especifica a continuación:

El fundamento matemático para la generación de números aleatorios está basado en el siguiente fundamento matemático de la transformada Wavelet:

$$f(\lambda, H) = cf|\lambda|^{1-2H} + O(|\lambda|^{\min(3-2H)/2)}) (48)$$

$$cf = \sigma^{2} (2\pi)^{-1} \sin(\pi H) gamma(2H+1) (49)$$

Dicha ecuación tiene como origen la relación del espectro de frecuencia de la potencia con la frecuencia y el parámetro de Hurst:

$$S(w) = c|w|^{-\gamma}$$

$$w \to 0$$
(50)

Una vez realizado los cálculos, a través de la transformada inversa de Fourier se obtienen las muestras.

En realidad, la transformada Wavelet no lleva la señal a una representación de integración por una función exponencial que representa la variable en frecuencia, cuya característica de ortogonalidad con la función en el tiempo, hace el efecto que la componente en frecuencia de la señal coincida con sus componentes armónicos de mayor potencia. En este caso, el del Wavelet, la función en el tiempo es multiplicada por otra señal del dominio del tiempo (es multiplicada varias veces, haciéndose el barrido con diferentes Wavelets provenientes de una Wavelet madre), pero que en si, a través de los escalamientos y los desvanecimientos, representa ese componente de frecuencia, y por tanto los Wavelet irán coincidiendo, por la característica de ortogonalidad con la micro estructura de la función del tiempo. En el caso de los Wavelet, un factor alto de compresión representará los componentes de baja frecuencia, mientras que uno bajo representa los altos componentes frecuenciales. A continuación están las líneas de código para solicitar la entrada a lo que originalmente era a través de parámetro de la función.

clc

00000000

display ('Se generará una matriz con n muestras, escalas de compresión y coeficientes Wavelets indicados')

n=10000000:

H=input('Número de Hurst:');

Scale=input ('Número de escala entre 1 y 16 ');

num coefs=input ('Numero de desvaneciencias del Wavelet :');

Las muestras provenientes de la transformada inversa Wavelet se almacenan en el vector sintético x.txt. Es necesario tener a consideración que al ejecutar este software externo, el vector sintético será el que corresponde al Wavelet y no la Transformada Inversa de Fourier, la cual si se encuentra el software orientado a objetos. Las siguientes líneas de código se insertaron para grabar el vector generado en x.txt, y poder manipularlo en el software orientado a objeto.

SS = real(iwt(z,rh,rg,Scale)); x=SS; save x.txt x - ascii; save H.txt H - ascii;save n.txt n - ascii;

> function FGNS = FGNDWSpectrum(lambda,H). Nombre del programa FGNDWSpectum. m

Lambda representa las frecuencias de las n muestras que se generarán. Este programa es pequeño. Las líneas de código fueron incluidas en el programa FGNDW.m, en la línea donde se hace el llamado a la rutina.

• function [h,g,rh,rg] = Daub(VanishingMoment). Nombre programa: Daub.m Retorna los coeficientes Wavelets Daubechies

Tabla 10. Aplicaciones .m desarrolladas

Aplicaciones .m desarrolladas por Franklin Planchart					
Módulos	Aplicación				
validacion_metodo.m	Calcula las varianzas de los vectores sintéticos activados para validar				
Activa_Validacion.m	Activa la validación del software a través de vectores sintéticos autosimilares				
ResumenH.m	Resumen de los factores H de todos los archivos .txt seleccionados.				

• Nombre del programa: Validacion_metodo.m function[val]=validacion_metodo(matriz)

líneas de código del programa:

00000000000

%Rutina de validación que compara los H generados sintéticamente con el %método de Fourier y los compara con las distintas formas de calcular H %En la fila final de la matriz val, se encuentran las desviaciones estándar %de las diferencias.

```
[f c]=size(matriz);
matriz2(1:f-1,1:c)=0;
matriz2=matriz(2:f,1:c);
val(f,1:c-5)=0;
for I=1:f-1
    val(I,I)=matriz2(I,I);
end
for J=2:c-4;
    acum=0;
for I=1:f-1;
    val(I,J-1)=((matriz2(I,J)-matriz2(I,1))^2)^.5;
    acum=acum+val(I,J-1);
end
val(f,J-1)=acum/(f-1);
save val.txt val -ascii
end
```

Observe que el archivo que compara todos los sintéticos, y el cual calcula desviación estándar de cada uno de los métodos con respecto al valor H con el cual se generó el vector, se guarda en val.txt. La entrada de esta función es "matriz".

La entrada "matriz" es proveniente del archivo matrix.txt. Simplemente cargue el archivo con load matrix.txt, y coloque la palabra matrix cuando llame la función.

En todo caso, un procedimiento sencillo ha sido activado para ahorrar pasos al investigador.

Esta función fue insertada en el módulo orientado a objeto, específicamente en autofractal_9, antecediéndole la respectiva activación para ejecutar este módulo. A partir de la línea 1203 se observa lo siguiente:

```
load validacion.mat
if strcmp(validacion,'si')
if strcmp(identif,'Vector autosimilar sintético');
load H.txt
matrix=[matrix;H m1 m2 m3 m4 m5 m6 m7 m8 m9];
end
```

save matrix.txt matrix -ascii

Seguido de la inserción de la función anterior en forma de continuación del programa y no como función.

Para asegurar que el módulo de almacenamiento de vectores sintéticos no quede activo, y destruya pruebas históricas generadas por el usuario, se procedió a bloquear la validación en el inicio del programa autofractal_1.fig en la línea 119. validacion='no':

save validacion.mat validacion

000000000

clc; close all

_

La siguiente es la salida de function[val]=validacion_metodo(matriz)

Donde matriz=matrix. Se puede colocar directo matrix.

```
MATLAB 7.9.0 (R2009b)
                                                                                                               File Edit Debug Parallel Desktop Window Help
 🖺 😅 🐰 🐚 🧖 🤊 🌣 📸 📆 😢 Current Folder: C:\Users\Franklin Planchart\Desktop\Documents\MATLAB
                                                                                                                  - ... (2)
 Shortcuts # How to Add # What's New
  >> validacion metodo (matrix)
      0.0010
               0.0065
                         0.0435
                                  0.0014
      0.0028
                0.0096
                         0.0083
                                   0.0026
                                             0.0594
      0.0193
               0.0066
                         0.0078
                                   0.0005
                                            0.0825
      0.0214
                0.0071
                         0.0167
                                   0.0144
                                             0.0140
      0.0252
                0.0096
                         0.0003
                                   0.0150
                                             0.0156
                0.0026
                         0.0084
      0.0011
                                   0.0279
                                             0.1045
      9.0200
                0.0292
                         0.0577
                                   0.0163
                0.0077
      0.0194
                         0.0283
                                   0.0313
                                             0.0045
      0.0123
                         0.0266
                                   0.0304
                                             0.0102
      0.0084
                0.0140
                         0.0349
                                   0.0024
                                             0.0337
      0.0152
                0.0088
                         0.0185
                                   0.0045
                                             0.0269
      0.0085
      0.0128
                0.0095
                         0.0255
                                   0.0703
                                             0.0497
      0.0047
                0.0188
                         0.0463
                                   0.0507
                                             0.0286
      0.0263
                0.0331
                         0.0804
                                   0.0174
                                             0.0153
                         0.1154
                                   0.0222
                                             0.0231
      0.0129
                0.0297
                         0.0391
      0.0386
                0.0875
                         0.1383
                                   0.0195
      0.0432
                         0.1036
                0.0652
                                  0.0121
                                             0.0193
      0.0825
                0.0947
                                  0.0427
                                             0.0345
      0.0224
              0.0270 0.0474
                                  0.0227
                                            0.0418
♠ Start
```

Figura 25. Resultado ejecución "validacion_metodo.m, function[]= validación metodo(matriz)"

Por ello, una vez que UD. entra al programa autofractal_1 y presione continuar, una protección ha sido activada y solo se puede romper desde afuera ejecutando:

```
Activa_validación.m. Código:

clc

disp('Después de activar, debe realizar las pruebas en la opción ')

disp ('Análisis múltiple, pruebas a un vector, Resumen H')

disp('allí se encuentra la rutina que almacena los vectores sintéticos')

disp('FGNIFourier que se irán generando.')

disp('Este programa tiene efecto solo si el programa autofractales ha sido')

disp('ejecutado')

%disp('La corrida anterior se guardará siguiendo el siguiente formato:')

%disp('valañomesdiahoramin.txt')

resp=input('Activar validacion con vectores sintéticos?si/no:','s');

if strcmp(resp,'si');

resp
load val.txt;
```

```
load matrix.txt
  %input('aja', 's')
  %matrix=0:
  %En matrix se guardarán todas las corridas que se hagan de pruebas
  %de vectores sintéticos creados con FGNI. La primera columna
  %guarda el valor de H y a continuación todos los tipos de H.
  %siguiendo el orden de los históricos
  %La última fila guarda el cálculo de la desviación Standard de los resultados
  % con respecto al valor H con que fue generada la muestra.
  matrix=0
  matrix(1,1:10)=0;
  save matrix.txt matrix -ascii
  validacion='si':
  save validacion, mat validacion
end
clc
```

Como se observa, borra la base histórica matriz, e inicializa el proceso. Se recomienda entonces salvar los históricos con el formato siguiente:

```
Save matrixco20xxyyzz.txt matrix
Save val20coxxyyzz.txt val
```

Donde co es un correlativo de dos dígitos, xx es el dia, yy es el mes y zz el año. Salve los históricos dado que ello es tiempo que se estuvo laborando en la máquina. Futuras versiones pueden automatizar la generación de múltiples vectores y probarlos, pero como este es un módulo para validar este software, no se profundizó en dicho esfuerzo.

En el caso de ser muestras de vectores provenientes del análisis con Wavelet, coloque una "w" que significa Wavelet:

```
Save matrixwco20xxyyzz.txt matrix
Save valw20coxxyyzz.txt val
```

Cuando se desee validar el software, se recomienda ejecutar los siguientes pasos:

- 1.- Ejecute autofractal_1 y presione continuar.
- 2.- Salga al modo comandos y ejecute Activa_validacion.m.

Este módulo borra el histórico de vectores sintéticos y el usuario entra a generar nuevos vectores.

- 3.- Concluida la generación de vectores sintético, diríjase al menú de análisis múltiple de tráfico
- 4.- Seleccione el menú: todas las pruebas a un vector
- 5.- Seleccione el menú de resumen de pruebas a un vector. Allí reside el módulo de Validacion_metodo.
- 6.- Genere un nuevo vector sintético
- 7.- Vuelva al paso 3.

