

**Desarrollo de un modelo generador de tráfico autosimilar para la
evaluación de los algoritmos de asignación de ancho de banda en WiMAX**

Tesista: Daniel Rabinovich

Director: Lic. Javier Díaz

Codirector: Ing. Luis Marrone

Tesis presentada para obtener el grado de
Magister en Redes de Datos
Facultad de Informática - Universidad Nacional de La Plata

Octubre de 2010

Reconocimientos

A mis directores de tesis Lic. Javier Díaz e Ing. Luis Marrone que me sugirieron con acierto el tema de la misma.

A las profesoras Lía Molinari y Paula Venosa quienes supieron asesorarme en el momento adecuado, y a las empleadas de la secretaría de postgrados de la Facultad de Informática para las cuales no hay palabras para describir su buena disposición hacia la gente.

A mis padres, hijos, hermanas y en especial a mi esposa, quienes tuvieron que soportar los efectos secundarios de la maestría.

Índice de contenidos

Lista de figuras	x
Lista de tablas	xvii
Glosario	xviii
1. Introducción.....	1
1.1 Objetivo.....	1
1.2 Motivación	1
1.3 Tecnología.....	2
1.4 Conocimientos deseables para el entendimiento de la presente tesis	3
1.5 Sobre los términos en inglés	3
1.6 Organización de la Tesis	3
2. Análisis estadístico de una muestra de tráfico autosimilar.....	5
2.1 Introducción	5
2.2 Análisis de autosimilaridad mediante los LDs.....	5
2.2.1 Conceptos sobre la regulación y la autosimilaridad	5
2.2.2 Autosimilaridad del tráfico de la muestra LBL-PKT4	6
2.3 Análisis de autosimilaridad mediante la función de autocorrelación	10
2.4 Análisis de la <i>heavy-tailness</i> mediante la CCDF.....	12
2.4.1 Función de distribución Pareto	13
2.4.2 CCDF de la muestra.....	16
2.5 Resumen sobre la comparación entre el tráfico por estación y el tráfico total ..	17
3. Análisis de las características del tráfico en un sistema WiMAX.....	19
3.1 Definiciones	19
3.2 Regulación en WiMAX	20
3.3 Interfaz de aire	21
3.4 Interfaz de cable	22
3.4.1 Tráficos que salen por las interfaces de cable.....	22
3.4.2 Tráfico de la interfaz de cable que sale de la BS	22

3.4.3	Tráfico de la interfaz de cable que sale de la SS.....	23
3.4.4	Tráficos que entran por las interfaces de cable	23
3.4.4.1	Tráfico de la interfaz de cable que entra a la SS.....	23
3.4.4.2	Tráfico de la interfaz de cable que entra a la BS	25
3.5	Influencia del tráfico UGS sobre la autosimilaridad.....	25
3.6	Influencia del tráfico UGS sobre la <i>heavy-tailness</i>	29
4.	Requerimientos del modelo para sintetizar tráfico	34
4.1	Parámetros que definen a un tráfico.....	34
4.2	Colas	34
4.3	Programa en Matlab y Simulink para evaluar tráfico	35
4.4	Análisis complementarios de la muestra LBL-PKT4	38
4.4.1	Análisis estadístico de la longitud neta de los paquetes	39
4.4.2	Ráfagas, <i>socks</i> , tiempos interarribo de paquetes por <i>sock</i>	40
4.4.2.1	Características del tráfico recibido por 10 estaciones de la muestra LBL-PKT4	40
4.4.2.2	Resumen de las características de la muestra completa LBL-PKT4 46	
4.4.2.3	Observaciones finales	47
5.	Desarrollo de un modelo generador de tráfico en Simulink.....	49
5.1	Elección del algoritmo del modelo para generar tráfico	49
5.1.1	Algoritmo M/Pareto	49
5.1.2	Algoritmo Pareto- <i>Modulated Poisson Process</i> (PMPP).....	50
5.2	Modelos en Simulink para los generadores M/Pareto y PMPP	52
5.2.1	Modelo en Simulink para el generador M/Pareto	53
5.2.2	Modelo en Simulink para el generador PMPP.....	63
6.	Simulación de un sistema SSQ FIFO	70
6.1	Introducción	70
6.2	Simulación de una SSQ con tráfico generado por el modelo M/Pareto	70
6.2.1	Procedimiento	71
6.2.2	Resultados	71

6.2.3	Análisis del coeficiente de Hurst H por el método IDC	75
6.2.4	Observaciones	75
6.3	Simulación de una SSQ con con tráfico generado por el modelo PMPP	77
6.3.1	Procedimiento	77
6.3.2	Resultados	78
6.3.3	Observaciones	81
7.	Modelo en Simulink para generar tráfico para ser aplicado al diseño de <i>schedulers</i> de WiMAX.....	83
7.1	Planificación de la prestación de servicios en WiMAX	83
7.1.1	Introducción	83
7.1.2	Clases de Servicios	83
7.1.3	<i>Scheduling</i> y adaptación del enlace	84
7.1.4	Prestación de servicios	86
	7.1.4.1 Comparación de la QoS entre IEEE 802.16 y IEEE 802.11..	86
	7.1.4.2 Aprovisionamiento de servicio y autoconfiguración.....	86
7.1.5	Desafíos en la implementación de la MAC de WiMAX y los Modelos de QoS	91
7.2	Información en los paquetes necesaria para el <i>scheduler</i>	92
7.3	Modelo de Simulink.....	93
7.3.1	Introducción	93
7.3.2	Arreglo de configuración para la distribución estadística de atributos con tiempos interarribo variables.....	95
7.3.3	Arreglo de configuración para la distribución estadística de atributos correspondiente a tráfico con tiempos interarribo constante	97
7.3.4	Arreglos de configuración para la distribución estadística de longitudes de paquetes.....	99
7.3.5	Modelo de Simulink para la asignación de atributos a los paquetes y generación de trafico UGS.....	101
	7.3.5.1 Cadena superior	103
	7.3.5.2 Cadena inferior	109
7.3.6	Efecto del trafico UGS en el comportamiento de la SSQ FIFO	111
8.	Conclusiones.....	113

8.1	Relativas a los objetivos planteados	113
8.2	Otros subproductos	113
8.2.1	Respecto a la autosimilaridad del tráfico	113
8.2.2	Respecto a la <i>heavy-tailness</i> de la CCDF	114
8.2.3	Efecto del agregado de un tráfico con tiempo interarribo constante a un tráfico autosimilar	114
9.	Trabajos a futuro.....	116
9.1	Análisis de la dispersión entre muestras de tráfico	116
9.2	Extracción del tráfico tipo CBR de una muestra	116
9.3	Modelos para sintetizar tráfico por aplicación.....	116
10.	Anexo 1 - Modelo de Simulink para analizar el comportamiento de una SSQ FIFO sometida a un tráfico dado.....	117
11.	Anexo 2 – Programas utilitarios en Matlab	122
11.1	Curva variancia-agregación	122
11.2	IDC (<i>Index of Dispersion for Counts</i>)	124
12.	Anexo 3 – <i>Scripts</i> para simulación con el modelo M/Pareto	125
13.	Anexo 4 – <i>Scripts</i> para simulación con el modelo PMPP	129
14.	Anexo 5 – <i>Script</i> para generar tráfico para WiMAX.....	133
15.	Anexo 6 – Reseña de WiMAX	137
15.1	Introducción	137
15.2	Historia.....	137
15.2.1	El WiMAX <i>Forum</i>	139
15.3	Las dos variantes de WiMAX.....	140
15.4	MAC de WiMAX	141
15.4.1	CS de la MAC	141
15.4.2	CPS de la MAC.....	143
15.4.2.1	UGS (<i>Unsolicited Grant Service</i>).....	143
15.4.2.2	rtPS (<i>real-time Polling Service</i>).....	144
15.4.2.3	ertPS (<i>extended real-time Polling Service</i>) (solo para <i>Mobile WiMAX</i>).....	144
15.4.2.4	nrtPS (<i>non real-time Polling Service</i>).....	144

15.4.2.5 BE (<i>Best Effort</i>)	145
15.4.3 Subcapa de Seguridad	145
15.5 PHY de <i>Fixed</i> WiMAX	145
15.6 PHY de <i>Mobile</i> WiMAX	147
15.7 Arquitectura de red WiMAX	153
15.8 <i>Multicast y broadcast</i>	155
15.9 <i>Next Generation</i> WiMAX.....	156
15.10 Equipamiento WiMAX y estado de la implementación	158
15.10.1 Estado de Implementación.....	158
15.10.2 Disponibilidad de equipos	159
16. Bibliografía.....	161

Lista de figuras

Fig. 1-1: Organización de la tesis.	4
Fig. 2-1: LD mostrando efectos de la congestión en el tráfico. Tomada de (6).	6
Fig. 2-2: LDs de los tráficos recibidos por cada una de las 10 estaciones seleccionadas (24,30% del tráfico total de la muestra LBL-PKT-4), la superposición del tráfico recibido por las 10 estaciones y el tráfico total de la muestra como referencia.	7
Fig. 2-3: LDs de los tráficos transmitidos por cada una de las 10 estaciones seleccionadas (28,55% del tráfico total de la muestra LBL-PKT-4), la superposición del tráfico transmitido por las 10 estaciones y el tráfico total de la muestra como referencia.	8
Fig. 2-4: log(ACF) versus log(k) del tráfico recibido por estaciones seleccionadas (24,30% de la muestra LBL-PKT-4).	11
Fig. 2-5: log(ACF) versus log(k) del tráfico transmitido por estaciones seleccionadas (28,55% de la muestra LBL-PKT-4).	12
Fig. 2-6: 99 muestras generadas por un simulador de la distribución Pareto con $\alpha=1,3$ y $b=1$	14
Fig. 2-7: CCDF de muestras generadas por un simulador de la distribución Pareto con $\alpha=1$ a 2 en pasos de 0,1.	15
Fig. 2-8: LD de muestras generadas por un simulador con distribución Pareto con $\alpha=1$ a 2 en pasos de 0,1.	16
Fig. 2-9: log(CCDF) versus log(Tiempo interarribo) del tráfico recibido por 10 estaciones seleccionadas (24,30% de la muestra LBL-PKT-4), la superposición de los mismos, el tráfico total y las pendientes de las asíntotas.	17
Fig. 2-10 log(CCDF) versus log(Tiempo interarribo) del tráfico transmitido por 10 estaciones seleccionadas (28,55% de la muestra LBL-PKT-4), la superposición de los mismos, el tráfico total y las pendientes de las asíntotas.	18
Fig. 3-1: Definiciones: interfaces de aire y de cable de un sistema WiMAX compuesto por una estación base (BB) y por una estación subscriptora (SS). Los términos entra o sale son siempre como los ve la estación WiMAX.	19
Fig. 3-2: Analogía hídrica mostrando el efecto congestivo de un enlace WiMAX con capacidad inferior a la conexión de cable que lo alimenta.	20
Fig. 3-3: CCDF del tiempo de transmisión de archivos en la WWW. La distribución es asintóticamente <i>heavy-tailed</i> con $\alpha_{SI} \approx 1,21$. Tomada de (16).	24
Fig. 3-4: LD tráfico total y LD tráfico total más un tráfico constante UGS con período 3ms.	26

Fig. 3-5: Variancias de los procesos m-agregados para el cálculo del coeficiente de Hurst. Tráfico total de la muestra LBL-PKT-4 y de la misma más un tráfico UGS con tiempo interarribo de 3ms.....	28
Fig. 3-6: CCDF del tráfico total de la muestra LBL-PKT-4 y de la misma más un tráfico UGS con tiempo interarribo de 3ms.....	29
Fig. 3-7: CCDF del tráfico total de la muestra LBL-PKT-4 y de la misma más un tráfico UGS con tiempo interarribo de 30ms.....	30
Fig. 3-8: CCDF del tráfico de la estación 32 de la muestra LBL-PKT-4 y de la misma más un tráfico UGS con tiempo interarribo de 80ms.....	30
Fig. 3-9: CCDF del tráfico recibido por 10 estaciones seleccionadas de la muestra LBL-PKT-4 a las cuales se adicionó un tráfico UGS con diferente período.	31
Fig. 3-10: LD del tráfico recibido por 10 estaciones seleccionadas de la muestra LBL-PKT-4 a las cuales se adicionó un tráfico UGS con diferente período.....	32
Fig. 4-1: Probabilidad de pérdida de paquetes en función de la capacidad de la cola para una muestra de tráfico real, denominado Ethernet, comparándolo con tráficos sintetizados con el algoritmo M/Pareto1, M/Pareto2 y Gaussiano. Tomada de (18).....	36
Fig. 4-2: Modelo de Simulink para analizar el comportamiento de una SSQ FIFO con un tiempo de servicio constante.....	36
Fig. 4-3: Resultado de aplicar el tráfico de la muestra LBL-PKT4 al modelo SSQ FIFO de Simulink de la Fig. 4-2, para diferentes intensidades de tráfico ρ en función de la capacidad de la cola.....	37
Fig. 4-4: Histograma de la cantidad de datos de los paquetes IP correspondientes a la muestra LBL-PKT4.....	40
Fig. 4-5: Paquetes IP registrados, correspondientes a la muestra LBL-PKT4. Cada punto representa uno o muchos paquetes recibidos por la estación destino. En una coordenada se indican las estaciones seleccionadas la 4, 5, 6, 14, 15, 19, 23, 24, 32, y 43, en otra la estación de donde proviene y en la tercera coordenada el instante en que se registró el paquete.	41
Fig. 4-6: Magnificación progresiva de un punto correspondiente a la estación destino 32 señalado en la Fig. 4-5.....	41
Fig. 4-7: La misma información que la Fig. 4-5 a la que se agrega circunferencias negras a los paquetes que presentan el flag Syn activado.	42
Fig. 4-8: Histograma de de lo cantidad de paquetes versus el tiempo interarribo entre paquetes de cada <i>sock</i>	44
Fig. 4-9: Detalle de la Fig. 4-8. Se ha marcado la barra que contiene a la moda que vale 0,48ms	44

Fig. 4-10: Cantidad de paquetes acumulados versus tiempo interarribo entre paquetes de cada <i>sock</i> . Los paquetes acumulados con tiempos interarribo menor a 10s son 210662 y constituyen el 96,7% de los 217854 paquetes seleccionados.....	45
Fig. 4-11: Función de distribución de la cantidad de paquetes versus tiempo interarribo entre paquetes de cada <i>sock</i> . Se ha marcado la mediana que vale 67,5ms.....	46
Fig. 4-12: Arriba, secuencia de paquetes correspondientes a 4 <i>socks</i> consecutivos enviados por la estación 4 y recibidos por la estación 19. Abajo, dos secuencias de paquetes de <i>sock</i> único recibidos por dos estaciones diferentes, nótese lo poco consistente que resulta asignar un <i>sock</i> a una ráfaga.....	46
Fig. 5-1: Diagrama en bloques del generador M/Pareto.....	50
Fig. 5-2: Diagrama de estado del modelo PMPP.....	51
Fig. 5-3: Ventana de diálogo del bloque Event- Based Random Number	53
Fig. 5-4: Modelo M/Pareto realizado en Simulink.....	54
Fig. 5-5: Ventana de diálogo del bloque Time-Based Entity Generator.....	55
Fig. 5-6: Ventana de diálogo del bloque <i>Set Attribute</i>	55
Fig. 5-7: Ventana de diálogo del bloque <i>Event-Based Sequence</i>	56
Fig. 5-8: Ventana de diálogo del bloque Infinite Server.....	56
Fig. 5-9: Ventana de diálogo del bloque Entity Sink.....	57
Fig. 5-10: Oscilograma de la cantidad de ráfagas superpuestas #n en el bloque <i>Infinite Server</i>	57
Fig. 5-11: Oscilograma del Tiempo interarribo de paquetes.....	58
Fig. 5-12: Vistas coordinadas en el tiempo del <i>Signal Scope1</i> y el <i>Instantaneous Entity Counting Scope</i>	58
Fig. 5-13: Ventana de diálogo del bloque <i>Time-Based Entity Generator1</i>	59
Fig. 5-14: Ventana de diálogo del bloque <i>Set attribute1</i>	59
Fig. 5-15: Ventana de diálogo del bloque <i>Output Switch</i>	60
Fig. 5-16: Ventana de diálogo del bloque <i>Single Server</i> . Se modificó los datos de la pestaña <i>Preemption</i>	61
Fig. 5-17: Ventana de diálogo del bloque <i>Discrete Event Signal to Workspace</i>	62
Fig. 5-18: Ventana de diálogo del bloque <i>Path Combiner</i>	62
Fig. 5-19: Ventana de diálogo correspondiente a los bloques <i>Time-Based Entity Generator</i> y <i>Time-Based Entity Generator1</i> y <i>Time-Based Entity Generator2</i>	63

Fig. 5-20: Modelo PMPP realizado en Simulink.....	64
Fig. 5-21: Ventana de diálogo correspondiente al bloque <i>Event-Based Random Number</i>	65
Fig. 5-22: Ventana de diálogo correspondiente al bloque <i>Event-Based Random Number1</i>	65
Fig. 5-23: Ventana de diálogo correspondiente al bloque <i>Input Switch</i>	66
Fig. 5-24: Ventana de diálogo del bloque <i>Discrete Event Signal to Workspace</i>	66
Fig. 5-25: Ventana de diálogo del bloque <i>Event-Based Sequence</i>	67
Fig. 5-26: Ventana de diálogo del bloque <i>Fcn</i>	67
Fig. 5-27: Oscilogramas producidos por los bloques <i>Signal Scope</i> , <i>Signal Scope1</i> , e <i>Instantaneous Entity Counting Scope</i>	68
Fig. 6-1: Curvas de Paquetes perdidos/Paquetes arribados versus la Capacidad de la cola....	72
Fig. 6-2: Curvas de Paquetes perdidos/Paquetes arribados versus la Capacidad de la cola....	73
Fig. 6-3: Curvas de Paquetes perdidos/Paquetes arribados versus la Capacidad de la cola en la búsqueda de mejores aproximaciones a la muestra LBL-PKT4.....	73
Fig. 6-4: Curvas de Paquetes perdidos/Paquetes arribados versus la Capacidad de la cola. Influencia de la intensidad de tráfico ρ y de la dispersión producida al correr varias veces la simulación con los mismos juegos de parámetros.....	74
Fig. 6-5: Curva IDC de las simulaciones más aproximadas a la muestra LBL-PKT4. $H=(Pen+1)/2=0,905$	74
Fig. 6-6: Curvas de Paquetes perdidos/Paquetes arribados versus la Capacidad de la cola....	78
Fig. 6-7: Curvas de Paquetes perdidos/Paquetes arribados versus la Capacidad de la cola....	78
Fig. 6-8: Curvas de Paquetes perdidos/Paquetes arribados versus la Capacidad de la cola....	79
Fig. 6-9: Curvas de Paquetes perdidos/Paquetes arribados versus la Capacidad de la cola. Se muestran las dos mejores aproximaciones por el método PMPP a la muestra LBL-PKT4 para $\rho=0,9$	80
Fig. 6-10: Comportamiento del tráfico sintético PMPP comparado con el de la muestra LBL-PKT4 cuando en el sistema SSQ FIFO se varía el tiempo de servicio μ y por ende la intensidad de tráfico $\rho=0,7$ $0,8$ $0,9$ y 1	80
Fig. 6-11: Curva IDC de las simulación más aproximada a la muestra LBL-PKT4 realizada con el modelo PMPP.....	81
Fig. 7-1: Mecanismo de QoS para servicios multimedia. Tomada de (23).....	84
Fig. 7-2: Modelo de referencia de gestión de red. Tomada de (23).....	87
Fig. 7-3: Estructura de la <i>wmanIfMib</i> . Tomada de (23).....	88

Fig. 7-4: Aprovisionamiento de flujos de servicio. Tomada de (23).....	90
Fig. 7-5: Arreglo At para introducción de atributos para el tráfico ya generado previamente.	95
Fig. 7-6: Arreglos de los atributos IPDest (IPD), IPSource (IPS) y DSCP (SC), con las secuencias pseudo aleatorias generadas por un <i>script</i> , conforme a la configuración especificada en el arreglo At.	97
Fig. 7-7: Arreglo AtCte para introducción de atributos para el tráfico con tiempos interarribo constante.	97
Fig. 7-8: Arreglos de los atributos IPDest (IPDcte) y IPSource (IPScte) con las secuencias pseudo aleatorias generadas por un <i>script</i> , conforme a la configuración especificada en el arreglo AtCte. El atributo DSCP es siempre 1.	98
Fig. 7-9: En el vector PacketLen están las longitudes de la carga útil TCP/UDP de cada paquete de la muestra de tráfico LBL-PKT4.....	99
Fig. 7-10: El arreglo Histo contiene la distribución de probabilidades para cada longitud de paquetes de la mencionada muestra, en la columna de la izquierda las longitudes de paquete y en la derecha las probabilidades correspondientes, la suma de todos los valores de la segunda columna totaliza uno.....	100
Fig. 7-11: Vector del atributo Len (LongPaq) con las secuencias pseudo aleatorias generadas por un <i>script</i> , conforme a la distribución definida en el arreglo Histo de la Fig. 7-10.....	100
Fig. 7-12: Ventana de diálogo del bloque <i>Event-Based Sequence</i> que genera una secuencia de números tomados del vector TiemposInterarribo cargado desde el <i>workspace</i> de Matlab. Los bloques <i>Event-Based Sequence1</i> , <i>Event-Based Sequence 2</i> y <i>Event-Based Sequence3</i> , poseen la misma ventana de diálogo salvo que el nombre de los vectores son IPS, IPD, y SC respectivamente.	101
Fig. 7-13: Modelo de Simulink para el marcado de paquetes con atributos y generación de tráfico con tiempo interarribo constante.....	102
Fig. 7-14: Ventana de diálogo del bloque <i>Time-Based Entity Generator</i> configurado para generar entidades con intervalos entre paquetes tomados del puerto t.....	103
Fig. 7-15: Ventanas de diálogo del bloque Set Attribute el cual marca los paquetes con las secuencias aleatorias disponible en sus <i>notifying ports</i> A1, A2 y A3.	104
Fig. 7-16: Ventana de diálogo del bloque <i>Path Combiner</i> . Este bloque combina flujos de entidades.....	104
Fig. 7-17: Ventana de diálogo del bloque <i>Event-Based Sequence6</i> que genera una secuencia de números tomados del vector LongPaq cargado desde el <i>workspace</i> de Matlab.....	105
Fig. 7-18: Ventanas de diálogo del bloque <i>Set Attribute2</i> el cual marca los paquetes con las secuencias aleatorias disponibles en su <i>notifying port</i> A1 provista por el bloque <i>Event-Based Sequence6</i>	105

Fig. 7-19: Ventana de diálogo del bloque <i>Get Attribute</i> . Este bloque recupera los atributos de cada paquete y los saca por sus puertos A1, A2, A3 y A4.	106
Fig. 7-20: Ventana de diálogo del bloque <i>Discrete Event Signal to Workspace</i> . El valor del campo <i>Variable name</i> es el nombre con que Simulink exportará al <i>workspace</i> el vector con el atributo IPSource. Las instancias 1,2 y 4 del mismo bloque cumplen idéntica función para los atributos IPDest, DSCP y Pkt Size.	106
Fig. 7-21: Ventana de diálogo del bloque <i>Entity Sink</i> con la salida número de entidades arribadas habilitada.	107
Fig. 7-22: Ventana de diálogo del bloque <i>Discrete Event Signal to Workspace3</i> . El valor del campo nombre de variable es el nombre con que Simulink exporta al <i>workspace</i> de Matlab una estructura cuyo elemento <i>Time.time</i> es un vector que almacena los tiempos de arribo de cada paquete al bloque <i>Entity Sink</i>	107
Fig. 7-23: Ventana de diálogo del bloque <i>Time-Based Entity Generator1</i> configurado para generar entidades a intervalos constantes, cargando el valor de dicho período de la variable <i>InterArrivalTimeCte</i> del <i>workspace</i>	108
Fig. 7-24: Ventana de diálogo del bloque <i>Set Attribute1</i> . Los atributos IPSource e IPDest son señales generadas por los bloques <i>Event-Based Sequence4</i> , y <i>Event-Based Sequence5</i> respectivamente, mientras que el atributo DSCP se fija en 1 ya que todas las entidades de esta cadena pertenecen a <i>flows</i> con QoS UGS.....	109
Fig. 7-25: Ventana de diálogo del bloque <i>Event-Based Sequence4</i> que genera una secuencia de números tomados del vector <i>IPScte</i> cargado desde el <i>workspace</i> de Matlab. El bloque <i>Event-Based Sequence5</i> , posee la misma ventana de diálogo salvo que el nombre del vector de donde se carga la secuencia de números es <i>IPDcte</i>	110
Fig. 7-26: Ventana de diálogo del bloque <i>Output Switch</i> . Este bloque conduce las entidades por el puerto de salida OUT1 hacia el <i>Path Combiner</i> de la cadena superior si el puerto de entrada P tiene valor 1, si P tiene valor 2 las entidades son derivadas al puerto de salida OUT2.....	110
Fig. 7-27: Curvas de Paquetes perdidos/Paquetes arribados versus la Capacidad de la cola.	111
Fig. 10-1: Modelo en Simulink de un sistema SSQ FIFO.....	117
Fig. 10-2: Ventana de diálogo del bloque <i>Event-Based Sequence</i>	117
Fig. 10-3: Ventana de diálogo del bloque <i>Time-Based Entity Generator</i>	118
Fig. 10-4: Ventana de diálogo de la solapa Statistics del bloque <i>Time-Based Entity Generator</i>	118
Fig. 10-5: Ventana de diálogo del bloque <i>To Workspace</i>	119
Fig. 10-6: Ventana de diálogo del bloque <i>Output Switch</i>	120
Fig. 10-7: Ventana de diálogo del bloque <i>FIFO Queue</i>	120

Fig. 10-8: Ventana de diálogo del bloque <i>Sink1</i>	120
Fig. 10-9: Ventana de diálogo del bloque <i>Single Server</i>	121
Fig. 15-1: Modelo de referencia del estándar IEEE 802.16. Tomada de (25).....	141
Fig. 15-2 Clasificación de paquetes realizada por la <i>Convergence Sublayer</i> . Tomada de (25).	142
Fig. 15-3: Descripción de subportadoras de OFDM. Tomada de (25).....	145
Fig. 15-4: Diferencia entre OFDM y OFDMA. Cada color representa datos de un usuario..	147
Fig. 15-5: Descripción en frecuencia de OFDMA. Subportadoras con el mismo color pertenecen al mismo subcanal. Tomada de (25).....	148
Fig. 15-6: Estructura general de un <i>frame</i> TDD de WiMAX. Tomada de (25).	149
Fig. 15-7: <i>Frame</i> TDD de OFDMA con múltiples zonas. Tomada de (25).....	150
Fig. 15-8: Sistema MIMO 3x2. Tomada de (25).....	152
Fig. 15-9: Modelo de referencia de red de WiMAX. Tomada de (25).....	154
Fig. 15-10: Redes WiMax fijas (en rojo) y móviles (en azul) que están en servicio o en implementación a Octubre de 2010. Fuente http://wimaxmaps.org/	159

Lista de tablas

Tabla 2-1 Parámetros más importantes de la distribución Pareto.	14
Tabla 4-1: Contenido de los campos de archivos de la muestra LBL-PKT4.	39
Tabla 4-2: Propiedades de los paquetes seleccionados de la muestra LBL-PKT4. Se seleccionó paquetes recibidos correspondientes a 10 estaciones, que constituyen el 24,3% de paquetes de la muestra.	43
Tabla 4-3: Propiedades del total de los paquetes recibidos de la muestra LBL-PKT4. La calificación recibido o enviado es fútil, ya que cuando se toma toda la muestra los paquetes recibidos son los mismos que los enviados.	47
Tabla 7-1: Algunas combinaciones de atributos de los paquetes para un adecuado funcionamiento del MAC <i>scheduler</i> de WiMAX.	92
Tabla 14-1: Arreglos con secuencia de atributos generados por programas invocados por script1.m	133
Tabla 15-1: Tamaños de la FFT para distintos anchos de banda del canal. Tomada de (25).	147
Tabla 15-2: Rendimiento espectral para <i>Mobile</i> WiMAX TDD. Los requerimientos del ITU para la <i>IMT-Advanced</i> están entre paréntesis. Tomada de (25).	157
Tabla 15-3: Rendimiento espectral en escenarios móviles para <i>Mobile</i> WiMAX TDD. Tomada de (25).	157
Tabla 15-4: Los cinco mayores operadores de WiMAX en el mundo. Tomada de (25).	158

Glosario

AAS	<i>Adaptive Antena Systems</i>
ACF	<i>Auto Correlation Function</i>
AMC	<i>Adaptive Modulation and Coding</i>
AP	<i>Access Point</i>
ASN	<i>Access Service Network</i>
BE	<i>Best Effort</i>
<i>beamforming</i>	lóbulo de la antena modificado electrónicamente
BS	<i>Base Station</i>
<i>burst</i>	ráfaga
CBR	<i>Constant Bit Rate</i>
CCDF	<i>Complementary Cumulative Distribution Function</i> , función de distribución acumulativa complementaria
CDF	<i>Cumulative Distribution Function</i> , función de distribución acumulativa
CID	<i>Connection ID</i>
CPS	<i>Common Part Sublayer</i>
CS	<i>service specific Convergence Sublayer</i>
CSN	<i>Connectivity Service Network</i>
DOCSIS	<i>Data Over Cable Service Interface Specification</i>
<i>downlink</i>	flujo de datos desde la BS hacia las SS
<i>dropping</i>	descarte de paquetes
ertPS	<i>extended real-time Polling Service</i>
FDD	<i>Frequency Division Duplexing</i>
FFT	<i>Fast Fourier Transform</i>
<i>frame</i>	trama de WiMAX
FUSC	<i>Full Usage of Subchannels</i>
<i>grant</i>	ancho de banda para transmitir datos en el UL
<i>handover</i>	proceso de cambio de celda
<i>heavy-tailed</i>	distribución en la que la caída de la curva de la CCDF es lenta
Hurst	coeficiente que indica el nivel de autosimilaridad de una distribución
IMT	<i>International Mobile Telecommunications</i>
ISI	<i>Inter Symbol Interference</i>
ITU	<i>International Telecommunications Union</i>

<i>LBL-PKT4</i>	muestra recolectada de tráfico entre el Lawrence Berkeley <i>Laboratory</i> y el resto del mundo de una hora de duración
LD	<i>Logscale Diagram</i>
LRD	<i>Long Range Dependencies</i>
M/Pareto	Markov/Pareto
MAC	<i>Media Access Control layer</i>
MBS	<i>Multicast and Broadcast Service</i>
MCBCS	<i>Multicast and Broadcast Multimedia Services</i>
MIB	<i>Management Information Base</i>
MIMO	<i>Multiple Input Multiple Output</i>
MMPP	Markov <i>Modulated Poisson Processes</i>
MS	<i>Mobile Station</i>
<i>multipath</i>	multitrayecto
nrtPS	<i>non real-time Polling Service</i>
OFDM	<i>Ortogonal Frequency Division Multiplexing</i>
OFDMA	<i>Ortogonal Frequency Division Multiple Access</i>
PDU	<i>Protocol Data Unit</i>
PHY	<i>Physical layer</i>
PMPP	Pareto-modulated <i>Poisson process</i>
<i>polling</i>	ranura tiempo frecuencia asignada a una terminal en el UL para que pueda solicitar ancho de banda si lo necesita
PUSC	<i>Partial Usage of Subchannels</i>
QoS	<i>Quality of Service</i>
RTG	<i>Receive/Transmit Transition Gap</i>
rtPS	<i>real-time Polling Service</i>
<i>scheduler</i>	módulo de software que planifica la asignación de ancho de banda a las estaciones subscriptoras en WiMAX
SDU	<i>Service Data Unit</i>
SLA	<i>Service Level Agreement</i>
<i>slot</i>	ranura, unidad mínima en el plano frecuencia tiempo de OFDMA
SM	<i>Spatial Multiplexing</i>
SNR	<i>Signal to Noise Ratio</i>
<i>sock</i>	cuádrupla estación origen, estación destino, puerto origen y puerto destino
SS	<i>Subscriber Station</i>
SSQ FIFO	<i>Single Server Queue First Input First Output</i>

STC	<i>Space Time Coding</i>
TDD	<i>Time Division Duplexing</i>
<i>throughput</i>	rendimiento o tasa de transferencia de acuerdo al contexto
<i>toolbox</i>	caja de herramientas de Simulink
TTG	<i>Transmit/Receive Transition Gap</i>
UGS	<i>Unsolicited Grant Service</i>
<i>uplink</i>	flujo de datos desde las SS hacia la BS
VoIP	<i>Voice over IP</i>
WiMAX	<i>Worldwide Interoperability for Microwave Access</i>
<i>workspace</i>	entorno de Matlab

1. Introducción

1.1 Objetivo

El objetivo de la tesis es elaborar un modelo destinado a la generación de tráfico de red para simulación de sistemas, total o parcialmente autosimilar, y que satisfaga determinadas características cuantitativas del tráfico que modela. Tasa promedio de paquetes, coeficiente de Hurst, comportamiento de las colas, y otros indicadores.

Además, cada paquete del tráfico generado, debe incluir la información adicional necesaria que permita analizar la eficiencia de los algoritmos de asignación de ancho de banda en WiMAX¹, en un entorno de simulación.

1.2 Motivación

Originalmente se orientó el trabajo de tesis al análisis de la eficiencia de diferentes algoritmos para la asignación de ancho de banda en WiMAX.

El módulo de software que asigna ancho de banda a las estaciones se denomina *scheduler* de WiMAX.

La información de los tiempos de arribo de los paquetes, la QoS (*Quality of Service*) de los mismos, la SS (*Subscriber Station*) que los origina o a cual va destinada, los SLA (*Service Level Agreement*) con cada estación o cliente, las condiciones radioeléctricas (propagación, ruido, interferencias, etc.), las solicitudes de ancho de banda pendientes de satisfacer, el estado de las colas, etc. constituyen la materia prima necesaria para alimentar al complejo y comprometido *scheduler* de WiMAX.

El estándar IEEE 802.16 (1), conocido como WiMAX, no impone ni sugiere los algoritmos de asignación de ancho de banda, dejando abierto el desarrollo de los mismos a los fabricantes como elemento diferenciador del desempeño de los equipos. Esto ha motivado que muchos investigadores se involucraran en el tema, desarrollando nuevos e ingeniosos algoritmos.

Para atacar el problema de diseño, y poder operar con flexibilidad sobre los parámetros que definen el tráfico es conveniente emplear un modelo para la generación de tráfico sintético, en vez de utilizar tráfico real, ya que este último es difícil de modificar y prácticamente imposible de parametrizar.

¹ *Worldwide Interoperability for Microwave Access*

Dicho modelo generador de tráfico debe contemplar la inclusión de toda la información necesaria en los paquetes que requiere el *scheduler* para su funcionamiento.

El resultado de la búsqueda de un modelo para la generación de tráfico con estas características, desarrollado y disponible para su uso, fue negativo.

Dada la importante magnitud de la tarea que implica el desarrollo de un modelo para generar tráfico adecuado, se resolvió reorientar el trabajo de tesis hacia el desarrollo del mismo, pero ya como objetivo.

1.3 Tecnología

Si bien existen varios programas para síntesis de tráfico, tal como Opnet, NS, etc. que contemplan dispositivos WiMAX, los detalles sobre la generación de tráfico no están suficientemente a la vista. Además al ser programas especializados, solo están disponibles en institutos o empresas donde existen investigadores o desarrolladores dedicados al tema, que pueden costear las licencias.

Para este desarrollo se decidió desde un comienzo emplear Matlab, y en particular Simulink, que es un programador gráfico con bloques funcionales y conectores.

Simulink posee una *toolbox* (caja de herramientas) llamada SimEvents la cual maneja el concepto de entidades. Las entidades representan en forma natural y eficaz a los paquetes que componen el tráfico. Además la *toolbox* posee una variedad importante de bloques para generar entidades, enrutarlas, encolarlas, manipularlas, etc. todo en el marco de una evolución temporal, permitiendo un manejo simple e intuitivo del tráfico, casi con la prescindencia total del el uso de fórmulas matemáticas.

Matlab² es un software con licencia para su uso, pero su amplio espectro hace que esté muy difundido, y que casi todas las universidades posean licencias adquiridas legalmente para su uso.

En el transcurso del desarrollo de la presente tesis también se han hecho evidente algunas debilidades de Matlab, especialmente en el manejo de vectores con algunos pocos millones de registros, mostrando errores cuya descripción no siempre se corresponden con la causa o simplemente colgadas de Matlab por falta de memoria libre de la computadora. Esto produjo una limitación en la cantidad máxima de paquetes manipulados a no más de un millón.

² Matlab, Simulink y SimEvents son marcas registradas por MathWorks.

1.4 Conocimientos deseables para el entendimiento de la presente tesis

Para la lectura de la presente es conveniente poseer conocimientos básicos de probabilidad y estadística, y algunas nociones sobre tráfico autosimilar.

También, es favorable para una mejor comprensión, poseer conocimientos sobre la teoría de los *logscale diagrams*, así como de Matlab y Simulink.

La información general sobre WiMAX está suficientemente cubierta en el Anexo 6 – Reseña de WiMAX, por lo que no es necesario poseer conocimientos previos.

1.5 Sobre los términos en inglés

Dada la característica técnica de la presente, el probable perfil de los lectores y que las tecnologías involucradas están descritas predominantemente en inglés, se ha renunciado a una traducción sistemática al español de los términos claves por un doble motivo. Primero, en muchos casos no existe una palabra o frase en español que represente adecuadamente la denominación original en inglés. Segundo, se entiende que de aplicarse una traducción no selectiva, el lector no podrá relacionar los términos que acá aparecieran con aquellos de las bibliografías, cuya mayor parte está en inglés.

Los términos en inglés se indican en *cursiva*.

Se ha evitado subestimar al lector de la tesis, traduciendo del inglés al español términos básicos o de uso muy generalizado en el campo informático.

Se ha agregado al comienzo un glosario o nomenclatura de siglas y locuciones inglesas para disipar dudas que pudieran surgir sobre alguna denominación.

1.6 Organización de la Tesis

En la Fig. 1-1 se muestra en un diagrama de flujo los elementos más importantes del desarrollo de la tesis, con la indicación de los capítulos correspondientes entre paréntesis.

Se omite en la figura, futuros trabajos, y los anexos donde se describen, el modelo en Simulink de la SSQ FIFO, se transcriben los diferentes programas utilitarios en Matlab y *scripts* para manejar los modelos construidos en Simulink, y se presenta una reseña de WiMAX.

Puede ser conveniente la lectura de la mencionada reseña para una mejor comprensión de los Cap. 3 y Cap. 7.

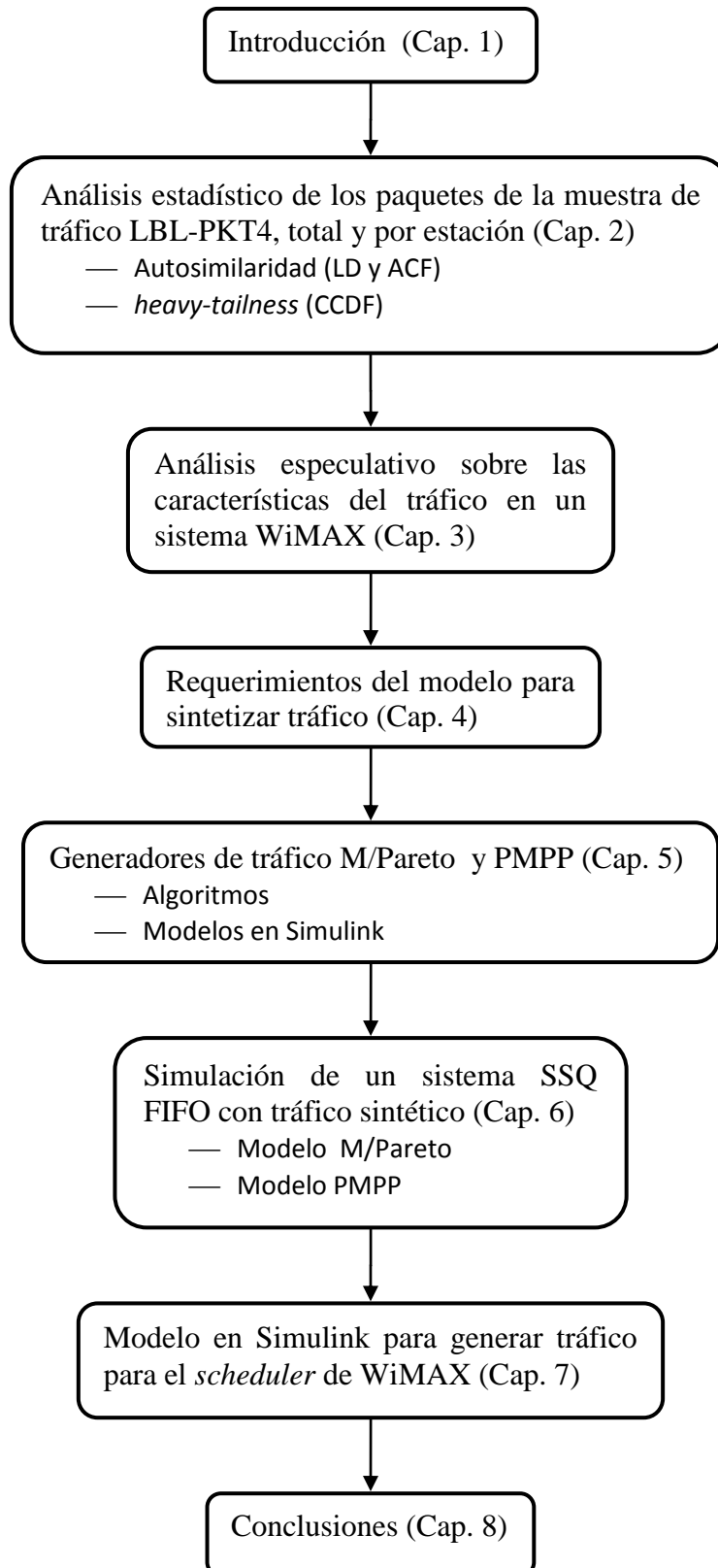


Fig. 1-1: Organización de la tesis.

2. Análisis estadístico de una muestra de tráfico autosimilar

2.1 Introducción

El objetivo del siguiente capítulo es analizar e investigar las características del tráfico que probablemente arribe a una estación WiMAX, por la interfaz de cable.

Como no se pudo acceder a trazas recolectadas, tanto del tráfico que arriba a la BS por la interfaz de cable, como el que arriba a la SS también por la interfaz de cable, se comenzará analizando una muestra de un tráfico que se entiende que puede ser suficientemente representativa del tráfico que arriba a la BS de WiMAX.

Por otro lado, el tráfico que arriba a una SS puede ser descripto por el tráfico que arriba a una sola estación tomado de la misma muestra.

Se seleccionó la muestra LBL-PKT-4, que es una traza de una hora entre el Lawrence Berkeley Laboratory y el resto del mundo. También influyó en la elección de esta muestra, el que contiene la información sobre la identidad de las estaciones origen y destino, y que el tráfico TCP/IP es amplia mayoría. Se trata de la misma muestra que empleó Paxon en (2), por lo tanto es una traza histórica. Se puede acceder a la misma siguiendo el enlace de (3).

Esta muestra pública, de tráfico autosimilar se descompuso en tráfico por estación y por sentido, mediante procedimientos de Matlab.

Posteriormente se analizó las características del tráfico total y del tráfico por estación, buscando las relaciones entre ambos.

El análisis consistió en determinar la autosimilaridad de los tráficos usando como herramientas los diagramas con escalas logarítmicas LDs (*Logscale Diagrams*), y estudiar el comportamiento de las dependencias de largo alcance LRD (*Long Range Dependencics*) observando las funciones de autocorrelación ACF (*Auto Correlation Functions*).

También se determinó las características *heavy-tailed* del tráfico por estación comparándolo con el tráfico total, mediante el análisis de las funciones de distribución acumulativa complementaria CCDF (*Complementary Cumulative Distribution Function*).

2.2 Análisis de autosimilaridad mediante los LDs

2.2.1 Conceptos sobre la regulación y la autosimilaridad

Como es sabido, referencias (4) y (5), los LDs grafican $Y(j) = \log_2 E(d_{jk}^2)$ versus $\log_2(j)$ u octava j . Si $Y(j)$ se se puede aproximar a una línea recta con pendiente α , el diagrama

corresponde a un tráfico con coeficiente de Hurst $H = \frac{1+\alpha}{2}$, si además $0,5 < H \leq 1$ el tráfico es autosimilar.

Se puede afirmar sin temor a equivocarse que un tráfico, no posee características autosimilares en todas sus escalas, cuando está sometido a limitaciones o regulaciones importantes que afectan a la mayoría de los emisores y en forma simultánea.