00000000000000

8.- Concluido todos los vectores, vuelva al nivel de comandos y ejecute: load val.txt. y load matrix.txt. Está disponible "val" y "matrix. En matrix Ud. observará todos los cálculos de H y en la última columna el valor H con el cual se creó el vector. Los otros son los valores H, con los diferentes métodos. En val observará el cálculo de las desviaciones estándar con respecto al H con que se generó el vector sintético.

En el caso de que UD esté generando vectores sintéticos y ejecutándoles las pruebas, y desee seguir en otro momento, puede salir del programa. Cuando lo abra de nuevo, no ejecute Activa_Validacion. Ejecute nuevamente autofractal_1.fig presione continúe y diríjase al nivel de comandos en Matlab. Ejecute los siguientes comandos:

```
validacion='si';
save validacion.mat validacion
```

Cuando se generan vectores sintéticos desde el programa autofractal_1.fig, solo existe la opción de generarlos con el método de la Transformada de Fourier, a través de la teoría de periodogramas. Si se desea probar con vectores sintéticos generados con la transformada Wavelet, debe ejecutar desde el nivel de comandos el programa FGNDW2.m, el cual es una versión modificada de FGNDW.m, y guarda el vector generado en x.txt en formato ASCII. Los otros pasos son iguales. Al finalizar las pruebas, salve el archivo en un histórico que solo el usuario puede borrar desde el nivel de comandos de Matlab o desde el sistema operativo.

El programa ResumenH.m, es una rutina sencilla que permite representar desde el nivel de comandos en Matlab las salidas del resumen del cálculo de H para diferentes muestras a las cuales se les realizó el cómputo masivo en el menú "Análisis Múltiple". A continuación se coloca el código:

```
clc
  load fileNamemat.mat
  load mattiempo.txt
  load matbyte.txt
  fprintf('**********
                           ********Resumen de los cálculos de los análisis de las
muestras seleccionadas ***
  fprintf('R/S
                 RS/Prom
                               V/T
                                       WLDEest
                                                   WE/Octava
                                                                          JI
                                                                                J2
                                                                                      N
Nombre de la muestra n')
  [fx,cx]=size(fileNamemat);
  [fxt,cxt]=size(mattiempo);
  [fxb,cxb]=size(matbyte);
  fprintf('\n')
  fprintf ('****Análisis: Tiempo entre llegadas****')
  for I=1:fx
 fprintf(\n%1.4f %1.4f
                              %1.4f
                                        %1.4f %1.4f
                                                            %1.4f
                                                                      %d
                                                                              %d
                                                                                      %d
', mattiempo(I, 1), mattiempo(I, 2), mattiempo(I, 3), mattiempo(I, 4), mattiempo(I, 5), mattiempo(I, 5)
(6), mattiempo(I,7), mattiempo(I,8), mattiempo(I,9)
  fprintf(fileNamemat(I, 1:cx), 'n')
  end
```

```
fprintf(\n\n') \\ fprintf('****An\'alisis: Cantidad de bytes por paquetes****') \\ for I=1:fx \\ fprintf('\n\%1.4f \%1.4f \%1.4f \%1.4f \%1.4f \%1.4f \%d \%d \%d ',matbyte(I,1),matbyte(I,2),matbyte(I,3),matbyte(I,4),matbyte(I,5),matbyte(I,6),matbyte(I,7), \\ matbyte(I,8),matbyte(I,9)) \\ fprintf(fileNamemat(I,1:cx),'\n') \\ end
```

En la figura 30 se observa la salida generada a nivel de comandos en Matlab.

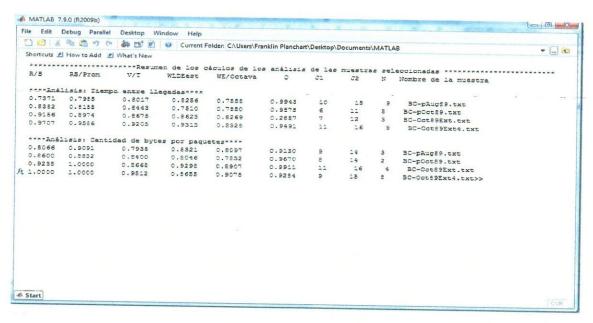


Figura 30. Resultado ejecución ResumenH.m

6.- Estructura del software con interfaz gráfica.

El software con interfaz gráfica al usuario une todos los programas .m para que dichos módulos sean llamados sin necesidad de conocer los detalles de las funciones o para ejecutar pruebas que son repetitivas, evitando tener que ejecutar módulo por módulo. La otra importancia es que coloca módulos de diferentes autores, y realiza adaptaciones de interés en algunos módulos, perfeccionando la salida.

Para ello, todos los módulos fueron adaptados para trabajar con dicha interfaz. El módulo LDEstimate fue modificado a fin de automatizar el proceso de iteración de octavas, el número de desvanecimientos del Wavelet y el parámetro de calidad del segmento seleccionado.

En la figura 31 se encuentra una descripción general de los módulos desarrollados en extensión .fig, encargados de llamar a las rutinas .m modificadas, adaptadas y optimizadas en algunos casos, manteniendo el núcleo del algoritmo de dichos módulos.

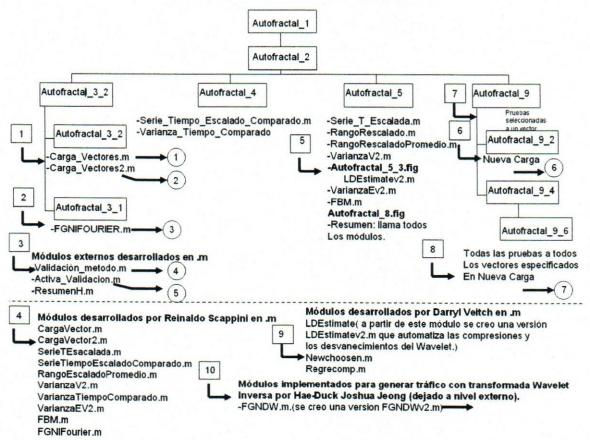


Figura 31. Estructura del Software de Análisis y Simulación de Tráfico Autosimilar

Módulos autofractales

La gráfica anterior se refleja en el software implementado en esa misma estructura, haciendo los llamados a las rutinas mencionadas anteriormente. En algunos casos, dada la incompatibilidad gráfica muy fuerte, se tuvo que copiar de nuevo el código, con la finalidad de poder cambiar la estructura del gráfico. Por eso, eso existe un sacrificio en la modularidad, por la imposibilidad de introducir mensajes de control del software y del usuario, y evitar errores de ejecución. Ejemplo, las dificultades en enviar mensajes al usuario de que una aplicación está en ejecución y debe esperar. Las rutinas de Callback en Matlab no aceptan entrar, colocar un mensaje y salir a ejecutar. Las cajas gráficas de interfaz deben ser cuidadosamente colocadas y manipuladas teniendo a consideración estos criterios.

La primera versión se realizó sin los controles a usuario, pero el software estaría destinado a no ser utilizado, dado que hay rutinas que son de larga duración y es necesario ir advirtiendo al usuario. Luego se introdujeron las cajas automáticas de control pero salvando estos obstáculos en los archivos con extensión .fig. En algunos casos se insertaron las cajas de control dentro de los mismos módulos .m.



Figura 32. Autofractal_1.fig. Pantalla inicial del software de Análisis y Simulación de Tráfico Autosimilar

Autofractal_3_1.fig:

Bifurca a dos rutinas: autofractal_3_1.fig y autofractal_3_2.fig.

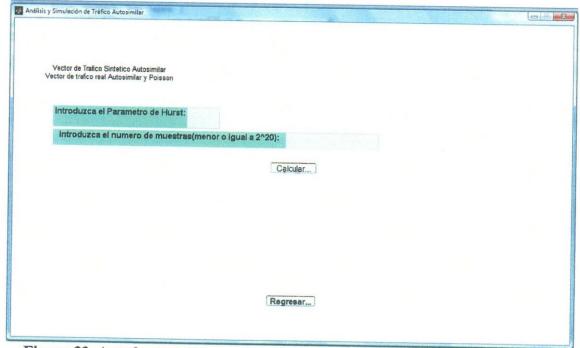


Figura 33. Autofractal_3_1.fig. Generación de vector autosimilar sintético con FGNIFOURIER.m

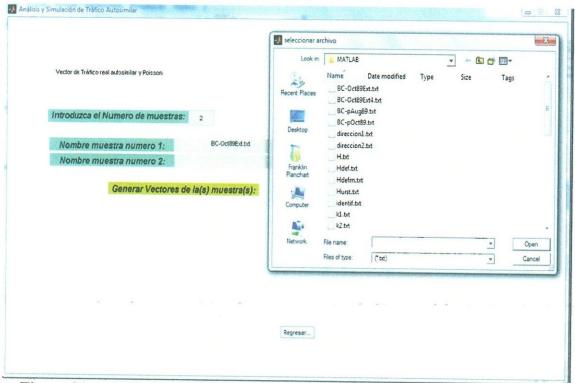


Figura 34. Autofractal_3_2.fig. Generación de vectores con las muestras 1 y 2.

De esta manera, la primera opción tiene dos bifurcaciones:

- -Generar un vector sintético (figura 33)
- -Extraer vectores de tamaño de los bytes que llegan y tiempo de llegadas (figura 34).
- Archivo.txt. Primera columna Tamaño bytes. Segunda columna tiempos de llegadas.
- -Vector sintético. Formato ASCII. Nombre: x.txt.

 Tamaño de muestra:n.txt. Parámetro de Hurts del vector sintético: H.txt

 -Muestra número 1 autosimilar:

vecs.txt ascii % Tamaño paquetes autosimilar save tieas.txt tiea –ascii % Tiempo entre llegadas autosimilar save xpoiss.txt xpois –ascii % Tamaño paquetes Poisson save tieaps.txt tieap –ascii % Tiempo entre llegadas Poisson

-Muestra número2:
Vecs2.txt ascii % Tamaño paquetes autosimilar
save tieas2.txt tiea –ascii % Tiempo entre llegadas autosimilar
save xpoiss2.txt xpois –ascii % Tamaño paquetes Poisson
save tieaps2.txt tieap –ascii % Tiempo entre llegadas Poisson

2 Se Introduce el número de muestras y el parámetro de Hurts

-Vector sintético. Formato ASCII. Nombre: x.txt.
Tamaño de muestra:n.txt. Parámetro de Hurts del vector sintético: H.txt

Variable "validación" tiene dos valores: "si", "no".

matrix.txt: matriz de diez columnas y n filas. Columnas 1 al 5 almacena parámetros
H con cada uno de los métodos en el siguiente orden: Varianza Tiempo, Rango Escala,
Rango escala Promedio, LDestimate, Varianza Wavelets octavas. Luego cuatro posiciones
para: factor de calidad, j1, j2, número de desvanecencias, y el valor H con el cual se generó
la muestra. Suposición: todas las muestras son de 1000000 de valores.