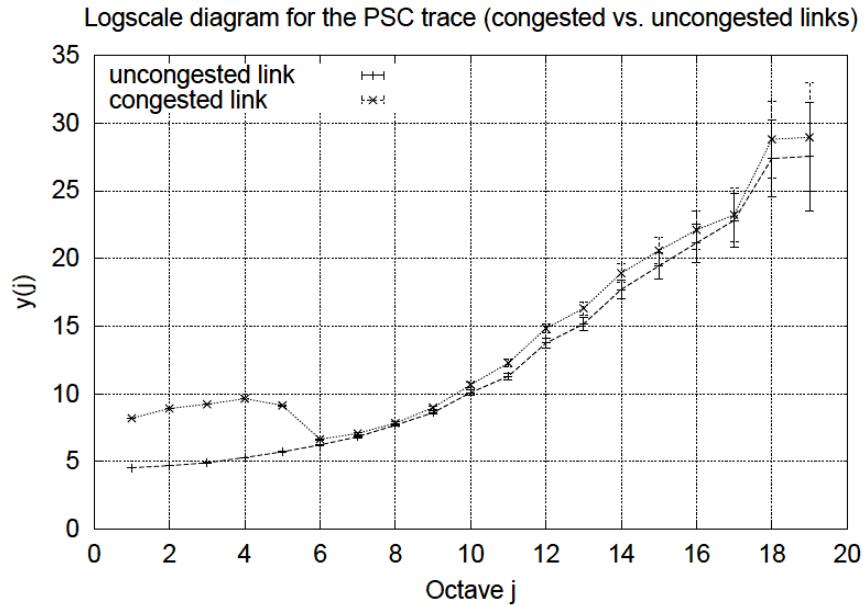


Fig. 2-1: LD mostrando efectos de la congestión en el tráfico. Tomada de (6).

Por ejemplo, una regulación “natural”, la impone la red cuando existe congestión, produciéndose una limitación en la velocidad. Uhlig y colaboradores en (6) y (7), muestran en los LDs, la influencia de los enlaces congestionados.

Las trazas de tráfico, Fig. 2-1, provenientes de PCS (Pittsburgh Supercomputing), muestra en las octavas 1 a 5, el efecto cuando el enlace está congestionado, en este caso el LD presenta características exponenciales, más que autosimilares.

2.2.2 Autosimilaridad del tráfico de la muestra LBL-PKT4

En la Fig. 2-2 se muestran los *logscale diagrams* del tráfico recibido, correspondientes a 10 estaciones seleccionadas, de las 1538 registradas en la muestra LBL-PKT4. Se han elegido las primeras 10 que tienen cada una más de 10000 paquetes recibidos, lo que corresponde a un 24,30% del total de paquetes. También se muestra el LD de la superposición de los tráficos transmitidos por las estaciones seleccionadas y el LD del tráfico total. Se han aproximado estos últimos LDs con unas rectas indicando la pendiente de las mismas.

En la Fig. 2-3 se muestra la misma información pero para el tráfico transmitido. Los paquetes transmitidos por estas mismas estaciones corresponden a un 28,55% de la muestra.

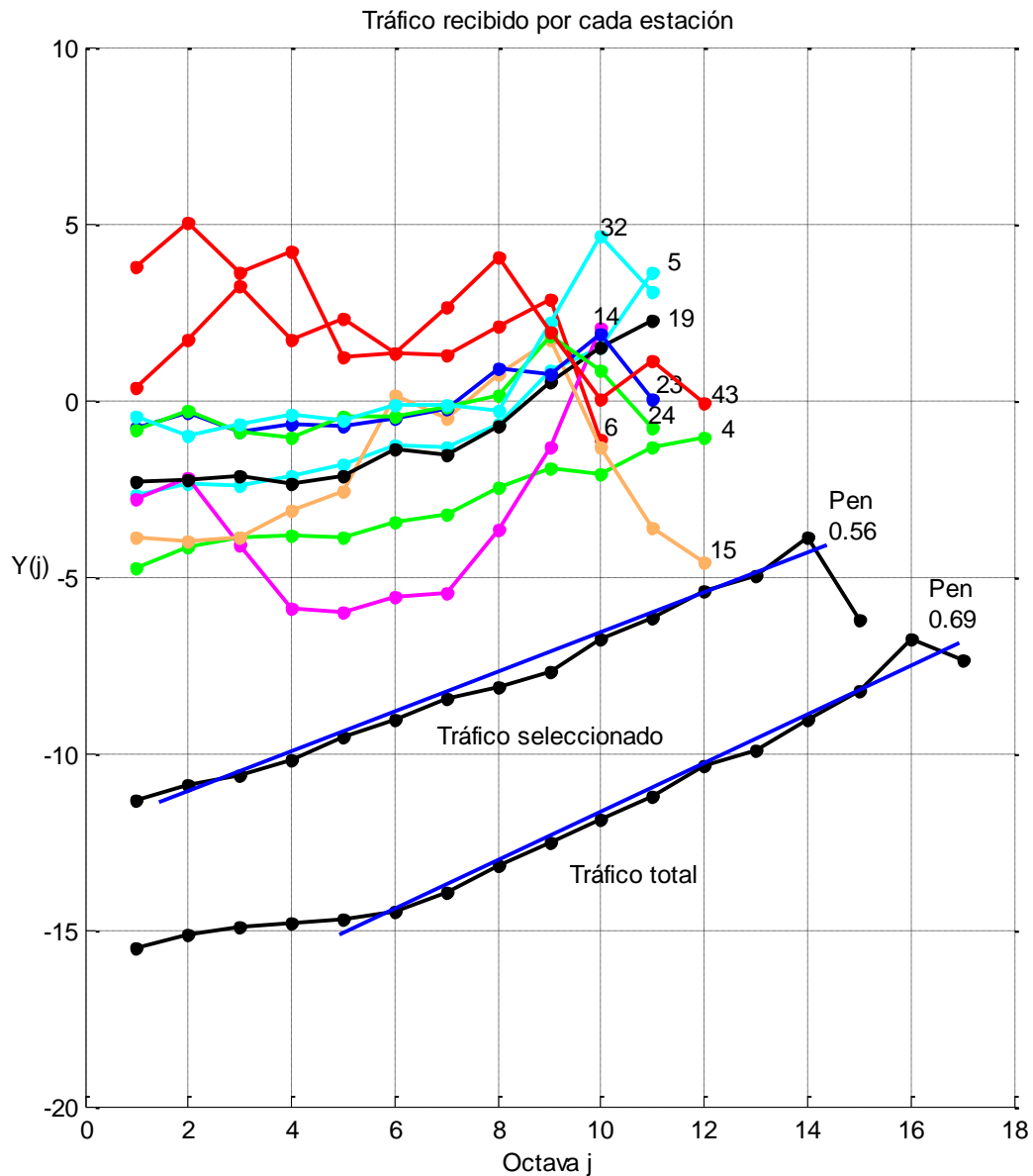


Fig. 2-2: LDs de los tráficos recibidos por cada una de las 10 estaciones seleccionadas (24,30% del tráfico total de la muestra LBL-PKT-4), la superposición del tráfico recibido por las 10 estaciones y el tráfico total de la muestra como referencia.

Un análisis de las Fig. 2-2 y Fig. 2-3 permite afirmar lo siguiente:

- En las estaciones seleccionadas de la muestra, el comportamiento estadístico del tráfico es muy parecido en ambos sentidos, tanto en cantidad como en calidad, ya sea comparando individualmente cada estación o comparando la superposición de las estaciones seleccionadas. Se especula que esto se debe a la proliferación de ACKs para segmentos individuales y/o tráfico interactivo.

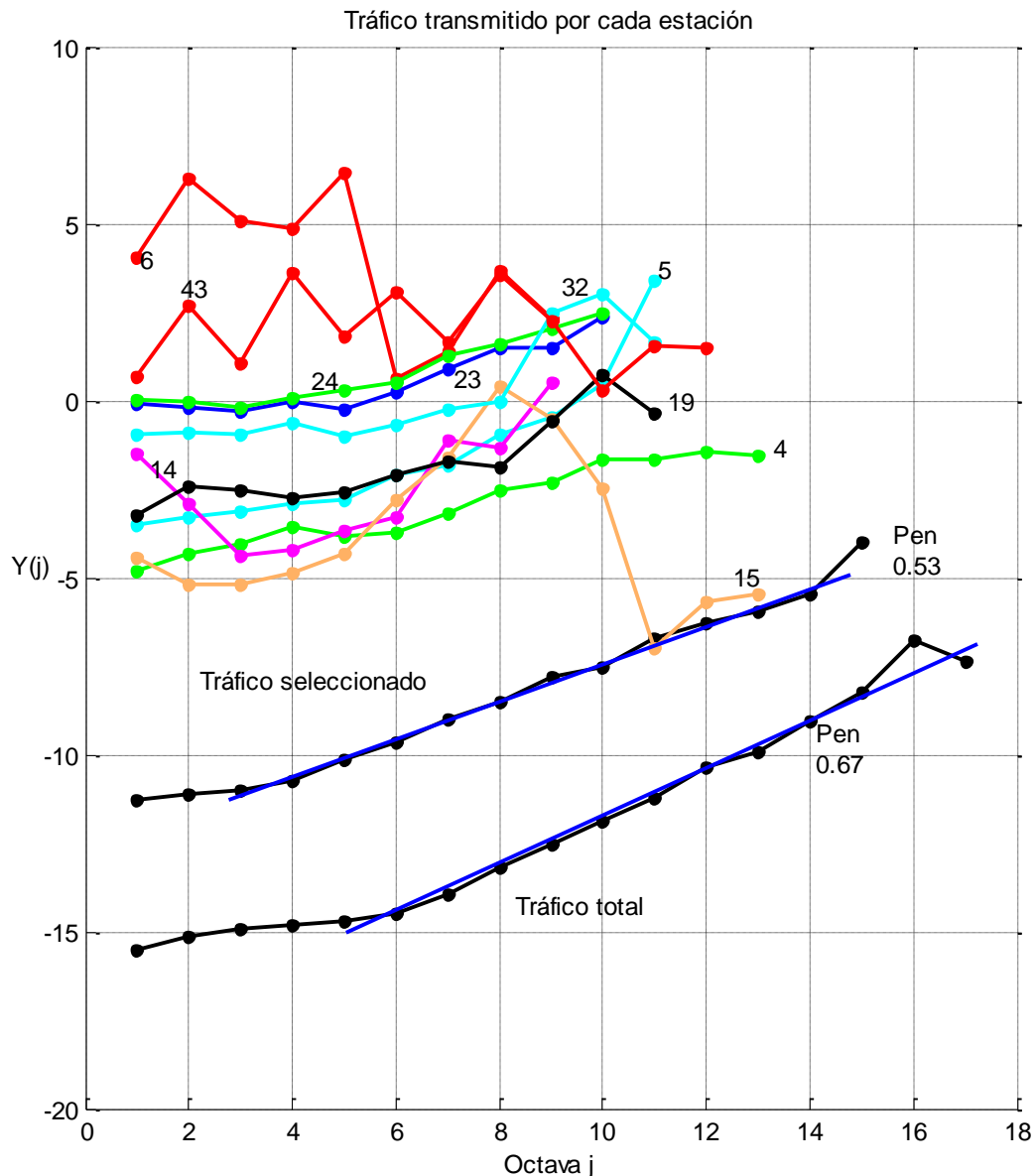


Fig. 2-3: LDs de los tráficos transmitidos por cada una de las 10 estaciones seleccionadas (28,55% del tráfico total de la muestra LBL-PKT-4), la superposición del tráfico transmitido por las 10 estaciones y el tráfico total de la muestra como referencia.

- El tráfico total es autosimilar entre la octava 6 y 15, $H = \frac{1+0,69}{2} = 0,845$. Entre las octavas 1 y 6, la pendiente es casi nula lo que puede corresponder a una distribución de tiempos interarribo exponencial. Esto último puede atribuirse a la concurrencia de varias causas, congestión, encolamiento de paquetes en los *routers*, etc. En general limitaciones provocadas por las características de los dispositivos de la red. Esto tiene sentido ya que los encolamientos debido a la congestión, si es pasajera, solo modifican el comportamiento del tráfico a corto plazo, y en los LDs las octavas bajas se corresponden con las componentes de alta frecuencia.
- El tráfico transmitido o recibido por las estaciones individuales, no puede considerarse autosimilar, excepto algunos tramos, como el tramo 5-9 de los

paquetes recibidos por la estación 4 o el tramo 1-5 de los paquetes enviados por la estación 5, por citar algunos ejemplos de los tramos más largos.

- Una pendiente global positiva, pero que no se aproxima a una línea recta, no indica autosimilaridad, indica únicamente que los coeficientes de detalle de la descomposición *wavelet* correspondientes a dilataciones grandes (o frecuencias bajas) tienen una presencia más importante que los coeficientes de dilataciones pequeños (o frecuencias altas).
- Otra característica importante observada en los LDs de la Fig. 2-2 y Fig. 2-3 es como un número relativamente pequeño de estaciones, alcanza para construir un tráfico con características autosimilar. Para el tráfico recibido de las estaciones seleccionadas $H = \frac{1+0,56}{2} = 0,78$. Para el tráfico transmitido $H = \frac{1+0,53}{2} = 0,765$.

La anterior observación muestra la robustez del fenómeno autosimilar. Es suficiente la superposición de muy pocas estaciones, en este caso solo 10 estaciones, cuyo tráfico individual no es autosimilar, aunque como se mostrará son en su mayoría *heavy-tailed*, construyen uno que sí lo es.

Es conocido que el protocolo TCP es un fuerte regulador del tráfico. Dispone de un sistema de control de flujo para que los receptores lentos moderen a los emisores rápidos, evitando ser saturados. El receptor indica al transmisor, en el campo *window* del encabezado TCP, la cantidad de bytes que puede recibir. En consecuencia, el emisor gradúa el tamaño de sus segmentos acorde a la capacidad de las colas y o capacidad de procesamiento del receptor (8 pág. 220).

TCP posee también un mecanismo de regulación cuando detecta congestión en la red. Existen dos indicadores de congestión por pérdida de paquetes, *timeout* en espera del reconocimiento (ACK) de un segmento, y recepción de ACKs duplicados (9 pág. 310).

El segundo se produce porque al notar el receptor la falta de un segmento de la secuencia, por cada nuevo segmento recibido tras la discontinuidad, el receptor responde con el mismo ACK correspondiente al último segmento recibido antes del faltante. Después de varios ACKs duplicados recibidos por el emisor, cuya cantidad depende de su configuración de TCP, se reenvía el segmento faltante y se reduce el ritmo de envío en la forma determinada por un algoritmo.

Existe un tercer indicador, aunque no necesariamente indica congestión ni pérdida de paquetes. Es el mensaje explícito ICMP *source quench*. El mensaje ICMP puede ser generado por un *router* (aunque algunas RFC lo desaconsejan en el caso de congestión porque consideran que la agrava) o una estación receptora cualquiera, cuando sus colas, o su

capacidad de procesamiento, están al límite o desbordadas. Tras recibir el mensaje, el transmisor reduce drásticamente el ritmo del envío de segmentos a esa conexión.

A pesar de estas regulaciones que impone el protocolo TCP, o cualquier otro protocolo de transporte, salvo congestión importante, el tráfico de la red no pierde su característica autosimilar (10 pág. 8).

2.3 Análisis de autosimilaridad mediante la función de autocorrelación

Considerando únicamente estadísticas de 2^{do} orden, una distribución autosimilar equivale a una distribución LRD. Una discusión sobre la diferencia entre ambas, se puede hallar en (6) y en (5 pág. 21).

Por definición, una distribución es LRD cuando la cola de la ACF $r(k)$ decae lentamente. Decaimiento lento implica que la sumatoria de las ACF no sea sumable

$$\sum_{k=-\infty}^{\infty} r(k) = \infty$$

donde

$$r(k) = \frac{\gamma(k)}{\sigma^2}$$

y

$$\gamma(k) = E(X_i - \mu)E(X_{i+k} - \mu)$$

siendo $\gamma(k)$ la autocovariancia. La sumatoria tiende a ∞ cuando se satisface que la cola de $r(k)$ decae hiperbólicamente

$$r(k) = Ck^{-\beta}$$

para $k \rightarrow \infty$, con $0 < \beta < 1$

Por otro lado en (10 pág. 18), se define que un proceso $X(t)$ es exactamente autosimilar de 2^{do} orden, con parámetro Hurst H , con $1/2 < H \leq 1$, si

$$\gamma(k) = \frac{\sigma^2}{2} ((k+1)^{2H} - 2k^{2H} + (k-1)^{2H})$$

para todo $k \geq 1$. Es simple demostrar (11 pág. 15) que para $k \rightarrow \infty$,

$$r(k) = Ck^{-(2-2H)}$$

por comparación $H = 1 - \frac{\beta}{2}$ y $\beta = 2 - 2H$

Empleando la misma muestra de tráfico LBL-PKT4, se calculó la ACF del tráfico de las primeras 10 estaciones que superaran cada una 10000 paquetes, para determinar su

autosimilaridad evaluando el comportamiento asintótico del diagrama doble logarítmico de las funciones de autocorrelación, Fig. 2-4 y Fig. 2-5.

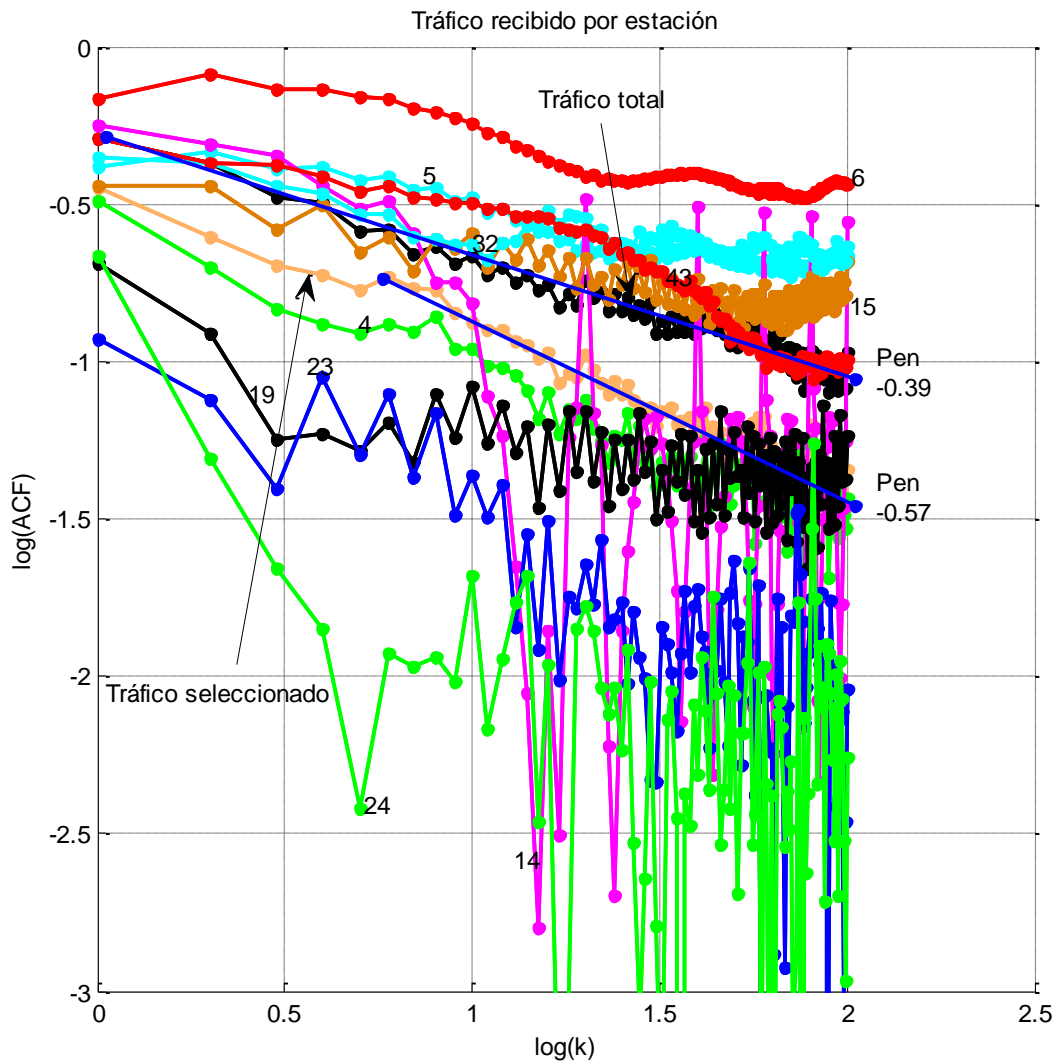


Fig. 2-4: $\log(\text{ACF})$ versus $\log(k)$ del tráfico recibido por estaciones seleccionadas (24,30% de la muestra LBL-PKT-4).

El análisis de las figuras permite afirmar que

- Solo presentan asíntotas claramente, el tráfico total, el tráfico seleccionado y en una forma menos pura la estación 4 en ambos sentidos. Estos comportamientos autosimilares ya se percibían en los LDs analizados previamente.
- El tráfico total presenta para el recibido y el enviado, un $H = 1 - \frac{0,39}{2} = 0,805$, valores un tanto inferiores comparados con el resultado obtenido del análisis con los LDs donde se obtuvo $H \approx 0,85$.
- Estas diferencias son comunes cuando se usan procedimientos diferentes para evaluar el coeficiente Hurst, También es muy importante la cantidad de valores de la muestra cuando H es alto. Una discusión sobre la exactitud de los estimadores del coeficiente de Hurst y sobre la cantidad de valores que debe tener una muestra para

un determinado nivel de confianza se puede hallar en (11) y en las referencias del mismo.

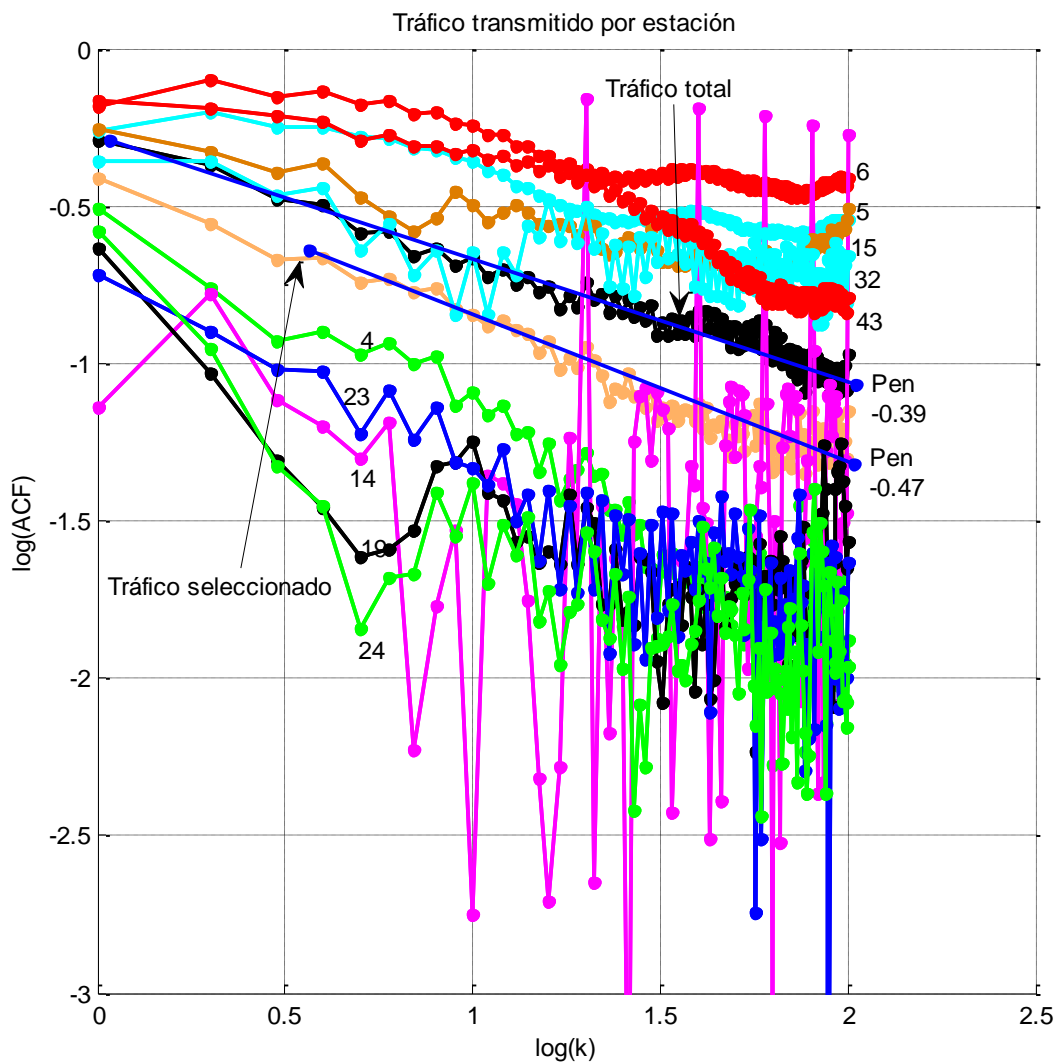


Fig. 2-5: $\log(\text{ACF})$ versus $\log(k)$ del tráfico transmitido por estaciones seleccionadas (28.55% de la muestra LBL-PKT-4).

— Tanto para el tráfico recibido como para el transmitido de las diez estaciones seleccionadas $H = 1 - \frac{0,47}{2} = 0,765$ y $H = 1 - \frac{0,57}{2} = 0,715$, la ACF decae más rápidamente que para el tráfico total.

2.4 Análisis de la *heavy-tailness* mediante la CCDF

Varios autores, han mostrado que el ingrediente imprescindible en la construcción de tráfico autosimilar, es que el tráfico individual que lo constituye, posea una distribución *heavy-tailed* (12) (13) (14). La realidad muestra, dada la robustez del fenómeno, que con una determinada masa crítica de tráfico por estaciones o conexiones que sean *heavy-tailed* es suficiente.

La calificación *heavy-tailed*, expresión que no tiene un equivalente en español de uso generalizado, se refiere a la caída lenta de la curva de la función de distribución acumulativa complementaria CCDF (*Complementary Cumulative Distribution Function*).

Sea X una variable aleatoria, que para el tráfico puede ser el tiempo interarribo o la cantidad de arribos por intervalo de tiempo, por ejemplo 100ms, con una función de distribución acumulativa CDF $F(x) = P[X \leq x]$ y función de distribución acumulativa complementaria CCDF $\overline{F}(x) = 1 - F(x) = P[X > x]$. La distribución es *heavy-tailed* si

$$P[X > x] \sim L(x)x^{-\alpha}, \quad x \rightarrow \infty$$

donde $0 < \alpha < 2$ es el índice o parámetro que define la forma de la cola. $L(x)$ es una función de x que varía lentamente en el infinito. Entonces una distribución es *heavy-tailed* si la CCDF decae asintóticamente hiperbólica, en escala doble logarítmica. Esto en contraste con distribuciones *light-tailed*, tales como la exponencial y normal las cuales poseen una caída exponencial.

El término asintótico se refiere a un diagrama de la CCDF en escalas doble logarítmicas, donde la pendiente de las asíntotas es $-\alpha$

$$-\alpha = \frac{d \log \overline{F}(x)}{d \log (x)}$$

2.4.1 Función de distribución Pareto

La más simple distribución *heavy-tailed* es la distribución Pareto. Existen muchas otras tales como la Log-normal, la Lévy, la Weibull, la Burr, la Cauchy, la *t-distribution*, la *skew lognormal cascade*, por nombrar las más reconocidas.

La función de densidad de probabilidad (PDF) de la distribución Pareto exhibe un comportamiento potencial,

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

donde α es el parámetro de forma, $\alpha > 0$, y b es el mínimo valor permitido de x , $0 < b \leq x$. Su CDF y CCDF están dadas respectivamente por

$$F(x) = P[X \leq x] = 1 - \left(\frac{b}{x}\right)^\alpha$$

$$\overline{F}(x) = P[X > x] = \left(\frac{b}{x}\right)^\alpha$$

es fácil mostrar que el valor medio $E(x) = \frac{b\alpha}{\alpha-1}$

En la Tabla 2-1, se muestra un resumen de los parámetros más importantes de la distribución Pareto, a los fines de esta tesis

Tabla 2-1 Parámetros más importantes de la distribución Pareto.

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

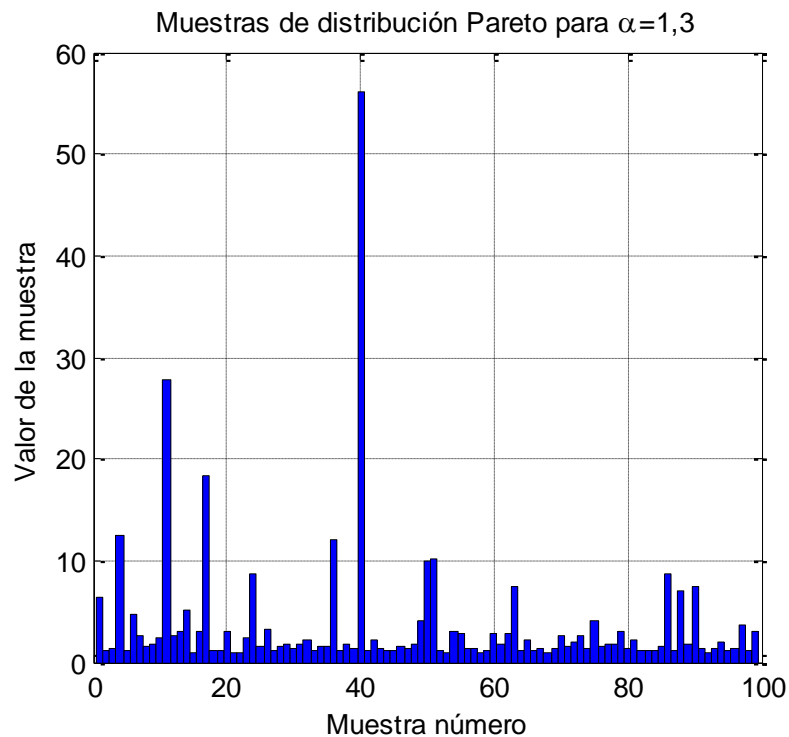


Fig. 2-6: 99 muestras generadas por un simulador de la distribución Pareto con $\alpha=1,3$ y $b=1$

En la Fig. 2-6 se grafican 99 muestras de una distribución Pareto, generadas en Matlab. Nótese que si bien la mayoría tiene un valor menor que 5, siempre hay posibilidad de valores extremadamente altos, debido a la variancia infinita de la distribución para $1 < \alpha \leq 2$.

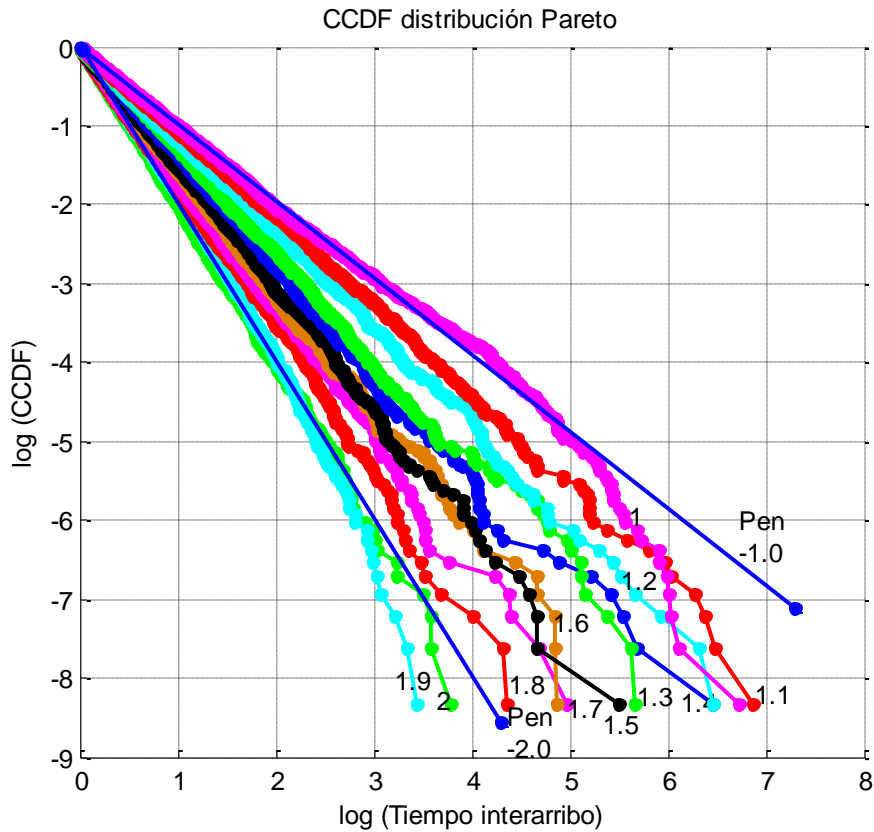


Fig. 2-7: CCDF de muestras generadas por un simulador de la distribución Pareto con $\alpha=1$ a 2 en pasos de 0,1.

En la Fig. 2-7 se grafican 11 curvas de la CCDF de una distribución Pareto, cada una obtenida de 2048 muestras generadas en Matlab, para $b = 1$ y $\alpha = 1$ a 2 en escalones de 0,1. También se ha graficado las pendientes -1 y -2 . La pendiente -2 corresponde a una distribución exponencial, y la -1 a la cola más pesada, que se emplea para generar tráfico con coeficiente de Hurst cercanos a 1.

Si $\alpha \leq 2$, la distribución tiene variancia infinita, y si $\alpha < 1$ la distribución además tiene valor medio infinito. A medida que α decrece una porción arbitrariamente larga de la PDF puede estar en la cola de la distribución. Cuando la función Pareto se aplica para modelar tráfico $1 < \alpha \leq 2$, el valor medio es finito y la variancia infinita.

En la Fig. 2-8 se muestra que un tráfico con un distribución Pareto no muestra ninguna tendencia a tener pendiente positiva en sus LDs, cualquiera sea su parámetro de forma α , por lo tanto un tráfico con una distribución *heavy-tailed*, no implica la autosimilaridad.

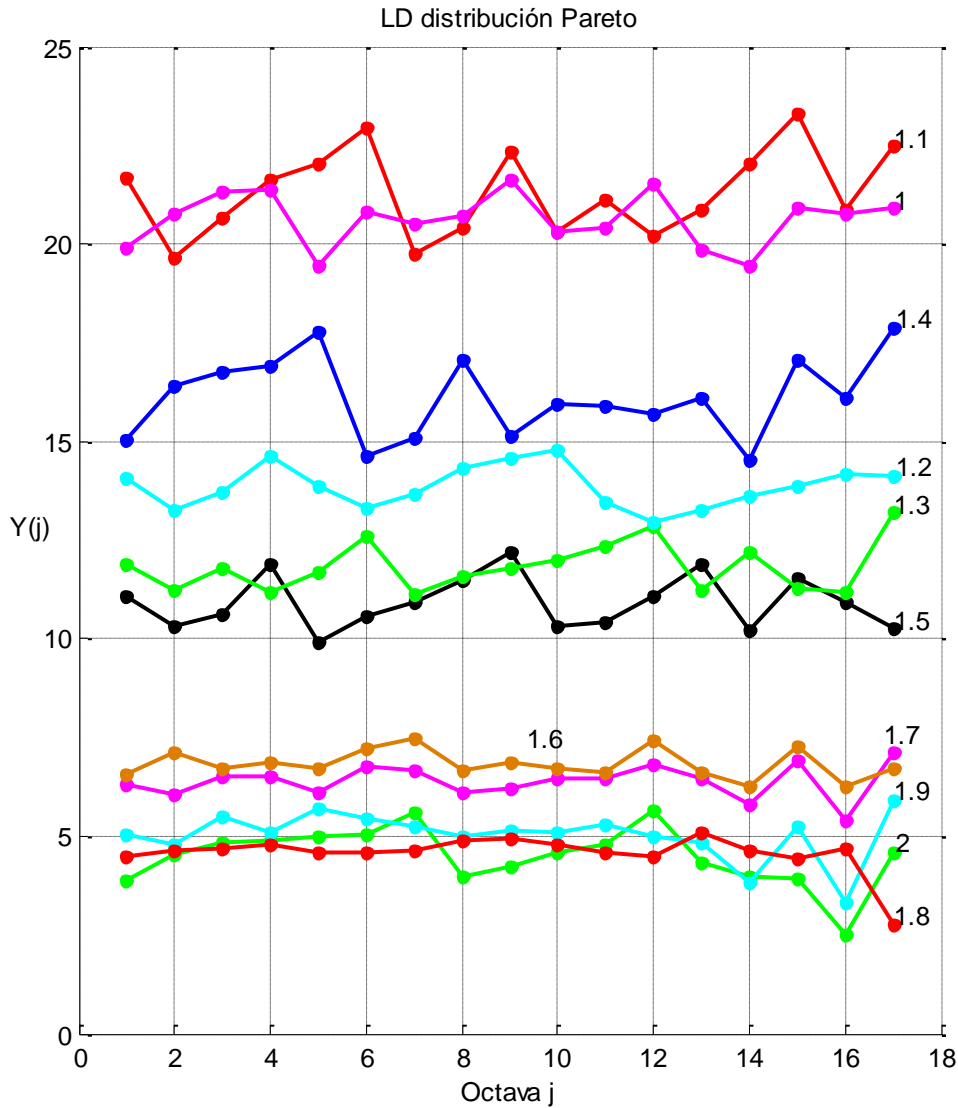


Fig. 2-8: LD de muestras generadas por un simulador con distribución Pareto con $\alpha=1$ a 2 en pasos de 0,1.

2.4.2 CCDF de la muestra

En las Fig. 2-9 y Fig. 2-10 se muestran los CCDF de tráfico recibido y transmitido respectivamente, de estaciones seleccionadas (las primeras 10 registradas con más de 10000 paquetes), tomadas de la muestra LBL-PKT-4. También se han indicado las pendientes $-\alpha$ en ambas figuras. En las mismas se aprecia que en promedio los tráficos individuales de las estaciones tiene una distribución asintóticamente *heavy-tailed*, $1 < \alpha < 2$, salvo algunas claras excepciones como las de las estaciones 14 y 5.

La superposición del tráfico, recibido y enviado, de las 10 estaciones seleccionadas es claramente *light-tail*, $\alpha \geq 2$, y mayor aún es α para el tráfico total.

Este resultado tiene lógica, ya que si se supone el extremo de que el tráfico sea tan grande que los paquetes viajen pegados uno tras otro, el tiempo interarribo sería constante y $\alpha = \infty$.

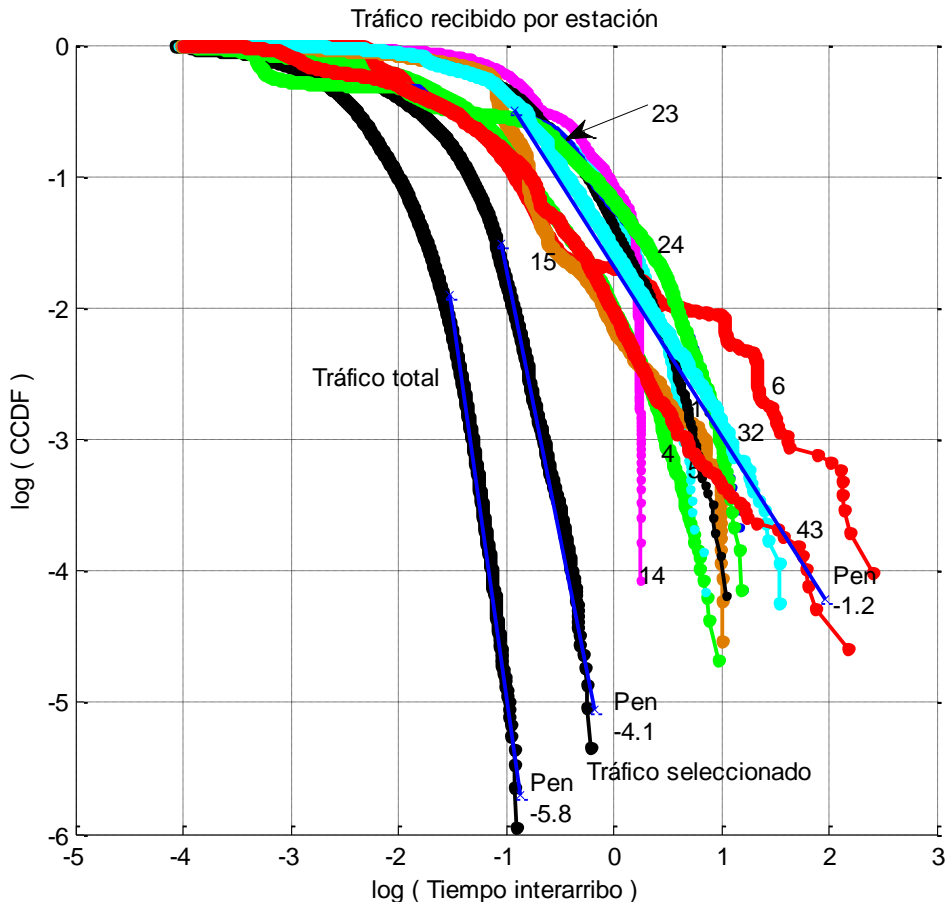


Fig. 2-9: $\log(\text{CCDF})$ versus $\log(\text{Tiempo interarribo})$ del tráfico recibido por 10 estaciones seleccionadas (24,30% de la muestra LBL-PKT-4), la superposición de los mismos, el tráfico total y las pendientes de las asíntotas.

2.5 Resumen sobre la comparación entre el tráfico por estación y el tráfico total

De lo analizado se concluye que

- El tráfico total compuesto de muchas estaciones es autosimilar, o autosimilar por tramos en el caso de regulaciones que afecten a muchos individuos simultáneamente, como en el caso de la congestión. Esto se confirmó con el análisis de los LDs y los diagramas de las ACFs. No hace falta abundar en justificaciones ya que esto ha sido confirmado reiteradamente por muchos investigadores, mediante análisis de muestras de tráfico recolectadas. No hay que descartar que el tráfico pueda ser no autosimilar por la naturaleza, probablemente temporal debida a un fenómeno no identificado, del tráfico de las estaciones que lo componen sin necesidad de que hubiera regulaciones importantes.

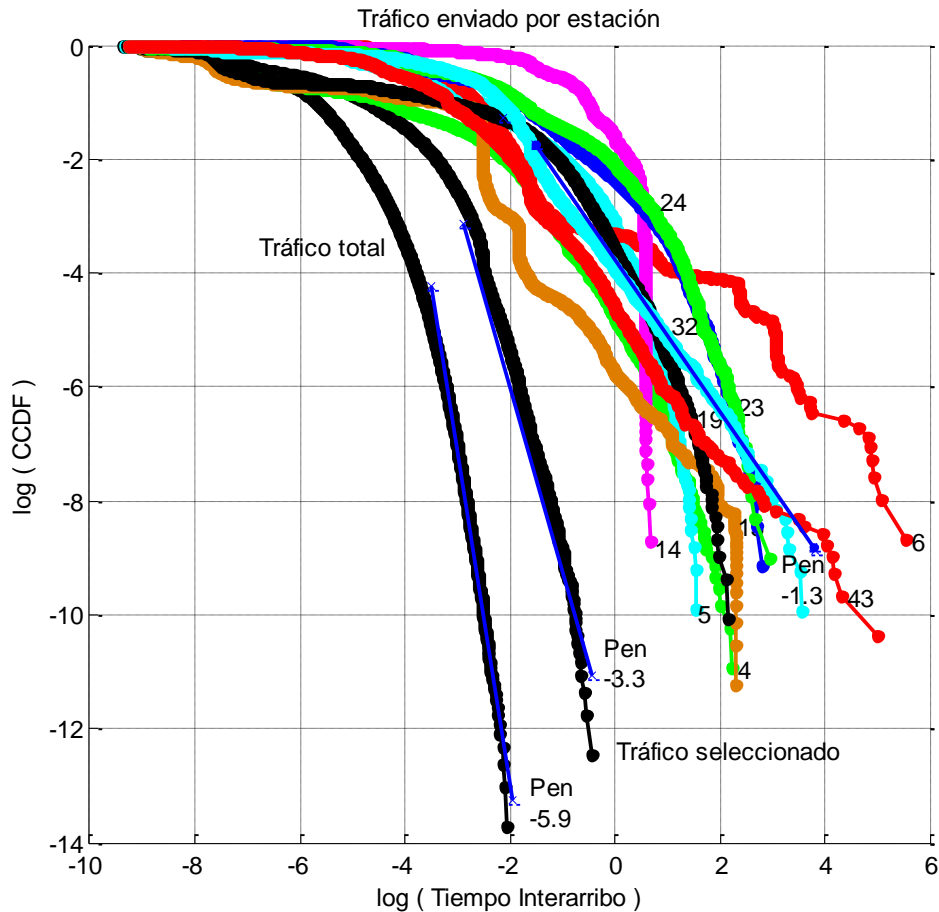


Fig. 2-10 $\log(\text{CCDF})$ versus $\log(\text{Tiempo interarribo})$ del tráfico transmitido por 10 estaciones seleccionadas (28.55% de la muestra LBL-PKT-4), la superposición de los mismos, el tráfico total y las pendientes de las asíntotas.