Variable "validación" (validacion.mat) tiene dos valores: "si", "no".

matrix.txt: matriz de diez columnas y n filas. Columnas 1 al 5 almacena parámetros
H con cada uno de los métodos en el siguiente orden: Varianza Tiempo, Rango Escala,
Rango escala Promedio, LDestimate, Varianza Wavelets octavas. Luego cuatro posiciones
para: factor de calidad, j1, j2, número de desvanecencias, y el valor H con el cual se generó
la muestra. Suposición: todas las muestras son de 1000.000 de valores.

val.txt: matriz de cinco columnas por n filas. Hasta la fila (n-1) contiene el cálculo de las
Varianzas con respecto al valor H con el cual se generó el vector sintético. La última fila
contiene la desviación estándar de cada método.

autofractal 4.fig:

Los ejemplos de las salidas de este módulo se han colocado en forma intencional para comparar un tráfico autosimilar y otros de Poisson. Observe las diferencias. Estas comparaciones tienen un valor científico, pero en el mediano plazo busca es una fijación, razonamiento y discusión en lo académico. Se pueden "observar" las semejanzas estadísticas en el tráfico autosimilar, y la ausencia de ráfagas en el tráfico Poisson, en la medida que se aumenta la escala temporal. Igualmente, en el método varianza tiempo, una inclinación de la línea recta hacia la izquierda es sinónimo de aumento del parámetro H, cuyo límite es 1.

Seleccione las	muestras a comparar
Muestra numero uno	Muestra numero dos
Vector Sintetico Autosimilar	Vector Sintetico Autosimilar
BC-Oct89Ext.txt_Autosimilar	○ BC-Oct89Ext4 txt_Autosimilar
BC-Oct89Ext.txt_tipo Poisson por MLE	○ BC-Oct89Ext4 txt_tipo Poisson por MLE
Procedimientos de comparacion	
Rango Escalado Comparado cantidad de Bytes	
Varianza Tiempo Comparado cantidad de Bytes	Ejecutar Comparacion
Varianza Tiempo Comparado cantidad de Bytes Rango Escalado Comparado tiempo entre llegadas	Ejecutar Comparacion
Varianza Tiempo Comparado cantidad de Bytes	Ejecutar Comparacion
Varianza Tiempo Comparado cantidad de Bytes Rango Escalado Comparado tiempo entre llegadas	Ejecutar Comparacion

Figura 35. Autofractal_4.fig. Comparación con métodos Escala/Tiempo y Varianza/Tiempo de las dos muestras y el vector sintético

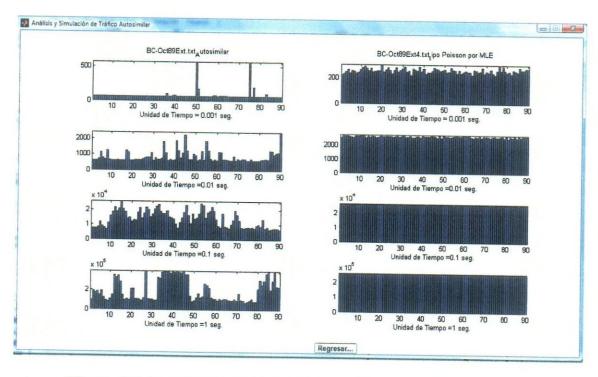


Figura 36. Autofractal_4.fig. SerieTiempoEsacaladoComparado.m

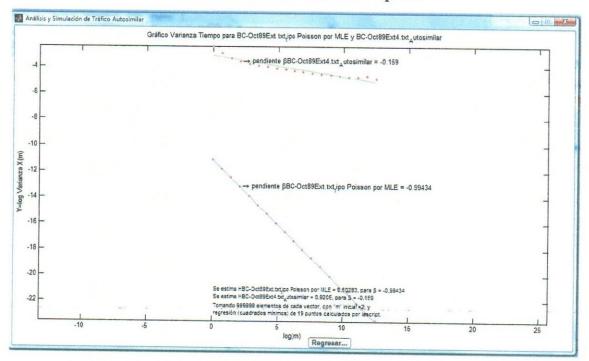


Figura 37. Autofractal_4.fig. VarianzaTiempoComparada.m

autofractal 5.fig

Las entradas a este módulo están basadas en los vectores generados. El único módulo que requiere mayor cantidad de entradas en LDestimate, al cual se dan las respectivas opciones para asegurar los parámetros de entrada. Los mismos son: el número de coeficientes y el nivel de compresión del Wavelet. El módulo LDestimate es llamado sucesivamente para todas las iteraciones posibles entre un J1 hasta un J2 máximo igual a 16. El usuario debe seleccionar el número de desvaneciencias. En la misma forma se da la opción de que el usuario observe todas las iteraciones en forma gráfica, y tiene gran importancia académica para observar lo que realmente realiza el módulo.

Archivos.txt, hasta un millón de muestras filtradas con analizador de protocolos. Muestra 1, Muestra 2 y Vector Sintético.

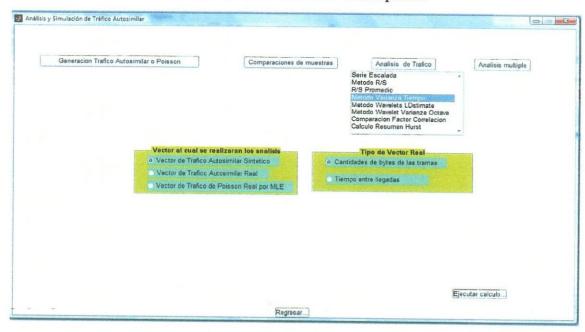


Figura 38. Autofractal_5.fig. Análisis de Tráfico individual de una muestra paso a paso. Cada prueba implica carga del vector.

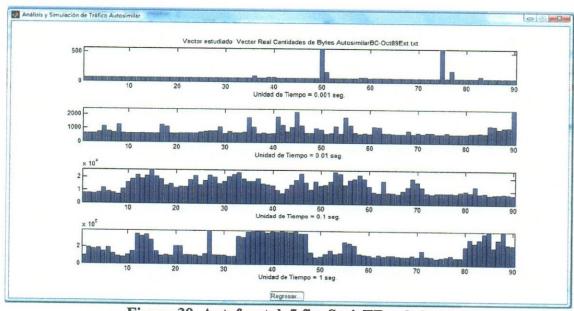


Figura 39. Autofractal_5.fig. SerieTEscalada.m

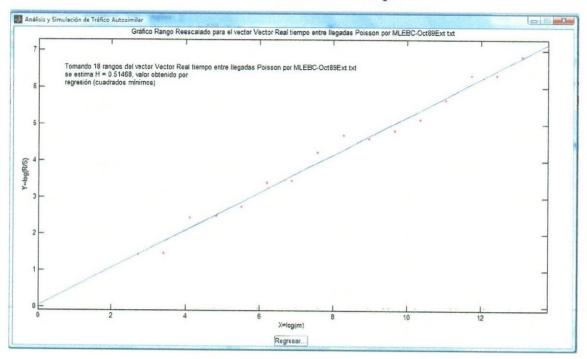


Figura 40. Autofractal_5.fig. RangoRescalado.m para una muestra tipo Poisson por MLE

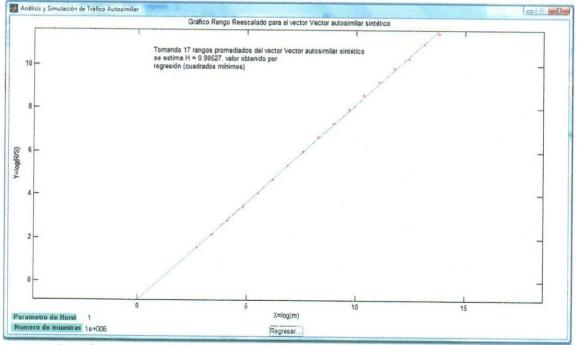


Figura 41. Autofractal_5.fig. RangoRescalado.m para un vector sintético

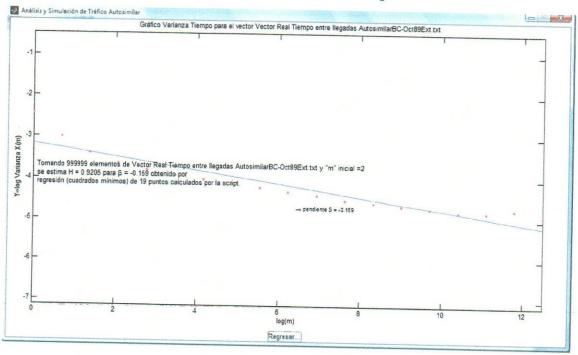


Figura 42. Autofractal_5.fig. VarianzaTiempo.m para una muestra real

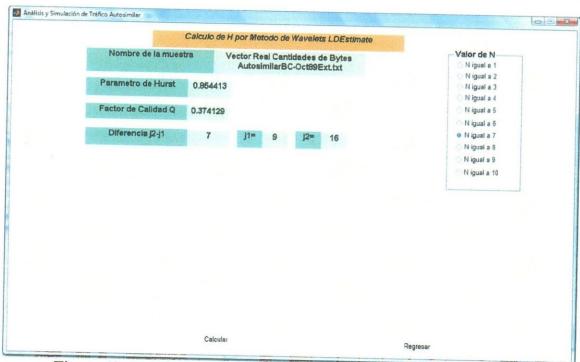


Figura 43. Autofractal_5.fig. LDestimatev2.m para una muestra real

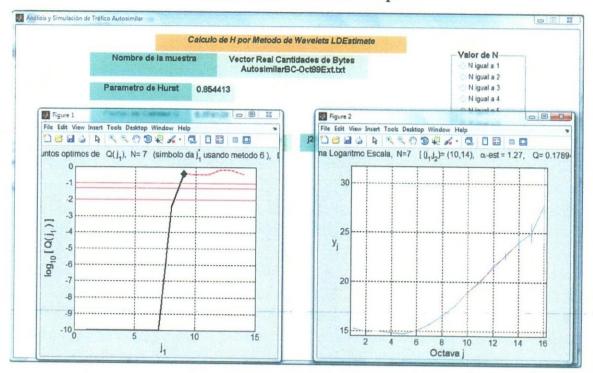


Figura 44. Autofractal_5_3.fig. LDestimatev2.m para una muestra real. Gráficos de Q y Varianza Coeficientes Wavelets

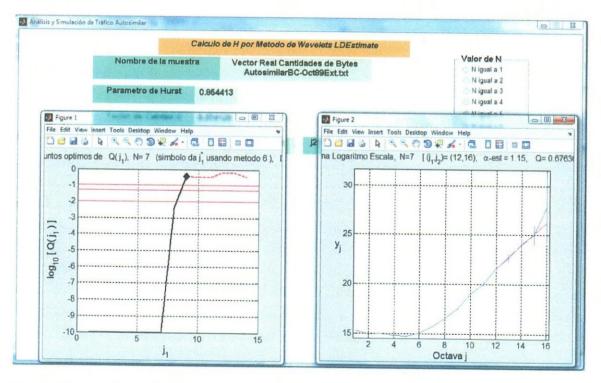


Figura 45. Autofractal_5_3.fig. LDestimatev 2.m para una muestra real. Gráficos de Q y Varianza Coeficientes Wavelets en un instante "t2".