- El tráfico por estación difícilmente pueda ser autosimilar, tal como queda en evidencia del análisis de los LDs y los diagramas de las ACFs. Se plantean las siguientes dos hipótesis justificativas que quedan abiertas para su prueba o discusión:
 - 1 Es insuficiente la cantidad y variedad de aplicaciones que generan tráfico de red, para que puedan provocar variabilidad en todas las escalas de tiempo. El comportamiento de una persona no es naturalmente multitarea, lo cual limita el solapamiento de aplicaciones que acceden al medio simultáneamente.
 - 2 El acceso a la red de las diferentes aplicaciones, pierde aleatoriedad debido al ordenamiento natural en el acceso a red impuesto por la utilización compartida de una cadena de recursos únicos, tal como el *stack* TCP/IP y un único acceso físico al medio, impidiéndose en forma natural explosiones de tráfico.
- El tráfico por estación en la mayoría de los casos es *heavy-tailed* tal como se muestra en el análisis de la pendiente de las curvas CCDF.
- Un resultado importante del análisis de la muestra LBL-PKT-4, especialmente para quién identifica equivocadamente autosimilaridad o LRD con una distribución *heavy-tailed*, es que el tráfico por estación es claramente *heavy-tailed*, pero el tráfico total claramente no lo es.

3. Análisis de las características del tráfico en un sistema WiMAX

Desde Leland y colaboradores (15) a esta parte, es aceptado que el tráfico en LAN, y en los accesos a Internet, no responde al simple y cómodo modelo de distribución de Poisson, sino más bien a una distribución autosimilar.

En esta parte se investigará si el tráfico que accede a WiMAX se puede considerar autosimilar o responde a otra distribución.

3.1 Definiciones

En la Fig. 3-1 se muestran las definiciones de interfaz de aire, interfaz de cable, tráfico que entra y tráfico que sale. Es común en el caso de las SSs (*Subscribe Stations*) portátiles, que el dispositivo WiMAX esté integrado con el hardware de la CPU del suscriptor, sin la existencia de un cable físico, tal es el caso de una *notebook* o de un teléfono celular con WiMAX. Sin embargo se seguirá usando esta denominación para diferenciar a la conexión, cualquiera sea ésta, de la de aire.

Nótese que los términos entra y sale se han definido cómo ve el sentido del tráfico las estaciones WiMAX, tanto para la BS (*Base Station*) como para la SS.

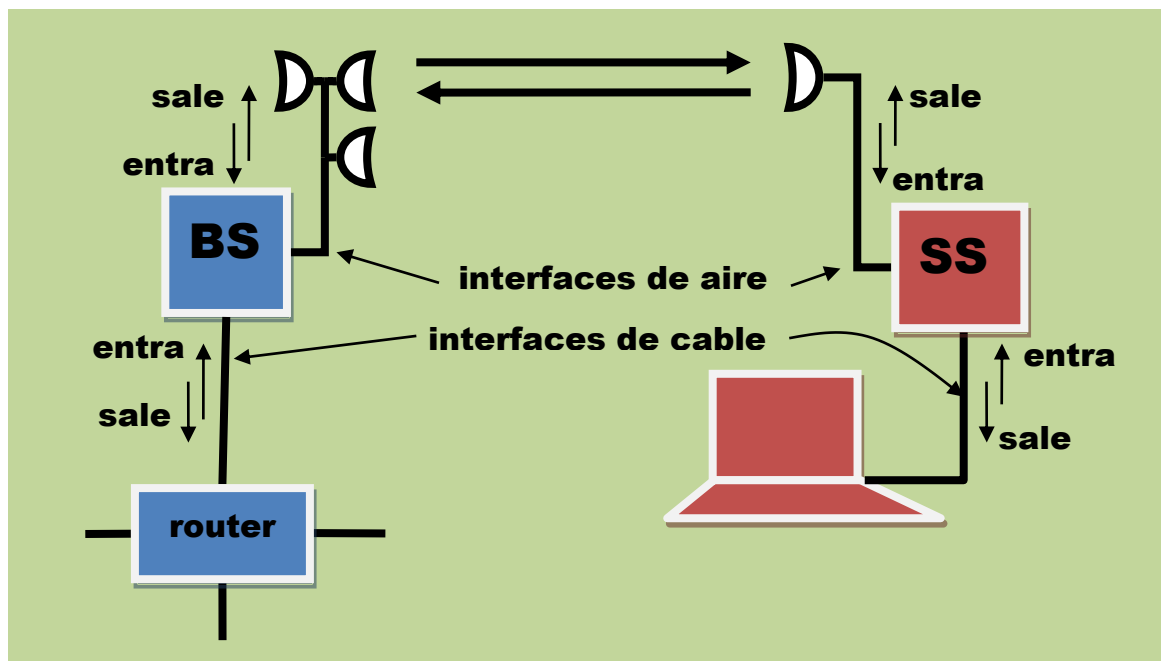


Fig. 3-1: Definiciones: interfaces de aire y de cable de un sistema WiMAX compuesto por una estación base (BB) y por una estación suscriptor (SS). Los términos entra o sale son siempre como los ve la estación WiMAX.

3.2 Regulación en WiMAX

En WiMAX, cuando el enlace está lo suficientemente solicitado como para que todas sus ranuras estén ocupadas todo el tiempo, se produce una regulación del tráfico similar a una congestión.

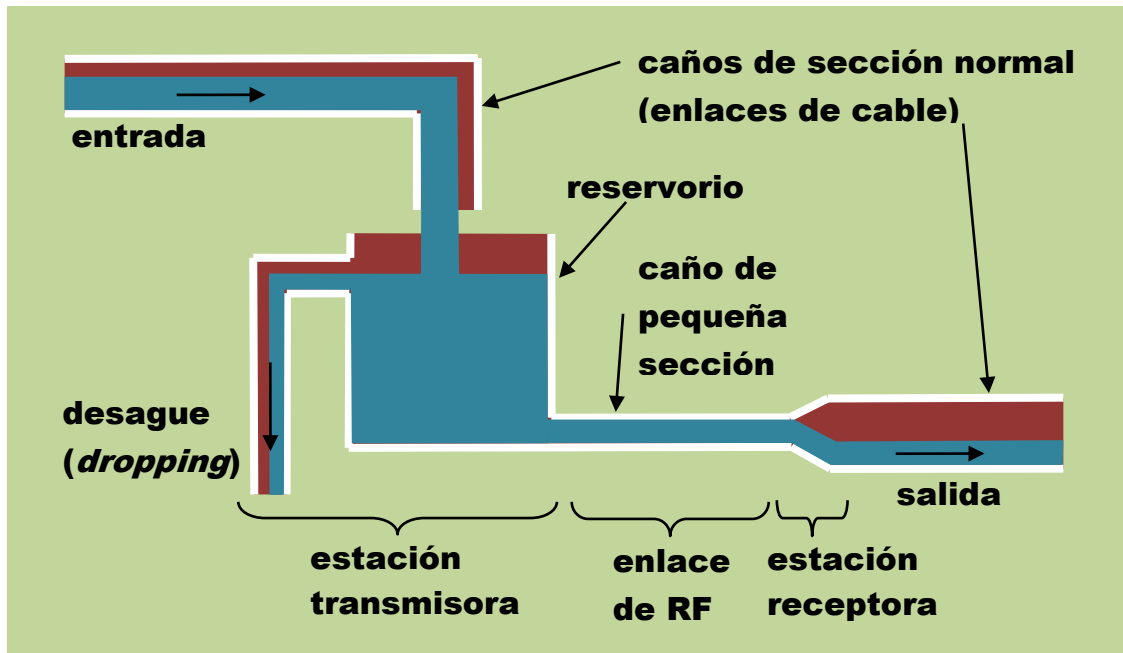


Fig. 3-2: Analogía hídrica mostrando el efecto congestivo de un enlace WiMAX con capacidad inferior a la conexión de cable que lo alimenta.

En la Fig. 3-2 se muestra una instalación hídrica en que un caño de sección grande alimenta a un caño de menor sección, donde se establece una analogía simplificada, solo en un sentido, con un enlace WiMAX congestionado. Los caños de sección normal representan las conexiones de red cableada, el caño de pequeña sección representa el estrechamiento producido por la limitada capacidad del enlace de RF. Si el caudal del caño de entrada es mayor que la capacidad del caño de pequeña sección, no existirá reservorio que no se rebalse por el desague transcurrido un tiempo. En una estación WiMAX el agua del desague equivale al descarte de paquetes o *dropping*.

En WiMAX cuando el flujo que entra es mayor que la capacidad del enlace, lo que se produce es una congestión selectiva. Debido a que es un sistema con controles de QoS, existe un esquema jerárquico donde los datos tienen diferentes prioridades según sea el tipo de información que transportan y a quienes pertenezcan (diferentes contratos, SLA *Service Level Agreement*).

Cuando existe congestión, existen conexiones a los cuales no se les modifica substancialmente el perfil de su tráfico, debido a su prioridad, mientras que el tráfico de otras conexiones sufren transformaciones substanciales, debido a la demora o *dropping*.

Se entiende que los paquetes que sufren congestión afectan el comportamiento en el corto plazo, ya que en mediano y largo plazo las capas superiores de los emisores, TCP y aplicación, autorregulan la frecuencia de emisión de sus paquetes para evitarla. Véase en los LDs de la Fig. 2-1, como la congestión no afecta las escalas grandes (frecuencias bajas).

3.3 Interfaz de aire

El mecanismo que usa WiMAX para controlar el tráfico en la interfaz de aire, consiste en la concesión de *slots* (ranura, unidad mínima en el plano frecuencia tiempo de OFDMA) conforme a las necesidades de los usuarios, arbitrado por el SLA.

Un sistema con QoS como WiMAX, es un poderoso regulador congénito, que busca tener siempre sus *slots* de aire rebosantes de bytes de información. Si la demanda para tráfico de alta prioridad no es suficiente para mantener repletos los *slots*, se conceden los *slots* sobrantes para tráfico con menor prioridad, siempre hambriento de ancho de banda.

Los enlaces de aire, frecuentemente son cuellos de botella de las redes. El *scheduler* de WiMAX, entre sus principales objetivos está el de mantener el máximo *throughput* posible, durante la mayor cantidad de tiempo.

Para el cumplimiento de este objetivo, la regulación que impone WiMAX en su interfaz de aire, es extrema en ambos sentidos, concediéndose byte por byte a los transmisores. En los horarios en que la demanda del enlace supera a la oferta, no es desatinado hablar de un *bitrate* casi constante, salvo alteraciones por cambios en las características de propagación. Nada más alejado de una distribución autosimilar.

Estas afirmaciones no excluyen que puede existir un tráfico autosimilar en la interfaz de aire de las estaciones subutilizadas o en horarios de poca demanda. Sin embargo, el análisis de la autosimilaridad del tráfico por la interfaz de aire, no es de mucho interés, ya que por más explosivo que pueda sea el tráfico por momentos, debido esto a la autosimilaridad del mismo, siempre será recortado al *bitrate* máximo del enlace.

3.4 Interfaz de cable

3.4.1 Tráficos que salen por las interfaces de cable

El objetivo de este trabajo es generar un tráfico que pueda ser útil para el desarrollo y prueba de la eficiencia de los *schedulers* de WiMAX. El tráfico que sale por las interfaces de cable de las estaciones WiMAX es el producto elaborado entregado por el *scheduler*, por lo tanto no se considerará sintetizar este tráfico.

Sin embargo no puede desconocerse que tienen un efecto indirecto, producido por la realimentación que se produce en las capas superiores, sobre el tráfico en la dirección opuesta.

Si el *scheduler* se ve obligado a descartar paquetes (*dropping*) ya sea porque sus colas están llenas o porque paquetes sensibles al tiempo (*deadline-sensitive*) no pudieron ser entregados en el plazo acordado (QoS), las capas superiores, TCP y aplicación, se encargarán de ralentizar los envíos advertidas por la pérdida de segmentos o demoras en las entregas.

El comportamiento del usuario también puede ser parte del lazo de realimentación. Considérese un tráfico interactivo, por ejemplo un chat entre dos estaciones con un enlace WiMAX en el camino. Si el tráfico que envía una estación se vuelve lento debido al enlace, el usuario de la otra estación interlocutora ralentizará sus envíos, dado que la interactividad impone una secuencialización de la información entre preguntas y respuestas.

3.4.2 Tráfico de la interfaz de cable que sale de la BS

El tráfico que sale por la interfaz de cable de la BS, a nivel bits, es el mismo que el que transporta el *uplink subframe* de un TDD (*Time Division Duplexing*) *frame* de WiMAX, esto es el mismo que ingresó a la BS por su interfaz de aire. Para los casos de interés, un enlace suficientemente demandado, el tráfico está constituido por un flujo de paquetes con un *bitrate* casi constante, salvo las siguientes situaciones a saber:

- Cambio en las condiciones de propagación que originen un cambio en la modulación o codificación adaptativa (MCS, *Modulation Coding Scheme*) de un número significativo de estaciones, por ejemplo una tormenta con lluvia.
- Modificación de las características promedio de las SSs que suben datos a la BS. Esto puede suceder por ejemplo si se cae una estación base de la celda colindante, y estaciones más lejanas, por lo tanto con esquemas de modulación y codificación más robustas, lo cual implica un *bitrate* menor, comienzan a interactuar con la BS.
- Cambio de la relación temporal entre el *download subframe* y el *upload subframe*. Es normal que esta relación no se modifique y se mantenga constante, por ejemplo 4 (*download*) a 1(*upload*), como en ADSL.

3.4.3 Tráfico de la interfaz de cable que sale de la SS

El tráfico que sale por la interfaz de cable de cada SS, al igual que el que sale de la BS, es el resultado del procesamiento realizado por el *scheduler*. Pero el primero es más variable que el segundo.

La unión o superposición de todos los tráficos que salen de las SSs, componen a nivel bits, el *downlink frame* de la BS. Que igual que el *uplink frame* es aproximadamente constante si el enlace está suficientemente demandado. Sin embargo el que la suma sea constante no impone que los sumandos lo sean.

Este tráfico que sale de la interfaz de cable de las SSs depende principalmente de

- Demanda de las aplicaciones que lo producen.
- Contrato con el proveedor del servicio, SLA.
- Condiciones de propagación.
- Cambios en la cantidad y calidad de SSs activas con la BS.

3.4.4 Tráficos que entran por las interfaces de cable

Estos son los tráficos que componen los insumos básicos de los *schedulers*.

El *scheduler* de la BS, en el caso de los paquetes que entran por la interfaz de cable de la BS es el encargado de asignar en que *frame* serán transportados. No necesariamente hay asignación, puede haber *dropping* como se comentó anteriormente.

En el caso de los paquetes que entran por la interfaz de cable de las SSs, puede suceder que el *scheduler* de la BS se encargue de la asignación en el *uplink*, de todas las conexiones de las SSs, *slot* por *slot*, o que se encargue únicamente de asignar lotes de *slots* a cada SS, y los *schedulers* individuales de cada SS de programar la asignación de cada conexión en los *slots* del lote que le fue asignado.

El que sea el *scheduler* de la BS o de cada SS el encargado de la asignación de *slots* en los *uplinks frames*, no modifica apreciablemente las características del tráfico que sale de las SSs, y menos por mecanismos indirectos a los que entran por las interfaces de cable.

3.4.4.1 Tráfico de la interfaz de cable que entra a la SS

Crovella y Bestavros (16) fueron unos de los primeros en buscar posibles causas, en los niveles más altos del sistema, para que el tráfico de la WWW sea autosimilar. Distribución del tamaño de los archivos, efectos del *caching*, preferencias de los usuarios, el efecto del “tiempo para pensar” de los usuarios, y superposición de muchas transferencias de archivos en una LAN.

Explican la distribución autosimilar del tráfico de red en términos de los tiempos de transmisión individual basados en el mecanismo descrito en (17). Este último artículo, puntualiza que un tráfico autosimilar puede ser construido multiplexando una gran cantidad de fuentes individuales SI-NO (*ON-OFF sources*) que tienen intervalos de SI transmisión y de NO transmisión con distribución *heavy-tailed*. Este mecanismo puede corresponder a una red de estaciones de trabajo, cada una de las cuales puede estar silenciosa o transfiriendo datos a una velocidad constante.

Ellos recogieron evidencia mostrando que el tamaño de los archivos transferidos, y por lógica también el tiempo para transferirlos, tienen una distribución *heavy-tailed*, esto es el tiempo de SI. También evidenciaron que el tiempo de NO es *heavy-tailed* aunque menos *heavy*.

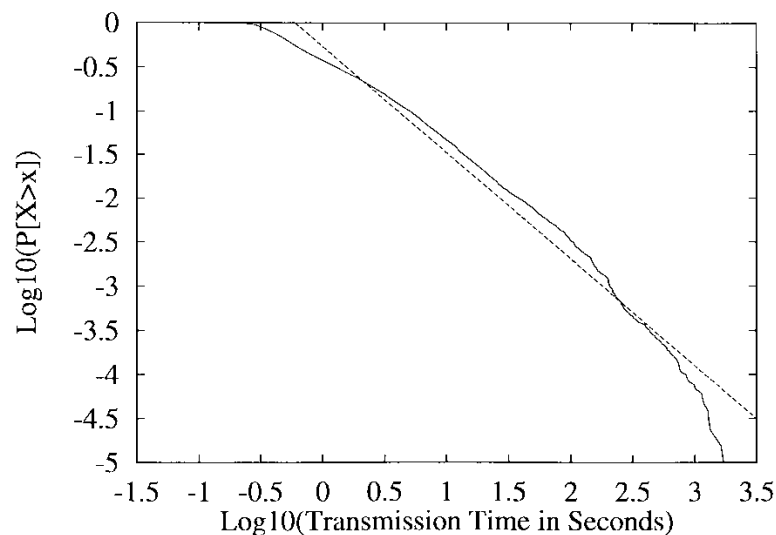


Fig. 3-3: CCDF del tiempo de transmisión de archivos en la WWW. La distribución es asintóticamente *heavy-tailed* con $\alpha_{SI} \approx 1,21$. Tomada de (16).

En la Fig. 3-3 se muestra la CCDF versus el tiempo de transmisión de los archivos, donde se aprecia el comportamiento asintótico *heavy-tailed* con $\alpha_{SI} \approx 1,21$ (tiempo SI). Un diagrama similar del tiempo de NO, muestra que el $\alpha_{NO} \approx 1,5$.

Con ayuda de $H = (3 - \min(\alpha_{SI}, \alpha_{NO}))/2$ (17), el coeficiente de Hurst resulta $H \approx 0,895$, el cual corresponde a un tráfico autosimilar. Este valor tan alto no pudo ser corroborado por las mediciones, debido a que no todo el tráfico de la WWW es por transferencia de archivos.

Se agrega por cuenta nuestra que el comportamiento *heavy-tailed* del tráfico por usuario (SS), no cambia porque el enlace sea mas lento ya que depende del comportamiento de los usuarios y de las aplicaciones de alto nivel, más que de la red en sí.

Las evidencias recogidas, entre otros, por (16) y las que resultan del análisis de las características *heavy-tailed* del tráfico por estación y sentido de la muestra LBL-PKT-4 realizados en este trabajo, no dejan mucho margen de duda para suponer que el tráfico generado por un usuario conectado a su SS, a pesar del efecto regulador de WiMAX, sea otro que una distribución *heavy-tailed* en la mayoría de los casos.

Por lo tanto así se considerará en el generador a desarrollar.

3.4.4.2 Tráfico de la interfaz de cable que entra a la BS

El tráfico que entra a BS es la superposición de los tráficos destinado a cada SS.

Si la cantidad de SSs activas con la BS es suficiente, y una proporción importante de los tráficos de cada SS tiene una distribución *heavy-tailed*, constituyen ingredientes suficientes para considerar al tráfico total que entra a la BS como autosimilar. Esto quedó corroborado en el análisis de la muestra LBL-PKT-4.

Cabe preguntarse porque es, o no es, importante considerar que un tráfico sea autosimilar.

Un tráfico es autosimilar cuando presenta marcados agrupamientos o ráfagas (*burstiness*) de paquetes en un amplísimo rango de escalas de tiempo. Picos de tráfico montan ondulaciones que a su vez cabalgan sobre hinchazones, etc. (15).

Como los procesos son aleatorios, existe probabilidad real de que en algún momento se sobrepongan todos los fenómenos simultáneamente, pico más ondulación más hinchazón, etc. generándose un explosivo aumento del caudal de paquetes, sin aumento del promedio. Esta explosividad puede agotar los recursos disponibles si han sido dimensionados para otra distribución menos explosiva, como la exponencial. Uno de los recursos que más sufren con esta situación son las colas.

Dado que hay marcadas posibilidades que el tráfico pueda ser autosimilar, siendo esta la hipótesis menos benévola en cuanto al dimensionamiento del sistema, se considerará diseñar un generador de tráfico con características autosimilares, basado en la superposición de los tráficos *heavy-tails* destinados a cada estación activa con la BS.

3.5 Influencia del tráfico UGS sobre la autosimilaridad

Es posible en WiMAX contratar un servicio de *bitrate* constante similar al CBR (Constant Bit Rate) de ATM. Este *scheduling service*, que se denomina UGS (*Unsolicited Grant Service*) fue concebido para soportar flujos que transportan paquetes de tamaño fijo en forma periódica, tal como la T1/E1 y VoIP sin supresión de silencios (1).

A continuación se analizará la influencia del agregado de un tráfico UGS sobre la autosimilaridad de un tráfico de referencia. Para llevar a cabo estos análisis se superpondrá un tráfico de paquetes con velocidad constante a la muestra LBL-PKT-4.

Al tráfico total de la muestra LBL-PKT-4, la cual tiene 896689 paquetes en 3600s, para exagerar el efecto, se le superpuso un tráfico constante con un período de 3ms, esto incrementa el número de paquetes en un 234%. En la Fig. 3-4 se grafican los LD con y sin tráfico UGS. Nótese que ambas curvas tiene prácticamente el mismo andar. El coeficiente de Hurst $H = \frac{1+0,69}{2} = 0,845$ no se ha alterado por la adición de un tráfico UGS. Hay un aumento en el *throughput* pero no hay cambio en la autosimilaridad, cuantificada por el coeficiente de Hurst.

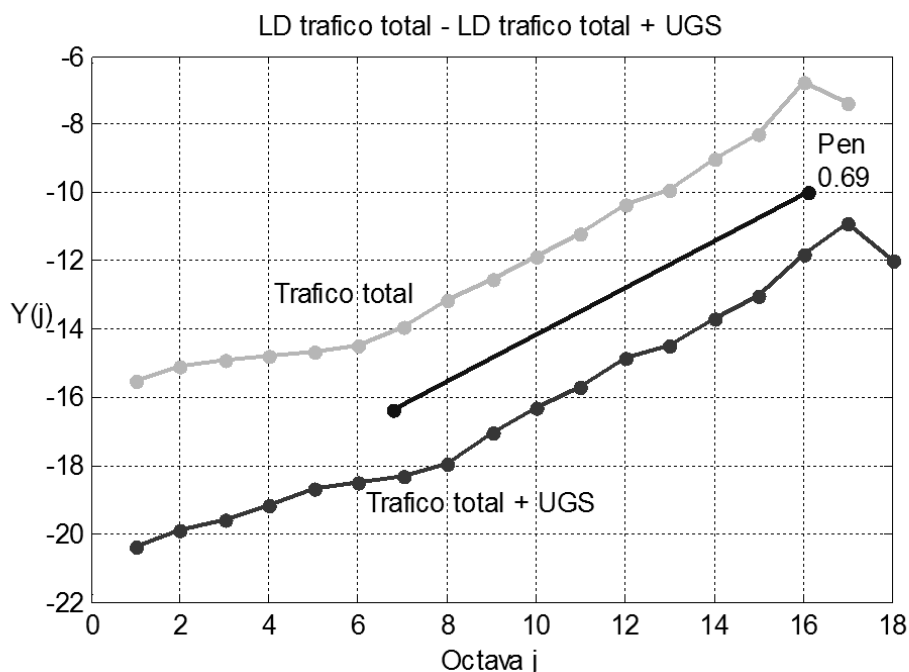


Fig. 3-4: LD tráfico total y LD tráfico total más un tráfico constante UGS con período 3ms.

Desde un punto de vista matemático, este último resultado no sorprende (7). Una importante propiedad de la *wavelet* madre (ψ_0) contra la no estacionaridad es que si esta tiene un número $N \geq 1$ de *vanishing moments* $\int t^k \psi_0(t) dt = 0$, donde $k=0, \dots, N-1$, permite eliminar cualquier tendencia polinomial hasta de grado $N-1$. En los LDs graficados en este trabajo siempre se ha usado *wavelets* Daubechies con $N=3$, lo cual suprime hasta un polinomio de grado 2 adicionado a la señal, y lógicamente suprime el efecto de un tráfico constante que corresponde a un polinomio de grado 0.

Los LDs de la Fig. 3-4, manifiestan una diferencia muy importante en los valores absolutos de los $Y(j)$, con y sin tráfico UGS agregado. Entendemos que esta observación

debería dar motivo suficiente para investigar el efecto de los valores absolutos de los $Y(j)$ de los LD, la cantidad de muestras y su relación con el tráfico constante (UGS) agregado al tráfico autosimilar.

El resultado de este análisis origina nuevos interrogantes relativo a las consecuencias sobre las colas. Algunos referentes proponen que el dimensionamiento de las colas se base únicamente en el *throughput* y el coeficiente de Hurst, suponiendo un tráfico autosimilar, y una probabilidad de pérdida de paquetes por rebasamiento de la cola dada.

Está claro que el coeficiente de Hurst calculado por el método LD no se modifica sensiblemente a pesar que la cantidad de paquetes se multiplicado por 2,34. Analicemos una hipótesis extrema, se agrega un tráfico UGS con un período mucho más pequeño por ejemplo 0,3ms, esto corresponde a $3600s/0,3ms=12$ millones de paquete en una hora, el resultado será que el *throughput*, se multiplica por más de 13 veces respecto al original, 896689 paquetes en 3600s. El tráfico original al ser minoritario ya no gravita, se tiene el mismo H, pero con un tráfico con velocidad de paquetes prácticamente constante. Si se calcula el tamaño de la cola con los parámetros: *throughput* y coeficiente de Hurst, suponiendo tráfico autosimilar e ignorando la existencia de tráfico con tiempo interarribo constante superpuesto, el resultado originaría un sobredimensionamiento en el tamaño de la cola, ya que el tráfico con comportamiento autosimilar es solo un pequeño porcentaje del tráfico total.

Bajo la sospecha de que el método LD como estimador de H, dada su dependencia con la cantidad de *vanishing moments* de la *wavelet* madre, pudiera no funcionar apropiadamente con el agregado de tráfico UGS. Se usará otro estimador para determinar el coeficiente de Hurst de los mismos tráficos que se corresponde con la Fig. 3-4, la pendiente de la variancia de las muestras agregadas.

El estimador variancia-agregación, se basa en la propiedad de los procesos autosimilares de que las variancias de los promedios de las muestras converge más lentamente a cero que m^{-1} (15)

$$\text{Var}(X^{(m)}) \sim cm^\beta$$

donde $c > 0$, $2 \leq m \leq \frac{n}{2}$, $\beta = 2H - 2$, $0 < \beta < 1$, n es la cantidad de muestras del proceso original. $X^{(m)}$ es el proceso obtenido por agregación del proceso original de la siguiente forma

$$X_i^{(m)} = \frac{1}{m} (X_{im-m+1} + \dots + X_{im}), i \geq 1$$

tomando logaritmos queda $\log(\text{Var}(X^{(m)})) \sim \beta \log(m) + \log(c)$, si se grafica

$$\log(\text{Var}(X^{(m)})) \text{ versus } \log(m), \text{ la pendiente es } \beta, \text{ y } H = 1 + \frac{\beta}{2}$$

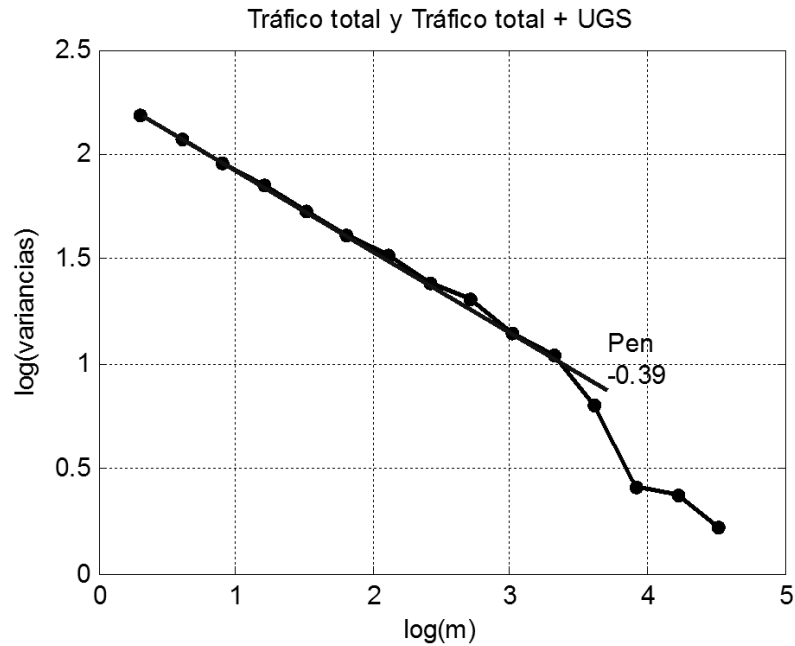


Fig. 3-5: Variancias de los procesos m-agregados para el cálculo del coeficiente de Hurst. Tráfico total de la muestra LBL-PKT-4 y de la misma más un tráfico UGS con tiempo interarribo de 3ms.

La Fig. 3-5 muestra el resultado de graficar el logaritmo de las variancias de los procesos agregados versus el logaritmo del nivel de agregación m para los dos casos, tráfico total y tráfico total más un tráfico constante (UGS), como las curvas son iguales solo se muestra una de ambas. $H = 1 - \frac{0,39}{2} = 0,805$, contra 0,845 por el método LD. En el Anexo 2 – Programas utilitarios en Matlab, se transcriben los programas en Matlab para el estimador variancia-agregación.

Este resultado, obtenido por un mecanismo más directo, pero menos eficiente, que la estimación por la pendiente LD (11), confirma que es correcta la estimación de H obtenida por el método LD. Las irregularidades al final de la curva se pueden atribuir a lo pequeña cantidad de muestras para calcular la variancia., cuando el nivel de agregación m es grande

Ya a la luz de este último resultado, es palmario indicar que las estadísticas de segundo orden como la variancia, y la autocovariancia, no pueden reflejar información sobre valores promedio, como el agregado de un tráfico constante al tráfico autosimilar.

Se concluye que para el dimensionamiento de las colas para un tráfico autosimilar se deberá tener en cuenta el porcentaje respecto al *throughput* (estadística de primer orden) total al cual describe cualquier indicador de segundo orden, tal como el coeficiente de Hurst, la variancia de los procesos agregados o la autocovariancia.

3.6 Influencia del tráfico UGS sobre la *heavy-tailness*

Investigaremos ahora la influencia de el tráfico constante (UGS) agregado, sobre la *heavy-tailness* del tráfico que envía una estación, para extrapolar los resultados si es posible, al tráfico de la interfaz de cable que entra a una SS.

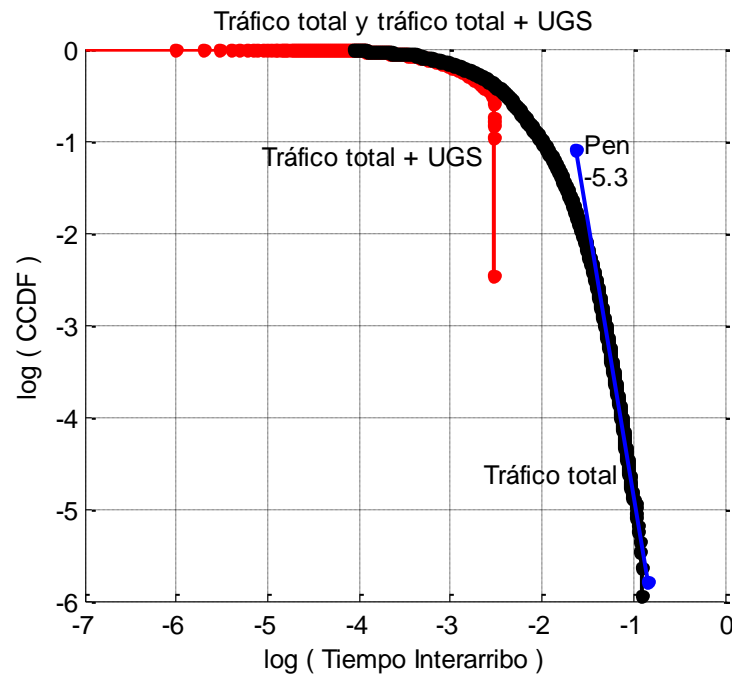


Fig. 3-6: CCDF del tráfico total de la muestra LBL-PKT-4 y de la misma más un tráfico UGS con tiempo interarribo de 3ms.

En la Fig. 3-6, se grafica la CCDF del tráfico total y el tráfico total + UGS, con los mismo datos de la Fig. 3-4.

Nótese que el tráfico UGS agregado causa un recorte vertical en $\log(3\text{ms}) = -2,52$, también se aprecia la proliferación de puntos con menor tiempo de interarribo debido a la inclusión del tráfico UGS, aunque sin modificar la forma de la CCDF apreciablemente antes del recorte.

Este comportamiento se ve más claro en la Fig. 3-7 donde el UGS agregado tiene un tiempo interarribo de 30ms. Se aprecia que no se modifica sensiblemente el andar de la curva del tráfico total hasta que sufre un recorte en $\log(30\text{ms}) = -1,52$.

Para el análisis del comportamiento de la CCDF de una sola estación se elije la estación 32, la cual presenta una asíntota muy clara, observar la Fig. 2-10. Como el tráfico de una única estación es mucho menor que el tráfico total, se agregará un tráfico UGS con tiempo interarribo de 80ms, que es congruente con una latencia máxima sugerida para VoIP de 80ms.

En la Fig. 3-8 se observa que el efecto es exactamente el mismo que con el tráfico total, hay un recorte en $\log(80\text{ms}) = -1,09$.

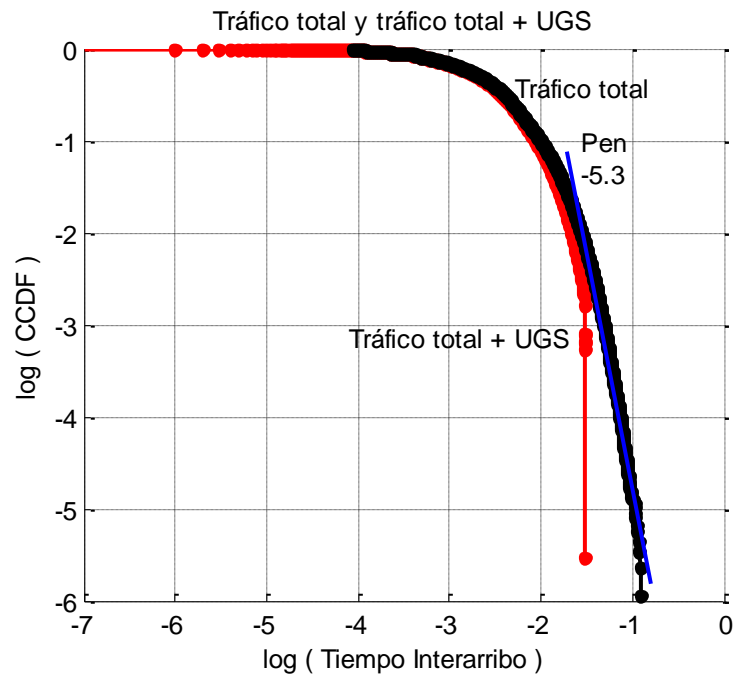


Fig. 3-7: CCDF del tráfico total de la muestra LBL-PKT-4 y de la misma más un tráfico UGS con tiempo interarribo de 30ms.

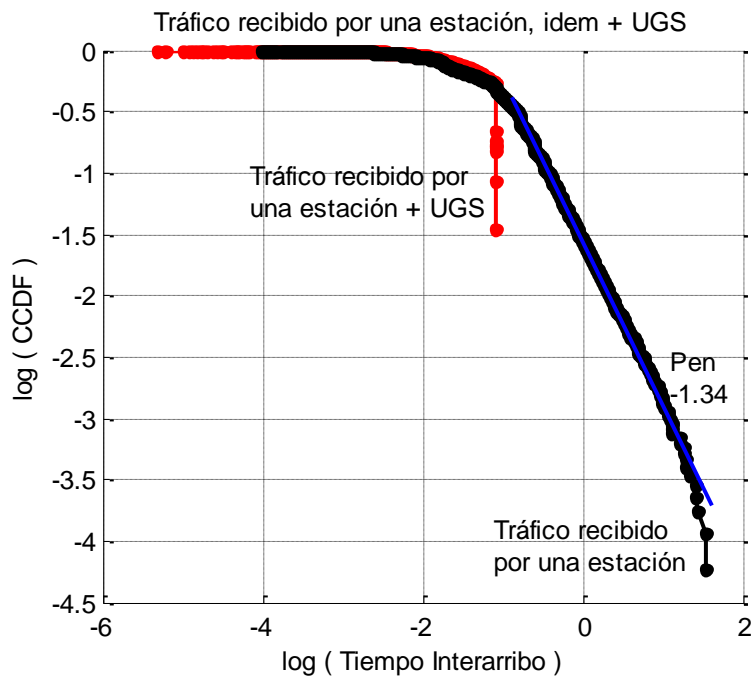


Fig. 3-8: CCDF del tráfico de la estación 32 de la muestra LBL-PKT-4 y de la misma más un tráfico UGS con tiempo interarribo de 80ms

Los gráficos de las Fig. 3-6, Fig. 3-7 y Fig. 3-8 muestran que cualquiera sea el tráfico, si se adiciona al mismo un tráfico UGS con un período dado, la correspondiente CCDF queda recortada en abscisa al logaritmo del período.

No resulta obvio observando las figuras nombradas en el párrafo anterior, que sucedería si a un tráfico autosimilar se le adiciona varios tráficos UGS con diferentes períodos. Este análisis nos permitirá establecer una analogía con el tráfico que entra a una BB, donde cada estación activa puede tener contratado un tráfico UGS con diferente *throughput*.

Se ha seleccionado de la muestra LBL-PKT-4 las primeras 10 estaciones con más de 10000 paquetes, y se le ha agregado a cada estación un tráfico UGS con un período fijo diferente, un valor randómico con distribución uniforme entre 300ms y 1s. Hubiera sido más real simular un tráfico VoIP con períodos entre 50ms y 80ms, pero los resultados gráficos obtenidos no son claros, dado que la muestra es pre VoIP.

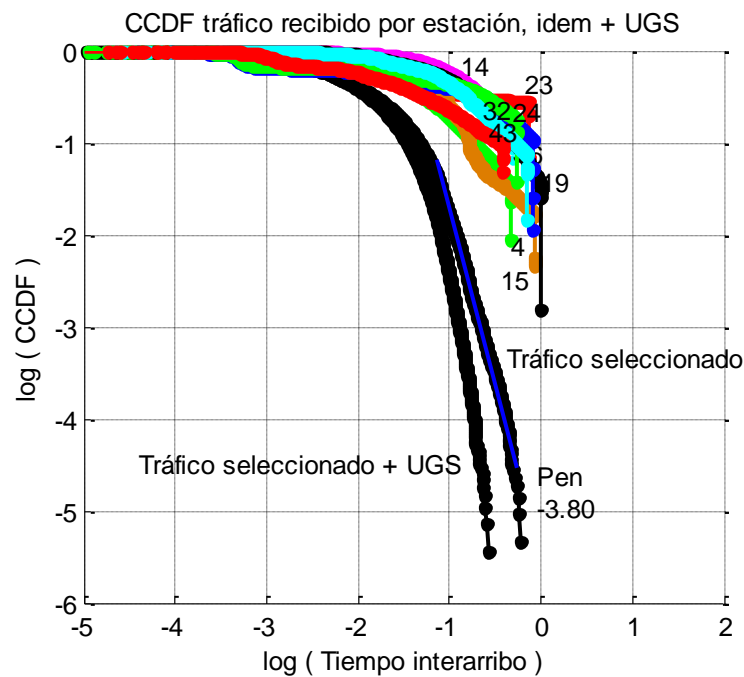


Fig. 3-9: CCDF del tráfico recibido por 10 estaciones seleccionadas de la muestra LBL-PKT-4 a las cuales se adicionó un tráfico UGS con diferente período.

En la Fig. 3-9 se han trazado los diagramas $\log(\text{CCDF})$ versus $\log(\text{Tiempo interarribo})$ de cada estación más el tráfico UGS, de todas las estaciones seleccionadas más su tráfico UGS y el de todas las estaciones seleccionadas como era originalmente.

Comparando la Fig. 3-9 con la Fig. 2-10, se aprecia que el tráfico recibido superpuesto de cada estación seleccionada más su tráfico UGS adicionado a cada estación sufre un recorte en abscisas al valor de la estación que tiene un período UGS menor, sin embargo el recorte no es abrupto como en el caso de un único tráfico UGS adicionado.

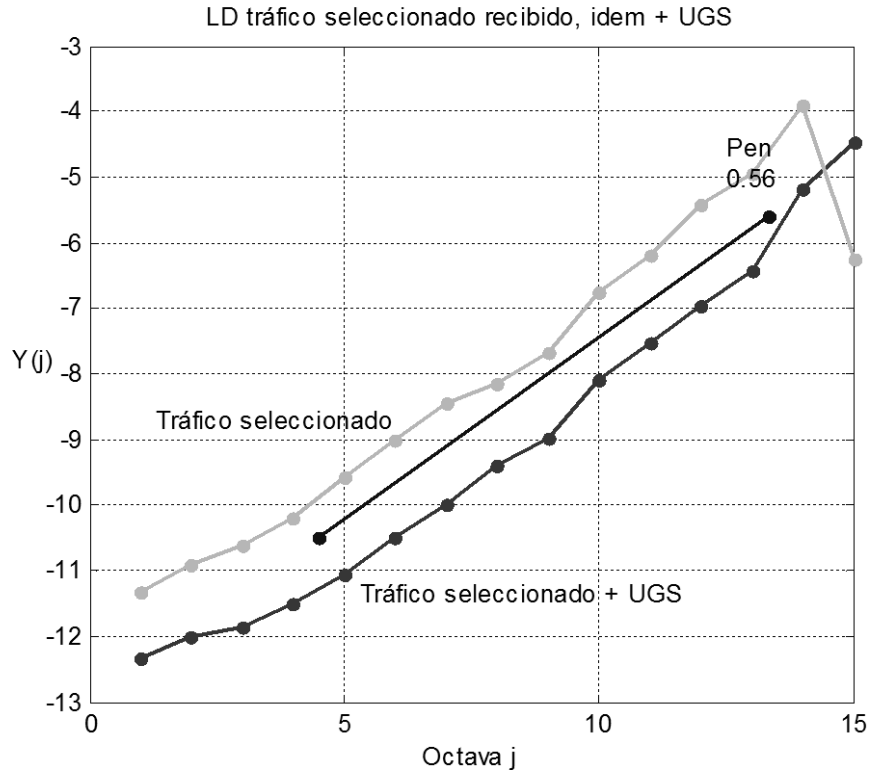


Fig. 3-10: LD del tráfico recibido por 10 estaciones seleccionadas de la muestra LBL-PKT-4 a las cuales se adicionó un tráfico UGS con diferente período.

En la Fig. 3-10 se han trazado los diagramas LDs para el tráfico recibido por las estaciones seleccionadas con y sin el tráfico UGS de cada una de ellas. Se observa que la pendiente es prácticamente la misma, lo mismo que el valor H determinado por este método.

Se concluye de las observaciones que

- El agregado de tráfico UGS produce una limitación de la CCDF en abscisa al logaritmo del menor período UGS que exista en la muestra. Si hay un único tráfico UGS, la limitación es abrupta, si hay muchos tráficos UGS con diferentes períodos, existe la limitación pero el recorte no es abrupto.
- Este efecto destruye la *heavy-tailness* de las curvas CCDF.
- Exista un único tráfico UGS o varios de diferentes períodos, los LDs no cambian sensiblemente su pendiente y por lo tanto su coeficiente de Hurst.