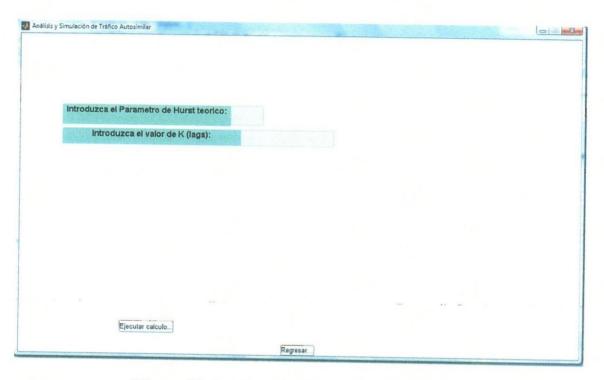


Figura 46. Autofractal_5.fig. Aplicación FBM.m

La figura 22 muestra la opción de generar un vector con la propiedad de la transformada de la correlación y compararla con la opción que UD. haya seleccionado: un vector sintético o una muestra.

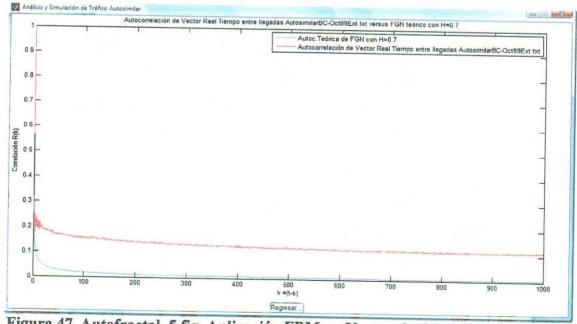


Figura 47. Autofractal_5.fig. Aplicación FBM.m. Vector sintético H=0.7 y muestra real BC-Oct89Ext.txt

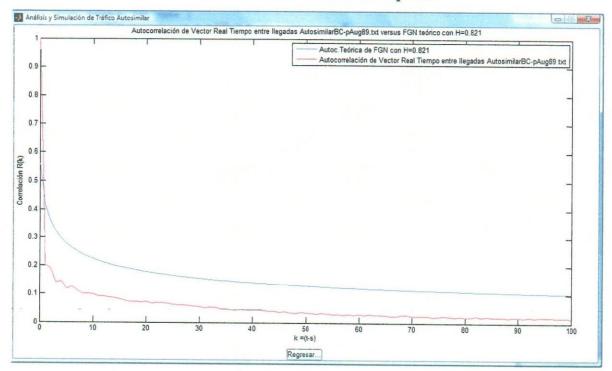


Figura 48. Autofractal_5.fig. Aplicación FBM.m con vector sintético de H=0.821 y muestra real BC-pAug89.txt

Seleccionando en "Análisis de tráfico" la opción "Comparación Factor de Correlación", se puede verificar el comportamiento de los generadores de tráfico FGNIFOURIER y FGNDWv2. Se seleccionará para ello parámetros H: 0.6 y 0.9.

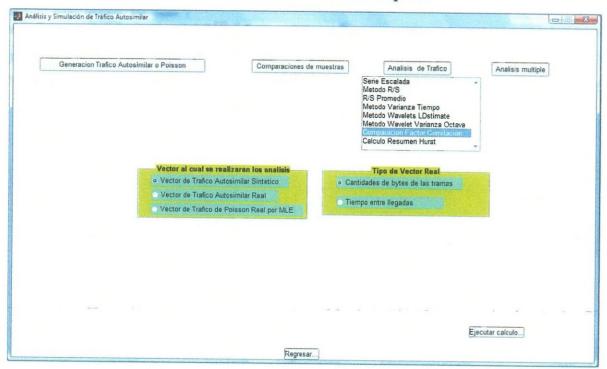


Figura 49. Autofractal_5.fig. Aplicación FBM.m

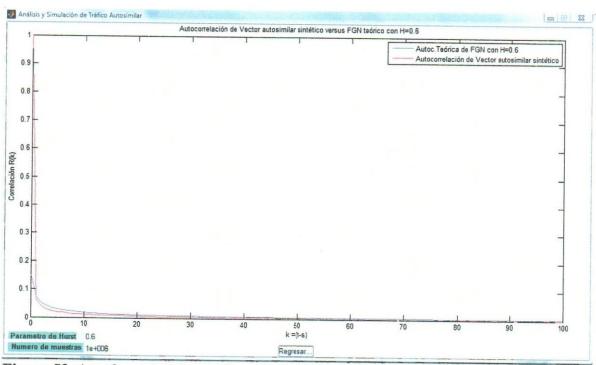


Figura 50. Autofractal_5.fig. Aplicación FBM.m a Vector sintético FGNIFOURIER.m de H=0.6 comparado con auttocorrelación teórica de 0.6

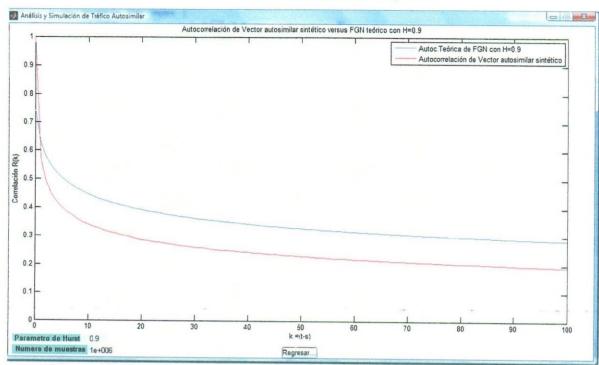


Figura 51. Autofractal_5.fig. Aplicación FBM.m a un vector sintético tipo FGNIFOURIER.m con H=0.9 y autocorrelación teórica con H=0.9

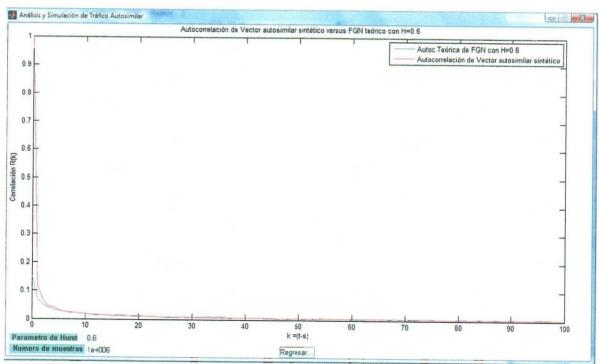


Figura 52. Autofractal_5.fig. Aplicación FBM.m a un vector sintético tipo FGNDWv2.m con H=0.6 y autocorrelación teórica con H=0.6

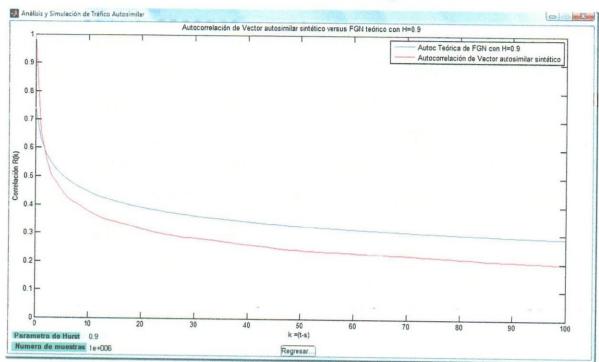


Figura 53. Autofractal_5.fig. Aplicación FBM.m a un vector sintético tipo FGNDWv2.m con H=0.6 y autocorrelación teórica con H=0.9

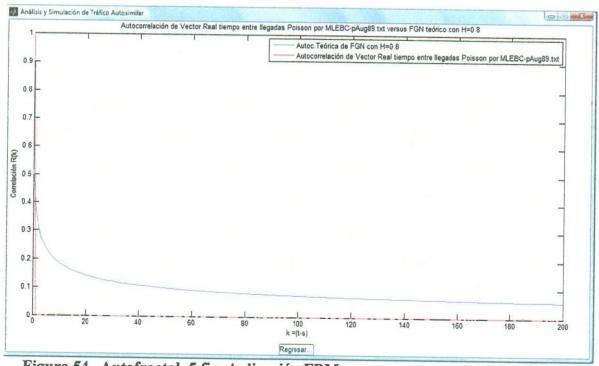


Figura 54. Autofractal_5.fig. Aplicación FBM.m a una muestra real tipo Poisson generada por MLE y autocorrelación teórica con H=0.8

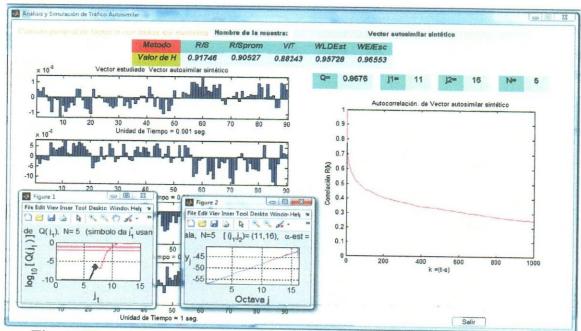


Figura 55. Autofractal_5. Llamado a autofractal_8.fig. resumen de todas las aplicaciones. Desactivación de los gráficos para cada método de H

La opción de Resumen ejecutará todos los módulos, activando los gráficos al final, solo los que son convenientes para un resumen.

autofractal_9.fig

La opción "Análisis múltiple" bifurca a dos acciones principales y una de soporte.

Las acciones principales son dos:

- Seleccionar Pruebas a un vector
- Ejecutar todas las pruebas a múltiples archivos seleccionados previamente

La ventaja de éste módulo, es que los vectores se cargan una sola vez y se le hacen todas las pruebas.

En el caso de la primera opción, permite obtener la salida gráfica por cada método seleccionado. En esta opción se encuentra la posibilidad de ejecutar igualmente el resumen, y si fue activada el proceso de validación, con los métodos explicados en validacion_metodo.m, los resultados de los cálculos H en los vectores sintéticos, se almacenarán en matrix.txt, y el cálculo de las varianzas con respecto a los vectores sintéticos almacenados en matrix.txt se guardarán en val.txt La segunda opción permite ejecutar el cálculo de todos los parámetros H y los guarda en los archivos explicados anteriormente. UD puede realizar nuevas aplicaciones estadísticas con dichos archivos. Sobre todo, es necesario realizar una aplicación que manipule "n" históricos generados. Esa sería una aplicación de gran interés en una investigación de tráfico.

Se ejecutan múltiples pruebas a un vector. Las entradas son las muestras grabadas en muestra 1, en lo que respecta al vector tamaño bytes, el vector tiempo entre llegadas

(autosimilar y Poisson por MLE, Maximum Likelikood Estimation) y el vector sintético autosimilar. Se ejecuta validación de vectores sintéticos, cuando se activa, y se almacenan en matrix.txt y val.txt. Refiérase a las figuras 56 y 57 para detallar la interfaz de usuario y la forma de selección de todas las pruebas a un vector.

Por otro lado, la opción de soporte es la referida como "nueva carga" y es donde se seleccionarán los archivos para la extracción de los vectores respectivos de tamaño de bytes y tiempo entre llegadas de paquetes. Hasta 50 archivos pueden ser seleccionados.

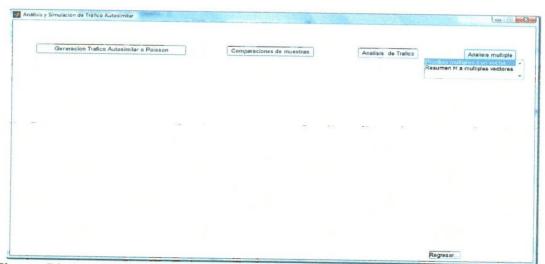


Figura 56. Autofractal_9.fig. Análisis múltiple para una muestra. Una sola carga.

Análisis múltiple para "n" muestras.

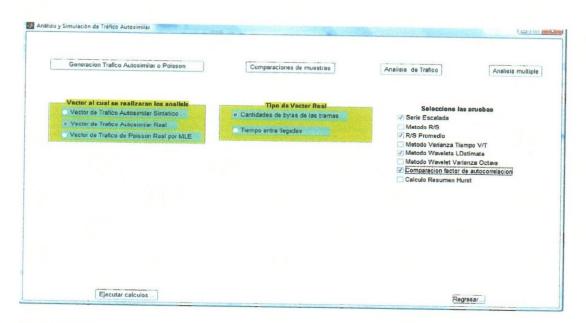


Figura 57. Autofractal_9.fig. Todas las pruebas seleccionadas a un vetor o muestra

7 Entradas de vectores sintéticos o muestra seleccionada.

autofractal 9 2.fig

El menú de soporte es la opción "Nueva carga". Permite seleccionar los archivos .TXT a los que se realizará el cálculo. Aparece cuando selecciona "Resumen H a múltiples vectores". Ud puede revisar los históricos.Pero si selecciona esta opción, los borrará.

- Archivos.txt, hasta un millón de muestras por archivo. Máximo 50 archivos filtrados con analizador de protocolos. Idealmente paquetes IPv4 o IPv6.
- 6 Se graban los nombres en fileNamemat.mat FileNamemat

A continuación se visualizan todas las ventanas. Los archivos se irán seleccionando uno a uno. Inicialmente el sistema advertirá que la nueva carga destruirá el archivo que mantiene los nombres del histórico. Salve primero los históricos de los archivos, que se encuentran en fileNamemat.mat, mattiempo.txt, matbyte.txt, siguiendo la nomenclatura nombreañomesdía, como se explicó en anteriormente.

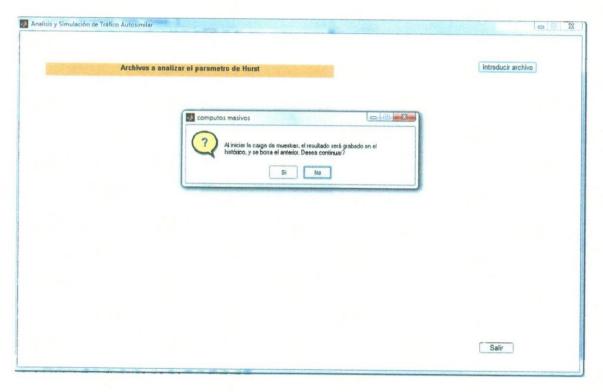


Figura 58. Autofractal_9_2.fig. Opción de nueva carga.

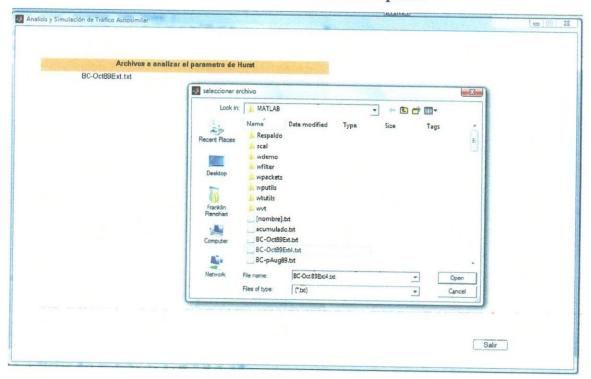


Figura 59. Autofractal_9_2.fig. Selección de los archivos .txt en nueva carga Una vez culminada la carga, se habilita la opción de "cómputo masivo" Ver figura 60.

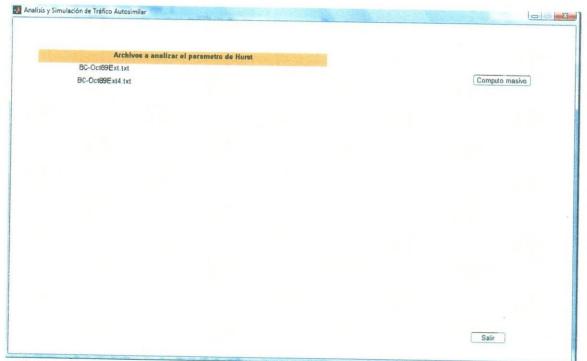


Figura 60. Autofractal_9_2.fig. Ejemplo de dos archivos.txt seleccionados. Puede seleccionar hasta 50 archivos

El menú de Resumen H a múltiples vectores va a tener como entrada la posibilidad de seleccionar archivos de muestra .TXT, por supuesto, filtradas de una gran muestra a través de un analizador de protocolos de comunicaciones. El límite es hasta 50 archivos.

El menú de Resumen H a múltiples vectores, desplegará los cálculos en forma iterativa a un archivo. Es decir, primero el cálculo de H al vector tamaño bytes y luego el vector tiempo entre llegadas. Al finalizar el cálculo de los "n" archivos, hasta un límite de 50, preguntará si UD. desea ver el histórico, dando dos opciones: observar archivo por archivo u observar el total. El archivo queda almacenado. UD. puede salir del menú y volver a examinar el histórico generado. Recuerde: salve el histórico desde el nivel de comandos antes de proceder a una nueva carga. A continuación se muestran las entradas y salidas para esta opción, seguida de las respectivas ventanas de interfaces gráficas.

autofractal_9_4.fig

Se ejecutan múltiples pruebas a múltiples vectores. Se llama a todos los módulos .m, originales y modificados.

- fileNamemat.mat son lo nombres de los archivos a los que se realizará el cómputo masivo. Fueron seleccionados en el menú "Nueva Carga".
- matbyte.txt y mattiempo .txt. contienen todos los parámetros H generados por los diferentes métodos, a los vectores seleccionados en el menú "Nueva Carga".

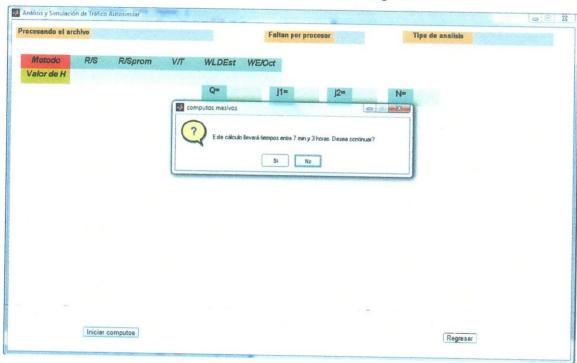


Figura 61. Autofractal_9_4.fig. Inicio de las pruebas múltiples. Se advierte el tiempo que durará la prueba

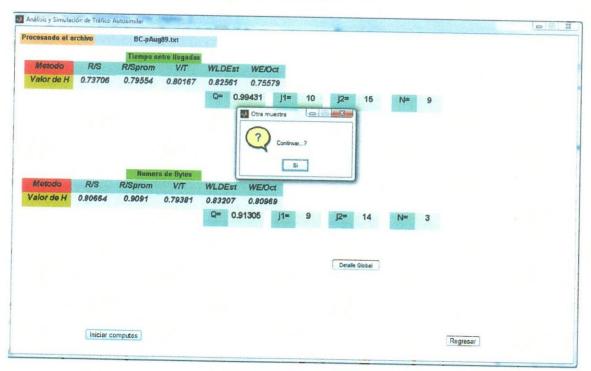


Figura 62. Autofractal_9_4. Ejemplo de fin de los cálculos a todas las muestras. Las mismas están grabadas. Puede ver una a una o ver las tablas generales.