Estos análisis confirman que el coeficiente de Hurst, y el *throughput* no es suficiente para dimensionar las colas. Hay que tener en cuenta si existe tráfico UGS, ya que de no tomarlo en cuenta se sobredimensionan las colas al considerarlo como contribuyente al tráfico autosimilar. Esta circunstancia ya fue señalada en varios artículos de Neame, Zukerman, etc. (14) y (18), quienes arribaron a una conclusión similar por otro camino.

Ellos al modelar un tráfico que produjera el mismo efecto sobre una cola de capacidad especificada y un servidor con un tiempo de servicio dado, sintetizaron un tráfico en base al

coeficiente de Hurst H , y la variancia del tráfico que se quiere simular, además de un λ de un proceso Poisson necesario para la construcción del modelo M-Pareto (Markov Pareto) o PPBP (Poisson Pareto *Burst Process*) que emplearon. Para que coincidieran los *throughput* del tráfico simulado con el del original, tuvieron que ajustar el modelo adicionando (o restando) un tráfico con CBR, que en el caso de WiMAX sería UGS, que como se mostró aquí no modifica el H ni la variancia.

4. Requerimientos del modelo para sintetizar tráfico

4.1 Parámetros que definen a un tráfico

Hay muchos parámetros o funciones estadísticas que definen un tráfico, por ejemplo

- *Throughput*
- Coeficiente de Hurst o fractabilidad
- Variancia versus nivel de agregación
- LRD (ACF) función de autocorrelación
- CCDF función de distribución acumulativa complementaria
- LD *logscale diagram* y otros estimadores gráficos de H

Es posible sintetizar tráficos que satisfagan uno o varios de los requisitos anteriores y de hecho existen muchos modelos para simulación que lo hacen.

Sin embargo desde un punto de vista técnico, y en particular con WiMAX, cualquier modelo es inválido si no simula correctamente el comportamiento de los dispositivos de red que se pretenden ensayar o diseñar, tal que el tráfico sintético produzca el mismo efecto que el tráfico real que se pretende recrear.

4.2 Colas

Considérese un *router* al cual concurren varias conexiones *gigabit* Ethernet. El *router* deberá encaminar cada paquete abriéndolo para identificar la dirección IP de destino y entregárselo al transmisor que corresponde a la ruta de destino. Ya constituye un reto tecnológico procesar un paquete cuyos bits arriban a una velocidad de 1bit por nanosegundo.

El problema se multiplica si arriban paquetes al *router* por varias entradas superponiéndose, el procesador del *router* no puede analizar dos o más paquetes simultáneamente porque no tiene la suficiente velocidad de procesamiento. La solución es almacenar los paquetes que arriban en una memoria intermedia hasta que puedan ser procesados. Esta memoria de almacenamiento intermedio o cola, es la solución histórica y universal para poder atender clientes cuyo tiempo interarribo, temporalmente, es menor al tiempo de servicio de los mismos.

Por supuesto, el problema parece no tener solución si el promedio a largo plazo del tiempo interarribo de clientes, es mayor que el promedio del tiempo de servicio de los mismos. La cola crecería hasta que la capacidad sea desbordada cualquiera sea esta.

Entonces, presentado así, los dispositivos de almacenamiento intermedio y en particular las colas, son quienes sufren con la variabilidad del tráfico, dejando el padecimiento por el

promedio del tiempo interarribo a los procesadores o servidores, ya que en este ítem, las colas son impotentes cualquiera sea su tamaño.

Entonces si el diseño o dimensionamiento de colas es el objetivo, no hay mejor parámetro que el comportamiento de las mismas para evaluar la calidad de un tráfico sintético. Desde este punto de vista es correcto afirmar que un buen generador de tráfico es aquel que produce el mismo efecto sobre las colas que el tráfico real que se pretende simular.

Dada la explosividad de arribo de paquetes producido por una tráfico con características autosimilar, pareciera que cualquier capacidad de una cola puede ser insuficiente para un pico de tráfico. Esto no es cierto porque nunca el *bitrate* de arribo va a superar la capacidad del enlace por el que ingresa, sin embargo existe una diferencia económica, que depende de la capacidad de la cola, y que pesa en el costo final de los productos.

Como siempre en cualquier diseño de un equipo, o sistema, hay requerimientos encontrados, comúnmente costo versus cualquier otro. En el caso específico de las colas, la capacidad de las mismas debe ser suficiente como para no produzca un degradación inaceptable en la performance del equipo, debida a la pérdida de paquetes por rebasamiento.

Desde una perspectiva ingenieril, está claro que no se busca un modelo que permita sintetizar un tráfico que satisfaga todos los parámetros estadísticos de un tráfico real, ya que esto puede ser un proceso sumamente complejo, mucho más que lo necesario, y por ende impráctico.

El objetivo como ingenieros es muy simple, construir un modelo para sintetizar tráfico que pueda ser aplicado en ámbitos de simulación, para el dimensionamiento de redes y evaluación del comportamiento de las mismas.

El bloque fundamental en la construcción de redes de telecomunicación modernas, es la cola de servidor único SSQ (*Single Server Queue*). Crear herramientas que puedan ser usadas para satisfacer el objetivo requiere modelos generadores de tráfico que puedan predecir en forma exacta el comportamiento del tráfico real en una SSQ (19). En particular se considerará colas con disciplina FIFO (*First Input First Output*) que son las más comunes en las redes.

4.3 Programa en Matlab y Simulink para evaluar tráfico

Por lo expresado en 4.2, un modelo generador sintetiza correctamente a un tráfico real dado, si ambos producen el mismo efecto sobre un sistema SSQ FIFO. Esto significa la misma pérdida de paquetes o bytes, para igual capacidad de cola e igual tiempo de servicio, siempre refiriéndose a valores estadísticos.

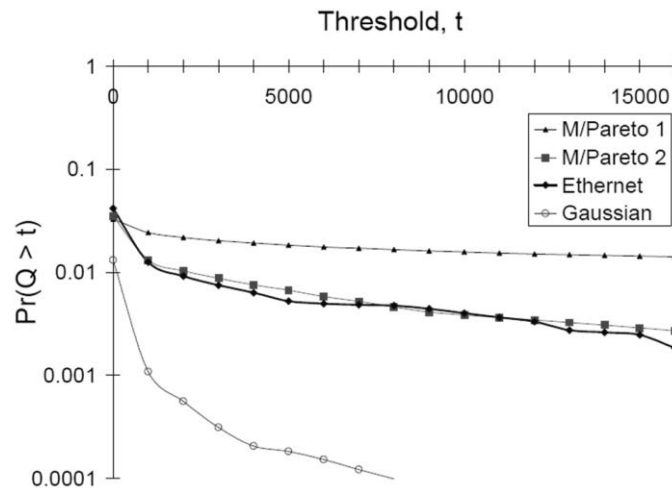


Fig. 4-1: Probabilidad de pérdida de paquetes en función de la capacidad de la cola para una muestra de tráfico real, denominado Ethernet, comparándolo con tráficos sintetizados con el algoritmo M/Pareto1, M/Pareto2 y Gaussiano. Tomada de (18).

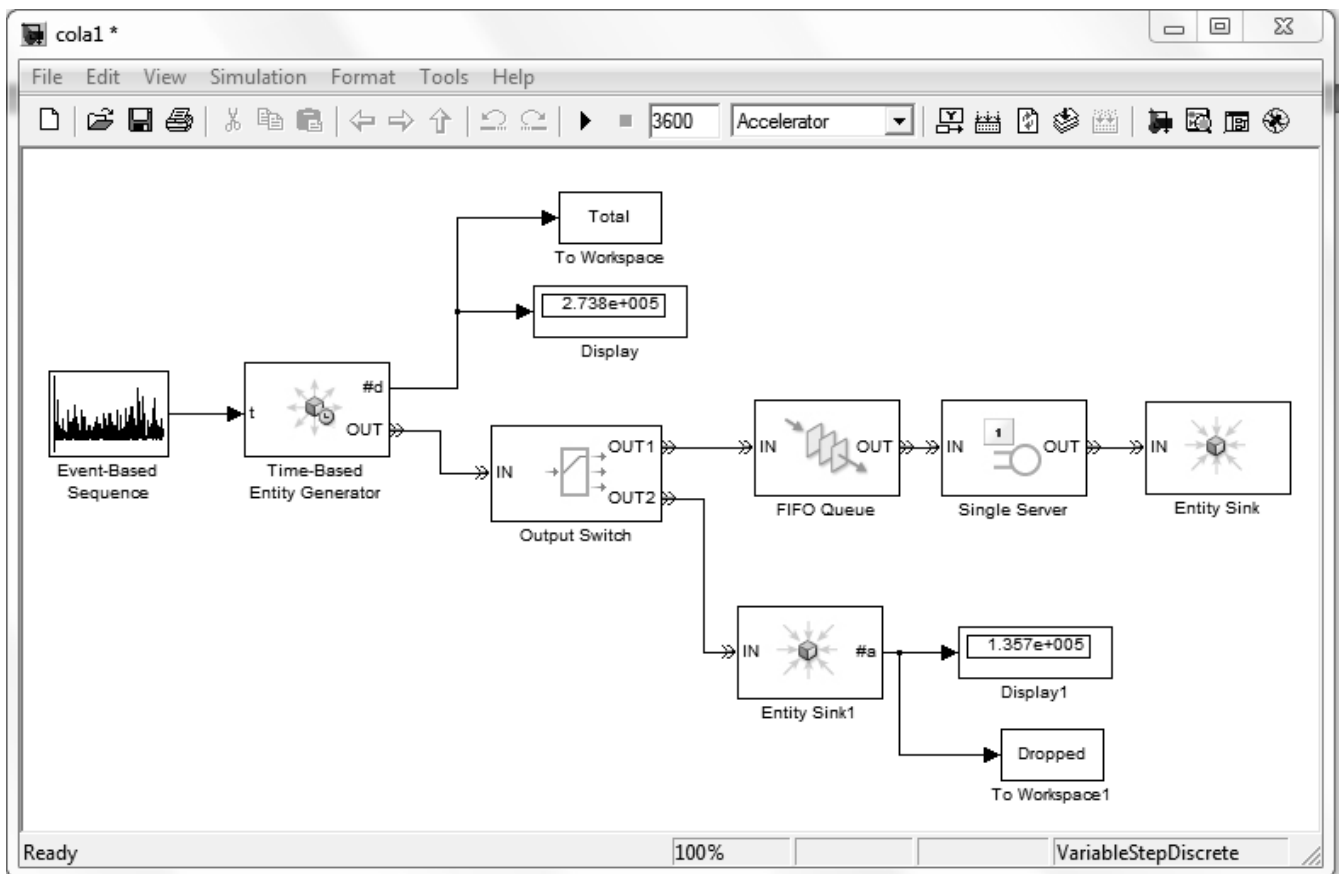


Fig. 4-2: Modelo de Simulink para analizar el comportamiento de una SSQ FIFO con un tiempo de servicio constante.

Se ha desarrollado a tal efecto un programa en Simulink, controlado desde el *workspace* de Matlab por un *script*. Esto último es necesario para correr reiteradamente el modelo, alimentado con parámetros diferentes desde el entorno de Matlab. También el *script* recolecta

la información y grafica la relación pérdida de paquetes a la cantidad total de paquetes que alimenta el sistema SSQ FIFO, en función del tiempo de servicio y de la capacidad de la cola.

Se ha definido que la información se presente en la misma forma que se realiza en (18), y que se muestra en la Fig. 4-1 de tal forma de poder comparar el andar de las curvas.

En la Fig. 4-2 se muestra el modelo³ de Simulink⁴ que se describe con detalle en el Anexo 1 - Modelo de Simulink para analizar el comportamiento de una SSQ FIFO sometida a un tráfico dado, en otros anexos se transcribe el *scripts* correspondientes para manejar el modelo desde el entorno de Matlab.

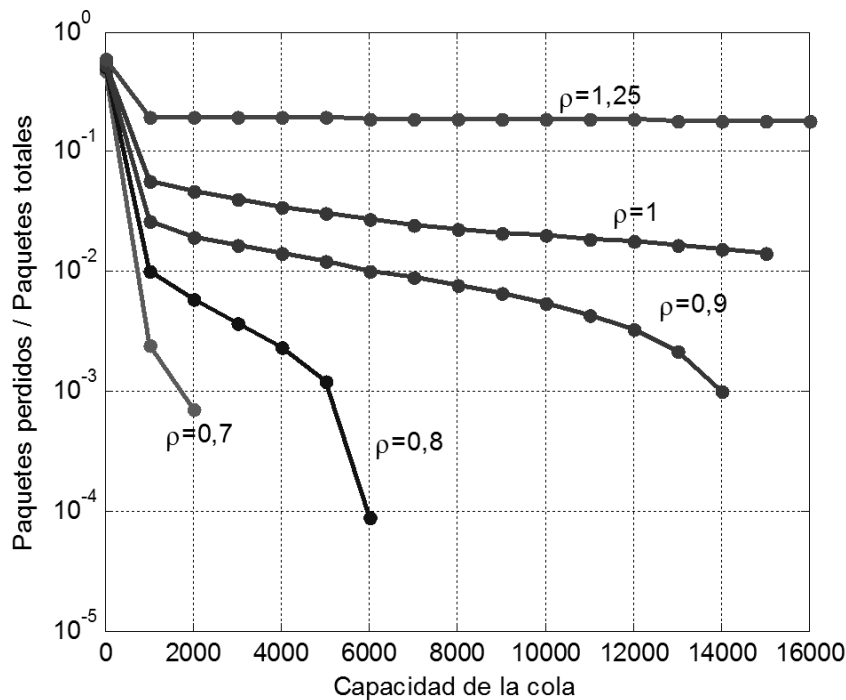


Fig. 4-3: Resultado de aplicar el tráfico de la muestra LBL-PKT4 al modelo SSQ FIFO de Simulink de la Fig. 4-2, para diferentes intensidades de tráfico ρ en función de la capacidad de la cola

Para probar el modelo de la SSQ elaborado con Simulink, se ha procesado con el mismo la muestra LBL-PKT4. Un *script* en Matlab que corre en el *workspace* abre el modelo cargándole la muestra y luego lo ejecuta iterativamente alimentándolo cada vez con diferentes valores de capacidad de cola y tiempo de servicio. En la Fig. 4-3 se muestra el resultado.

Las distintas trazas se corresponden a diferentes intensidades de tráfico ρ . Advierta la semejanza con la de la Fig. 4-1.

³ Puede crear confusión el que a los programas gráficos de Simulink se los denomine modelos, por lo tanto se debe interpretar el sentido de la palabra modelo en el contexto en el cual se la nombra.

⁴ Se deja en inglés las denominaciones de los bloques de Simulink para no confundir a un potencial usuario del modelo, ya que de ser traducidos no se podría encontrar con facilidad los bloques en las librerías.

Hay que resaltar dos hechos de la Fig. 4-3. Primero la alta sensibilidad de las curvas con la intensidad de tráfico ρ . Segundo, cuando el ρ es mayor que 1, ningún tamaño de cola alivia la pérdida de paquetes tal como era de esperarse. Para $\rho=1,25$, el andar es casi horizontal, para una capacidad de cola de 16000 la relación de pérdida vale 0,179. Si el tiempo de simulación fuera lo suficientemente largo, $t \rightarrow \infty$, para que la cantidad de paquetes que quedan en la cola fuera despreciable en relación a los perdidos la curva terminaría en una recta horizontal de valor 0,2

$$\frac{\text{Paquetes perdidos}}{\text{Paquetes totales}} = \frac{\lambda t - \mu t - \text{cola}}{\lambda t} \approx \frac{\lambda - \mu}{\lambda} = \frac{\rho - 1}{\rho} = \frac{0,25}{1,25} = 0,2$$

4.4 Análisis complementarios de la muestra LBL-PKT4

El próximo objetivo es desarrollar un modelo en Matlab y Simulink para el generador de tráfico.

Para evaluar el comportamiento del modelo, una prueba importante es sintetizar un tráfico que tenga el mismo *throughput*, y que produzca el mismo efecto en un sistema SSQ que el tráfico de la muestra LBL-PKT4 el cual se toma como referencia.

En la medida que los otros parámetros del tráfico generado, tales como el coeficiente de Hust, CCDF, variancia, etc. se aproximen al tráfico de la muestra consideraremos el tráfico generado por el modelo más parecido al real.

Además de los análisis ya realizados a la muestra LBL-PKT4 relativas a su autosimilaridad, y *heavy-tailness* es necesario realizar otros adicionales. Las necesidades fueron surgiendo naturalmente a medida que se desarrollaba el modelo, salvo que aquí se condensan para que la información no quede dispersa.

Una gran cantidad, quizá la mayoría, de los generadores de tráfico autosimilar para simulación se basan en que los paquetes arriban en forma de ráfagas, cuyos tiempos interarribo responden a una distribución exponencial, y sus duraciones a una distribución de Pareto. Por lo tanto se considera necesario investigar los parámetros de las ráfagas presentes en la muestra, para aplicarlos a las distribuciones mencionadas.

Interesan la velocidad promedio de arribo de las ráfagas, la duración promedio de cada ráfaga, la velocidad de arribo promedio de los paquetes de cada ráfaga y la distribución estadística de la cantidad de bytes de datos de cada paquete. Se comenzará por el último ítem respetando el orden con cual se realizó el presente trabajo.

4.4.1 Análisis estadístico de la longitud neta de los paquetes

Tabla 4-1: Contenido de los campos de archivos de la muestra LBL-PKT4.

Nombre de Archivo	Time stamp	Source Host	Dest. Host	Source Port	Dest. Port	Data bytes	Flags	Seq Num 1	Seq Num 2	Ack Seq Num	Cantidad de paquetes
lbl-pkt4.tcp	x	x	x	x	x	x					862946
lbl-pkt4.sf	x	x	x	x	x		x	x	x	x	(13699)
lbl-pkt4.udp	x	x	x	x	x						33744
lbl-pkt4.encap	x										303055
lbl-pkt4.other	x										121140

Dado que los únicos paquetes que tienen información sobre la cantidad de bytes de datos que transportan, son aquellos que llevan segmentos TCP, se limitará el análisis únicamente a estos.

En la Tabla 4-1 se muestra la información provista por la muestra LBL-PKT4, desmenuzada por archivos. En la última columna se indica la cantidad de paquetes en cada archivo. Los paquetes del archivo lbl-pkt4.sf también están incluidos en lbl-pkt4.tcp pero con diferente información.

Lo que sigue son tres líneas de texto tomadas arbitrariamente del archivo lbl-pkt4.tcp

5.839397 4 22 2331 119 46

5.852912 75 32 2097 25 0

5.853508 7 8 80 1170 512

la segunda línea es un segmento TCP de inicio de conexión a los 5,852912 segundos entre las estaciones origen 75 (puerto 2097), y la estación destino 32 (puerto 25) con 0 bytes de datos.

La línea que se muestra debajo de este párrafo, es la correspondiente descrita en el archivo lbl-pkt4.sf que complementa la información del lbl-pkt4.tcp, donde los autores han eliminado la cantidad de bytes de datos, agregándose el *flag S*, los números de secuencia y de Ack. Nótese que al ponerse 0 en la cantidad de bytes del segmento en cuestión, significa que no se está tomando en cuenta los 20 bytes de encabezado TCP.

5.852912 75 32 2097 25 S 1110080000:1110080000 0

En la Fig. 4-4, se muestra el histograma de los bytes de datos registrados en la muestra. Llamen la atención tres líneas. La línea de 0 bytes que es la más numerosa y se corresponde con todos los paquetes sin datos como Acks, inicio y fin de sesión, etc. Otro valor que abunda mucho en la muestra es 512 bytes, si a este valor le sumamos los 20 bytes del encabezado TCP más los 20 bytes del encabezado Ethernet, se corresponde con el MTU (*Maximum Transmission Unit*) de 552 bytes que es el máximo del protocolo SLIP para las conexiones de

internet por *dialup*. Al último está la línea de 1460 bytes, la cual se corresponde con el MTU de Ethernet de 1500 bytes (9).

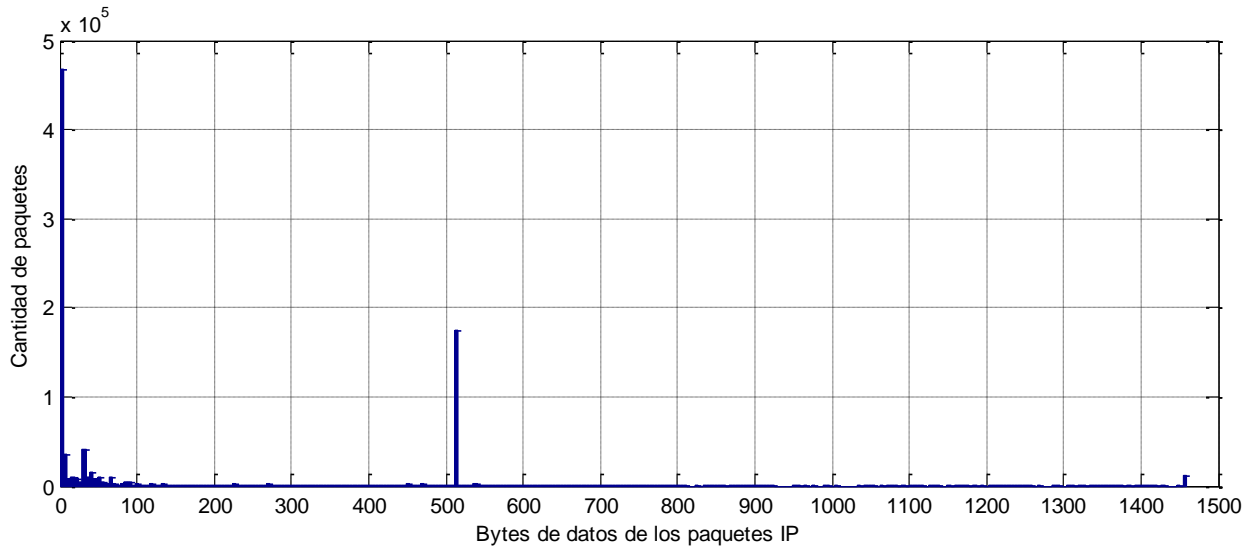


Fig. 4-4: Histograma de la cantidad de datos de los paquetes IP correspondientes a la muestra LBL-PKT4.

4.4.2 Ráfagas, *socks*, tiempos interarribo de paquetes por *sock*⁵

Lo primero a investigar es a que conjuntos de paquetes atribuimos la denominación de ráfaga.

Con el fin de buscar relaciones entre ráfagas y estaciones, y aproximarnos al uso que se le dará al generador, se ha seleccionado de la muestra LBL-PK4 el tráfico TCP y UDP recibido por 10 estaciones, pero cuyo volumen es suficientemente significativo, ya que compone el 24,3% de paquetes de toda la muestra.

4.4.2.1 Características del tráfico recibido por 10 estaciones de la muestra LBL-PKT4

En la Fig. 4-5 se muestra un diagrama en 3-D donde cada punto es un paquete (o muchos paquetes sobrepuestos) y en las coordenadas se indica la estación destino, la estación origen y el momento en que se registró el paso del mismo.

Los paquetes son muy numerosos, aproximadamente 217000, por eso si se realiza una ampliación de un punto elegido arbitrariamente de la Fig. 4-5 donde se muestra la zona con ampliación escalonada se encuentra que el mismo puede representar varios paquetes tal como se aprecia en la Fig. 4-6.

Se denominará, en este contexto, *sock* a la cuádrupla estación origen, estación destino, puerto origen y puerto destino. Se ha usado un esquema de colores por cada *sock* diferente. Si

⁵ Para el presente análisis de la muestra LBL-PKT4 se han desarrollado expresamente programas en Matlab, pero se entiende que por sus características poco específicas, no ameritan incluirlos en esta presentación.

la secuencia de paquetes entre una estación origen y una destino es de un mismo color, por ejemplo fucsia, significa que todos los paquetes transmitidos por la estación origen tienen el mismo *sock*.

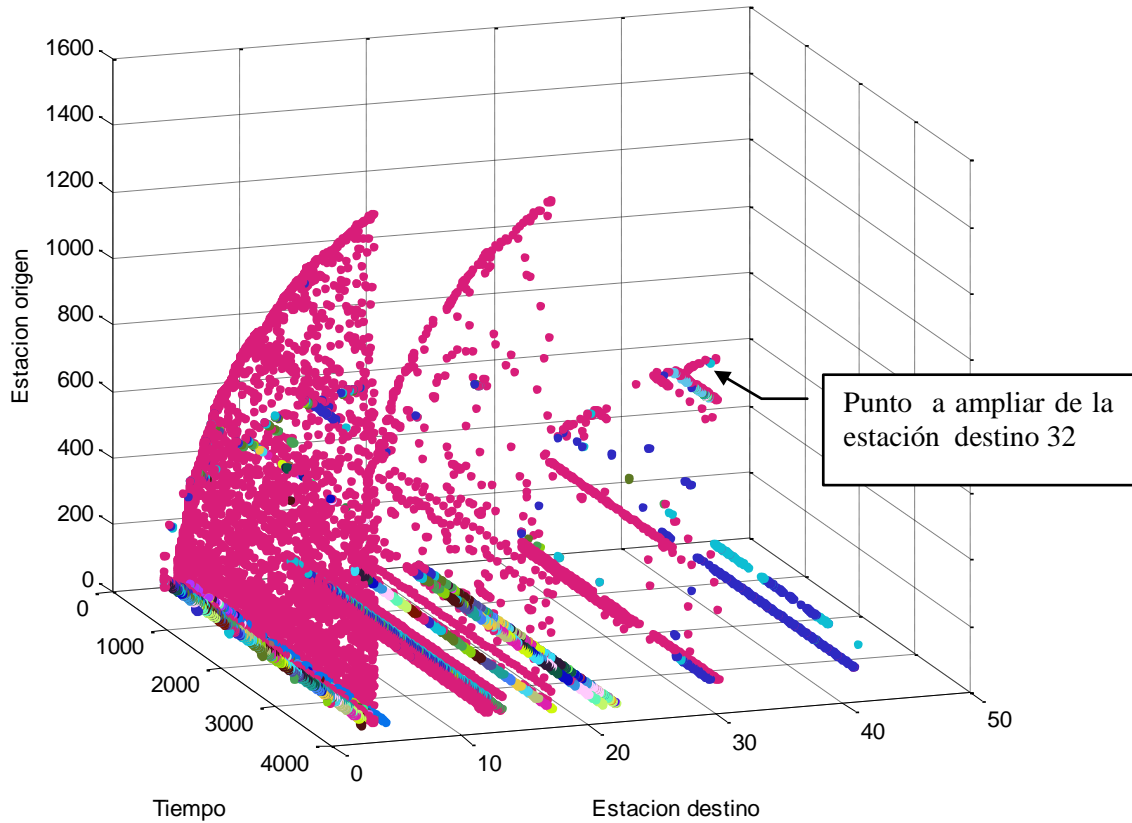


Fig. 4-5: Paquetes IP registrados, correspondientes a la muestra LBL-PKT4. Cada punto representa uno o muchos paquetes recibidos por la estación destino. En una coordenada se indican las estaciones seleccionadas la 4, 5, 6, 14, 15, 19, 23, 24, 32, y 43, en otra la estación de donde proviene y en la tercera coordenada el instante en que se registró el paquete.

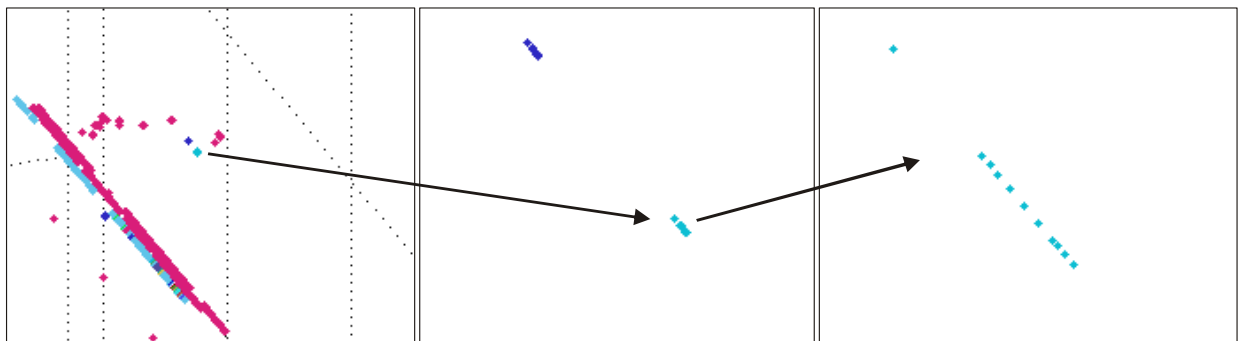


Fig. 4-6: Magnificación progresiva de un punto correspondiente a la estación destino 32 señalado en la Fig. 4-5.

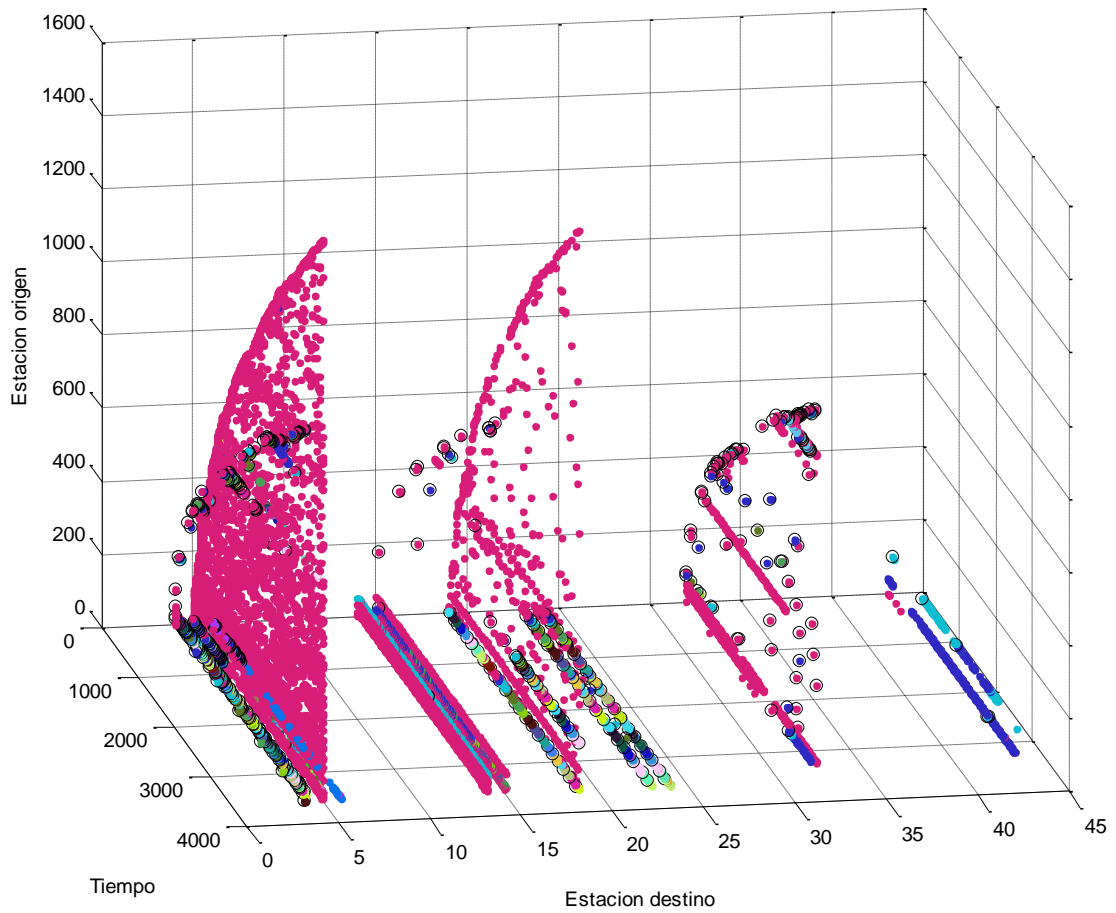


Fig. 4-7: La misma información que la Fig. 4-5 a la que se agrega circunferencias negras a los paquetes que presentan el flag Syn activado.

En la Fig. 4-7, se presenta la misma información que la Fig. 4-5, pero incorporando mediante un círculo negro los paquetes que tienen el *flag* Syn activado, lo que permite identificar inicio de sesiones TCP.

Es intuitivo advertir que las estaciones 5, 19, y 32, generan algún tipo de servicio a las estaciones restantes, ya que reciben paquetes de la mayoría.

Nótese especialmente para las estaciones destino 5 y 19 de la Fig. 4-7, que todos los paquetes son sockets únicos por cada estación origen, pero con ausencia total de *flags* Syn activados. Esto se debe a que el protocolo es UDP.

Como resultado de la aplicación de una herramienta desarrollada en Matlab especialmente para el caso, se condensa en la Tabla 4-2 las características básicas de los paquetes seleccionados analizados de la muestra LBL-PKT4.

Tabla 4-2: Propiedades de los paquetes seleccionados de la muestra LBL-PKT4. Se seleccionó paquetes recibidos correspondientes a 10 estaciones, que constituyen el 24,3% de paquetes de la muestra.

	Parámetro	Valor	Unidad
1	Nombre de la muestra	LBL-PKT4	
2	Tiempo durante el cual se recolectó la muestra	3600	s
3	Cantidad total de paquetes de la muestra, únicamente TCP+UDP	896690	paquetes
4	Cantidad de estaciones seleccionadas para analizar el tráfico recibido	10	estaciones
5	Cantidad de paq. TCP+UDP recibidos por las estaciones seleccionadas	217854 (24,3%)	paquetes
6	Cantidad de paq. TCP recibidos por las estaciones seleccionadas	207791 (95,4%)	paquetes
7	Cantidad total de paq. UDP recibidos por las estaciones seleccionadas	10063 (4,6%)	paquetes
8	Cantidad de paquetes promedio por estación seleccionada TCP+UDP	21785	paquetes
9	Cantidad de <i>flags</i> Syn	577	paquetes
10	Cantidad de paquetes promedio por <i>flag</i> Syn	378	paquetes
11	Cantidad de <i>socks</i> TCP+UDP	2530	<i>socks</i>
12	Cantidad de <i>socks</i> TCP	602	<i>socks</i>
13	Cantidad de <i>socks</i> UDP	1928	<i>socks</i>
14	Cantidad de paquetes promedio por <i>sock</i> TCP+UDP	86	<i>socks</i>
15	Cantidad de paquetes promedio por <i>sock</i> TCP	345	<i>socks</i>
16	Cantidad de paquetes promedio por <i>sock</i> UDP	5	<i>socks</i>
17	Tiempo promedio duración de un <i>sock</i>	454	s
18	Tiempos interarribo entre paquetes de un mismo <i>sock</i>		
	Moda	0,00048	s
	Mediana	0,06753	s
	Promedio	5,33775	s

En la Fig. 4-8 se muestra un histograma de los tiempos interarribo entre paquetes de cada *sock*, tomados aisladamente, sin entremezclar paquetes de diferentes *socks*. Solo se toman en cuenta los *socks* que tienen dos o más paquetes recibidos de las estaciones seleccionadas, ya que un único paquete no define un tiempo interarribo. Se ha graficado hasta 0,1s que comprende más del 60%.

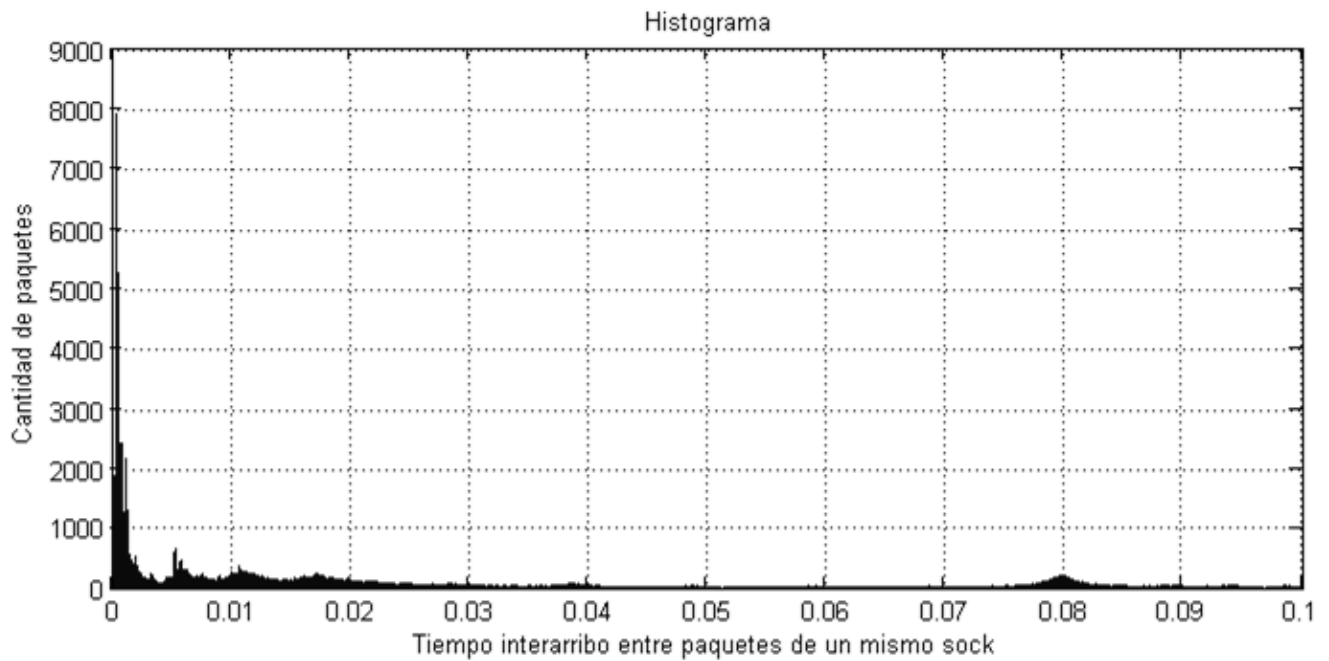


Fig. 4-8: Histograma de de lo cantidad de paquetes versus el tiempo interarribo entre paquetes de cada *sock*.

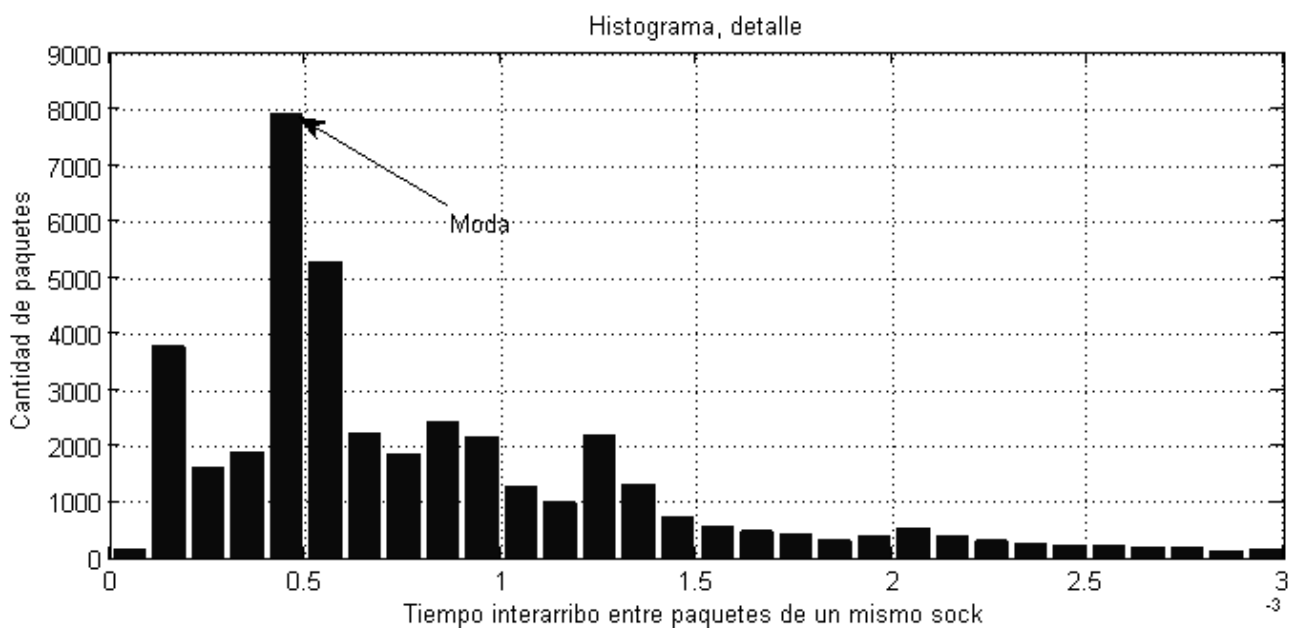


Fig. 4-9: Detalle de la Fig. 4-8. Se ha marcado la barra que contiene a la moda que vale 0,48ms

En la Fig. 4-9 se muestra un detalle del histograma de la Fig. 4-8. Se indica la moda, que se corresponde con el *bin* que contiene mayor cantidad de paquetes acumulados. La cantidad de paquetes acumulados hasta el valor de la moda es solo del 6,1% del total seleccionado de la muestra, por lo tanto no es un valor suficientemente representativo del tiempo interarribo del total de paquetes recibidos de las estaciones seleccionadas de la muestra.

Aún menos representatividad que la moda de los tiempos interarribo, lo tiene el promedio o expectativa, que vale 5,33s (ver Tabla 4-2). Para avalar lo expresado basta suponer que un solo tiempo interarribo entre dos paquetes de cualquier *sock* valga, exagerando ya que la muestra toma solo 3600s, un millón de segundos, esto elevaría el promedio casi al doble, 9,98s.

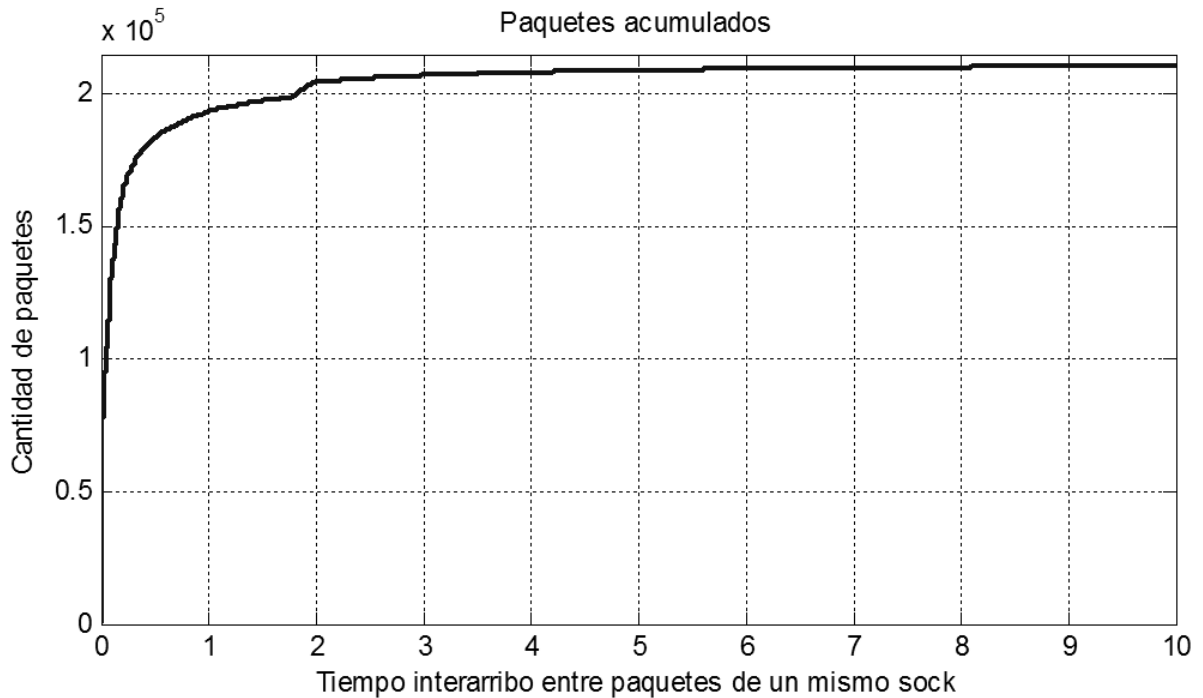


Fig. 4-10: Cantidad de paquetes acumulados versus tiempo interarribo entre paquetes de cada *sock*. Los paquetes acumulados con tiempos interarribo menor a 10s son 210662 y constituyen el 96,7% de los 217854 paquetes seleccionados.

En la Fig. 4-10 se muestra la cantidad de paquetes acumulados en función del tiempo interarribo entre paquetes de un mismo *sock*. Si bien existen tiempos interarribo cercanos al tiempo total de la muestra, los tiempos mayores que 10s componen solo el 3,3% de los paquetes recibidos por las estaciones seleccionadas.