Una vez que UD. seleccione la opción de "ver los registros uno a uno", el sistema pregunta si desea hacer el barrido nuevamente. Esta aplicación en realidad tiene un carácter histórico, por el resultado de las investigaciones de cómo manejar tablas en Matlab. Esta facilidad se descubrió posteriormente, dada la dificultad inicial en manejar los índices de los títulos. Pareció de interés dejarla por la forma de mostrar la información al usuario.

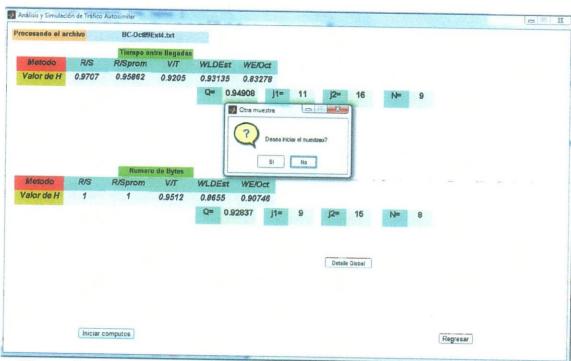


Figura 63. Autofractal_9_4.fig. Opción de nuevo barrido a los registros de las muestras

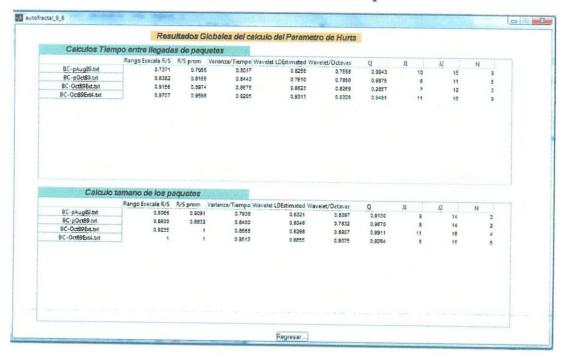


Figura 64. Autofractal_9_6.fig. Formato de los cálculos masivos en tablas

Detalles de entradas 4, 9 y 10

- Nombre del archivo.txt
- Nombre del archivo.txt, número de inicio de compresión, número final de compresión, número de desvanecimiento del Wavelet.
- Número de muestras, número de desvanecimiento, parámetro de Hurst.

7.- Validación del software

El procedimiento de validación del software se realizará en esta primera instancia, mediante la siguiente metodología:

- -20 muestras de cálculo con cada uno de los métodos de para determinar H por FGNIFOURIER (Fourier).
- -20 muestras de cálculo con cada uno de los métodos para determinar H por FGNDWv2 (wavelet).
- -4 muestras Bellcore Morristown Research and Engineering, las mismas utilizadas en el trabajo de W. E. Leland and D. V. Wilson[1] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the self-similar nature of ethernet traffic (extended version)", IEEE/ACM Transactions on Networking, vol.2, pp.1–15, Feb. 1994.. Estas trazas

ampliamente conocidas y mencionadas en muchas de las publicaciones hechas sobre este tema, están disponibles en el siguiente sitio llamado The Internet Traffic Archive:

http://ita.ee.lbl.gov/html/contrib/BC.html. En particular se trata de cuatro capturas de tráfico, llamadas BC-pAug89.TL; BC-pOct89.TL; BCOct89Ex,TL y BC-Oct89Ext4.TL. La descripción detallada de las trazas, se pueden ver, tanto en la página del link, como en el trabajo citado.

-Vector acumulado de tamaño de bytes proveniente de un cálculo sobre BC-Aug89.TXT. Nota: todos los archivos .TL deben ser copiados en .TXT o .dat para que el software los pueda tratar.

-Vector fgn8.dat, colocado por Darryl Veitch para realizar pruebas de validación.

7.1.- Validación con vectores sintéticos FGNIFOURIER:

Se generaron 20 muestras y se calcularon los parámetros H con todos los métodos, desde el menú "Análisis Múltiple", Pruebas a un vector. Se activó la validación con el programa: Activa_validacion.m.

Tabla 11. Cálculo de H com los cinco métodos para vectores FGNIFOURIER.m

FGNI	R/S	R/S Prom	V/T	LDestimate	WLog/Esc	Q	j1	j2	N
0.5000	0.5383	0.5256	0.5102	0.5068	0.5714	0.9955	6.0000	11.0000	5.0000
0.5800	0.5607	0.5699	0.5459	0.5502	0.6383	0.9828	4.0000	9.0000	8.0000
0.5500	0.5272	0.5451	0.5151	0.5304	0.6553	0.9997	3.0000	14.0000	9.0000
0.5900	0.5742	0.5757	0.5568	0.5722	0.7040	0.9910	6.0000	11.0000	5.0000
0.6000	0.5965	0.5923	0.5765	0.5986	0.6215	0.9994	10.0000	16.0000	6.0000
0.6500	0.6363	0.6461	0.6022	0.6811	0.6373	0.9742	11.0000	16.0000	7.0000
0.6800	0.6909	0.6602	0.6447	0.6675	0.6493	0.9787	7.0000	12.0000	9.0000
0.6900	0.6985	0.6788	0.6650	0.6593	0.6824	0.9292	5.0000	10.0000	8.0000
0.7000	0.6663	0.6736	0.6646	0.6915	0.7483	0.9803	7.0000	16.0000	4.0000
0.7500	0.7409	0.7420	0.7291	0.7405	0.7559	0.9970	6.0000	16.0000	2.0000
0.7800	0.7664	0.7541	0.7276	0.7609	0.7366	0.9993	7.0000	12.0000	8.0000
0.7900	0.7419	0.7427	0.7376	0.7969	0.7600	0.8766	7.0000	12.0000	1.0000
0.8000	0.7436	0.7507	0.7305	0.7536	0.7752	0.9988	11.0000	16.0000	8.0000
0.8100	0.7590	0.7756	0.7658	0.8010	0.7789	0.9560	11.0000	16.0000	4.0000
0.8800	0.8489	0.8725	0.8340	0.8723	0.8575	0.9668	5.0000	11.0000	4.0000
0.8900	0.8660	0.8789	0.8684	0.8576	0.8779	0.9543	11.0000	16.0000	3.0000
0.9000	0.8773	0.8549	0.8266	0.8848	0.8700	0.9419	10.0000	16.0000	2.0000
0.9000	0.8773	0.8549	0.8266	0.8848	0.8700	0.9419	10.0000	16.0000	2.0000
0.9100	0.8550	0.8780	0.8656	0.9014	0.9188	0.9991	5.0000	11.0000	8.0000
0.9500	0.9020	0.9099	0.8829	0.9740	0.9157	0.9483	11.0000	16.0000	1.0000
0.9800	0.8993	0.8980	0.8455	0.9755	0.9489	0.9913	5.0000	10.0000	2.0000
0.9900	0.9234	0.9403	0.9038	0.9751	0.9520	1.0000	11.0000	16.0000	9.0000
1.0000	0.9591	0.9208	0.8888	1.0000	0.9658	0.9928	6.0000	16.0000	2.0000

La columna 1 es el valor de H generado en el vector sintético. Las siguientes cinco columnas son los métodos cálculo de H. Las última cuatro columnas son el factor de calidad, los niveles de compresión j1 y j2 y el número de desvaneciente del Wavelet Daubechies.

La siguiente tabla refleja el cálculo de cada uno de los factores H con los diferentes métodos, con respecto al factor H con el cual se calculó la serie a raíz de la Transformada de Fourier. La última línea, encerrada en rojo, calcula el promedio de las desviaciones

estándar. En este caso, LDestimate resultó tener la desviación menor. Es decir, es el método más exacto, lo que está acorde a los resultados y conclusiones de las múltiples pruebas realizadas en los trabajos referidos en la bibliografía.

Tabla 12. Cálculo de desviación estándar para vectores FGNIFOURIER.m

R/S	R/Sprom	V/T	LDestim	Var/Esc
0.0383	0.0256	0.0102	0.0068	0.0714
0.0193	0.0101	0.0341	0.0298	0.0583
0.0228	0.0049	0.0349	0.0196	0.1053
0.0158	0.0143	0.0332	0.0178	0.1140
0.0035	0.0077	0.0235	0.0014	0.0215
0.0137	0.0039	0.0478	0.0311	0.0127
0.0109	0.0198	0.0353	0.0125	0.0307
0.0085	0.0112	0.0250	0.0307	0.0076
0.0337	0.0264	0.0354	0.0085	0.0483
0.0091	0.0080	0.0209	0.0095	0.0059
0.0136	0.0259	0.0524	0.0191	0.0434
0.0481	0.0473	0.0524	0.0069	0.0300
0.0564	0.0493	0.0695	0.0464	0.0248
0.0510	0.0344	0.0442	0.0090	0.0311
0.0311	0.0075	0.0460	0.0077	0.0225
0.0240	0.0111	0.0216	0.0324	0.0121
0.0227	0.0451	0.0734	0.0152	0.0300
0.0227	0.0451	0.0734	0.0152	0.0300
0.0550	0.0320	0.0444	0.0086	0.0088
0.0480	0.0401	0.0671	0.0240	0.0343
0.0807	0.0820	0.1345	0.0045	0.0311
0.0666	0.0497	0.0862	0.0149	0.0380
0.0409	0.0792	0.1112	0.0000	0.0342
0.0320	0.0296	0.0512	0.0162	0.0368

7.2.- Validación con Vectores sintéticos FGNDWv2

El siguiente es un análisis de cálculo del parámetro H con todos los métodos, pero con muestras generadas a través de la transformada inversa Wavelet. Para esta prueba se fijó el máximo nivel de compresión a 16, y el número de desvanecimientos en 10.