La Fig. 4-11 es un detalle de la Fig. 4-10, la ordenada está en tanto por uno y se amplió la escala de tiempos. En ella se ha señalado la mediana que vale 67,5ms. El 50% de los paquetes tiene un tiempo interarribo menor que la mediana y el otro 50% mayor que la mediana.

Como se mencionó anteriormente, se busca relacionar los datos obtenidos de analizar estadísticamente la muestra LBL-PKT4 con los conceptos de ráfaga que se manejan en los generadores de tráfico para simulación.

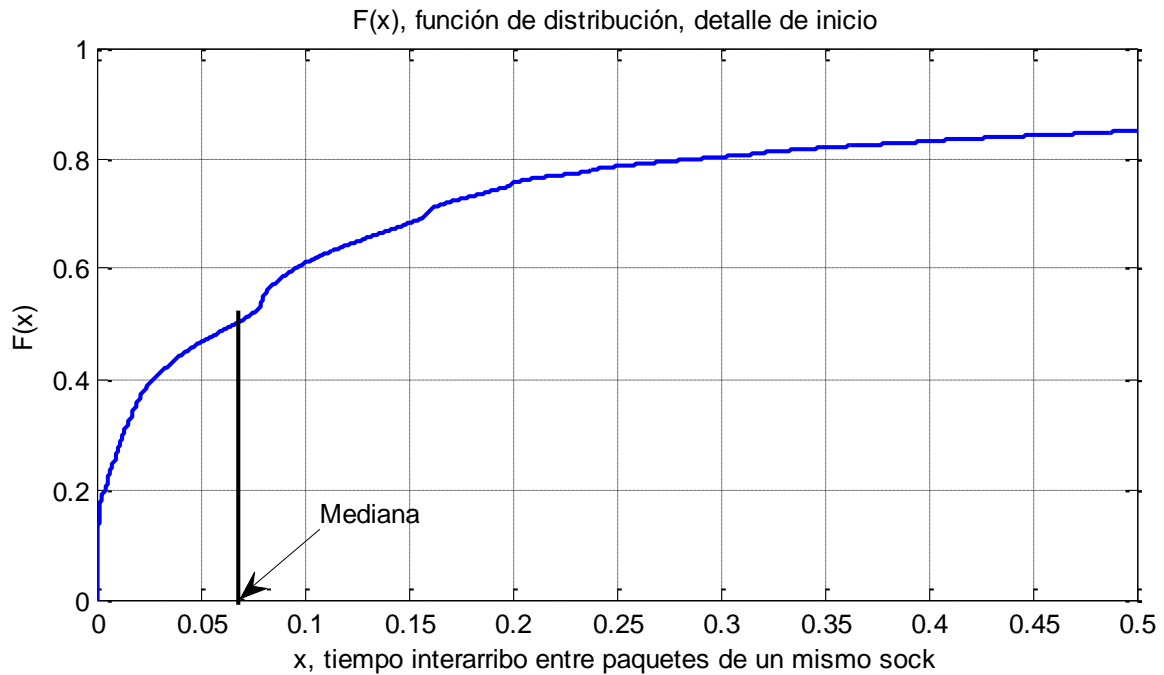


Fig. 4-11: Función de distribución de la cantidad de paquetes versus tiempo interarribo entre paquetes de cada sock. Se ha marcado la mediana que vale 67,5ms.

Algunos de estos parámetros son tiempo promedio de duración de ráfaga, velocidad promedio de aparición de nuevas ráfagas, velocidad promedio de paquetes por ráfaga y los parámetros de segundo orden de los nombrados, tales como las variancias y el coeficiente de Hurst.

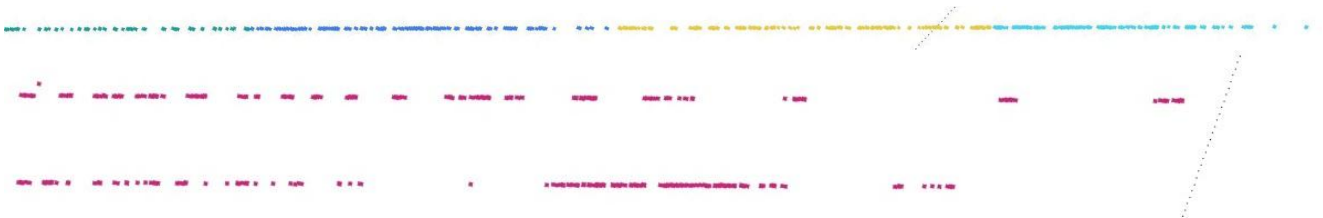


Fig. 4-12: Arriba, secuencia de paquetes correspondientes a 4 socks consecutivos enviados por la estación 4 y recibidos por la estación 19. Abajo, dos secuencias de paquetes de sock único recibidos por dos estaciones diferentes, nótese lo poco consistente que resulta asignar un sock a una ráfaga.

4.4.2.2 Resumen de las características de la muestra completa LBL-PKT4

El coeficiente de Hurst vale $H=0,81$ determinado por la pendiente de la variancia de los procesos m-agregados, mostrado en la Fig. 3-5. El comportamiento de una SSQ alimentado por el tráfico de la muestra LBL-PKT4 está graficado en la Fig. 4-3. El resto de las características está en la Tabla 4-3.

Tabla 4-3: Propiedades del total de los paquetes recibidos de la muestra LBL-PKT4. La calificación recibido o enviado es fútil, ya que cuando se toma toda la muestra los paquetes recibidos son los mismos que los enviados.

	Parámetro	Valor	Unidad
1	Nombre de la muestra	LBL-PKT4	
2	Tiempo durante el cual se recolectó la muestra	3600	s
3	Cantidad total de paquetes de la muestra, únicamente TCP+UDP	896690	paquetes
4	Cantidad de estaciones seleccionadas para analizar el tráfico recibido	1525	estaciones
5	Cantidad de paq. TCP+UDP recibidos por las estaciones seleccionadas	896690	paquetes
6	Cantidad de paq. TCP recibidos por las estaciones seleccionadas	862946 (96.2%)	paquetes
7	Cantidad total de paq. UDP recibidos por las estaciones seleccionadas	33744 (3.8%)	paquetes
8	Cantidad de paquetes promedio por estación seleccionada TCP+UDP	588	paquetes
9	Cantidad de <i>flags Syn</i>	6141	paquetes
10	Cantidad de paquetes promedio por <i>flag Syn</i>	146	paquetes
11	Cantidad de <i>socks</i> TCP+UDP	13183	<i>socks</i>
12	Cantidad de <i>socks</i> TCP	6237	<i>socks</i>
13	Cantidad de <i>socks</i> UDP	6946	<i>socks</i>
14	Cantidad de paquetes promedio por <i>sock</i> TCP+UDP	68	<i>socks</i>
15	Cantidad de paquetes promedio por <i>sock</i> TCP	138	<i>socks</i>
16	Cantidad de paquetes promedio por <i>sock</i> UDP	5	<i>socks</i>
17	Tiempo promedio duración de un <i>sock</i>	268	s
18	Tiempos interarribo entre paquetes de un mismo <i>sock</i>		
	Moda	0,00013	s
	Mediana	0,08008	s
	Promedio	4,00424	s

4.4.2.3 Observaciones finales

De la información recabada se pueden realizar las siguientes observaciones relacionadas con las propiedades de las ráfagas:

- Como se aprecia en la Fig. 4-7, existen muchísimos *socks* que no comienzan con paquetes con el *flag Syn* activado. Existen dos razones. La primera es que la mayoría de los *socks* son UDP, 1928 versus 602 TCP. La segunda es que la muestra es de solo 3600s y algunas sesiones se establecieron antes de comenzar la captura, 602-577=

25. Evidentemente la información de los *flags* Syn no es aplicable para el cálculo de la cantidad de ráfagas ya que no las representa. Otro motivo de discrepancia si se intentan asociar sesiones con ráfagas son los Syn fallidos de iniciar una sesión.
- Lamentablemente tampoco los *socks* parecen representar apropiadamente a las ráfagas. El análisis visual de una buena cantidad de *socks* ampliando la Fig. 4-5, permiten establecer que en la mayoría de los casos los *socks* están compuestos de muchas ráfagas tal como se muestra en la Fig. 4-12, también se observa en la secuencia de arriba de la misma figura que no siempre hay una separación clara entre *socks*, en este caso suponer que un *sock* es una única ráfaga es tan válido como suponer que los cuatro *sock* también constituyen una única ráfaga. Sin embargo como se supone que el ajuste del modelo generador de tráfico se realizará variando los parámetros con algún método heurístico, la cantidad de *socks* puede ser un prudente valor inicial para la cantidad de ráfagas, aunque probablemente inferior al que finalmente se determine.
 - La información estadística sobre los tiempos interarribo entre paquetes de un mismo *sock* parece un tanto más confiable para nuestros fines, ya que en la forma que se obtuvieron los datos, el concepto de *socks* solo se aplicó para evitar la superposición de conexiones TCP en la determinación de los tiempos interarribo de paquetes de una única conexión, lo cual es correcto. Se obtuvieron 3 valores estadísticos de primer orden, la moda, la mediana y la media o promedio. La moda y la media se descartan por no ser representativos, quedando la mediana como candidato para ser un buen valor inicial de la velocidad de paquetes por ráfaga.

Habiéndose analizado los conceptos sobre la cantidad de ráfagas y tiempo interarribo de paquetes de una misma ráfaga, se avanza con el diseño del modelo para simulación de tráfico.

5. Desarrollo de un modelo generador de tráfico en Simulink

5.1 Elección del algoritmo del modelo para generar tráfico

Originalmente se consideró desarrollar un modelo basado en alguna variedad o arreglo de MMPP (*Markov Modulated Poisson Processes*) los cuales acotados en tiempo, generan exitosamente un tráfico LRD, con características autosimilar, muy parecido al tráfico real, tal como es registrado en un *router* de la periferia de la LAN con acceso a Internet (20) (21).

Los modelos basados en MMPP para la generación de tráfico, seducen por su relativa fácil implementación en simuladores y también porque permiten en muchos casos un ataque analítico exitoso. Sin embargo las distribuciones Markov, por si solas no tienen el “gen” para crear naturalmente un tráfico LRD. Lo pueden simular muy bien, pero solo en un tiempo acotado debido a su característica *memoryless*. LRD implica naturalmente todo lo contrario, el futuro depende del pasado.

Mandelbrot, creador del término “autosimilar” (o fractal), sugirió originalmente que la superposición de muchas fuentes que exhiben el efecto Noah o síndrome de la variancia infinita, produce un flujo autosimilar (11). En la referencia (15), Leland y colaboradores emplearon este método para proveer una explicación de la autosimilaridad del tráfico en términos de la naturaleza del tráfico generado por cada fuente individual. Ellos sugirieron que cada fuente individual que contribuye a un tráfico autosimilar puede ser representada por la simple abstracción SI-NO, donde estas fuentes SI-NO exhiben el mencionado efecto en que los períodos de NO transmisión como los de SI transmisión tienen variancia infinita característica de las distribuciones *heavy-tailed*, cuyo exponente más sencillo y popular es la distribución Pareto.

Existen una buena cantidad de modelos para generar tráfico que se basan en este concepto. Se expondrá dos de ellos que llaman la atención por ser simples y representativos. El M/Pareto (Markov/Pareto) desarrollado por Addie, Neame y Zukerman (18) y Pareto-*modulated Poisson process* (PMPP) desarrollado por Le-Ngoc y Subramanian (13).

5.1.1 Algoritmo M/Pareto

Consta de un generador de ráfagas donde los tiempos entre inicio de cada ráfaga están dados por el valor de una variable aleatoria Poisson (exponencial⁶) de valor medio λ . La duración de

⁶ Se admite que la distribución Poisson y la exponencial son equivalentes, cuando se habla del fenómeno en general, pero numéricamente no son lo mismo. La Poisson describe la probable cantidad de paquetes a arribar en un intervalo de tiempo dado, y la exponencial describe el tiempo entre paquetes.

cada ráfaga está dada por el valor de una variable aleatoria Pareto de factor de forma α y valor medio $\mu = \frac{b\alpha}{\alpha-1}$. En 2.4.1 se describe con detalle la función Pareto.

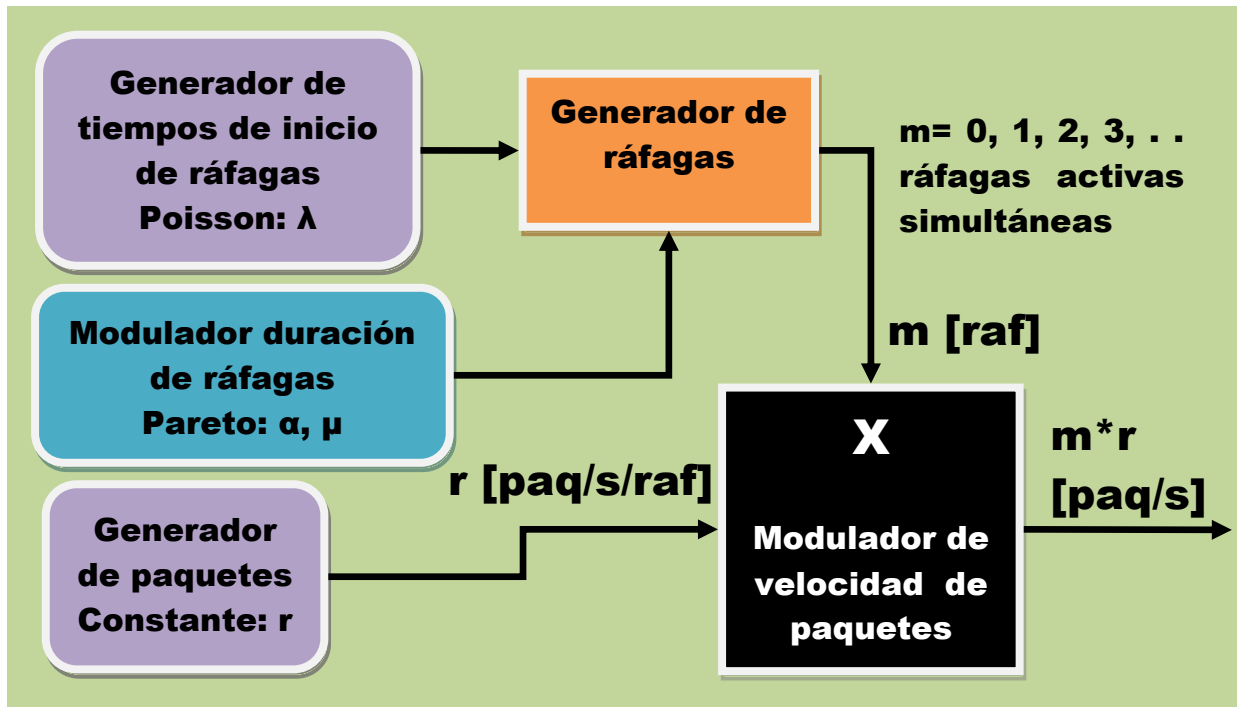


Fig. 5-1: Diagrama en bloques del generador M/Pareto.

A continuación se describe el funcionamiento tomando como referencia la Fig. 5-1.

Las ráfagas se superponen en el tiempo. Cuando se inicia una ráfaga puede haber ninguna, una o muchas ráfagas activas que se iniciaron anteriormente. La cantidad de ráfagas activas superpuestas $m=0,1,2,\dots$ modulan la salida de un generador de paquetes con velocidad constante r [paq/s/raf]. A las salida del modulador se obtiene una producción de paquetes con una velocidad $m*r$, donde r es la velocidad promedio de paquetes por cada ráfaga.

Si el modelo se ejecuta durante un tiempo t , lo suficientemente largo, la cantidad de paquetes generados debería ser próxima a $n \approx t\lambda r\mu$.

H está relacionado con la pendiente $-\alpha$ de la asíntota de la CCDF de la duración de cada ráfaga por $H = \frac{3-\alpha}{2}$ (15) de donde $\alpha = 3 - 2H$.

En este modelo el agregado de fuentes se maneja incrementando λ .

5.1.2 Algoritmo Pareto-Modulated Poisson Process (PMPP)

En este modelo los tiempos de transmisión de las fuentes individuales de tráfico, son caracterizados por una distribución Pareto. En 2.4.1 se describen detalles de la función Pareto.

Ha sido probado en (22) que si $1 < \alpha < 2$ para los tiempos de SI transmisión y NO transmisión de los procesos individuales de cada fuente, el resultado de la superposición de los mismos constituyen un tráfico autosimilar con coeficiente de Hurst $H = \frac{3-\alpha}{2}$.

Esta última ecuación brinda una relación valiosa, a los fines de esta tesis, entre los procesos de las fuentes individuales y el tráfico resultante de la agregación de los mismos. La autosimilaridad es medida por el coeficiente de Hurst H . Los procesos autosimilares con $0,5 < H < 1$ presentan una función de autocorrelación con dependencias de largo alcance (LRD). Los procesos con un valor alto de H , cercano a 1, están constituidos por ráfagas altamente explosivas, mientras que aquellos cuyo valor de H es bajo, cercano a 0,5, esto no sucede.

En el método PMPP, el tráfico agregado se aproxima por un doble proceso de Poisson. En este doble proceso el tráfico se origina conmutando entre dos fuentes que generan paquetes con una distribución de Poisson en el tiempo, con valores medio respectivamente λ_1 y λ_2 . La mejor forma de analizarlo es con un diagrama de estados, Fig. 5-2.

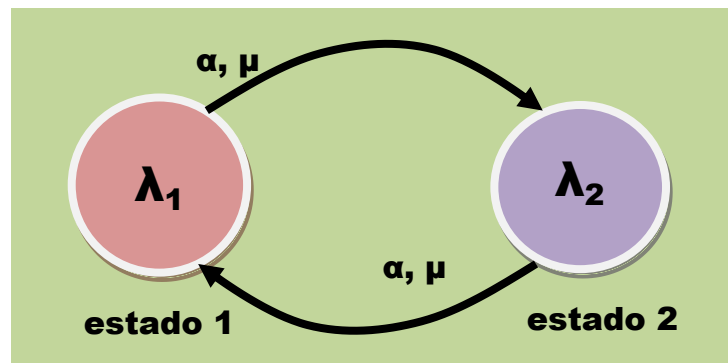


Fig. 5-2: Diagrama de estado del modelo PMPP.

En el estado 1 se generan paquetes a una velocidad promedio de λ_1 paquetes/s mediante un proceso exponencial. El tiempo de permanencia en este estado depende del valor de una variable aleatoria con una distribución Pareto, cuyo factor de forma vale α y valor medio $\mu = \frac{b\alpha}{\alpha-1}$. Transcurrido el tiempo definido por el valor de la variable Pareto en el estado 1, se conmuta al estado 2, donde se generan paquetes a una velocidad promedio λ_2 permaneciendo en este estado el tiempo definido por un nuevo valor de la variable aleatoria Pareto. Finalizado este período se conmuta al nuevamente al estado 1 y así sucesivamente.

H está relacionado con la pendiente $-\alpha$ de la asíntota de la CCDF de la distribución Pareto correspondiente a la duración de cada ráfaga por $H = \frac{3-\alpha}{2}$ (15).

En este modelo el agregado de fuentes se maneja simplemente aumentando los valores de λ_1 y λ_2 .

La cantidad de paquetes generados debería ser próxima a $n = t \frac{\lambda_1 + \lambda_2}{2}$ donde t , es el tiempo que se considera, λ_1 y λ_2 son las velocidades promedio de generación de paquetes conforme a un proceso de Poisson (exponencial). La ecuación anterior solo define la suma de λ_1 y λ_2 .

El valor de λ_1 (o de λ_2), además de α y μ , son los tres parámetros de ajuste necesarios para darle la forma necesaria a la curva paquetes perdidos a paquetes totales versus Capacidad de la cola (ver Fig. 4-1 y Fig. 4-3) para que produzca el mismo efecto en una SSQ que el tráfico que se desea simular.

5.2 Modelos en Simulink para los generadores M/Pareto y PMPP

Como Simulink no dispone de una función de librería para generar distribuciones Pareto, éstas se generarán en el *workspace* de Matlab con funciones de librería de Matlab. Una vez generada la secuencia Pareto queda disponible para el modelo Simulink como un arreglo en el *workspace*. Las funciones en Matlab para generar las secuencias Pareto se transcriben en el Anexo 3 – *Scripts* para simulación con el modelo M/Pareto y Anexo 4 – *Scripts* para simulación con el modelo PMPP.

Se informa que se desarrollaron varios modelos en Simulink para crear distribuciones Pareto secuenciales (no *batch*), pero fueron desestimados, ya que no pudieron competir con la eficiencia y simplicidad de las funciones de librería de Matlab. Además los mismos resultaron algo rebuscados, como para exponerlos en este contexto, donde entre otros objetivos se busca mostrar la sencillez y practicidad de Simulink, y en especial del *toolbox* SimEvents para el tratamiento y simulación del tráfico.

5.2.1 Modelo en Simulink para el generador M/Pareto

Para analizar la arquitectura y el funcionamiento refiérase a la Fig. 5-4. El modelo consta de dos cadenas de entidades cada una con su generador *Generator* y su *Sink* (los *Sinks* no son obligatorios pero se han puesto por formalidad). En la cadena inferior se generan las ráfagas y en la superior los paquetes.

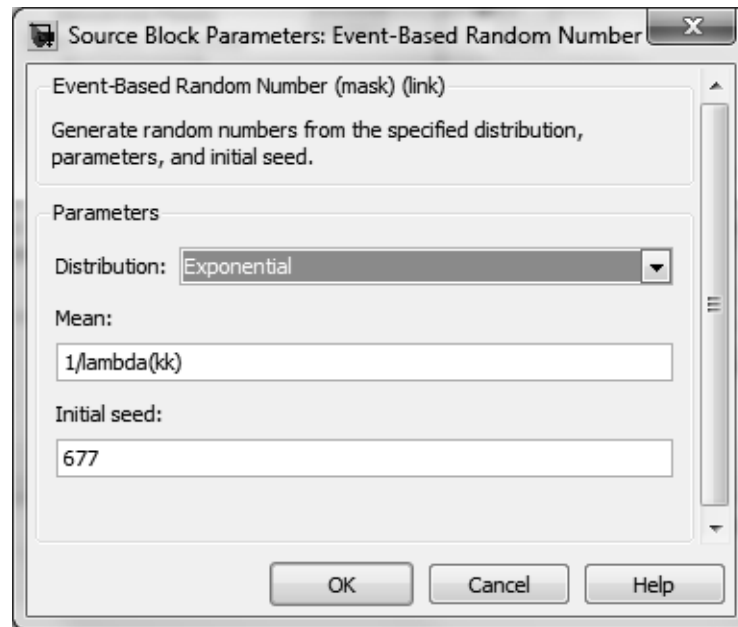


Fig. 5-3: Ventana de diálogo del bloque Event- Based Random Number

Si se clikea dos veces sobre el bloque *Event-Based Random Number* se abre una ventana de diálogo con las propiedades, como se aprecia en la Fig. 5-3. Se seleccionó distribución exponencial con una media de tiempo interarribo de $1/\lambda(kk)$. $\lambda(kk)$ es un elemento de un arreglo, ya que el modelo será manipulado desde Matlab mediante un *script* para automatizar la simulación. En el momento de ejecutarse el modelo, el arreglo λ debe estar presente (cargado en memoria) en el *workspace* de Matlab, de otra forma se producirá un *runtime error*. Este generador provee los tiempos interarribo de las ráfagas conforme a una distribución exponencial.

La salida de este bloque toma sucesivos valores conforme a esta distribución, pero solo se hace presente un nuevo valor, después que una entidad haya partido del bloque subsiguiente, el *Time-Based Entity Generator*, cuya ventana de diálogo se muestra en la Fig. 5-5. Una nueva entidad partirá de este último bloque transcurrido el tiempo correspondiente al presente valor de t , y así sucesivamente. El puerto t del bloque *Time-Based Entity Generator* se lo denomina *notifying port* por esta propiedad.

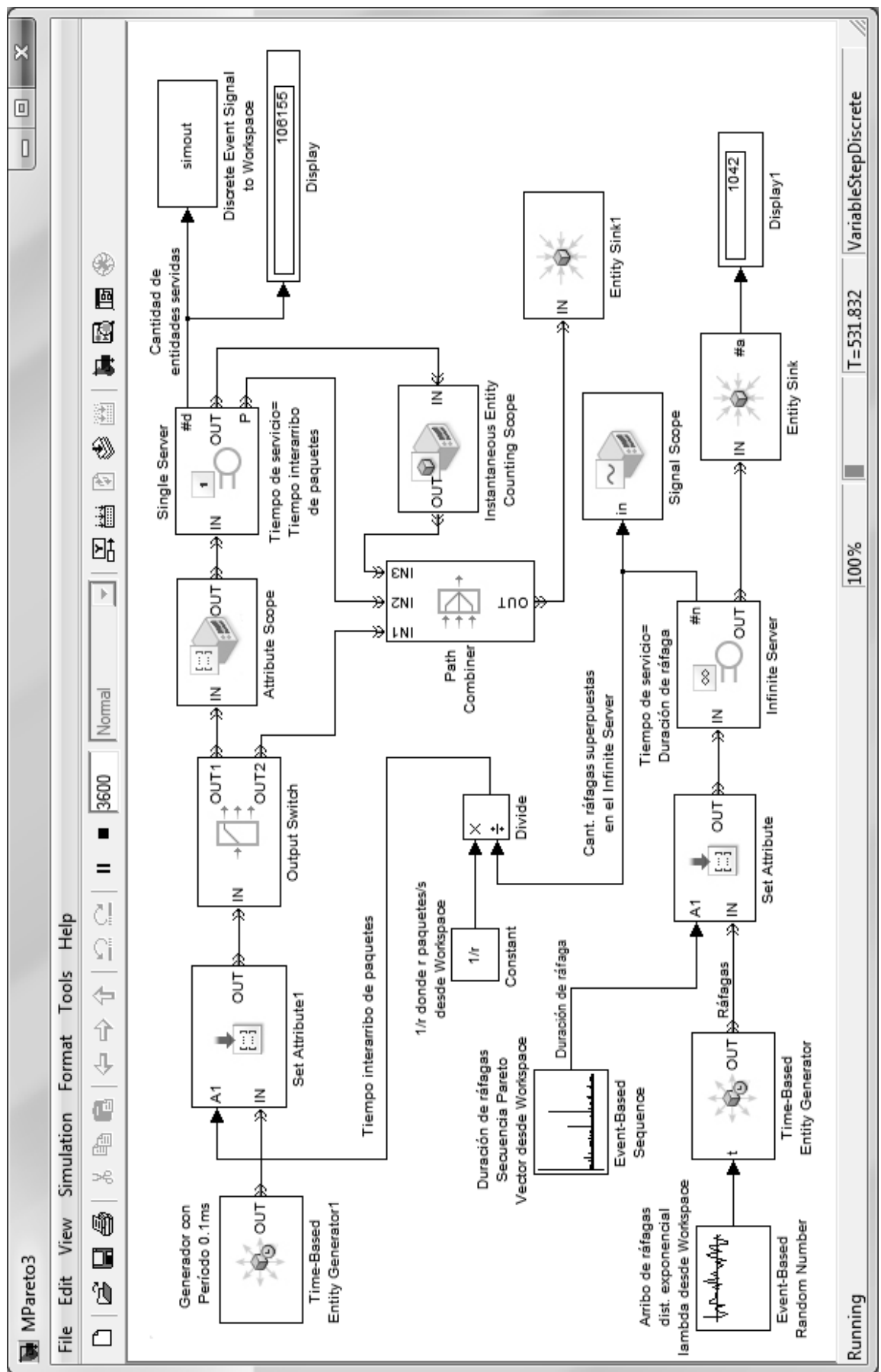


Fig. 5-4: Modelo M/Pareto realizado en Simulink.

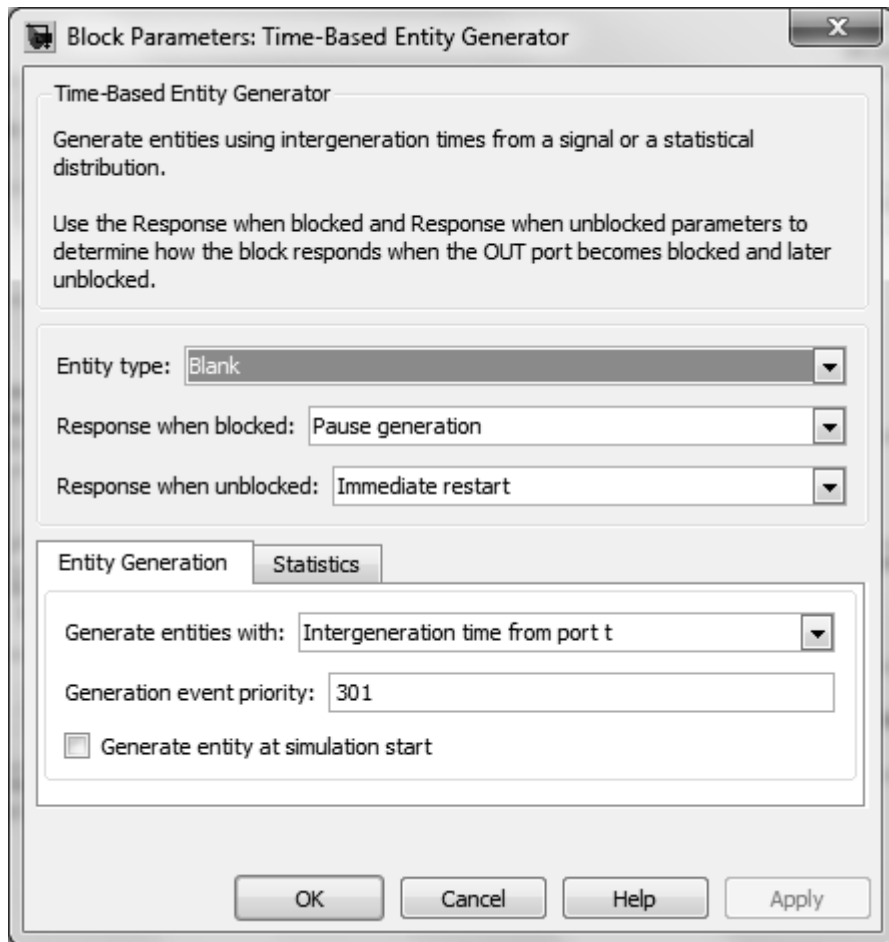


Fig. 5-5: Ventana de diálogo del bloque Time-Based Entity Generator.

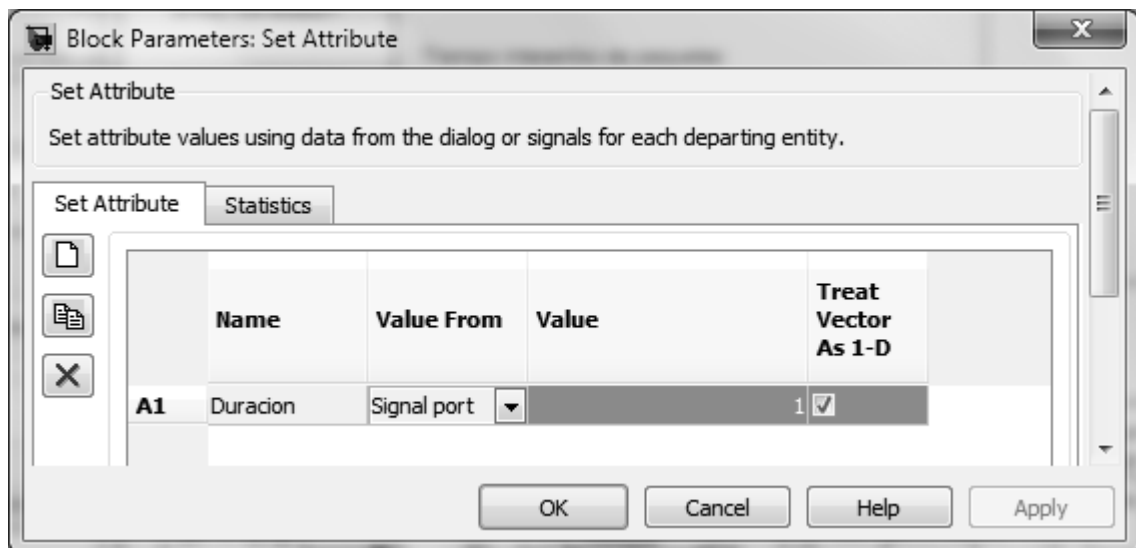


Fig. 5-6: Ventana de diálogo del bloque Set Attribute.

Entre el *Time-Based Entity Generator* y el *Infinite Server* se intercala el bloque *Set-Attribute*, Fig. 5-6. Este bloque crea una propiedad a cada entidad, si no fue creada previamente, asignándole un valor conforme a su *notifying port* de entrada A1. En este caso a cada ráfaga

se le asigna una duración dada por el bloque *Event- Based Sequence*, cuya secuencia con distribución Pareto es tomada del arreglo *VectorPareto* del *workspace* de Matlab, Fig. 5-7.

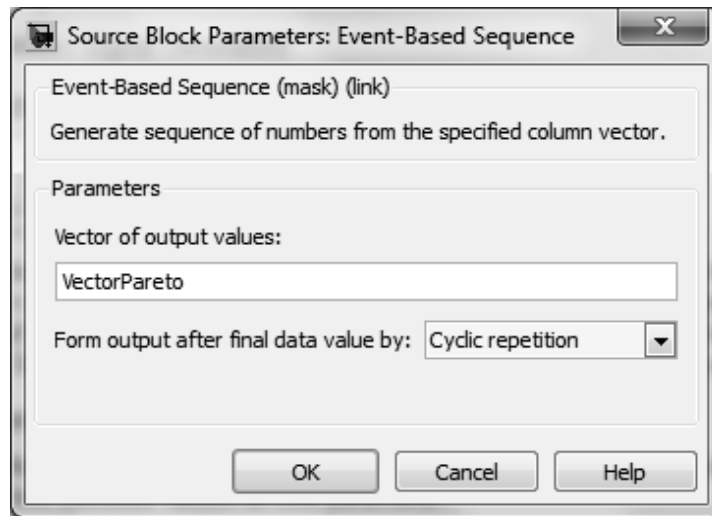


Fig. 5-7: Ventana de diálogo del bloque *Event-Based Sequence*.

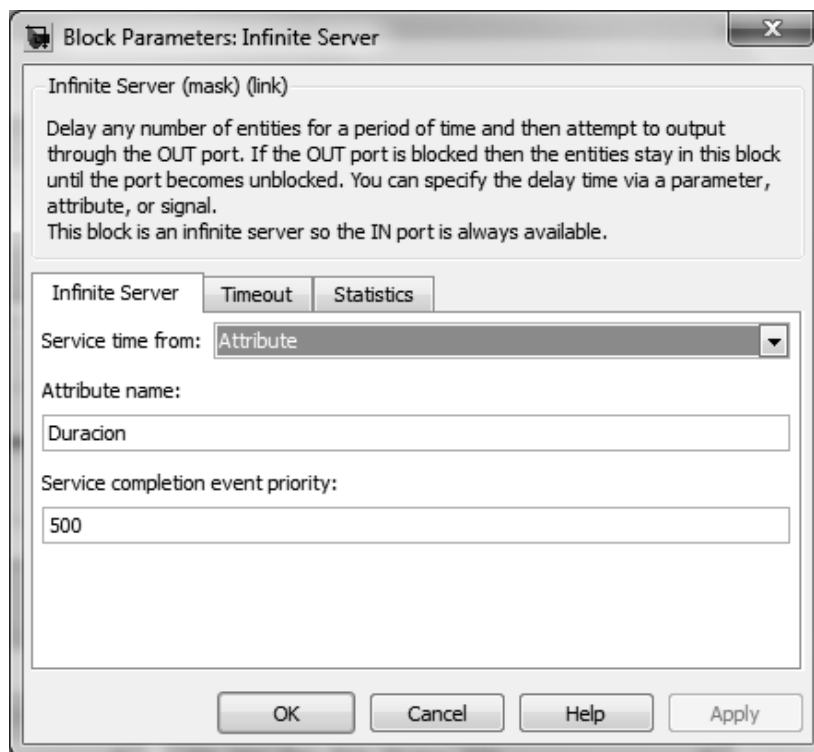


Fig. 5-8: Ventana de diálogo del bloque *Infinite Server*.

El bloque que sigue es el ya mencionado *Infinite Server*, ver Fig. 5-8. Es un servidor compuesto de infinitos servidores idénticos en paralelo. Se configura para que el tiempo de servicio se corresponda con la duración establecida por la distribución Pareto, consignada en

el atributo Duracion de cada entidad que arribe. La salida estadística #n indica la cantidad de ráfagas que están siendo servidas simultáneamente en el bloque.

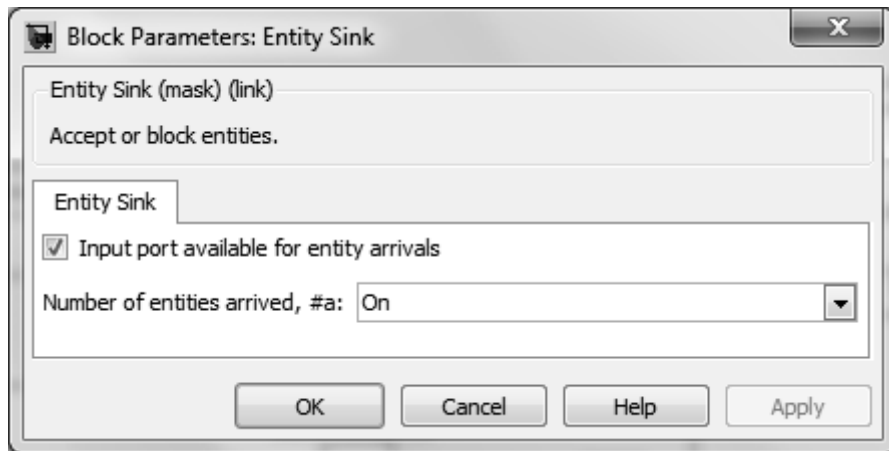


Fig. 5-9: Ventana de diálogo del bloque Entity Sink.

Las entidades, cumplida su función se descartan en el bloque sumidero *Entity Sink*, Fig. 5-9, cuya salida estadística #a alimenta a un bloque *Display1* que informa la cantidad de entidades (ráfagas) arribadas del bloque anterior.

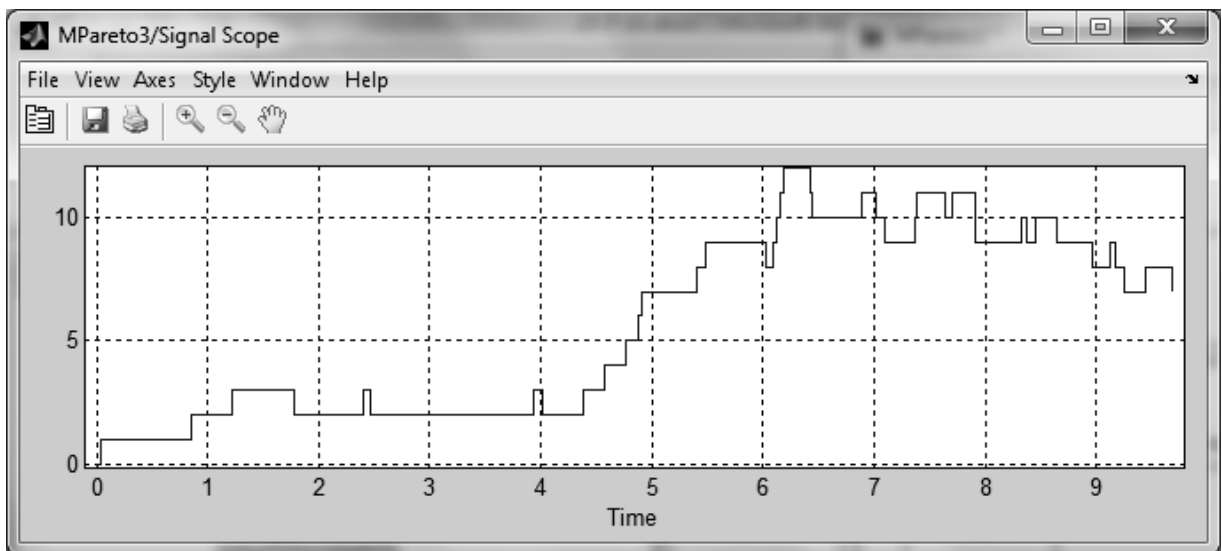


Fig. 5-10: Oscilograma de la cantidad de ráfagas superpuestas #n en el bloque *Infinite Server*.

A la salida #n del bloque *Infinite Server* se le ha conectado un *Signal Scope* para investigar si el proceso funciona correctamente. En la Fig. 5-10 se muestran los valores de #n de 0 a 10s. Por ejemplo en el tiempo de corrida del modelo igual a 5s, hay 7 ráfagas que están siendo servidas simultáneamente.

La información #n deberá estar disponible para fijar el tiempo de servicio del bloque *Single Server* de la cadena superior. Pero #n no representa el tiempo interarribo de paquetes,

sino la cantidad de ráfagas activas simultáneamente, por ello con auxilio del bloque *Divide* y el bloque *Constant* (cuyo valor *r* se obtiene del *workspace*) se calcula el Tiempo interarribo de paquetes = $1/(r * \#n)$, cuyo oscilograma producido por el *Signal Scope1* se muestra en la Fig. 5-11, nótese la inversión respecto al oscilograma de la Fig. 5-10.

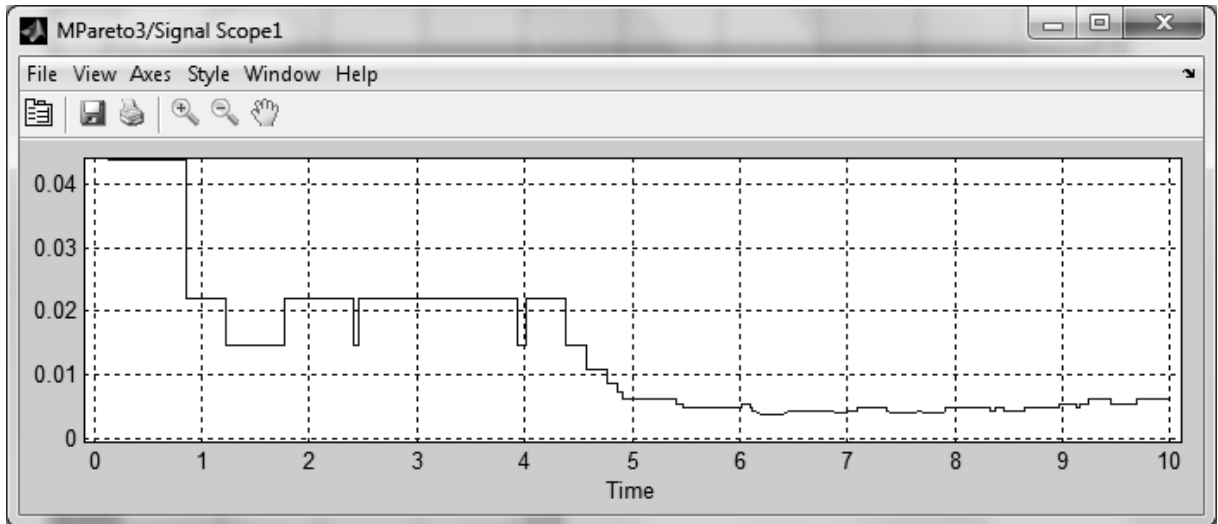


Fig. 5-11: Oscilograma del Tiempo interarribo de paquetes.

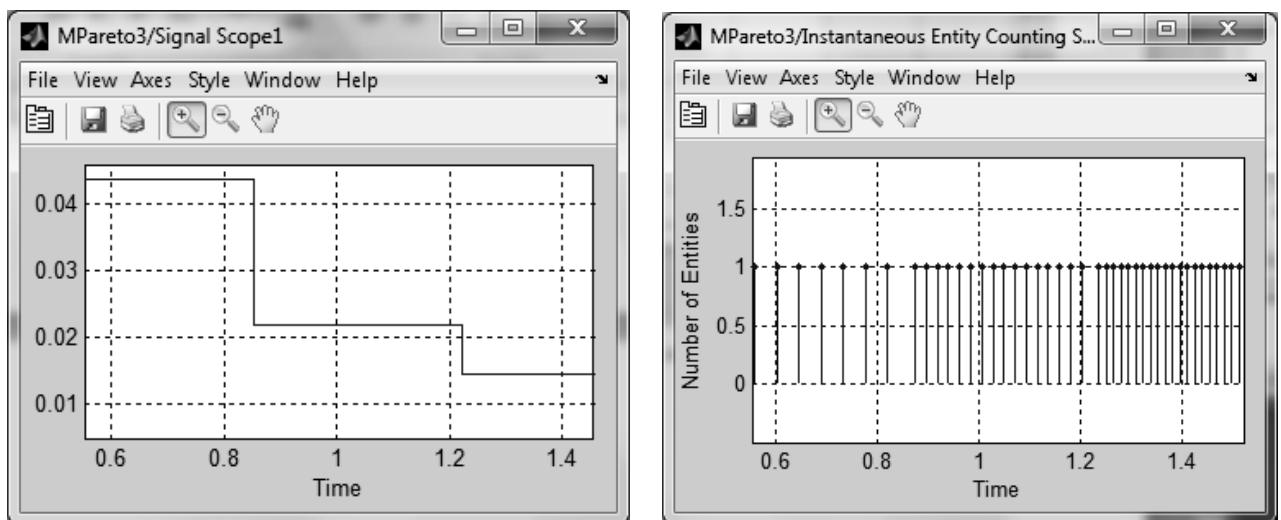


Fig. 5-12: Vistas coordinadas en el tiempo del *Signal Scope1* y el *Instantaneous Entity Counting Scope*.

En la Fig. 5-12 se muestra un detalle, de 0,6 a 1,4s, de la Fig. 5-11 y su correspondiente efecto en el bloque *Instantaneous Entity Counting Scope*.

Se analiza ahora la cadena superior. A la izquierda comienza con un bloque *Time-Based Entity Generator1*, el mismo es un generador de intervalos de tiempo de 0,1ms, Fig. 5-13. Se seleccionó 0,1ms como el mínimo tiempo interarribo posible entre paquetes para el tráfico

que se desea simular. Sea una conexión Ethernet de 100Mbps/s, supóngase paquetes

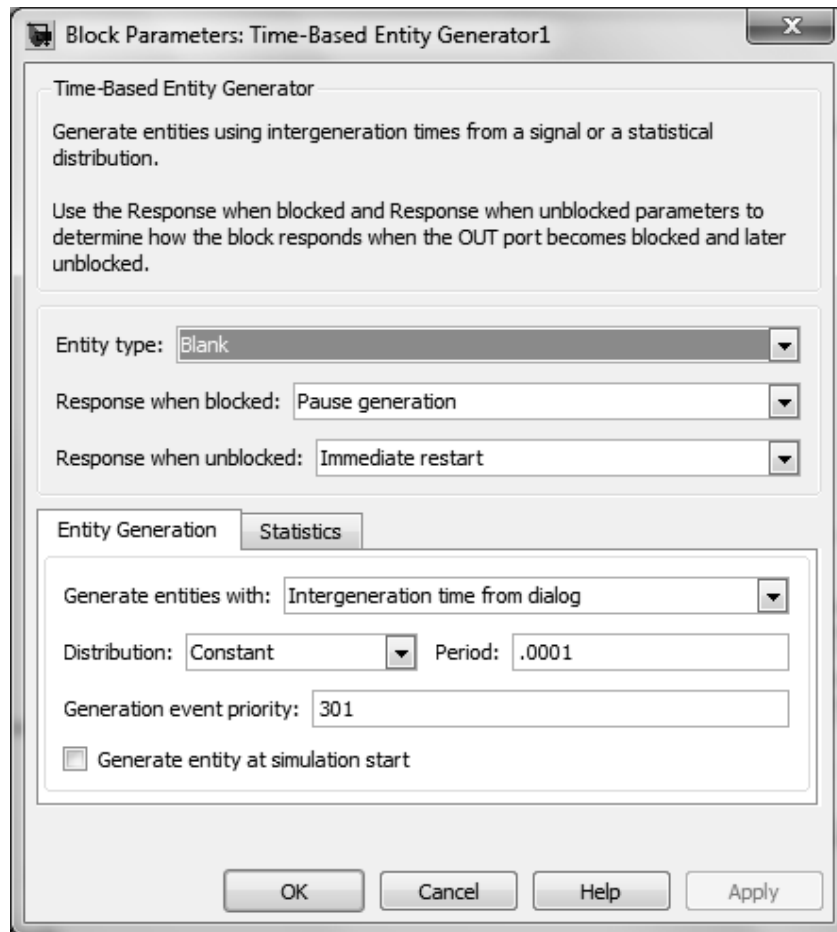


Fig. 5-13: Ventana de diálogo del bloque *Time-Based Entity Generator1*.

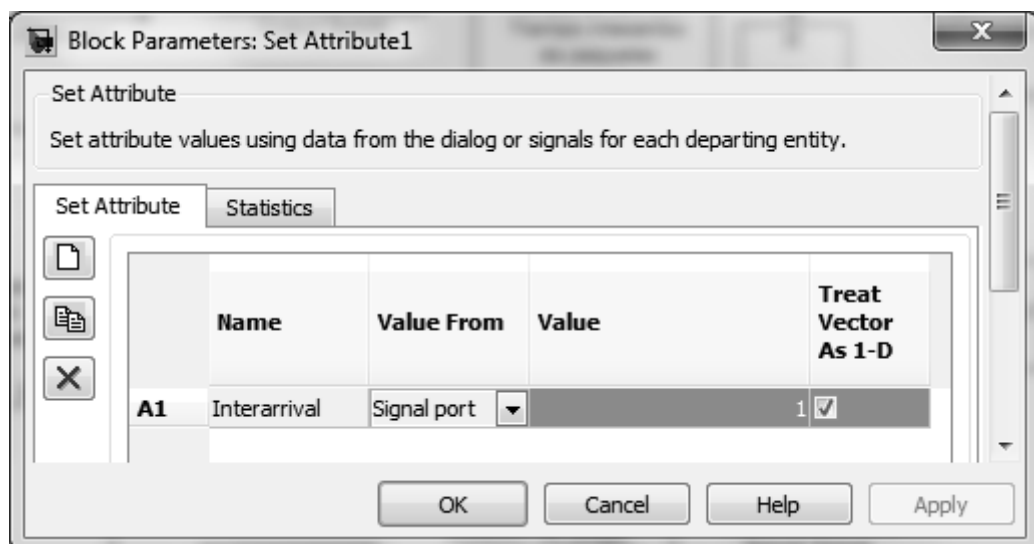


Fig. 5-14: Ventana de diálogo del bloque *Set attribute1*.

únicamente de 1000 bytes, el tiempo interarribo de paquetes no puede ser menor que $\frac{1000\text{byte}}{\frac{100\text{Mbit}}{\text{s}} \cdot \frac{8\text{byte}}{\text{bit}}} = 1,25\mu\text{s}$, en este caso 0,1ms es congruente con el tráfico que se quiere simular que es el almacenado en el archivo LBL-PKT4. Cuanto mayor es este tiempo más rápida es la simulación. Básicamente la función del bloque es disponer siempre paquetes frescos, para alimentar al bloque *Single Server*.

Le sigue un bloque *Set Attribute1* que marca los paquetes con el tiempo interarribo provisto por la cadena inferior del modelo, en la Fig. 5-14 se muestra la ventana de diálogo correspondiente.

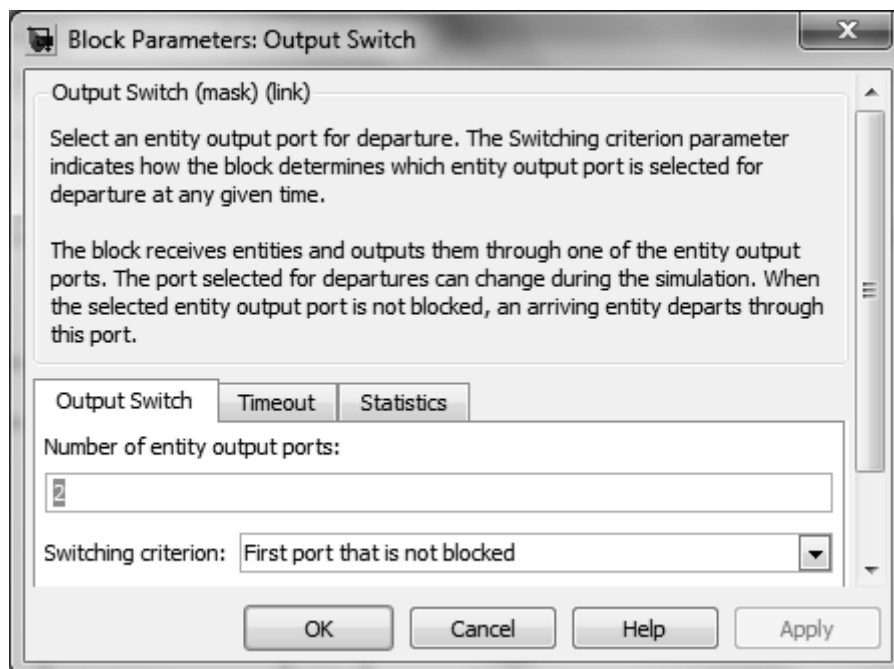


Fig. 5-15: Ventana de diálogo del bloque *Output Switch*.

El bloque que siguiente es un *Output Switch*, Fig. 5-15. Si el *Single Server* está bloqueado sirviendo un paquete el paquete es desviado al sumidero *Entity Sink1* por el puerto OUT2 de esta forma el *Time-Based Entity Generator1* nunca quedará bloqueado.

La cadena continúa con el bloque *Single Server*, cuyo tiempo de servicio queda definido por el atributo consignado en cada paquete cuyo nombre es Interarrival, Fig. 5-16.

Fue necesario modificar los datos de la solapa *Preemption* por una sutil cuestión que complicó el diseño. Cuando la cantidad de ráfagas superpuestas en el *Infinite Server* es nula, el tiempo interarribo de paquetes es infinitamente grande o un valor que por programa se pueda asignar arbitrariamente para representar el tiempo interarribo mas grande que pueda suceder. Si tras esta situación a los pocos milisegundos arriban muchas ráfagas al *Infinite Server*, #n se incrementa marcándose los paquetes que emergen del bloque *Time-Based Entity*

Generator1 con un tiempo interarribo de paquetes pequeño, pero estos son descartados por el OUT2 del bloque *Output Switch* debido a que el puerto IN del *Single Server* es del tipo *notifying port*, y este último bloque nombrado no aceptará paquetes hasta que haya servido el que tiene en curso cuyo tiempo de servicio es un valor elevado.

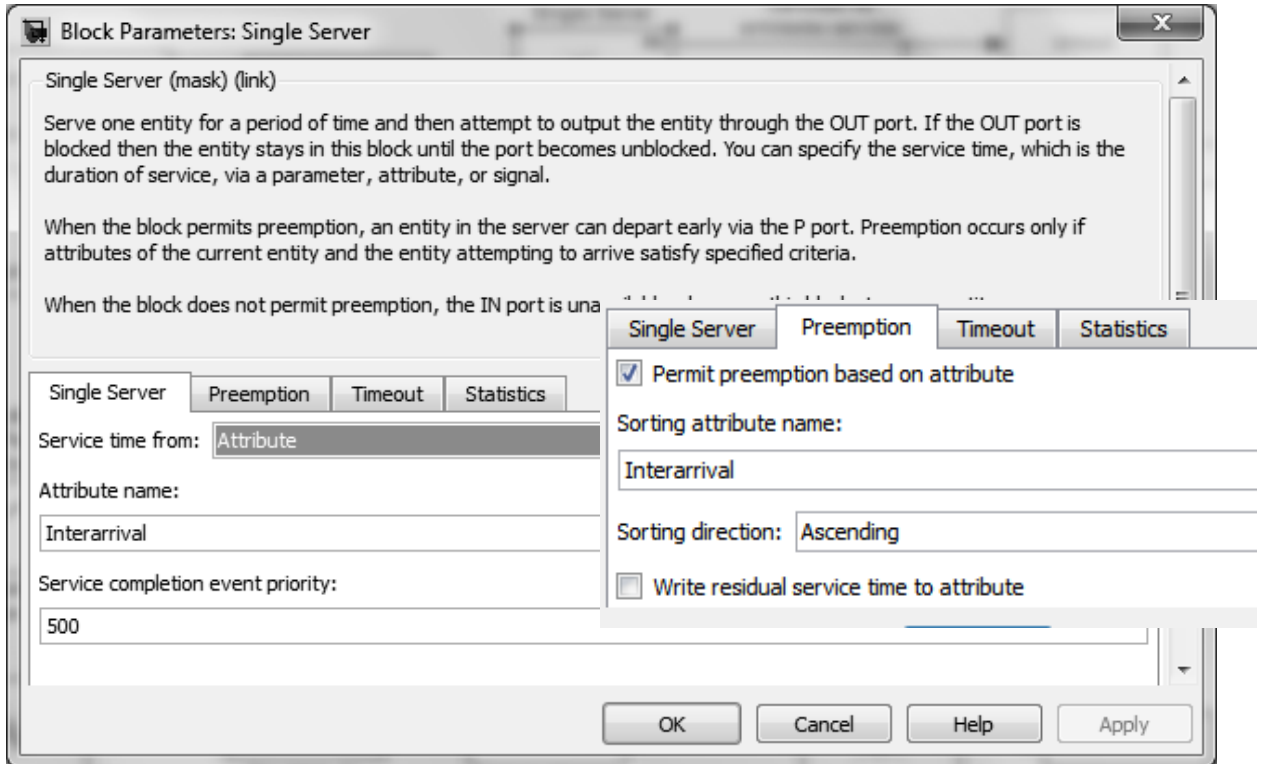


Fig. 5-16: Ventana de diálogo del bloque *Single Server*. Se modificó los datos de la pestaña *Preemption*.

La situación planteada es que un único paquete puede detener el tráfico de muchos otros. Esta situación no parece ajustarse a la realidad, pero puede ser discutible esta opinión, de hecho puede generar espaciamientos que reproduzcan mas fielmente el tráfico que el generador pretende simular, situación que no se ha investigado especialmente en lo que respecta el efecto sobre las colas.

Al permitirse *Preemption* por los paquetes con menor tiempo interarribo como atributo (*Attribute name* Interarrival), si el paquete presente posee como atributo un tiempo interarribo superior se descarta por el puerto P desbloqueando el *Single Server* para el que llega.

La salida estadística #d del *Single Server* es enviada al *workspace* por el bloque *Discrete Event Signal to Workspace*, Fig. 5-17, en forma de una estructura bautizada simout, en cuyo vector simout.time está el producto final de este modelo. Se agrega a la misma salida un bloque *Display* que monitorea la cantidad de paquetes generados mientras se ejecuta el modelo.

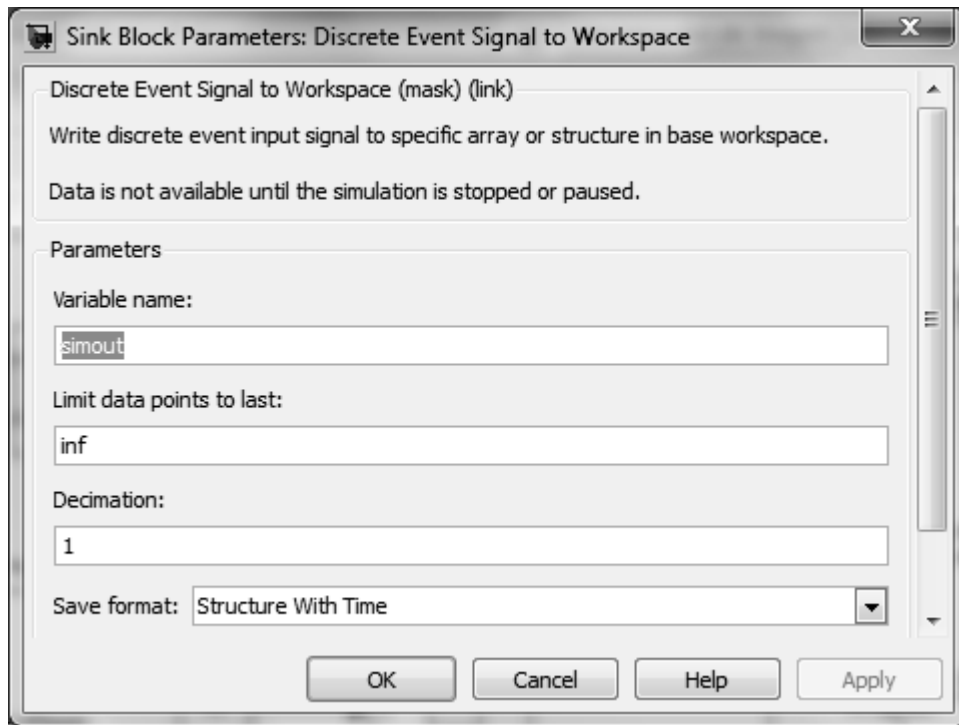


Fig. 5-17: Ventana de diálogo del bloque *Discrete Event Signal to Workspace*.

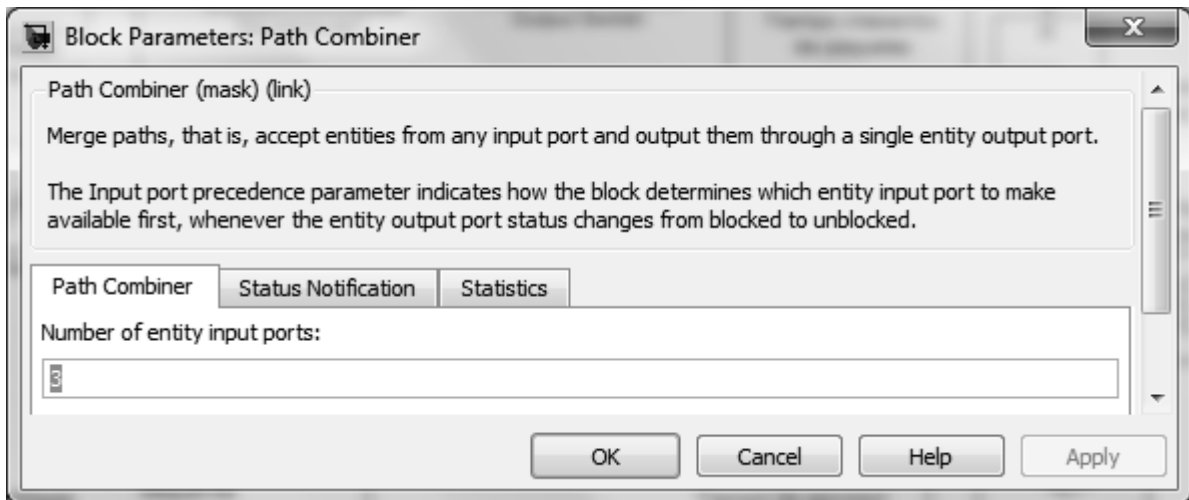


Fig. 5-18: Ventana de diálogo del bloque *Path Combiner*.

El *Path Combiner*, Fig. 5-18, combina los paquetes servidos con los no servidos para su deposición en el bloque *Entity Sink1* que es idéntico al de la Fig. 5-9, excepto por el nombre.

5.2.2 Modelo en Simulink para el generador PMPP

Para analizar la arquitectura y el funcionamiento refiérase a la Fig. 5-20. El modelo está compuesto por tres cadenas de entidades. En la parte inferior, las entidades de ambas cadenas representan paquetes. En la parte superior las entidades de la cadena representan ráfagas.

Los bloques gemelos *Time-Based Entity Generator* y *Time-Based Entity Generator1*, generan entidades con un tiempo interarribo que depende del valor de sus *notifying ports* t en el momento de generarse, Fig. 5-19. Solo aceptarán un nuevo valor de t cuando el bloque *Input Switch* habilite la entrada correspondiente. Los bloques *Event-Based Random Number* y *Event-Based Random Number1* generan valores discretos con distribución exponencial, con valores medios $1/\lambda_1$ y $1/\lambda_2$ respectivamente, Fig. 5-21 y Fig. 5-22.

Estos valores no son generados si alguno de los *notifying ports* t de los *Time-Based Entity Generators* subsiguientes tiene su salida de entidades OUT bloqueada. Ni bien se desbloquean el puerto de entrada t acepta un nuevo valor.

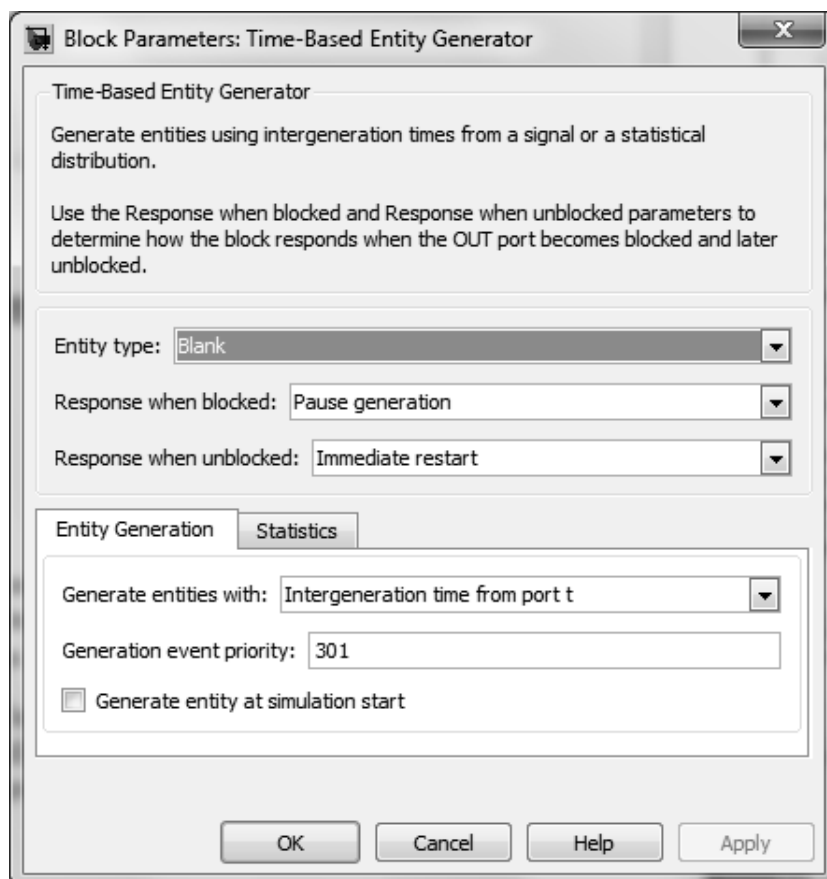


Fig. 5-19: Ventana de diálogo correspondiente a los bloques *Time-Based Entity Generator* y *Time-Based Entity Generator1* y *Time-Based Entity Generator2*.

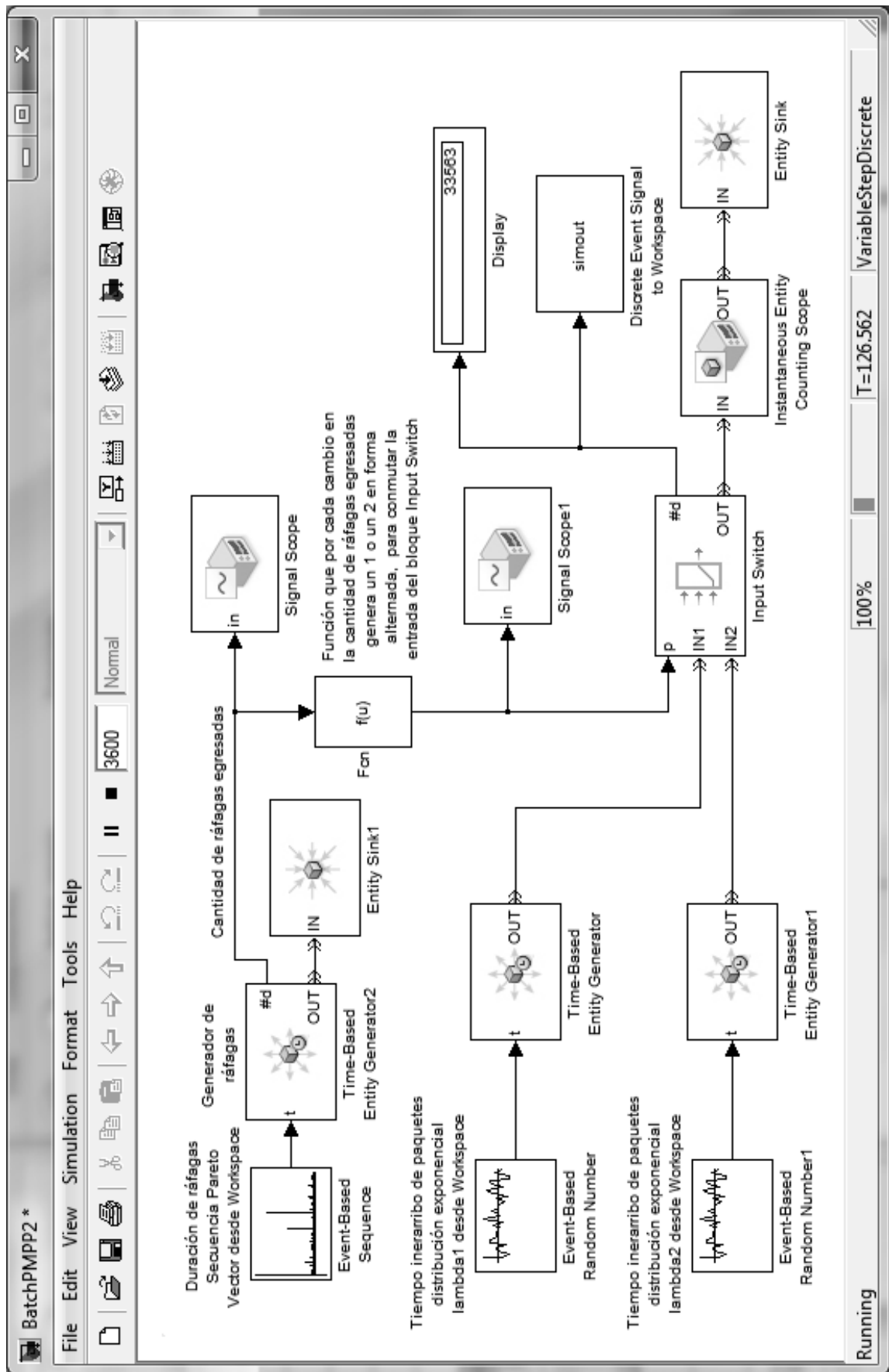


Fig. 5-20: Modelo PMPP realizado en Simulink.

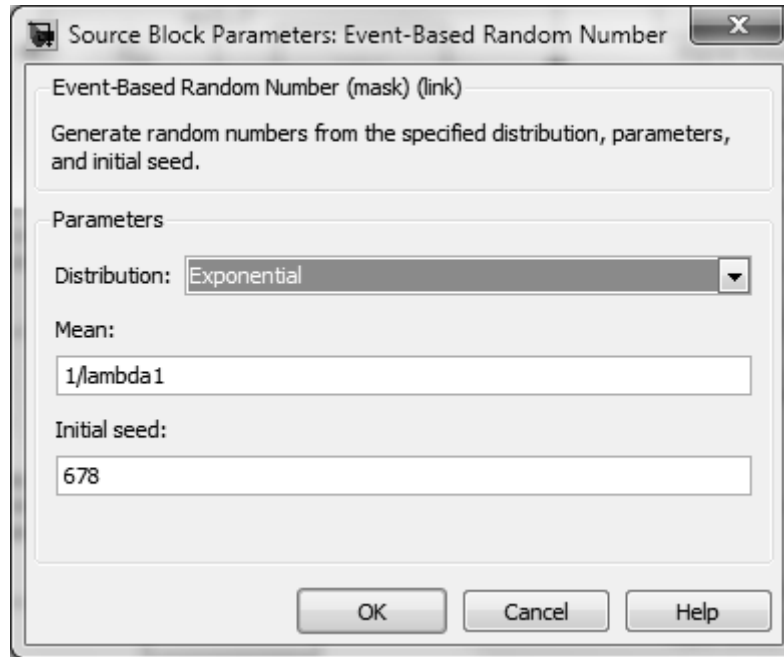


Fig. 5-21: Ventana de diálogo correspondiente al bloque *Event-Based Random Number*.

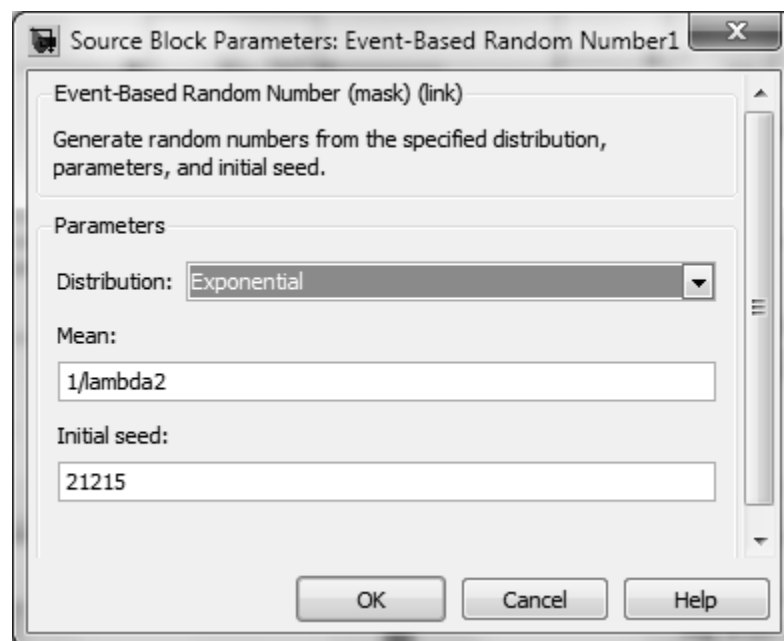


Fig. 5-22: Ventana de diálogo correspondiente al bloque *Event-Based Random Number1*.

Las dos cadenas inferiores se aúnan en el bloque *Input Switch*, cuya entrada se selecciona conforme al valor de la entrada P, que solo puede ser 1 o 2, Fig. 5-23. El puerto de estadística #d informa la cantidad de paquetes egresados del bloque.

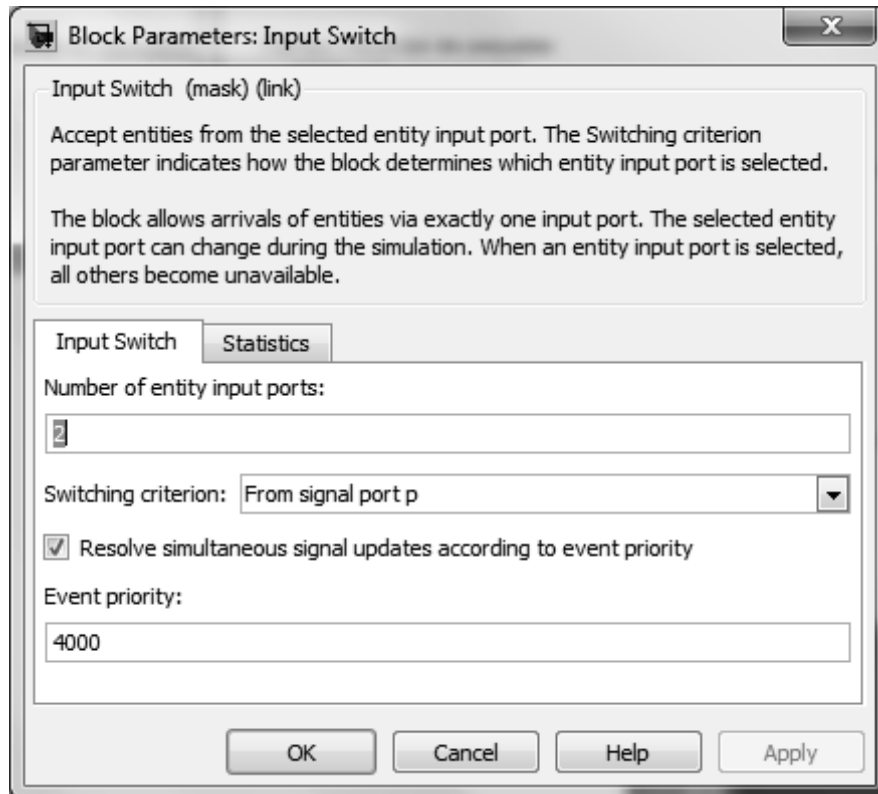


Fig. 5-23: Ventana de diálogo correspondiente al bloque *Input Switch*.

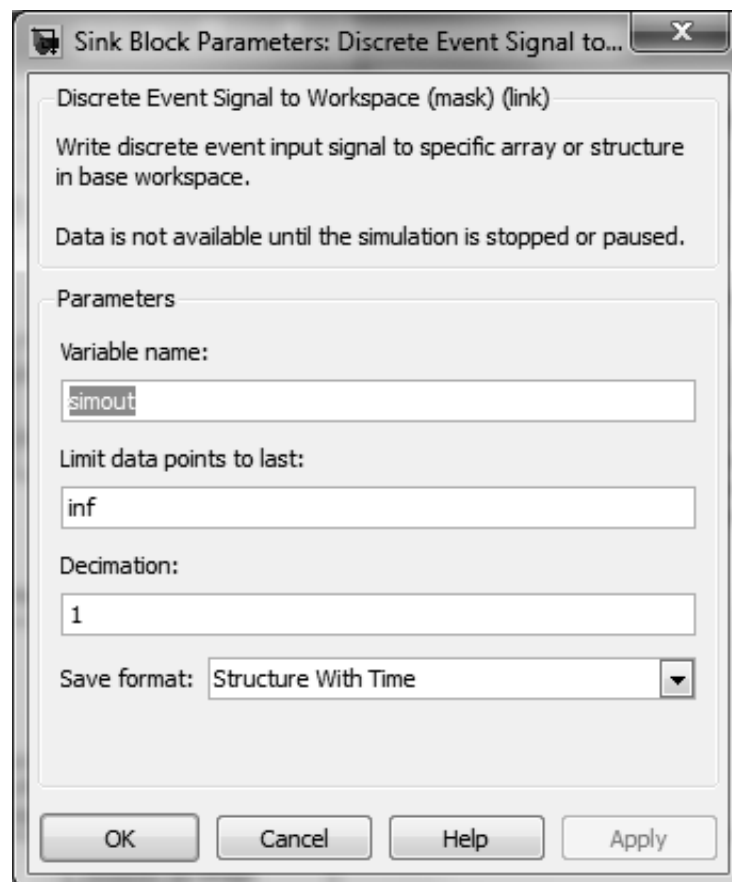


Fig. 5-24: Ventana de diálogo del bloque *Discrete Event Signal to Workspace*.

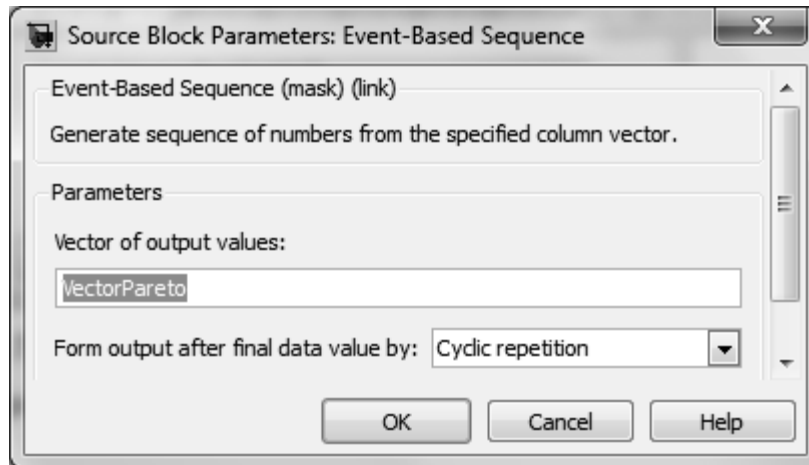


Fig. 5-25: Ventana de diálogo del bloque *Event-Based Sequence*.

El puerto OUT del *Input Switch* se conecta a un *Instantaneous Entity Counting Scope* que permite monitorear el arribo de paquetes mientras se corre el modelo. A su vez este último bloque conduce las entidades al bloque sumidero *Entity Sink*.

La salida estadística #d provee la secuencia temporal de paquetes del tráfico simulado, la misma es enviada al *workspace* por el bloque *Discrete Event Signal to Workspace* en forma de una estructura bautizada *simout*,

Fig. 5-24. Este es el producto final de este modelo.

La misma salida se deriva al al bloque *Display* que monitorea la cantidad de paquetes generados.

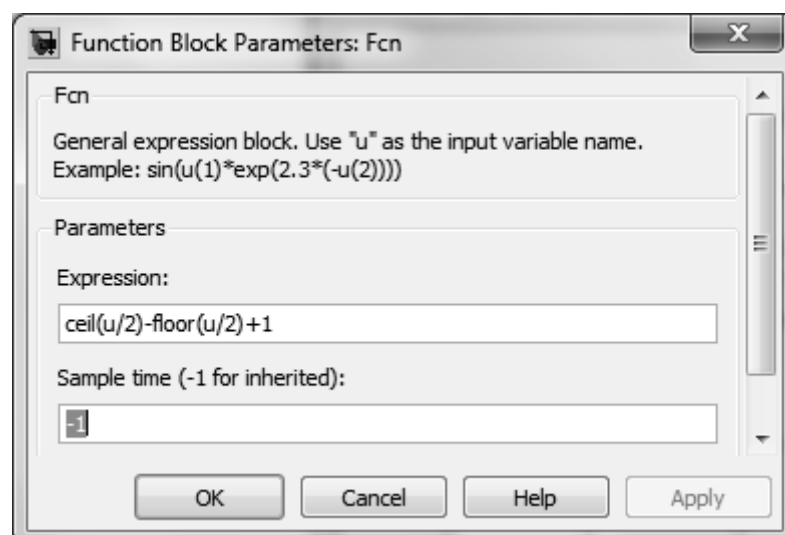


Fig. 5-26: Ventana de diálogo del bloque *Fcn*.

La cadena superior comienza con el bloque *Event-Based Sequence*, Fig. 5-25. Este bloque carga desde el *workspace* un arreglo de tiempos interarribo denominado *VectorPareto*, generado previamente en Matlab conforme a una distribución Pareto.

La salida de este bloque se conecta al *notifying port* del bloque *Time-Based Entity Generator2*, Fig. 5-19. Este bloque genera entidades las cuales representan ráfagas con tiempos interarribo definidos por el valor de la señal presente en el puerto *t*.

La salida OUT conduce las entidades generadas al bloque sumidero *Entity Sink1*.

La salida estadística #d del bloque *Time-Based Entity Generator2* se incrementa en una unidad con cada ráfaga egresada.

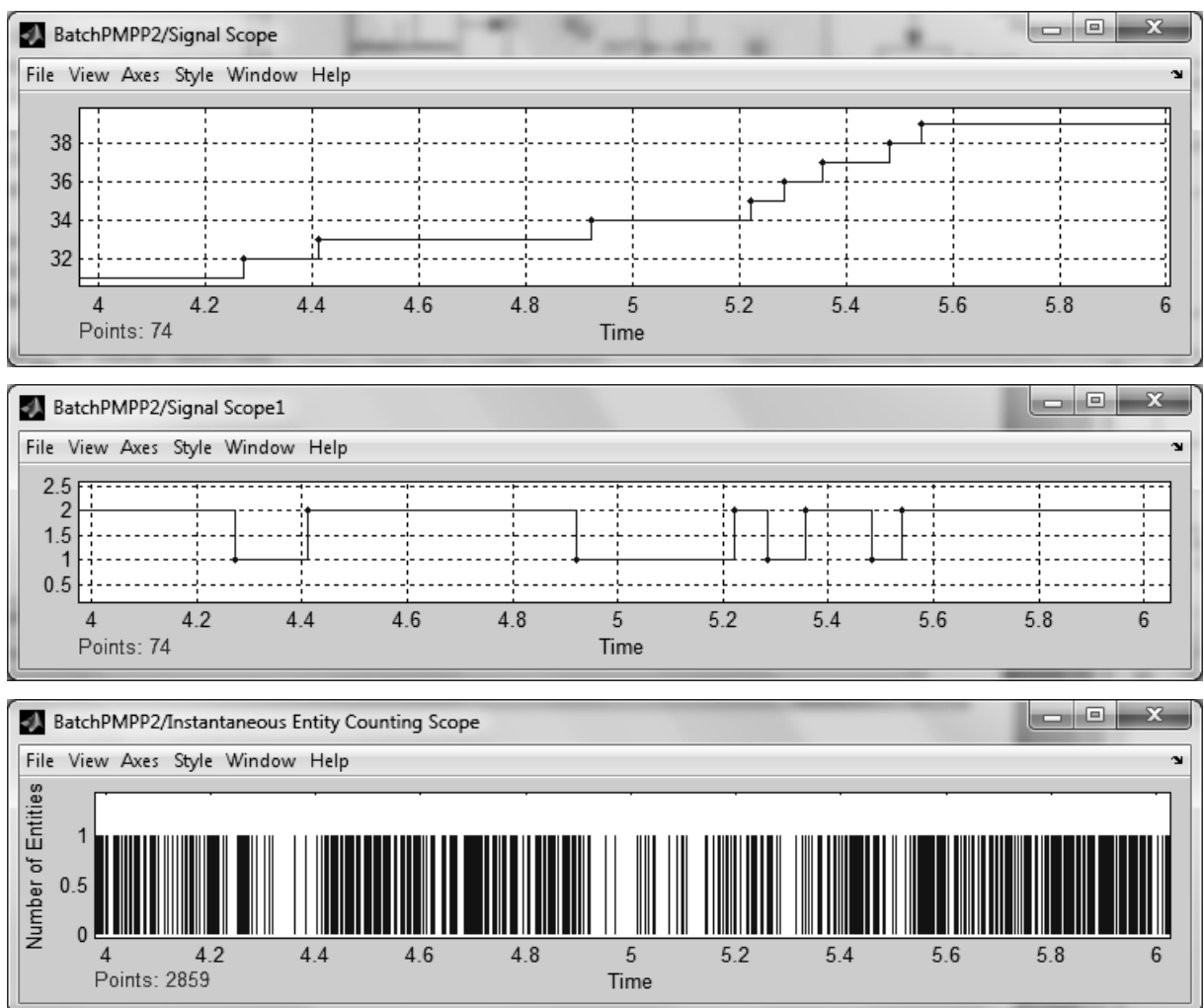


Fig. 5-27: Oscilogramas producidos por los bloques *Signal Scope*, *Signal Scope1*, e *Instantaneous Entity Counting Scope*.

Como se describió previamente el generador PMPP consiste en alternar dos generadores de tiempos interarribo de paquetes con distribución exponencial, de valores medios $1/\lambda_1$ y $1/\lambda_2$ respectivamente, cuyos tiempos de conexión alternados son fijados por una

secuencia aleatoria con distribución Pareto. Cada vez que el *Time-Based Entity Generator2* genera una entidad (ráfaga) se incrementa en una unidad su salida estadística #d.

El bloque *Fcn* (función) convierte cada incremento de #d en una señal de valor 1 o 2 en forma alternada, Fig. 5-26. La expresión de Matlab $\text{ceil}(u/2)-\text{floor}(u/2)+1$ cumple esta función. La función integrada en Matlab $\text{ceil}()$ devuelve el entero superior más próximo al valor de la expresión entre paréntesis, en cambio $\text{floor}()$ devuelve el entero inferior más próximo.

En la Fig. 5-27, se muestran los oscilogramas producidos por los bloques *Signal Scope*, *Signal Scope1*, y *Instantaneous Entity Counting Scope*. Se han seleccionado escalas convenientes y alineado verticalmente las imágenes para comparar la información.

En *Signal Scope* se muestra la señal #d incrementándose monótonamente de uno en uno cada vez que egresa una entidad. El *Signal Scope1* muestra el efecto del bloque *Fcn*, transformando a #d en una sucesión 1, 2, 1, 2,... necesaria para encauzar a la salida OUT del bloque *Input Switch*, en forma alternada, la secuencia de paquetes producida por cada uno de los dos generadores.

En el oscilograma del bloque *Instantaneous Entity Counting Scope* se aprecia, a pesar de que la separación entre paquetes es aleatoria, la conmutación de los dos generadores cuyos tiempos de intergeneración poseen valores medio $1/\lambda_1$ y $1/\lambda_2$ diferentes. Para el ejemplo la relación entre los valores medios es exactamente 4,2.

6. Simulación de un sistema SSQ FIFO

6.1 Introducción

El objetivo de la simulación es sintetizar tráfico cuyo comportamiento aplicado a una SSQ FIFO, con un tiempo de servicio y capacidad de la cola dados, tenga la misma probabilidad de pérdidas de paquetes por rebasamiento de la cola, que el tráfico al cual se pretende simular.