Tabla 13. Cálculo deH con los cinco métodos para vectores FGNDWv2.m

FGNI	R/S	R/S Prom	V/T	LDestimate	WLog/Esc	Q	j1	j2	N
0.5000	0.4981	0.4935	0.4565	0.5014	0.6972	0.9938	4.0000	9.0000	4.0000
0.5500	0.5472	0.5596	0.5417	0.5526	0.6094	0.9617	7.0000	14.0000	2.0000
0.5800	0.5993	0.5866	0.5722	0.5795	0.6625	0.9661	6.0000	11.0000	8.0000
0.5900	0.6114	0.5971	0.5733	0.6044	0.6040	0.9842	10.0000	16.0000	4.0000
0.6000	0.6252	0.6096	0.6003	0.6150	0.6156	0.8227	8.0000	16.0000	1.0000
0.6500	0.6511	0.6474	0.6416	0.6779	0.7545	0.9977	10.0000	16.0000	5.0000
0.6800	0.6600	0.6508	0.6223	0.6963	0.7569	0.9994	8.0000	13.0000	6.0000
0.7000	0.6806	0.6923	0.6717	0.7313	0.7045	0.9618	11.0000	16.0000	2.0000
0.7500	0.7623	0.7425	0.7234	0.7196	0.7398	0.9626	11.0000	16.0000	2.0000
0.7800	0.7884	0.7660	0.7451	0.7776	0.7463	0.9953	6.0000	14.0000	8.0000
0.7900	0.8052	0.7812	0.7715	0.7945	0.7631	0.7901	8.0000	14.0000	3.0000
0.8000	0.7915	0.7789	0.7689	0.8359	0.8261	0.9919	10.0000	16.0000	5.0000
0.8500	0.8372	0.8405	0.8245	0.7797	0.8997	0.9782	11.0000	16.0000	9.0000
0.8800	0.8847	0.8612	0.8337	0.8293	0.8514	0.9542	11.0000	16.0000	7.0000
0.8900	0.8637	0.8569	0.8096	0.9074	0.8747	0.9075	11.0000	16.0000	3.0000
0.9000	0.8285	0.8290	0.7846	0.9222	0.8769	0.9713	8.0000	13.0000	8.0000
0.9500	0.9371	0.9203	0.9109	0.9138	0.9468	0.4396	11.0000	16.0000	8.0000
0.9800	0.9088	0.8865	0.8624	0.9329	0.9629	0.9576	11.0000	16.0000	5.0000
0.9900	0.9022	0.9052	0.8755	0.9928	0.9694	0.9979	11.0000	16.0000	2.0000
1.0000	0.9608	0.9296	0.8968	0.9654	0.9772	0.8469	11.0000	16.0000	2.0000

A continuación se reflejan los cálculos de la desviación estándar para cada una de las muestras, con respecto al parámetro H con el cual se calculó la serie. Según los datos con los cuales se calculó dicha desviación, el método LDestimate, transformada Wavelet, es más exacto.

Tabla 14. Cálculo de desviación para vectores FGNDWv2.m

R/S	R/Sprom	V/T	LDest	Var/Esc
0.0019	0.0065	0.0435	0.0014	0.1972
0.0028	0.0096	0.0083	0.0026	0.0594
0.0193	0.0066	0.0078	0.0005	0.0825
0.0214	0.0071	0.0167	0.0144	0.0140
0.0252	0.0096	0.0003	0.0150	0.0156
0.0011	0.0026	0.0084	0.0279	0.1045
0.0200	0.0292	0.0577	0.0163	0.0769
0.0194	0.0077	0.0283	0.0313	0.0045
0.0123	0.0075	0.0266	0.0304	0.0102
0.0084	0.0140	0.0349	0.0024	0.0337
0.0152	0.0088	0.0185	0.0045	0.0269
0.0085	0.0211	0.0311	0.0359	0.0261
0.0128	0.0095	0.0255	0.0703	0.0497
0.0047	0.0188	0.0463	0.0507	0.0286
0.0263	0.0331	0.0804	0.0174	0.0153
0.0715	0.0710	0.1154	0.0222	0.0231
0.0129	0.0297	0.0391	0.0362	0.0032
0.0712	0.0935	0.1176	0.0471	0.0171
0.0878	0.0848	0.1145	0.0028	0.0206
0.0392	0.0704	0.1032	0.0346	0.0228
0.0241	0.0271	0.0462	0.0232	0.0416

Los resultados anteriores se evidencian en la misma forma en la tabla 15, según Vecchio R.[12].

Tabla 15. Validación métodos cálculo H. Fuente[12]

	Sen	e de 16384 p	untos	Serie de 8192 puntos			Serie de 4096 puntos		
Vaior de H	Cálculo de H	Cálculo de F	Cálculo de H	Cálculo de H	Cálculo de H	Cálculo de H	Cálculo de H	Cálculo de H	Cálculo de H
RMDM	Ondita	R/S plot	VTP	Ondita	R/S plot	VTP	Ondita	R/S plot	VTP
0,50	0,4954	0,4742	0,4878	0,5046	0,5202	0.4649	0.5283	0,5431	0.4519
0,55	0,5453	0,5234	0,531	0,5546	0,5702	0,5087	0,5677	0,5898	0,4965
0,60	0,5953	0.5773	0,5751	0,6046	0,6211	0,5533	0,6177	0,6368	0,5415
0,65	0.6453	0,6343	0,6194	0,6545	0,6731	0,5983	0.6678	0,6843	0,5864
0,70	0,6953	0.6854	0,6635	0,7045	0,7193	0,6431	0,7152	0,7292	0,6307
0.75	0.7453	0.7332	0,7068	0,7546	0,7642	0,6871	0,7653	0,7717	0,6337
0,80	0,7953	0.7812	0,7488	0,8057	0,8089	0,7295	0.8153	0,8134	0,7149
0.85	0.8453	0.8285	0,7887	0,8546	0,8525	0,7698	0,8652	0,8542	0,7535
0,90	0,8953			0,9046	0,8941	0,807	0,9163	0,8933	0,7893
0,95	0.9453			0,9546	0,9331	0,8407	0.9652	0,9251	0,8217

7.3.- Validación mediante cálculos en archivos provenientes de Bellcore Morristown Research and Engineering http://ita.ee.lbl.gov/html/contrib/BC.html

A continuación se colocan los resultados de análisis múltiple a las pruebas provenientes de archivos de Bellcore Morristown Research and Engineering. La comparación se hace con respecto a una ejecución de la versión original de los módulos, realizada por Ricardo Scappini [7] en el año 2010:

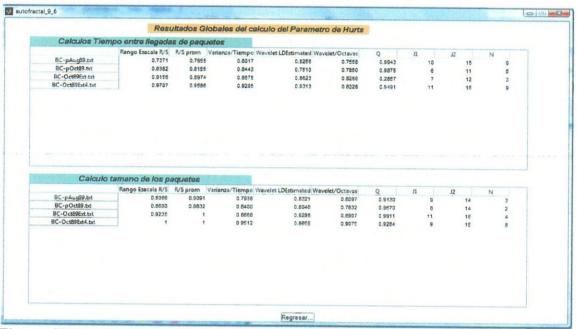


Figura 65. Pruebas a muestras reales de laboractorios Bellcore con autofractal 9_6:

Pruebas múltiples a múltiples vectores.

Los archivos correspondientes se pueden comparar con las corridas de los módulos ejecutadas por Leonardo Scappini. La única diferencia se observa en LDestimate, dado que Scappini utilizó un método visual, y el trabajo realizado en esta investigación permite comparar a través de múltiples matrices y calcular los rangos óptimos, considerando que una diferencia (j1-j2)< 4 no es deseable. Así Q de mayor, es conveniente, según las experimentaciones, tener una diferencia mayor o igual cinco. Darryl Veitch lo advierte a través del módulo Regrescomp.m dando mensajes de rango no óptimo. La decisión aquí estriba en un resultado experimental, determinándose que ese rango pudiese ser satisfactorio. De todas formas se considera que el módulo LDEstimatev2.m debe ser probado con otros archivos disponibles en Internet.

Tabla 16. Resultados obtenidos por Leonardo Scappini, Fuente[7]

Archivo	R/S	V/T	LDestimate	Var/escala	J1	J2
pAug89.TL	.737	.802	.821	.756	9	16
pOct89.TL	.839	.844	.755	.785	6	16

Las diferencias en LDestimate se deben a que en este caso están realizadas automáticamente, mientras que Ricardo Scappini lo realizó por aproximación gráfica.

7.4.- Validación mediante un vector acumulado.

Este es un caso en el que la tendencia del parámetro de Hurts es 1. El archivo se guardo en acumulado.txt: acumulado=cumsum(pAug89(:2)) . Se seleccionó la columna de tamaño de paquetes. Este vector se encuentra almacenado. El procedimiento es grabarlo en x.txt y se le hace la prueba de cálculo.

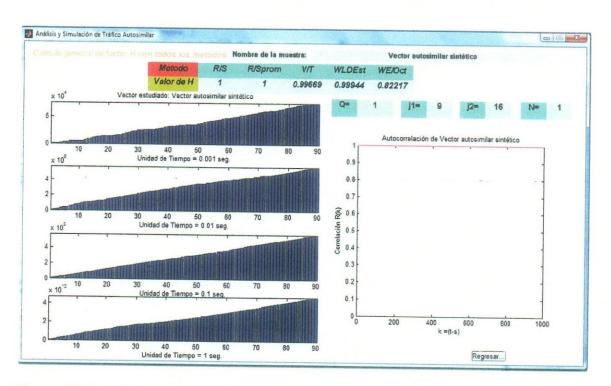


Figura 66. Resultado de Análisis de Resumen H, del menú de análisis múltiple, todas las pruebas a un vector acumulado en tamaño bytes. BC-pAug89.txt.

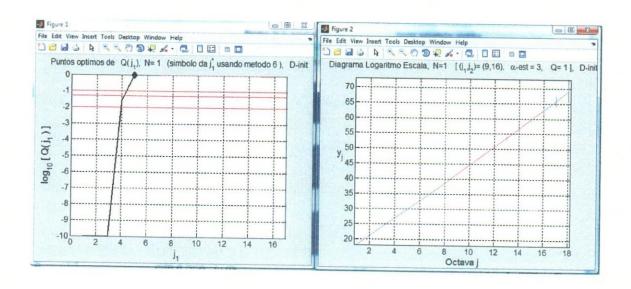


Figura 67. Resultado de Análisis de Resumen H, del menú de análisis múltiple, todas las pruebas al vector acumulado en tamaño bytes BC-Aug89.txt.

En el caso de la ejecución del software de Ricardo Scappini el resultado fue H=.733 con LDestimate, resultado que no concuerda con los otros métodos, donde el valor de H siempre da superior a .9. Ello es debido a la ecuación faltante en el software. Como consecuencia el módulo está en prueba mediante la operación automática de generación de compresiones y desvanecimientos de los Wavelets.

7.5.- Prueba del archivo fgn8.dat, cargado directamente por Darryl Veitch Se procedió a cargar este archivo en x.txt para ejecutar las pruebas como si fuera un vector sintético.