Es un hecho que el comportamiento de la cola depende primordialmente de la intensidad de tráfico $\rho = \frac{\text{tiempo de servicio}}{\text{tiempo interarribo promedio}} = \frac{\mu}{\lambda}$. Normalmente no es de interés para los diseñadores el comportamiento de la cola cuando $\rho \ll 1$ ya que en estas condiciones una capacidad de cola pequeña es suficiente para que las probabilidades de pérdida de paquetes sea un valor despreciable. Tampoco es de mucho interés un valor $\rho > 1$ ya que las pérdidas pasado un transitorio, son constantes e independientes de la capacidad de la cola. Normalmente para evaluar el comportamiento de las colas sometidas a un tráfico dado las comparaciones se establecen para una intensidad de tráfico $\rho = 0,9$

6.2 Simulación de una SSQ con tráfico generado por el modelo M/Pareto

En la sección 5.1.1 se estableció que si el modelo de tráfico M/Pareto se ejecuta durante un tiempo t lo suficientemente extenso, la cantidad de paquetes generados debería ser próxima a $n \approx t\lambda r\mu$. Donde λ es la media de la distribución exponencial que determina el inicio de cada ráfaga, r la cantidad de paquetes por segundo promedio de cada ráfaga y μ la media de la distribución Pareto que regula la duración de cada ráfaga.

También se estableció que el coeficiente de Hurst H está relacionado con la pendiente $-\alpha$ de la asíntota de la CCDF de la duración de cada ráfaga por $H = \frac{3-\alpha}{2}$ de donde $\alpha = 3 - 2H$.

El hecho de que el coeficiente de Hurst sea similar al valor de la muestra que se quiere simular se considerará como una condición deseable pero secundaria comparada con el comportamiento de la cola. Además como quedó expuesto en 3.5 y 3.6 el agregado de tráfico UGS no modifica el coeficiente de Hurst y destruye el *heavy-tailness* de la CCDF eliminando la asíntota, por lo cual la relación $H = \frac{3-\alpha}{2}$ queda inválida.

En particular, como ya se observará, este modelo para simulación no es muy afortunado en honrar la relación anterior. De las observaciones realizadas al azar en el dominio del tiempo del tráfico de paquetes generado, salvo al inicio siempre hay una o más ráfagas presentes. Esta única ráfaga siempre presente se puede interpretar como un UGS de r

paquetes por segundo, por lo tanto cuando se desee corroborar el H del tráfico sintetizado, no se obtendrá una relación exacta como la que dicta la formula referida previamente.

Un análisis profundo de esta discrepancia escapa al alcance de este trabajo, además se ve desmotivado por el mejor comportamiento, en general, del otro generador propuesto.

6.2.1 Procedimiento

El procedimiento a seguir es el siguiente

- 1 Se establece valores iniciales de λ , r , μ y α obtenidos de los análisis de la muestra que se pretende simular
- 2 Se genera el tráfico sintético con el modelo de Simulink M/Pareto
- 3 Se somete el tráfico a un sistema SSQ para $\rho=0,9$
- 4 Se compara con comportamiento del tráfico que se pretende simular aplicado al mismo sistema y en las mismas condiciones
- 5 Se evalúan las diferencias y se determina en forma heurística nuevo juego de valores λ , r , μ y α
- 6 Se repite de 2 a 5 hasta que las diferencias sean aceptadas

Las condiciones iniciales se toman de los análisis realizados al tráfico de la muestra LBL-PKT4. Ver el resumen en la Tabla 4-3, la ecuación $n \approx t\lambda r \mu$ y las fórmulas de los valores medios de la distribución exponencial y Pareto.

$$\text{Factor de forma de Pareto: } \alpha = 3 - 2H = 3 - 2 * 0,81 = 1,38$$

Media de la distribución exponencial: $\lambda = \frac{13183}{3600} = 3,66 = \frac{\text{ráfagas}}{s}$ donde 13183 es la cantidad total de *socks* de la muestra

Paquetes por segundo por ráfaga: $r = \frac{1}{0,08008} = 12,5 \frac{\text{paq}}{s}$ donde 0,08008 es la mediana de los tiempos interarribo entre paquetes de un mismo *sock*

$$\text{Media de la distribución Pareto: } \mu = \frac{n}{t\lambda r} = \frac{896690}{3600 * 3,66 * 12,5} = 5,45s$$

Inicialmente se corrieron los modelos de Simulink M/Pareto3.mdl para crear el tráfico y cola1.mdl para verificar el comportamiento de la cola manualmente. Posteriormente se creó un *script* para automatizar el procedimiento automata2.m, la función *paretogen3.m* para generar el vector Pareto, y *figuras.m* para dibujar la pérdida de paquetes versus la capacidad de la cola. Estos programas están transcritos en el Anexo 3 – *Scripts* para simulación con el modelo M/Pareto.

6.2.2 Resultados

En la Fig. 6-1 se grafica la primeras simulaciones buscando aproximar el comportamiento de

la cola del tráfico sintético al de muestra LBL-PKT4. En ordenada se indica la relación paquetes perdidos a paquetes arribados a la cola, y en la abscisa capacidad de la cola en paquetes. Se fueron ensayando distintas combinaciones de los parámetros del tráfico M/Pareto, resultando la de $\alpha = 1,38$ $\mu = 5,45$ $\lambda = 2$ muy buena. También se grafica el comportamiento de LBL-PKT4 para compararlo con el tráfico simulado. En todos los casos la intensidad de tráfico es $\rho = 0,9$.

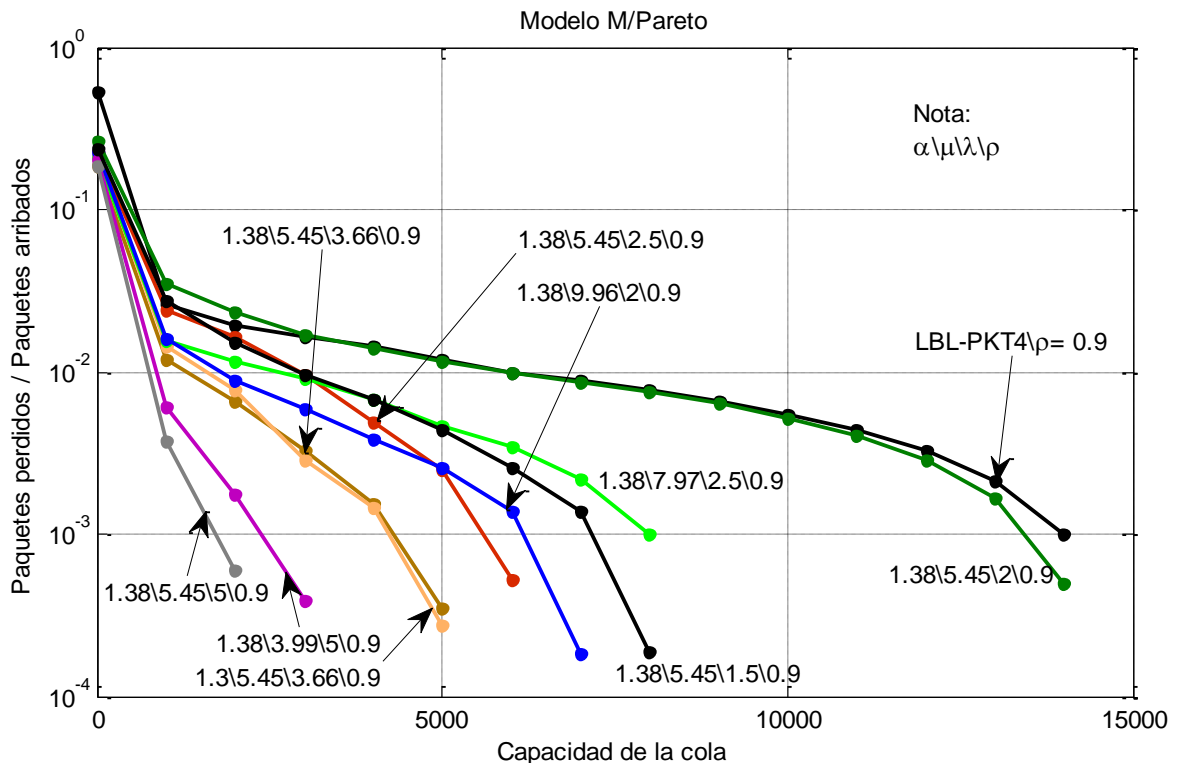


Fig. 6-1: Curvas de Paquetes perdidos/Paquetes arribados versus la Capacidad de la cola.

En la siguiente corrida, Fig. 6-2, se buscó observar el efecto de variar α y por lo tanto H. Se usaron los siguientes valores de parámetros $\alpha = [1.3 \ 1.5]$, $\mu = [4 \ 4.5 \ 5]$, y $\lambda = [1.8 \ 2 \ 2.2]$.

En la Fig. 6-3 se parametrizó el *script* con los siguientes valores: $\alpha = [1,3 \ 1,4]$, $\mu = [4,3 \ 4,5 \ 4,7]$, $\lambda = [2 \ 2,1 \ 2,2]$ y $\rho = [0.9]$ buscando mejores aproximaciones.

En la Fig. 6-4 se muestran dos efectos. El primero es la dispersión importante de las curvas al correr la simulación varias veces con el mismo juego de parámetros.

El segundo es el comportamiento del tráfico simulado cuando se modifica la intensidad de tráfico ρ . Para intensidades pequeñas el comportamiento es más aproximado al del tráfico LBL-PKT4 que para la unitaria.

Los autores de este modelo (18) sugieren reajustar los parámetros si se modifica ρ sensiblemente.

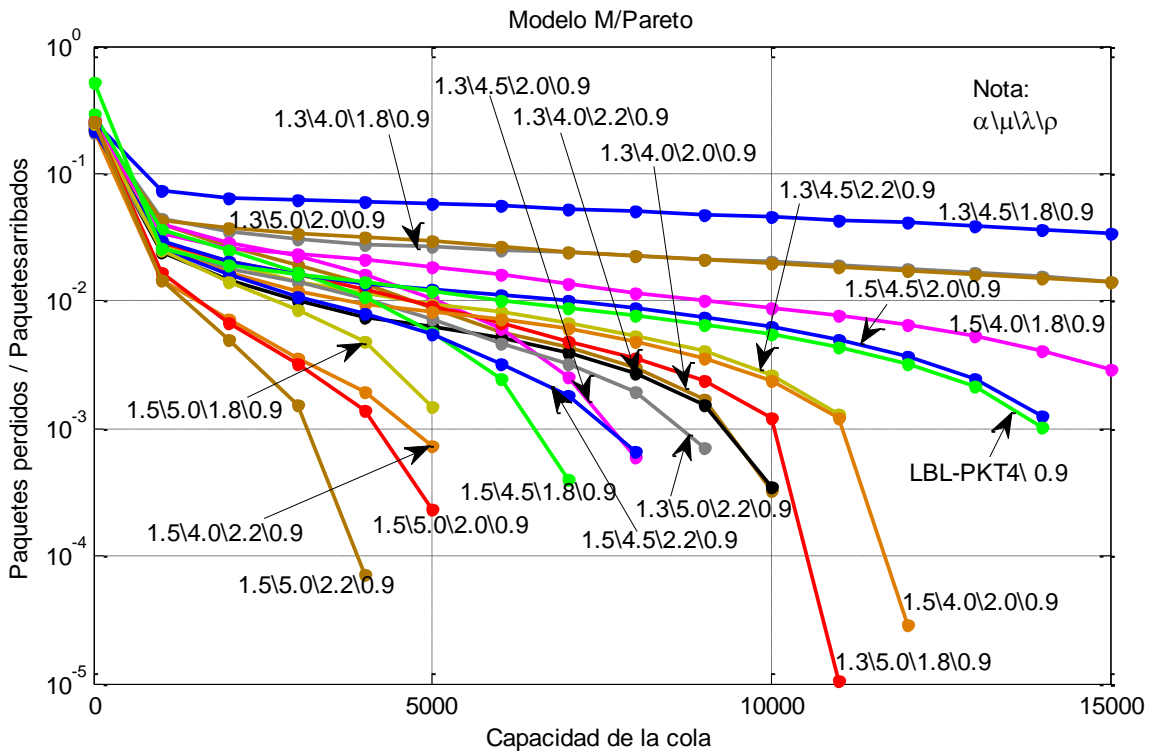


Fig. 6-2: Curvas de Paquetes perdidos/Paquetes arribados versus la Capacidad de la cola.

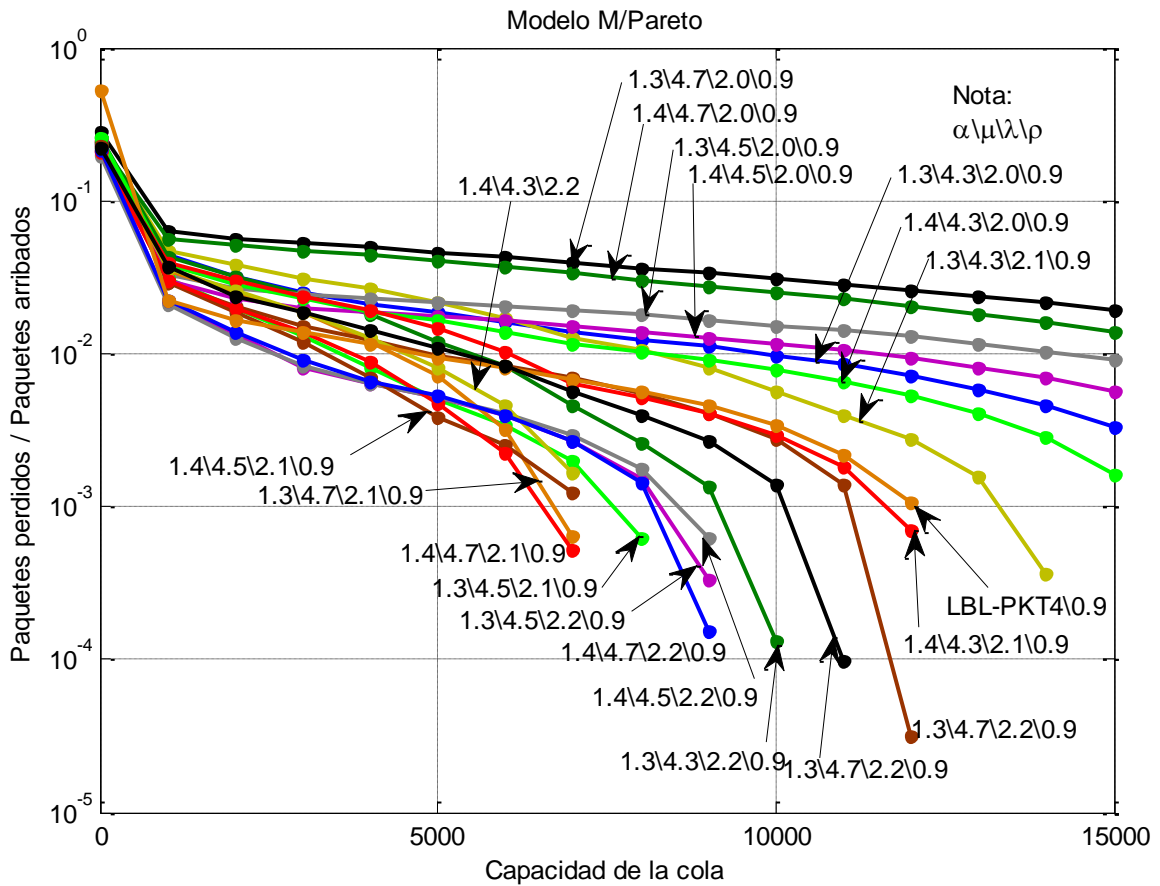


Fig. 6-3: Curvas de Paquetes perdidos/Paquetes arribados versus la Capacidad de la cola en la búsqueda de mejores aproximaciones a la muestra LBL-PKT4

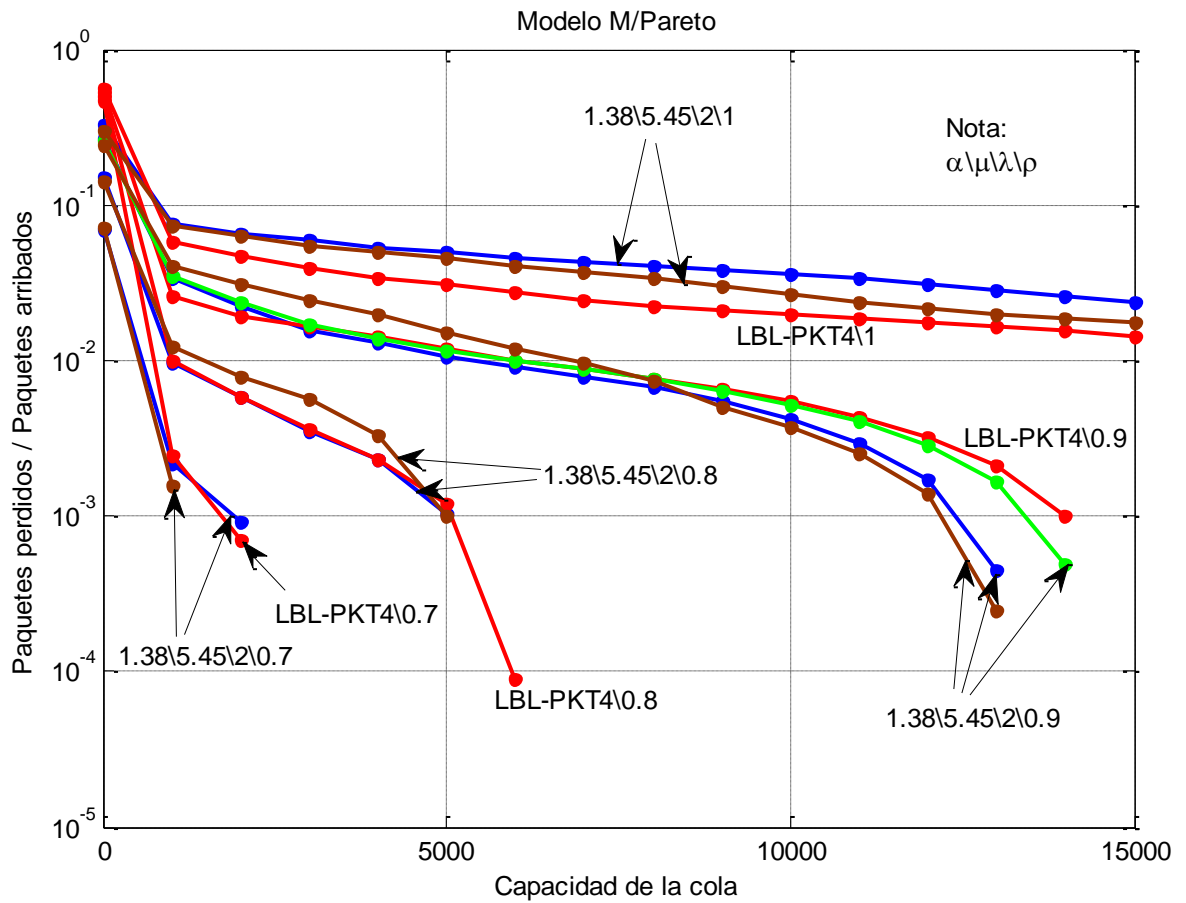


Fig. 6-4: Curvas de Paquetes perdidos/Paquetes arribados versus la Capacidad de la cola. Influencia de la intensidad de tráfico ρ y de la dispersión producida al correr varias veces la simulación con los mismos juegos de parámetros.

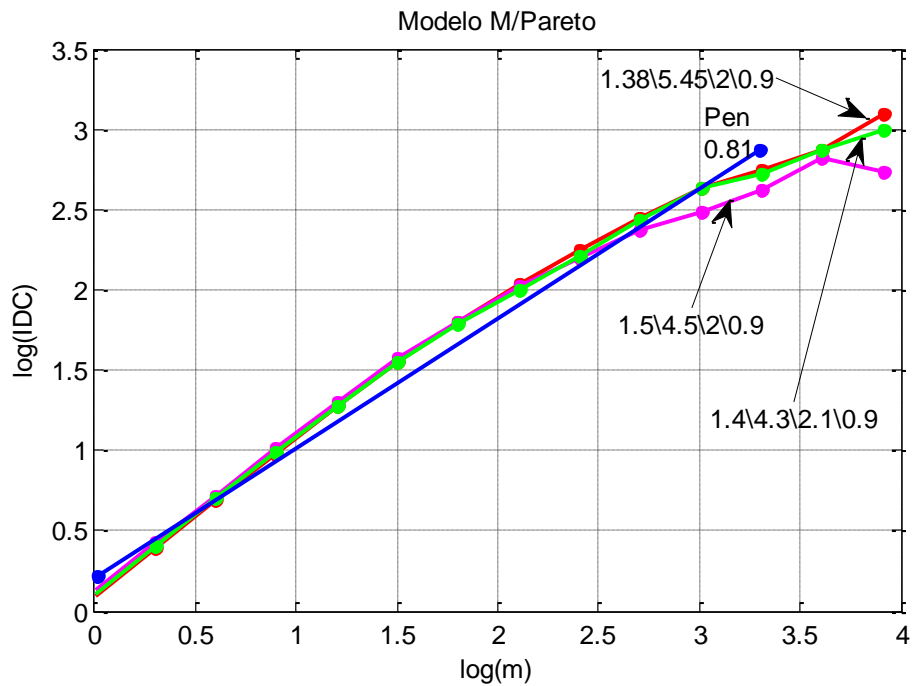


Fig. 6-5: Curva IDC de las simulaciones más aproximadas a la muestra LBL-PKT4. $H=(Pen+1)/2=0,905$

6.2.3 Análisis del coeficiente de Hurst H por el método IDC

Aunque el método IDC (*Index of Dispersion for Counts*) (11) (13) no es el más rápido, muchos autores lo emplean por la curva suave que produce, lo que permite acomodar fácilmente una asíntota para determinar fácilmente su pendiente, la que se relaciona con el coeficiente de Hurst por la siguiente fórmula $H = \frac{(Pen+1)}{2}$. La transcripción del código en Matlab se muestra en el Anexo 2 – Programas utilitarios en Matlab.

Las tres curvas IDC de la Fig. 6-5 tienen casi la misma pendiente lo cual es anómalo. Si se calcula H con la pendiente de las curvas IDC resulta un valor de $H=0,95$ muy superior al teórico dado por la fórmula $H = \frac{3-\alpha}{2}$. Calculando H con esta última ecuación resulta $H=0,81$ ($\alpha=1,38$), $H=0,8$ ($\alpha=1,4$) y $H=0,75$ ($\alpha=1,5$). Se estima que este efecto se debe a lo expuesto previamente en la sección 6.1.

6.2.4 Observaciones

Observando las figuras se puede afirmar lo siguiente relativo al modelo M/Pareto para simulación:

- Todas las curvas mostradas son producidas por aproximadamente la misma cantidad de paquetes en un hora. Llama la atención la alta sensibilidad del comportamiento de la cola a pequeños cambios de los parámetros.
- En la Fig. 6-1 la mejor aproximación a LBL-PKT4\0,9 es 1,38\5,45\2\0,9 en la Fig. 6-2 es 1,5\4,5\2\0,9 y en la Fig. 6-3 es 1,4\4,3\2,1\0,9. Esto indica que tráficos con parámetros diferentes pueden producir un comportamiento similar de la cola.
- Especialmente en la Fig. 6-1 y Fig. 6-2 queda claro que un α menor que se corresponde con un H mayor, empeora el comportamiento de la cola aumentando las pérdidas de paquetes por rebasamiento.
- La Fig. 6-4 permite inferir que si se desea un modelo que reproduzca con exactitud un tráfico dado, hay que ajustarlo para el valor de la intensidad de tráfico ρ del tráfico que se desea simular.
- En la Fig. 6-4 también se observa que la dispersión de las curvas al correr el modelo varias veces indica que el juego de parámetros que reproduzca más exactamente el tráfico que se desea simular se debería obtener repitiendo n veces la simulación con el mismo juego y graficando el promedio de los n valores de cada capacidad. Observe que la curva verde de parámetros 1,38\5,45\2\0,9 que ajusta muy bien a la LBL-PKT4, no parece tan buena en sus versiones azul y marrón.
- El modelo no ajusta muy bien para capacidades bajas de la cola, especialmente para capacidad de la cola en 0 paquetes. Este desajuste es propio de la naturaleza del modelo debido a que fija una granulometría mínima de tiempo interarribo de paquetes, que se fijó para las presentes simulaciones en 0,1ms, mientras que el tráfico real puede ser aún aleatorio en esta pequeña escala de tiempo. Este desajuste pierde importancia rápidamente al aumentar la capacidad de la cola.

— El coeficiente de Hurst no representa adecuadamente el tráfico sintetizado

6.3 Simulación de una SSQ con con tráfico generado por el modelo PMPP

De la misma manera que en la simulación con el modelo M/Pareto en 6.2, la simulación con el modelo PMPP consiste en generar tráfico PMPP con características similares al de la muestra LBL-PKT4 y verificar su comportamiento en un sistema SSQ FIFO comparándolo con el comportamiento del tráfico de la mencionada muestra. Se hará énfasis en un análisis para una intensidad de tráfico $\rho=0,9$, pero también se verificará el tráfico sintético para distintos valores de ρ comparándolo con el de la muestra.

De la misma forma que con el modelo M/Pareto, se verificará las características autosimilares del tráfico generado, con el método IDC.

6.3.1 Procedimiento

El procedimiento a seguir es el siguiente

- 1 Se establece valores iniciales de $\frac{\lambda_1+\lambda_2}{2}$, μ y α obtenidos de los análisis de la muestra que se pretende simular. Se establece una relación arbitraria entre λ_2/λ_1
- 2 Se genera el tráfico sintético con el modelo de Simulink PMPP
- 3 Se somete el tráfico a un sistema SSQ para $\rho=0,9$
- 4 Se compara con comportamiento del tráfico que se pretende simular aplicado al mismo sistema y en las mismas condiciones
- 5 Se evalúan las diferencias y se determina en forma heurística nuevo juego de valores α , μ y λ_2/λ_1
- 6 Se repite de 2 a 5 hasta que las diferencias sean tolerables

Las condiciones iniciales se toman de los análisis realizados al tráfico de la muestra LBL-PKT4. Ver el resumen en la Tabla 4-3, la ecuación $n = t \frac{\lambda_1+\lambda_2}{2}$ y las fórmulas de los valores medios de la distribución exponencial y Pareto.

Factor de forma de Pareto: $\alpha = 3 - 2H = 3 - 2 * 0,81 = 1,38$

Media de distribución Pareto: se tomará como número de ráfagas a la cantidad de *socks* 13183. Suponiendo que cada inicio de ráfaga o terminación de ráfaga provoca un cambio en la velocidad $\lambda_1 \rightarrow \lambda_2$ o $\lambda_2 \rightarrow \lambda_1$, entonces $\mu = \frac{3600}{2*13183} = 0,136s$

Velocidad media de los paquetes: $\frac{\lambda_1+\lambda_2}{2} = \frac{n}{t} = \frac{896690}{3600} = 249 \frac{\text{paq}}{s}$

Para automatizar los puntos 2, 3 y 4 del procedimiento se diseñó un *script* el cual está transcrito en el Anexo 4 – *Scripts* para simulación con el modelo PMPP.

6.3.2 Resultados

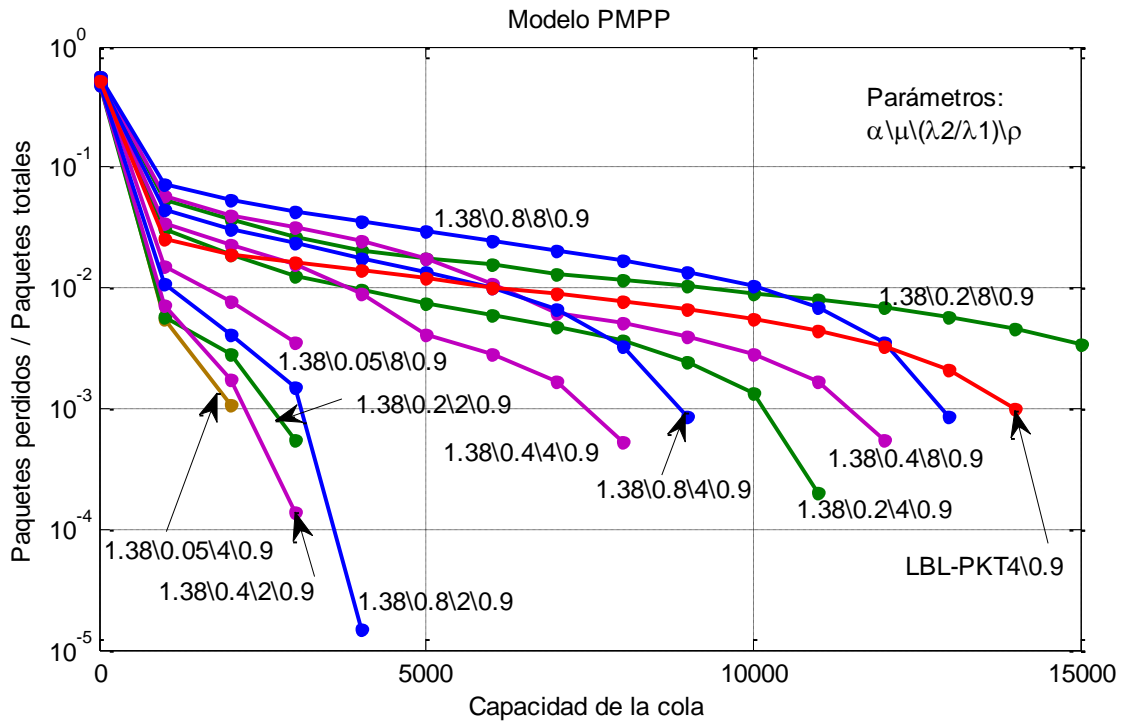


Fig. 6-6: Curvas de Paquetes perdidos/Paquetes arribados versus la Capacidad de la cola.

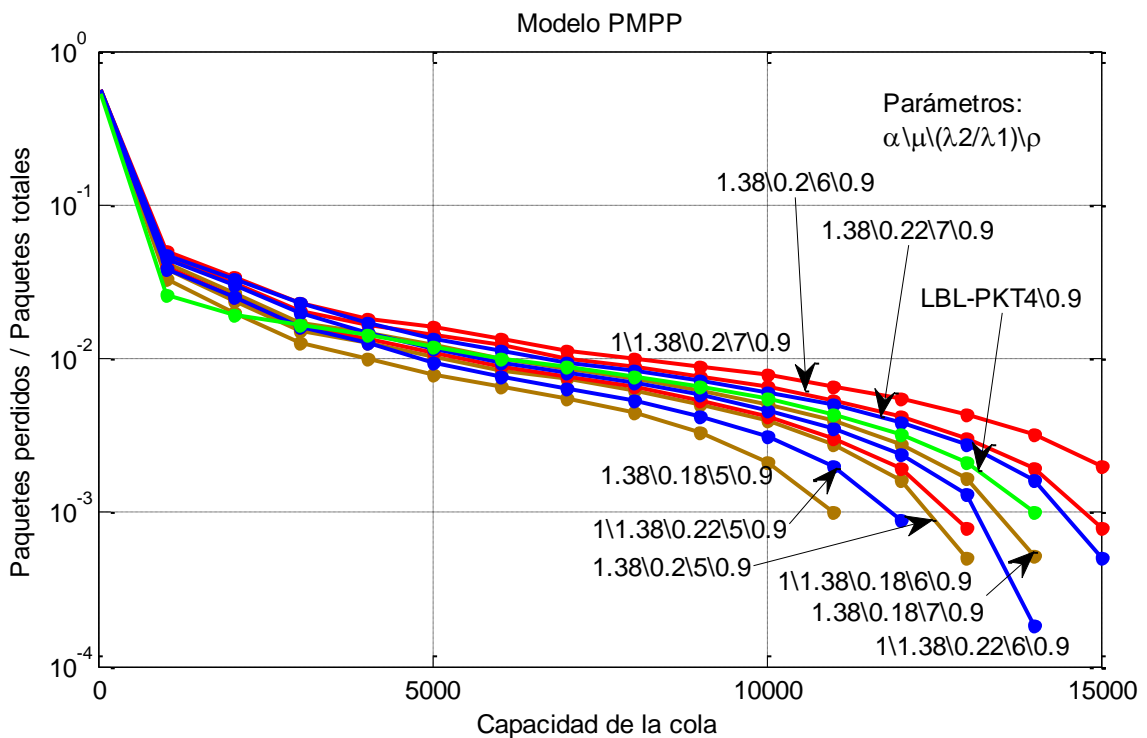


Fig. 6-7: Curvas de Paquetes perdidos/Paquetes arribados versus la Capacidad de la cola.

En la Fig. 6-6 se grafica el comportamiento de la cola aplicando el tráfico sintético elaborado con el modelo PMPP buscando simular el tráfico de la muestra LBL-PKT4. En ordenada se

indica la relación paquetes perdidos a paquetes arribados, y en la abscisa capacidad de la cola en paquetes. Se ensayaron diferentes combinaciones de los parámetros de μ y λ_2/λ_1 .

Se parametrizó el *script* con los siguientes valores $\alpha = [1,38]$, $\mu = [0,05 \ 0,1 \ 0,2 \ 0,4 \ 0,8]$, y $\lambda_2/\lambda_1 = [2 \ 4 \ 8]$ buscando la mejor aproximación. La parametrización que más se aproxima al comportamiento de la LBL-PKT es la $\alpha = 1,38$, $\mu = 0,2$, $\lambda_2/\lambda_1 = 8$.

La Fig. 6-7 muestra el resultado de una segunda simulación afinando los parámetros para una mejor aproximación el comportamiento de la cola. La curva más aproximada es la $\alpha = 1,38$, $\mu = 0,18$, $\lambda_2/\lambda_1 = 7$.

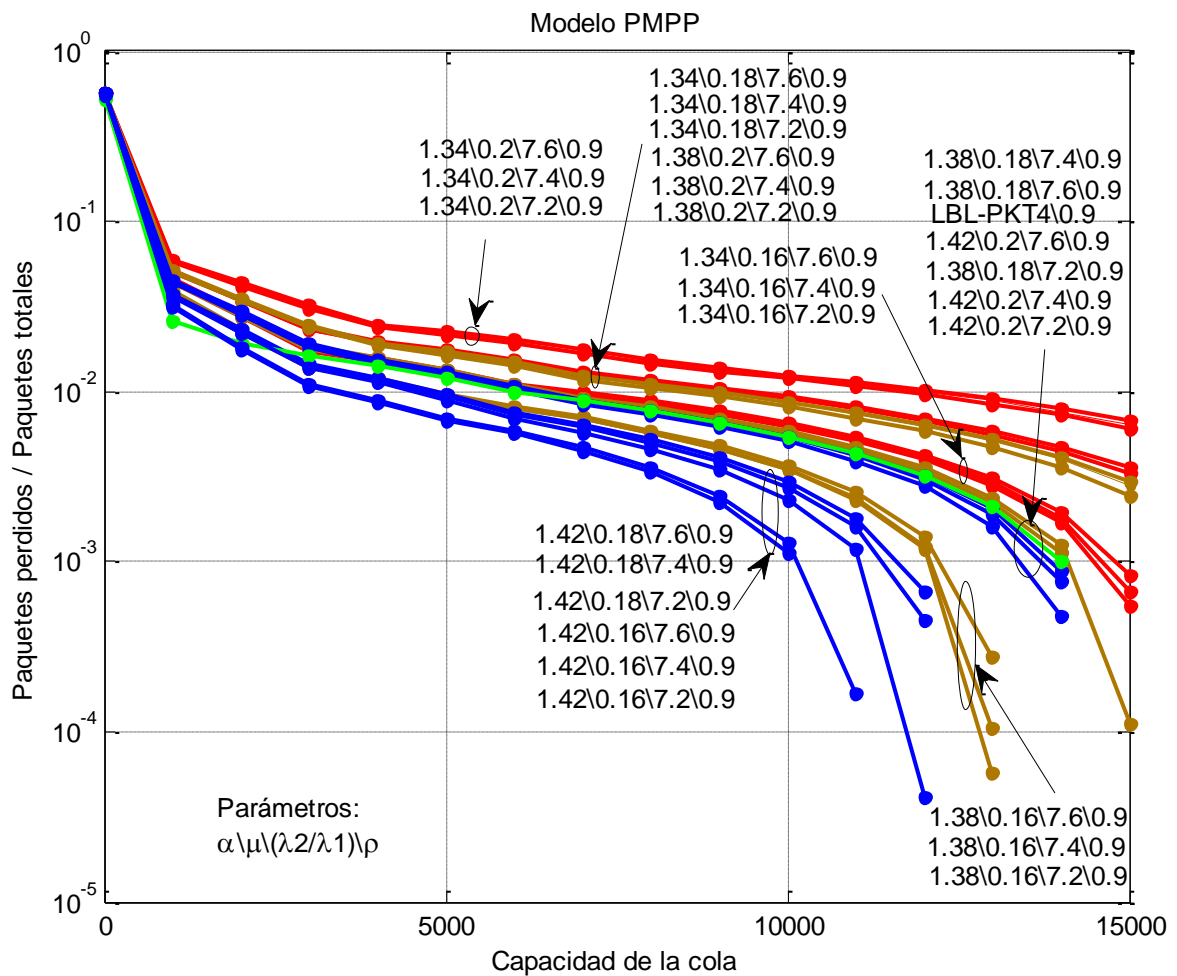


Fig. 6-8: Curvas de Paquetes perdidos/Paquetes arribados versus la Capacidad de la cola.

En la Fig. 6-8 se muestran resultados de una tercera simulación ajustando aún más los parámetros. Las curvas que más se aproximan corresponden a los juegos de parámetros $\alpha = 1,38$, $\mu = 0,18$, $\lambda_2/\lambda_1 = 7,6$ y $\alpha = 1,42$, $\mu = 0,2$, $\lambda_2/\lambda_1 = 7,6$. También se ensayaron pequeñas variaciones de α , correspondiendo siempre el más pequeño (coeficiente de Hurst mayor) el tráfico que más pérdidas produce en la cola.

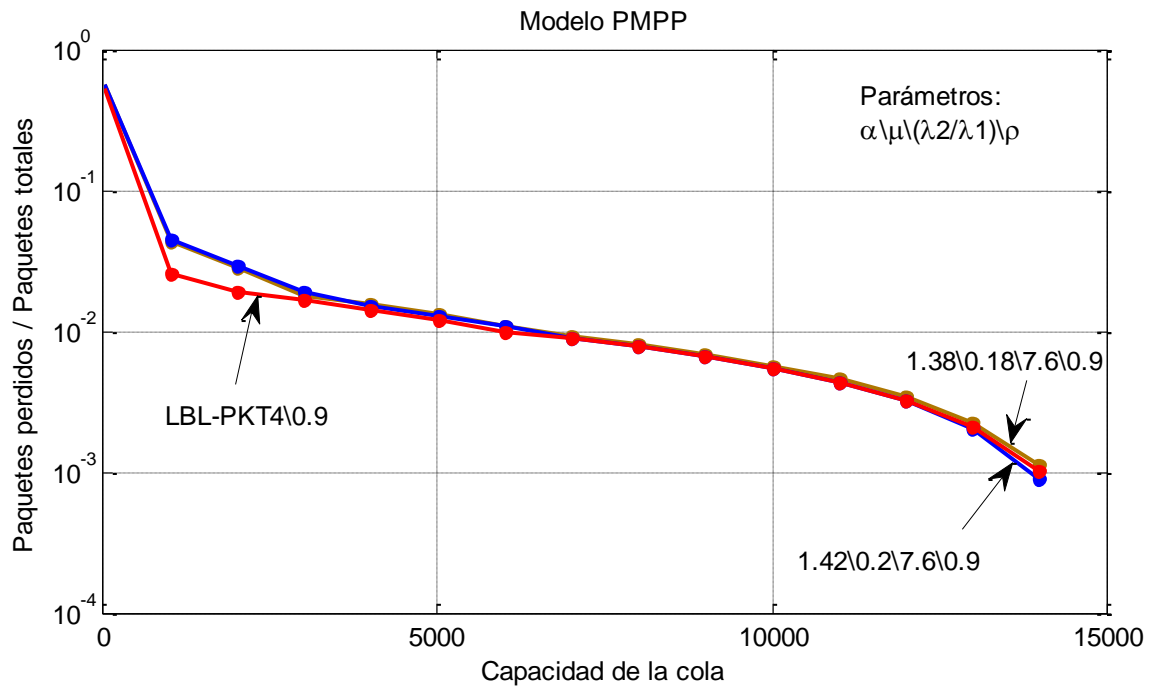


Fig. 6-9: Curvas de Paquetes perdidos/Paquetes arribados versus la Capacidad de la cola. Se muestran las dos mejores aproximaciones por el método PMPP a la muestra LBL-PKT4 para $\rho=0,9$.

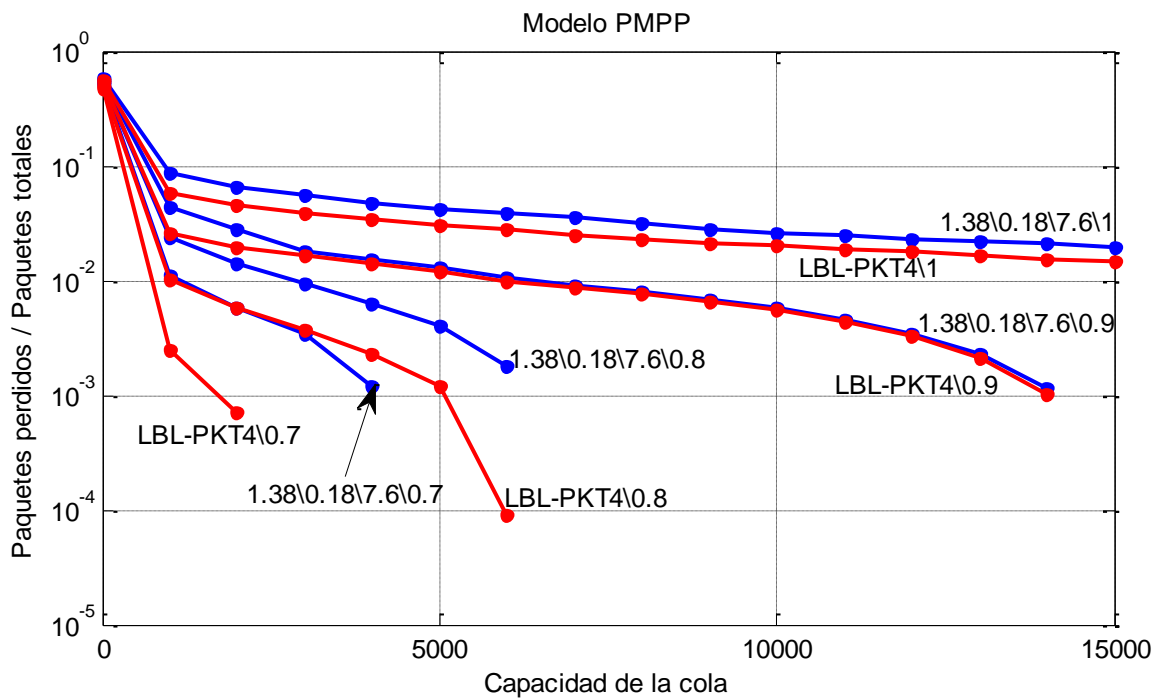


Fig. 6-10: Comportamiento del tráfico sintético PMPP comparado con el de la muestra LBL-PKT4 cuando en el sistema SSQ FIFO se varía el tiempo de servicio μ y por ende la intensidad de tráfico $\rho=0,7$, $0,8$, $0,9$ y 1 .

En la Fig. 6-9 se muestra con detalle la curva LBL-PKT4\0.9 y las mencionadas que se ubicaron más próximas. La curva con parámetros $\alpha=1,38$, $\mu=0,18$, $\lambda_2/\lambda_1=7,6$ se entiende

que es la más aproximada. Se registra una diferencia apreciable únicamente para capacidades de cola de 1000 y 2000 paquetes. Esta diferencia podría ser atendible si la capacidad de la cola de diseño del sistema está alrededor de estos valores.

En la Fig. 6-10 se muestra el comportamiento del tráfico sintético comparado con el de la muestra LBL-PKT4 para intensidades de tráfico $\rho=0,7$ $0,8$ $0,9$ y 1 . Para intensidades mayores que $0,9$ la aproximación se mantiene no así para valores menores. Por esta razón se sugiere ajustar nuevamente los parámetros para cada cambio de intensidad de tráfico ρ . Conservando los mismos parámetros cuando varía ρ daría lugar a diseños conservadores ya que para valores cercanos a la unidad de la intensidad de tráfico ρ , es mayor la pérdida con el tráfico sintético que con el real.

En la Fig. 6-11 se muestra la determinación gráfica del coeficiente de Hurst, mediante el método IDC. La concordancia con el de la muestra LBL-PKT4 es muy buena.