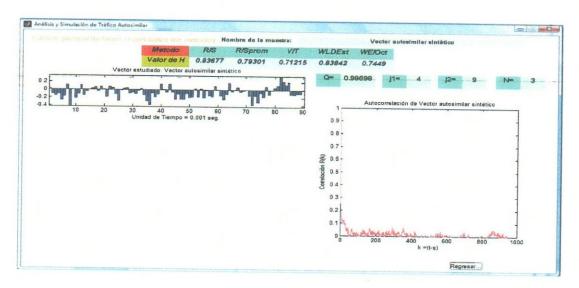


Figura 68. Pruebas al vector fgn8.dat de Darryl Veitch mediante la opción de análisis múltiple, todas las pruebas a un vector, Resumen H.

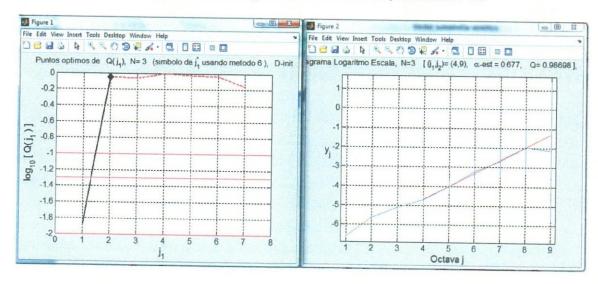


Figura 69. Pruebas al vector fgn8.dat de Darryl Veitch mediante la opción de análisis múltiple, todas las pruebas a un vector, Resumen H. Resultados Ricardo Scappini[7]

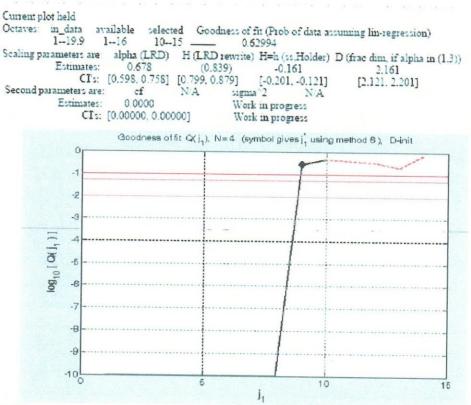


Figura 70. Resultados de Ricardo Scappini ejecutando LDEstimate.m. Fuente [7].

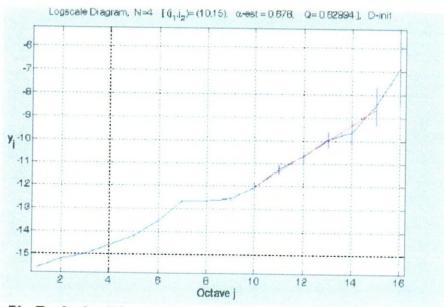


Figura 71. Rsultados Ricardo Scappini ejecutando LDEstimate.m. Fuente [7]

8. Conclusiones

El software implementado para cálculo del parámetro H, con interfaz gráfica a usuario, se desarrolló empleando utilerías elaboradas en Matlab por diversos científicos a nivel mundial, y realizando modificaciones para hacerlas compatibles a nivel de lenguaje orientado a objetos. En la misma forma se introdujo una automatización propia al módulo desarrollado por Darryl Veitch.

El módulo desarrollado por Darryl Veitch para cálculo del factor H basado en Wavelets, resultó el más exacto, según las pruebas de validación. Dicho módulo fue modificado con el objetivo de obtener una solución óptima al cálculo de H, acorde al nivel de compresión y de desvanecencia del Wavelet.

Se desarrollaron aplicaciones particulares para validar el software, con resultados que se consideran satisfactorios.

Este software tiene un alto componente orientado a la enseñanza y la academia, para un nivel del octavo o noveno semestre de Ingeniería de Telecomunicaciones o Eléctrica. El componente científico e ingenieril permite realizar pruebas múltiples a vectores provenientes de las muestras recolectadas en campo a través de un analizador de protocolo de comunicaciones.

Se considera que la herramienta se puede habilitar para ser utilizada en el análisis de tráfico de cualquier red que se considere tenga una tendencia a la autosimilaridad, a fin de determinar el parámetro H a nivel de las diferentes estructuras de topología de red: conmutación, enrutamiento, acceso y transmisión. El último, transmisión, se puede aplicar cuando las arquitecturas de red están implementadas en protocolos MPLS directas sobre fibra óptica, como la MetroEthernet. Tener MPLS activo no es un requisito, dado que una opción es ASON (arquitectura Óptica propuesta por el ITU-T, en la cual MPLS es un subconjunto dentro de toda la norma) pero se considera que una red de un operador debe tener una forma de establecer los circuitos y que los núcleos incorporen el potencial de los protocolos de enrutamiento para que la red reduzca las mallas virtuales permanentes, dando más poder a la señalización.

9.-Recomendaciones

Iniciar pruebas de medición sobre arquitecturas de redes a nivel nacional con el software implementado. Igualmente, introducir este software para ser utilizado, probado y modificado por la Academia Universitaria.

Luego de esta fase, iniciar proyectos para la automatización de recolección de muestras y sus análisis en tiempo real.

Seguida de esta fase, analizar junto a los proveedores la factibilidad técnica económica de que todos los componentes de red tengan incorporada la asignación dinámica de buffer en ambiente autosimilar fractal.

Un proyecto de mayor reto conlleva a plantear o rediseñar los modelos de comunicaciones, con el objetivo de que las interfaces entre las capas de comunicaciones introduzcan el análisis predictivo. De esta manera, protocolos como el TCP no esperarán a una reacción ante la congestión en las redes, tal como sucede actualmente, sino que las ventanas de control del TCP se adaptarían a una forma más flexible que no implique un impacto en la reducción súbita del máximo de velocidad a transmitir. El algoritmo del TCP está diseñado para actuar ante una contingencia en ambiente de tráfico Poisson: es decir, la reacción. En un ambiente caótico es necesario analizar y producir un nuevo protocolo con estas consideraciones, para aprovechar la posibilidad de acción proactiva. En este sentido, es necesario analizar el RFC 3168, que introduce un cambio en las banderas de TCP y en el campo DSCP (Servicio Diferenciado en IP) para aplicar métodos de Notificación Explícita de Congestión en el mundo IP hacia el TCP. Dicho método, trabajando en conjunto con algoritmos de predicción causará un alto impacto en la optimización de los servicios ofrecidos por las redes de Telecomunicaciones. La característica de semejanza en las diferentes escalas de tiempo facilitará el diseño de nuevas implementaciones en los protocolos de comunicaciones, y es una teoría que está siendo recomendada por diversos científicos.

Dado que los sistemas de señalización están migrando igualmente a IP, e inclusive, los datagramas del sistema de señalización número 7, pudiesen estar siendo igualmente impactados, el estudio del tráfico para ambientes autosimilares es importante para la optimización del servicio, y de esta manera, facilitar las migraciones e integraciones de redes, al igual que reducir los riesgos de degradación del servicio.

10.- Programa de actividades

0000

0 0 00000000000

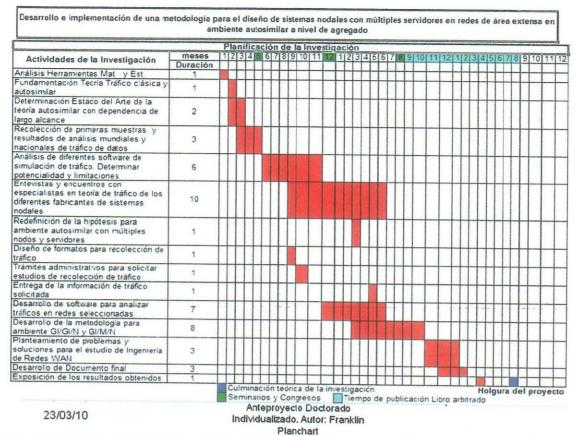


Figura 72. Cronograma de actividades para realización del Doctorado Individualizado en ingeniería en la UCV

11.- Bibliografía

-

00000

[1]Janeuski, Toni. Mobile Comunication series Traffic análisis and design of gíreles IP networks. 2003

[2]Diaz Campion, Gabriel Jesús. Teoría de Tráfico Telefónico. 2010.

[3]Malo, Martnez Julio. Análisis de Teoría de Ondículas orientada a las aplicaciones de Ingeniería Eléctrica: Fundamntos. Julio 2002.

[4] Alzate Marco. Modelos de tráfico en análisis y control de redes de Comunicaciones. Universidad Distrital Francisco José Caldas. S/F.

[5]Castro, R; Lopez,M; Martinez, J. Universidad Politécnica de Madrid, España. Universidad de Tarapacá, Chile. Martinez, M. Universidad Simón Bolivar. Introducción a la Transformada Wavelet.

[6] Hae-Duck, Joshua Jeong. Modelling of Self-Similar Teletraffic for simulation Universidad de Canterbury. 2002.

- [7]Ramón Scappini, Reinaldo José. Studio del tráfico autosimilar orientado a la simulación mediante la utilización de Wavelets u onditas. Facultad Informática Universidad Nacional de la Plata. Agosto 2010.
- [8] W. Willinger and T.G. Kurtz. The changing nature network traffic: Scaling Phenomenon. S/F
- [9] Veitch,D; Abry, P. A. Wavelet-based joint estimator of the parameter of long range dependence information theory. IEEE transactions on Volume 45, Issue 3, Apr 1999. Page(s) 878-897.
- [10]Jeong, H-D; Mc Nickle,D and Pawlikowski, K. Fast self-similar teletraffic conference base on FGN and Wavelets, in proceedings of IEEE International conference on networks (Iconn 99). Austria.
- [11]Planchart Coronado, Franklin Jesús. Algoritmo de simulación y análisis de tráfico autosimilar con dependencia de largo alcance tipo G/M/N. Congreso Internacional de Científicos en Cuba. Diciembre 2010.
- [12] Vecchio, R. "Modelización de tráfico auto-similar y evaluación de sus efectos en el tamaño de las colas. Tesis de Maestría, Instituto Tecnológico de Buenos Aires. Escuela de post-Grado",2003.
- [13] Bravo, J.; Marrone L. Tráfico Autosimilar. Método Algebráico para asignación Dinámica de Buffer . S/F.
- [14] Shoichiro Nakamura. Análisis Numérico y Visualización Gráfica con Matlab.Prentice-Halls Hispanoamericana. 1997.
- [15] (Norros,I. "On the Use of Fraccional Brownian Motion in the Theory of Conecctionless Network". IEEE Journal on Selected Areas in Comunication, Agosto 1995
- [16] Aguilar, P. Jairo. Componentes de la Ingeniería de Tráfico. Unión Internacional de Telecomunicaciones (ITU-T). 2008.