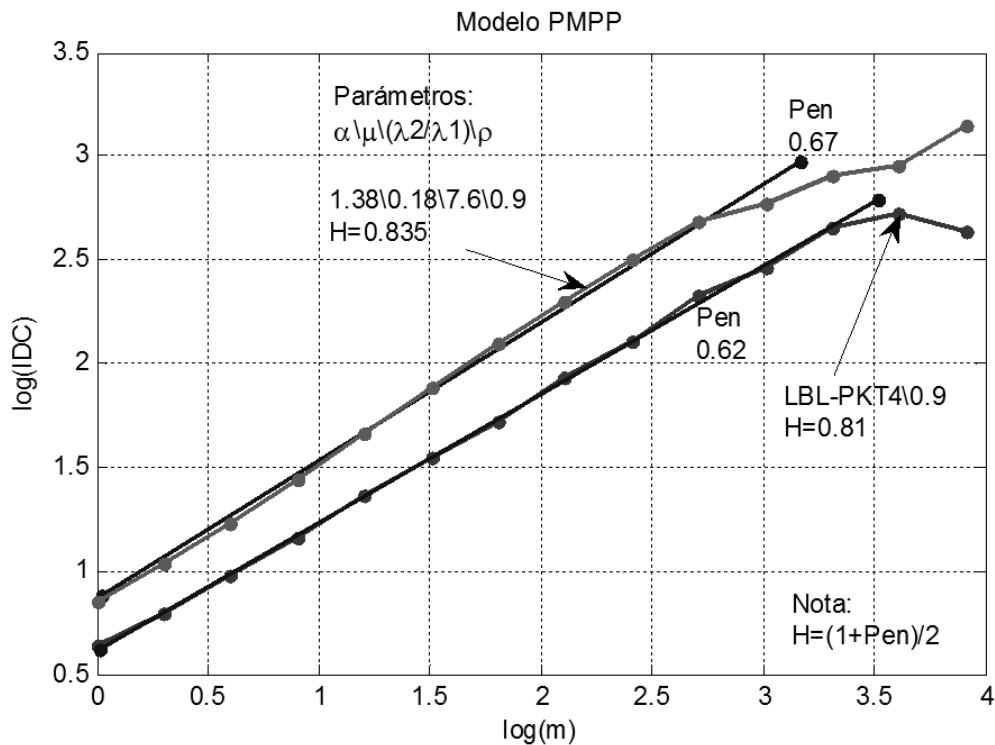


Fig. 6-11: Curva IDC de las simulación más aproximada a la muestra LBL-PKT4 realizada con el modelo PMPP

6.3.3 Observaciones

Observando las figuras se puede afirmar lo siguiente relativo al modelo PMPP para simulación:

- Las Fig. 6-6, Fig. 6-7, Fig. 6-8 y Fig. 6-9 muestran que los resultados de simulación encajan muy bien no solo para capacidades grandes de la cola, también reproduce

muy bien el comportamiento de la LBL-PKT4 desde capacidad cero de la cola, situación que no sucedía con el modelo M/Pareto.

- Las simulaciones variando la intensidad de tráfico presentan buen comportamiento para valores de $\rho > 0,9$ tal como se aprecia en la Fig. 6-10.
- La Fig. 6-11 muestra que el valor del coeficiente de Hurst determinado por el método IDC, es apenas superior al de la muestra LBL-PKT4. Además la rectitud de la curva manifiesta un comportamiento autosimilar puro. No es así en las curvas IDC del tráfico generado por el modelo M/Pareto, Fig. 6-5, no solo produce un valor del coeficiente de Hurst bastante mayor que el de la muestra LBL-PKT4 y la anomalía observada de que no se modifica H al variar α , sino que además el andar de las mismas no es muy recto.
- Se verificó que este modelo es bastante sensible a las diferencias en la repetición de simulaciones con el mismo juego de parámetros si no se reinician los generadores de números aleatorios con cada simulación. Se realizaron simulaciones repetidas con el mismo juego de parámetros pero con diferentes semillas para el caso de la distribución exponencial que se realiza en Simulink y sin reinicializar el generador de números aleatorios de Matlab. Los resultados fueron desalentadoramente diferentes con cada corrida, aunque esto en principio no debiera sorprender en demasía por la naturaleza aleatoria de los fenómenos.
- Se comprobó que el modelo M/Pareto, ver Fig. 6-4, repite más parecido con iteraciones no reinicializadas. Sin entrar a una discusión matemática, se arriesga a brindar la siguiente explicación. En M/Pareto un valor exageradamente grande de la variable Pareto solo superpone una ráfaga durante mucho tiempo a las que están presentes con permanencias Pareto muchos menores, por lo tanto su efecto es promediado con las otras ráfagas presentes. En cambio en el modelo PMPP el efecto de un tiempo de ráfaga Pareto con un valor disparatado (como es lógico en esta distribución con variancia infinita) no tiene atenuación alguna en su efecto, ya que no hay superposición con otras, por eso es dable esperar mayor dispersión de corrida a corrida en el modelo PMPP que en el modelo M/Pareto.
- Este tema libera inocentemente el pestillo de una caja de Pandora. ¿Cuál de los dos modelos reproduce mejor la realidad? Pero la realidad solo se puede evaluar tomando muestras de la realidad, que presenten aproximadamente la misma cantidad de paquetes en el mismo intervalo de tiempo y aplicando dicho tráfico al sistema SSQ FIFO de ensayo.
- Este tema, y la observación ya realizada oportunamente de que la dispersión de las curvas al correr los modelos varias veces indicaría que el juego de parámetros que reproduciría más exactamente el tráfico que se desea simular se obtendría repitiendo n veces la simulación con el mismo juego y graficando los promedios de los n valores de cada capacidad, quedan para ser analizados en futuros trabajos, en los que se disponga de más y mejores muestras de tráfico y un equipamiento informático adecuado que permita reducir drásticamente los tiempos de simulación. Por lo tanto se excluye este tema de la exposición, reinicializando los generadores de números aleatorios de Matlab y Simulink con cada simulación.

7. Modelo en Simulink para generar tráfico para ser aplicado al diseño de *schedulers* de WiMAX

7.1 Planificación de la prestación de servicios en WiMAX

7.1.1 Introducción

En el Anexo 6 – Reseña de WiMAX, se desarrolla un breve resumen en el que se describe los elementos básicos del sistema WiMAX. Es recomendable leer el mencionado anexo previo a la lectura de este capítulo.

La principal fuente de información para el desarrollo de la presente sección proviene de un artículo de la revista *Intel Technology Journal* (23), siendo uno de sus autores también redactor de la IEEE 802.16-2009, lo que acredita valor al mismo.

Se busca definir la información necesaria que debe ser incluida en los paquetes del tráfico a generar, para que el MAC *scheduler* pueda realizar su tarea.

7.1.2 Clases de Servicios

La MAC provee diferenciación de QoS para los diferentes tipos de aplicaciones que pueden operar sobre redes 802.16. El estándar define los siguientes cinco tipos de servicios. Para una descripción de los mismos ver 15.4.2.1 a 15.4.2.5.

- UGS (*Unsolicited Grant Service*).
- rtPS (*real-time Polling Service*)
- ertPS (*extended real-time Polling Service*) (solo para *Mobile WiMAX*)
- nrtPS (*non real-time Polling Service*)
- BE (*Best Effort*)

Cada conexión entre la SS y la BS se le asigna una clase de servicio como parte de la creación de la conexión. Cuando los paquetes son clasificados en la subcapa de convergencia, la conexión a la cual son asignados se elige en función del tipo de garantías de QoS que requiere la aplicación.

En la Fig. 7-1 muestra el mecanismo de clasificador de WiMAX para la asignación de QoS a servicios multimedia, como ejemplos incluye voz TDM, VoIP, vídeo *streaming*, TFTP, HTTP y correo electrónico. En la Fig. 7-1, se omite el ertPS ya que el artículo fuente de donde fue tomada, data de Agosto de 2004 cuando todavía la norma no incluía *Mobile WiMAX* y por ende este servicio, pero esta circunstancia no menoscaba la validez de la referencia.

Hay dos tipos de mecanismos de *polling*:

- *Unicast*: Cuando se realiza un *polling* individual a una SS, se le asigna ancho de banda para que envíe sus mensajes de requerimiento de ancho de banda.
- Por contención: se utiliza cuando el ancho de banda disponible es insuficiente para un *polling* individual a muchas SS inactivas SS. La asignación es *multicast* o *broadcast* a un grupo de SSs las que tienen que competir por la oportunidad de enviar peticiones de ancho de banda.

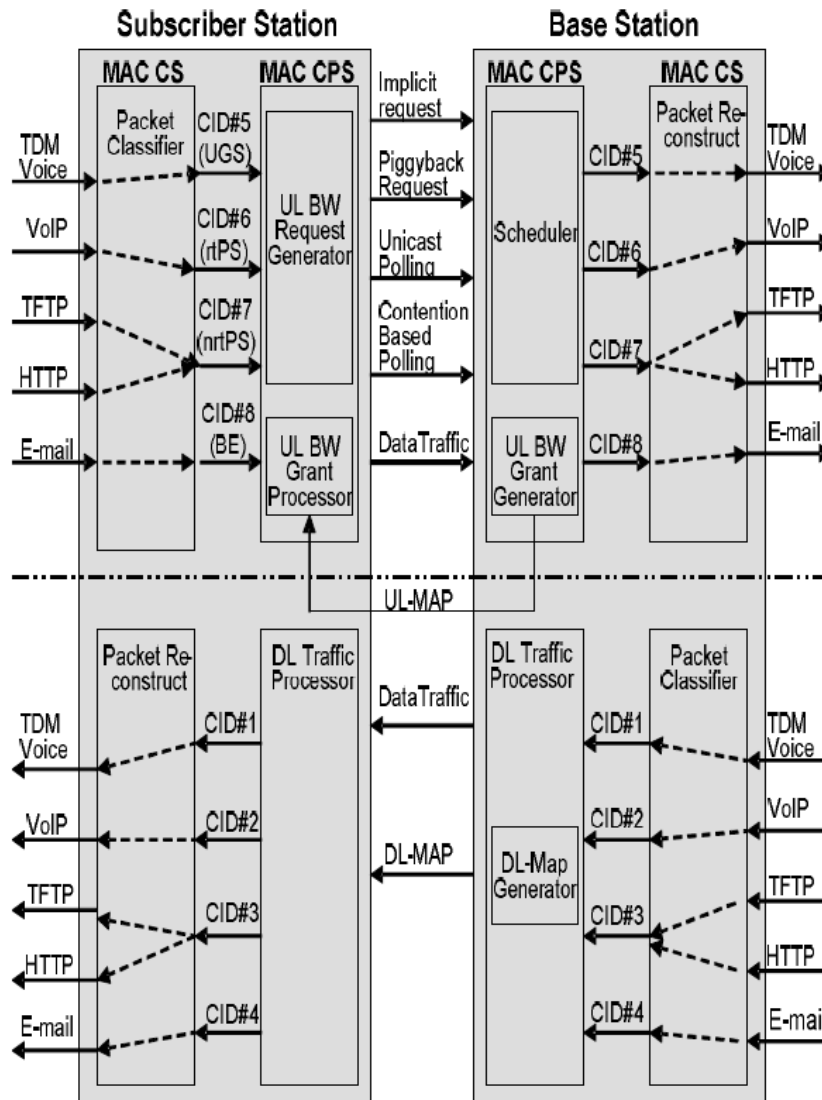


Fig. 7-1: Mecanismo de QoS para servicios multimedia. Tomada de (23).

7.1.3 Scheduling y adaptación del enlace

El objetivo del *scheduling* y la adaptación del enlace es proveer el deseado tratamiento de QoS para el tráfico que atraviesa el enlace aéreo, optimizando la utilización de los recursos inalámbricos.

El *scheduling* en la MAC 802.16 se divide en dos tareas relacionadas: la planificación de la utilización enlace inalámbrico entre las SSs y el *scheduling* individual de los paquetes entre la BS y las SSs.

El *scheduler* corre en la BB y se considera parte de la capa MAC de la estación base. El mismo define el contenido los *subframes downlink* y *uplink* de cada *frame*. Cuando se usan modos opcionales tales como transmitir con diversidad, por ejemplo AAS (*Adaptive Antena System*) y MIMO (*Multiple Input Multiple Output*), la MAC debe dividir el *uplink* y el *downlink* en diferentes zonas. Zona de transmisión con diversidad normal, zona AAS, y zona MIMO, para acomodar las SSs a ser servidas con cada uno de estos modos.

Habiendo dividido los *subframes* en zonas, el *scheduler* asigna oportunidades de transmisión individuales a cada SS, dentro de la zona en que ellas operan. En OFDM, las oportunidades de transmisión para cada estación son ranuras de tiempo, mientras que en la OFDM UL y OFDMA UL y DL, las oportunidades de transmisión son ranuras de tiempo dentro de subcanales individuales. Ver Fig. 15-4.

Cuando la BS emplea AAS con SDMA (*Spatial Diversity Multiple Access*), una ranura de tiempo determinada en un subcanal dado puede ser asignada a varias de SSs. Esto significa que el problema bidimensional del *scheduling* (con ranuras de tiempo a lo largo de uno de los ejes y subcanales a lo largo de la otra) se convierte en un problema de tres dimensiones, donde el tercer eje es el eje espacial.

La MAC debe determinar cuál de las SSs posee características ortogonales respecto al resto, que la hacen buena candidata para compartir las mismas combinaciones de ranuras definidas en el plano subcanal/tiempo.

El *scheduler* debe determinar los perfiles apropiados de los *bursts* (ráfagas) para la comunicación inalámbrica con cada SS. La BS monitorea la relación señal/ruido (SNR) e incrementa o reduce el nivel de la tasa de codificación y modulación, manteniendo la relación en niveles adecuados, para el tráfico con cada una de las SSs. Con esto busca obtener el más alto *throughput* posible, manteniendo al mismo tiempo un determinado nivel de BER.

El *scheduler* del enlace inalámbrico determina los requerimientos individuales de ancho de banda de cada SS, teniendo en cuenta la clase de servicio de cada conexión y el estado de las colas en la BS y la SS.

La BS monitorea sus propias colas para determinar los requerimientos de ancho de banda del *downlink*, y utiliza varios mecanismos diferentes de comunicación (tales como *polling* y requerimientos de ancho de banda no solicitados) para mantenerse informado de los requerimientos de ancho de banda de las SSs para el *uplink*.

Finalmente, existe un *scheduler* de paquetes en la BS y en cada SS. Este *scheduler* planifica los paquetes de las colas de cada conexión en las oportunidades de transmisión asignadas a cada SS en cada *frame*.

7.1.4 Prestación de servicios

En primer lugar, se explora las diferencias en los mecanismos de QoS entre la 802.11 y 802.16, con miras a señalar los retos asociados a una implementación en gran escala de WiMAX. A continuación, se describe la arquitectura de la prestación de servicios utilizando las funcionalidades de la MAC.

7.1.4.1 Comparación de la QoS entre IEEE 802.16 y IEEE 802.11

La característica principal de una red WiFi es su simplicidad. Cada una de las SS se puede desplazar a cualquier AP (*Access Point*, punto de acceso), casi sin intervención del usuario.

Sin embargo, la simplicidad también viene con limitaciones. Incluso con la mejoras de la QoS en el estándar 802.11e, solo soporta un número limitado parámetros de calidad de servicio (es decir, ocho prioridades del usuario) y un sola conexión. La 802.11 se basa en una arquitectura distribuida, donde la operación de la MAC está coordinada entre los APs y las SSs.

Por otra parte, WiMAX se basa en una arquitectura de control centralizado, donde el *scheduler* en la BS posee un control completo de los medios de acceso inalámbrico con todas las SSs. WiMAX puede soportar múltiples conexiones que se caracterizan por un conjunto completo de parámetros de QoS. Además, WiMAX provee el clasificador de paquetes para asignar estas conexiones a las diferentes aplicaciones de usuarios e interfaces, tales como Ethernet, TDM, ATM, IP, VLAN, etc.

Sin embargo, la rica diversidad de características y la flexibilidad que brinda WiMAX también aumentan la complejidad en la implementación y prestación de servicios de banda ancha a las redes de acceso inalámbricas tanto fijas como móviles.

En los apartados siguientes se describen la prestación de servicios, autoconfiguración y la WiMAX *Management Information Base* (MIB).

7.1.4.2 Aprovisionamiento de servicio y autoconfiguración

En la Fig. 7-2 se muestra el modelo de referencia de gestión de las redes de acceso inalámbrico de banda ancha (BWA, *Broadband Wireless Access*). El modelo consiste en un

sistema de gestión de red (NMS, *Network Management System*), nodos administrados, y una base de datos de los flujos de servicio (*Service Flow Database*).

Los nodos administrados BS y SS colectan y almacenan los objetos gestionados en el formato IEEE 802.16 MIB. Los objetos gestionados se ponen a disposición de los NMS mediante el protocolo SNMP (*Simple Network Management Protocol*).

La *Service Flow Database* contiene la información de QoS asociada a cada flujo de servicio, las que son tenidas en cuenta por la BS y SS en la creación de conexiones de transporte cuando un servicio es aprovisionado o una SS entra en la red.

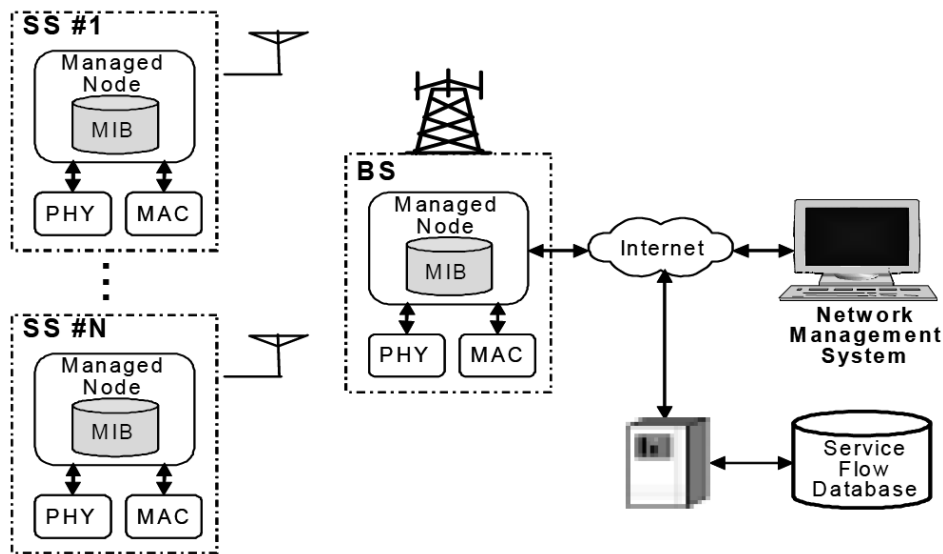


Fig. 7-2: Modelo de referencia de gestión de red. Tomada de (23).

La Fig. 7-3 muestra la estructura de la MIB de wmanIfMib para 802.16. La wmanIfMib está compuesta de tres grupos:

- wmanIfBsObjects: Este grupo contiene los objetos gestionados a ser implementados en la BS.
- wmanIfSsObjects: Este grupo contiene los objetos gestionados a ser implementados en la SS.
- wmanIfCommonObjects: Este grupo contiene objetos gestionados para ser implementados en la BS y la SS.

wmanIfMib contiene las siguientes tablas para soportar el aprovisionamiento de flujos de servicio.

wmanIfBsProvisionedSfTable

Esta tabla contiene la información de flujo de servicio pre-aprovisionado que se utilizará para crear conexiones cuando un usuario entra en la red.

- SS MAC *address*: un único identificador de las SS para asociar el flujo de servicio con una SS.
- Sentido: la dirección de este flujo de servicio (*uplink* o *downlink*).
- Índice de clase de servicio: un puntero al parámetro de QoS establecidos para dicho flujo de servicio.
- Estado del flujo de servicio: hay tres estados (aprovisionado, admitido, y activado) indicando si el recurso está preparado, admitido, o activo.

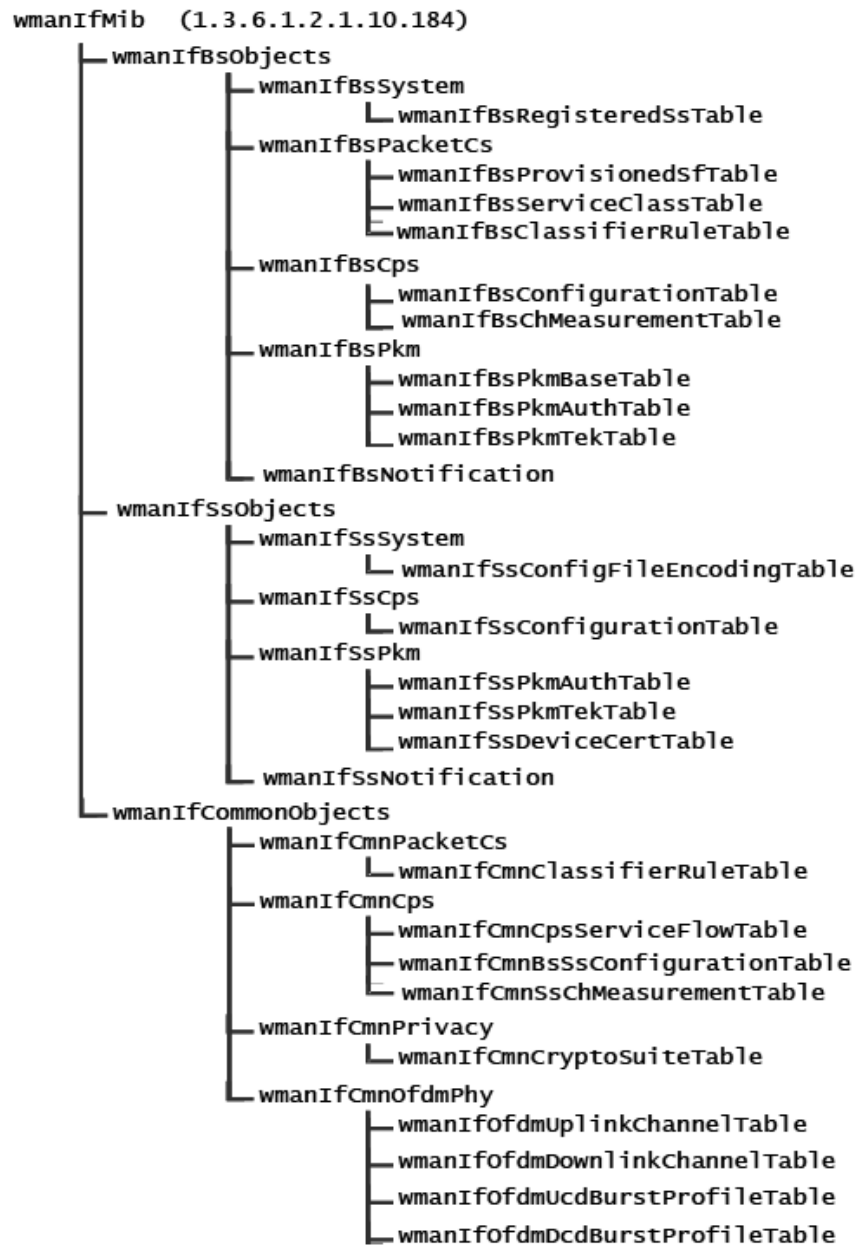


Fig. 7-3: Estructura de la wmanIfMib. Tomada de (23).

wmanIfBsServiceClassTable

Esta tabla contiene los parámetros de calidad de servicio que están asociados con cada flujo de servicio. Los principales parámetros incluidos son las siguientes:

- Prioridad de tráfico: El valor (0 .. 7) especifica la prioridad asignada a un flujo de servicio. Cuando dos flujos de servicio tienen idénticos parámetros de QoS, el flujo de servicio con mayor prioridad debería sufrir la menor demora y una mayor preferencia en los *buffers*.
- *Maximum sustained rate*: Especifica el pico de velocidad en la transferencia de flujo de servicios en bits por segundos.
- *Maximum traffic burst*: Especifica el máximo tamaño de *burst* que puede ser transportado.
- *Minimum reserved rate*: La tasa en bits por segundo promediada en el tiempo, que especifica la mínima cantidad de datos por unidad de tiempo a ser transportados en el flujo de servicio.
- *Jitter* tolerado: Especifica la variación máxima en la demora para el flujo de servicio.
- *Maximum latency*: especifica la máxima demora entre la recepción de un paquete por el BS o SS en su interfaz de red y el reenvío del paquete por su interfaz de RF.

wmanBsClassifierRuleTable

Esta tabla contiene las reglas para que el clasificador de paquetes asigne los paquetes del *downlink* y el *uplink* al flujo de servicio.

- En el sentido *downlink*, cuando se recibe un paquete de la red, el clasificador en la BS puede utilizar la dirección MAC o la dirección IP para determinar a cuál de las SS se debe enviar el paquete, y el parámetro TOS (*Type of Service*), o el DSCP (*Differentiated Service Code Point*) para seleccionar el flujo de servicio con apropiada QoS.
- En el sentido UL, cuando la BS recibe un paquete de las instalaciones del cliente, el clasificador de la SS puede usar la dirección MAC origen/destino o la dirección IP y número de puerto, TOS/DSCP, VLAN (*Virtual Local Area Network*) ID para asignar el paquete al flujo de servicio que soporte la QoS requerida.

Es muy importante en la implementación de WiMAX minimizar la intervención del cliente y los costos de la puesta en marcha por el prestatario. Lo que sigue describe las características del aprovisionamiento del servicio mediante la configuración de la *Provisioned Service Flow Table*, *Service Class Table*, y *Classifier Rule Table*, descritas anteriormente, para permitir auto instalación y autoconfiguración.

Cuando un cliente se suscribe al servicio, deberá comunicar al proveedor de servicios la información de cada flujo de servicio incluyendo el número de conexiones UL/DL con las tasas de transferencia de datos y demás parámetros de QoS, junto con el tipo de aplicaciones (por ejemplo, Internet, voz o video) que el cliente tiene intención de correr.

El proveedor de servicios de pre-aprovisionará los servicios registrando la información de los flujos de servicio en la *Service Flow Database*. Cuando la SS ingresa al sistema

completando con la BS los procedimientos de entrada a la red y autenticación, la BS bajará la información de los flujos de servicio desde la *Service Flow Database*.

La Fig. 7-4 proporciona un ejemplo que describe cómo se registra la información de los flujos de servicio. Las tablas 7A, 7B y 7C indican que dos de las SSs, identificadas por su dirección MAC 0x123ab54 y 0x45fed1, han sido previamente aprovisionadas. Cada una de las SSs tiene dos flujos de servicios, identificados por sfIndex, con los parámetros de calidad de servicio asociados que se identifican por QoSIndex 1 y 2 respectivamente

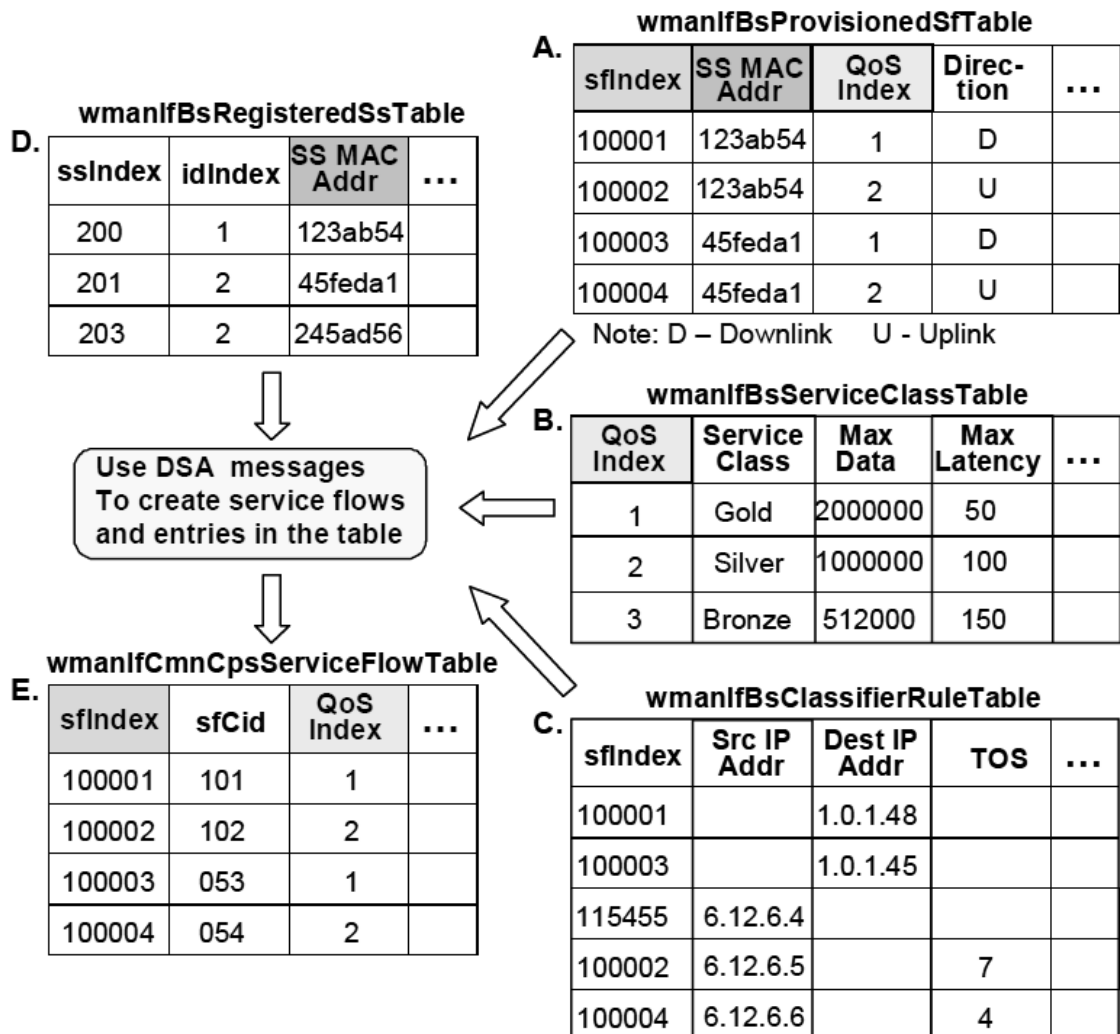


Fig. 7-4: Aprovisionamiento de flujos de servicio. Tomada de (23).

Los campos de QoSIndex apuntan a una entrada de QoS en la wmanIfBsService ClassTable la que contiene tres niveles de calidad de servicio: *Gold*, *Silver* and *Bronze*. A su vez los campos sfIndex apuntan a la entrada de la wmanBsClassifierRuleTable, indicando qué reglas se deben ser usadas para clasificar los paquetes en el flujo de servicio dado.

Cuando las SS con la dirección MAC 0x123ab54 se registra en el BS, esta crea una entrada en el `wmanIfBaseRegisteredTable` en la tabla 7D. Con la información de la dirección MAC de la SS, la BS será capaz de encontrar la información de los flujos de servicio que ya ha sido pre-aprovisionados en las tablas 7A, 7B y 7C.

El BS usará un mensaje *Dynamic Service Addition* (DSA) para crear los flujos de servicio para `sfIndex` 100001 y 100002, con la información pre-aprovisionada del flujo de servicio. Esto crea dos entradas en `wmanIfCmnCpsServiceFlowTable` en la tabla 7E. El flujo de servicio estará disponible para que el cliente envíe el tráfico de datos.

7.1.5 Desafíos en la implementación de la MAC de WiMAX y los Modelos de QoS

Las tareas realizadas por el protocolo 802.16 MAC se pueden dividir más o menos en dos categorías diferentes: actividades periódicas frecuentes (por cada *frame*) y actividades aperiódicas que se desarrollan cada muchos *frames*. Las actividades tales como el *scheduling*, empaquetamiento, fragmentación, y ARQ (*Automatic Repeat Query*), se realizan con la granularidad de los *frames*, y están condicionadas rigurosos plazos de tiempo real. Se deben completar a tiempo para la transmisión en el *frame* al cual están asociadas.

Por el contrario, las actividades aperiódicas se ejecutan conforme a temporizadores que no están asociados con un *frame* específico o el período de un *frame*, por lo tanto no tienen plazos estrictos que cumplir.

Las dos categorías de las tareas descritas anteriormente interactúan en que las actividades poco frecuentes normalmente afectan a las muy frecuentes. Por ejemplo el registro y asociación de una SS con una BS, ocurre tras el intercambio de muchos mensajes, que resultan en la creación de muchas conexiones y estados asociados entre la SS y la BB. Estas conexiones pueden incluir actividades que deban ser ejecutadas cada un período de *frame*, tales como estado de fragmentación, retransmisiones ARQ y empaquetamiento.

Además de soportar la QoS y funcionalidades MAC descritas anteriormente, las implementaciones plantean un conjunto de desafíos de virtualización a las que las funcionalidades de la MAC deben satisfacer.

En concreto, se espera que en el intervalo de la configuración del sistema, sea posible configurar un sistema único para el tratamiento de múltiples canales de aire como instancias MAC separadas. Así, una única BS (e implementaciones MAC asociadas) podrían, por ejemplo utilizar dos canales de 10 MHz en paralelo en dos instancias MAC diferentes. Este tipo de virtualización es necesario porque la utilización y asignación del ancho de banda

disponible de aire es altamente dependiente de las políticas de transporte, la carga del sistema y el entorno de RF.

7.2 Información en los paquetes necesaria para el *scheduler*

Una observación cuidadosa del apartado apartado 7.1.4.2, la información de la tabla *wmanBsClassifierRuleTable*, y la Fig. 7-4 transcriptos de (23) permiten deducir que los paquetes que arriban a una estación WiMAX no necesitan transportar ninguna información adicional para que el *scheduler* pueda planificar con eficiencia el mapeo de paquetes en el *frame*.

La información necesaria está limitada a aquella normalmente incorporada en los segmentos y paquetes del protocolo TCP/IP.

Los paquetes no necesitan transportar ninguna información adicional, en su encabezado o cuerpo a la que está prevista en la correspondientes RFC publicadas por la IETF. El único cambio consiste en la real puesta en función del campo DSCP (*Differential Service Code Point*), el que reemplaza a su versión anterior TOS (*Type of Service*), para que el *scheduler* pueda seleccionar el *service flow* que soporte la QoS acordada.

Tanto en el DL como en el UL, el TOS o DSCP permite seleccionar la QoS requerida por el paquete. El DSCP informa implícitamente el tipo de aplicación.

El otro dato importate es la SS que origina o a la cual va destinado el paquete, en el caso del DL es la IP de la SS destino. En el caso del UL puede ser la dirección MAC origen o el IP origen. Esta información está relacionada con el contrato de nivel de servicio (SLA, *Service Level Agreement*) de la SS con el prestatario del servicio. Puede la aplicación requerir UGS pero el usuario por razones económicas solo acordó servicio BE.

Tabla 7-1: Algunas combinaciones de atributos de los paquetes para un adecuado funcionamiento del MAC *scheduler* de WiMAX.

Sentido	IP origen	IP destino	MAC origen	TCP Port origen	TCP Port destino	DSCP/TOS	VLAN ID
<i>downlink</i>		X				X	
<i>downlink</i>		X		X			
<i>downlink</i>	X	X		X			
<i>downlink</i>		X					X
<i>uplink</i>	X					X	
<i>uplink</i>	X				X		
<i>uplink</i>			X		X		
<i>uplink</i>	X						X

Puede que el DSCP o TOS no esté disponible por alguna razón, por ejemplo un servidor de la web no tiene ninguna obligación ni preocupación en enviar el DSCP o envía el más alto desaprensivamente para obtener alguna ventaja, o en el *uplink* las aplicaciones de los usuarios no están preparadas para incorporar un valor adecuado al DSCP. En estos casos el *scheduler* puede recurrir para clasificar los paquetes por el número de puerto TCP origen en el *downlink*, o el puerto destino en el UL, los que indican en forma implícita la aplicación y por ende la QoS asociada.

Se entiende que con el tiempo los *schedulers* serán cada vez más sofisticados e irán avanzando en el desarrollo de nuevos algoritmos para la clasificación automática de los paquetes, incluso pudiendo llegar al análisis de contenido, si no está encriptado.

Como sea la información que se necesite, actualmente no se necesita más de tres informaciones o atributos en los paquetes para que el *scheduler* pueda cumplir su función cabalmente.

En la Tabla 7-1 se muestra algunas de las combinaciones de atributos de los paquetes que se entiende aceptables para el adecuado funcionamiento del scheduler de WiMAX.

Cuales atributos de los paquetes se toman en cuenta para el *scheduling*, es una cuestión abierta reservada a los fabricantes de equipos, en este trabajo se considerará el uso de no más de tres atributos de los paquetes, aunque el modelo a desarrollar será lo suficientemente abierto para admitir la incorporación de más de tres.

Esto es importante ya que define la información, que hay que adicionar a los tiempos de arribo, que debe acarrear cada paquete de un tráfico sintético para ser aplicado al desarrollo y prueba de los *schedulers* de WiMAX que es el objetivo de este trabajo.

7.3 Modelo de Simulink

7.3.1 Introducción

Los *schedulers* de WiMax han sido tratados profusamente en la literatura. Sin embargo ninguna compañía fabricante de equipos o módulos para WiMAX ha hecho público sus algoritmos con cierto detalle. Es tanta la influencia de este bloque en la performance de los equipos y en los requerimientos de capacidad de procesamiento y de memoria de los controladores embebidos, lo cual se traduce directamente en costo. Esto hace que dichos algoritmos se constituyan en un secreto muy bien guardado de la industria moderna de las telecomunicaciones.

Sin embargo no es un secreto la información que los mismos requieren para satisfacer la QoS y optimizar el rendimiento de los equipos. Mucha información que necesita el *scheduler* no se toma de los paquetes que viene de las capas superiores, sino que se genera en la misma capa MAC, por ejemplo niveles de recepción, relación señal ruido, pedido de ancho de banda, *handover*, etc. Realmente la capa MAC de WiMAX realiza muchos y muy complicados procesos, los cuales son absolutamente transparentes a las capas superiores, entre ellos la planificación de las comunicaciones para atender compromisos acordados previamente para satisfacer los niveles de QoS acordados con los clientes.

A los fines de proveer una herramienta adecuada para el desarrollo y la optimización de los *schedulers* de WiMAX se ha desarrollado un modelo en Simulink, manipulado con un *script* desde el *workspace* de Matlab. El funcionamiento del *script* de Matlab y el modelo en Simulink se analizará a continuación, pero el código del *script* de Matlab se transcribe en el Anexo 5 – *Script* para generar tráfico para WiMAX.

Como se concluye en 7.2 los atributos de los paquetes que se toman en cuenta para el *scheduling* es una cuestión abierta reservada a los fabricantes de equipos, pero que importa el uso de no más de tres atributos de los paquetes. Por lo tanto en el modelo desarrollado, se considera hasta tres atributos por paquete cualquiera que sean o se usen. Dos de ellos se denominan arbitrariamente IPSource e IPDest (ya que dependen si es *uplink*, *downlink* o de las decisiones del fabricante de equipos, además estas denominaciones pueden ser convenientes para el uso del simulador en aplicaciones que no sean WiMAX), y el tercer atributo DSCP. Además se incorpora un cuarto atributo con la longitud o carga útil de los paquetes, referidos a los segmentos TCP para compatibilizar esta información con la de muestras de tráfico registradas de la realidad, lo cual completa la información necesaria transportada por los paquetes para que el *scheduler* realice su tarea.

El *script* de Matlab y el modelo de Simulink que se propone, trabaja con la información generada por cualquier simulador de tráfico, que proporcione tiempos interarribo o de arribo de paquetes, tal como el modelo PMPP descrito anteriormente, esta información la toma Simulink de un vector almacenado previamente en el *workspace* de Matlab cargado previamente por el *script*.

Los paquetes generados por el simulador de tráfico se marcan con atributos conforme a reglas de juego predefinidas incorporadas en el código del *script* y en los bloques del modelo de Simulink, parametrizadas con la información de configuración almacenada en un arreglo de Matlab con datos ingresados por el usuario del modelo.

El modelo incorpora la posibilidad de agregar tráfico con tiempos interarribo constante para simular el servicio UGS que soporta CBR tal como emulación T1/E1 y VoIP sin supresión de silencios. La información de configuración para este tráfico se introduce en un segundo arreglo, con un formato similar al primero.

En un tercer arreglo se almacena una secuencia de longitudes de paquetes generados previamente mediante una distribución arbitraria. En el ejemplo desarrollado aquí, las longitudes de los paquetes se generan usando la distribución recabada de un histograma correspondiente a la muestra LBL-PKT4.

7.3.2 Arreglo de configuración para la distribución estadística de atributos con tiempos interarribo variables

Tras múltiples pruebas intentando crear una GUI comprensiva para la entrada de datos surgió la convicción de que la mejor forma consiste en emplear como entrada y modificación de datos al mismo editor de variables de Matlab, que tiene un manejo simple, eficaz e intuitivo, el cual permite además una extensa variedad de operaciones.

Hay que expresar claramente que Matlab no es un producto orientado a crear aplicaciones comerciales enlatadas en un .exe o un instalador, sino más bien un entorno plástico, potente y agradable para el desarrollo de rutinas matemáticas orientado al desarrollo y la investigación.

	1	2	3	4
1	2	5	12	24
2	0.1000	0.2000	0.3000	0.4000
3	[6,7,8,9]	[10,11,12,13,14]	[15,16,17,18,19,20]	[21,22]
4	[0.5000,0.3000,0.1000,0.1000]	[0.2000,0.2000,0.3000,0.1500,0.1500]	[0.7000,0.2000,0.0250,0.0250,0.0250,0.0250]	[0.5000,0.5000]
5	[0.0300,0.0700,0.2000,0.3000,0.4000]	[0.0300,0.0700,0.2000,0.3000,0.4000]	[0.1000,0.2000,0.3000,0.1500,0.2500]	[0.0300,0.0700,0.2000,0.3000,0.4000]
6	[0.0300,0.0700,0.2000,0.3000,0.4000]	[0.0300,0.0700,0.2000,0.3000,0.4000]	[0.2500,0.2500,0.2500,0.2400,1.0000e-03]	[0.0300,0.0700,0.2000,0.3000,0.4000]
7	[0.0300,0.0700,0.2000,0.3000,0.4000]	[0.0300,0.0700,0.2000,0.3000,0.4000]	[0.2000,0.2000,0.2000,0.2000,0.2000]	[]
8	[0.0300,0.0700,0.2000,0.3000,0.4000]	[0.0300,0.0700,0.2000,0.3000,0.4000]	[0.2000,0.2000,0.2000,0.2000,0.2000]	[]
9	[]	[0.0300,0.0700,0.2000,0.3000,0.4000]	[0.8000,0.1000,0.0200,0.0200,0.0600]	[]
10	[]	[]	[0.5000,0,0,0,0.5000]	[]
11				

Fig. 7-5: Arreglo At para introducción de atributos para el tráfico ya generado previamente.

El arreglo de celdas (*cells*) para la introducción y modificación de los valores que definen la generación de los atributos IPSource, IPDest y DSCP se muestra en la Fig. 7-5. En la primera fila están la IPSource con denominación absolutamente arbitraria, 2, 5, 12 y 24. Si se desea agregar más IPSource se deben introducir más columnas a la derecha en el arreglo.

En la segunda fila están las probabilidades correspondientes, en tanto por uno, en que los paquetes serán marcados. La suma horizontal debe ser igual a 1.

Por ejemplo, si el paquete es marcado con IPSource=12, el vector [15, 16, 17, 18, 19,20] define los valores posibles para IPDest conforme a la distribución arbitraria definida por el usuario que está en la celda inmediatamente inferior:

[0.7000, 0.2000, 0.0250, 0.0250, 0.0250, 0.0250].

Las filas siguientes son relevantes tantas como cantidad de IPDest que hay en la tercera fila. En este caso hay 6 valores de IPDest y sus correspondientes 6 filas:

15→[0.1000, 0.2000, 0.3000, 0.1500, 0.2500]

16→[0.2500, 0.2500, 0.2500, 0.2400, 0.0010]

17→[0.2000, 0.2000, 0.2000, 0.2000, 0.2000]

18→[0.2000, 0.2000, 0.2000, 0.2000, 0.2000]

19→[0.8000, 0.1000, 0.0200, 0.0200, 0.0600]

20→[0.5000, 0, 0, 0.5000]

Los valores de IPDest no necesitan ser consecutivos como es el caso del ejemplo. Cada fila tiene 5 valores de probabilidad correspondientes a cada una de las QoS definidas en la norma IEEE802.16-2009 (1: UGS, 2: ertPS, 3: rtPS, 4: nrtPS y 5: BE).

Supóngase que el “azar” selecciona IPDest=19, el vector [0.800, 0.100, 0.0200, 0.0200, 0.0600] define cuales son las probabilidades que tiene el paquete de pertenecer a un *flow* con QoS: UGS, ertPS, rtPS, nrtPS y BE respectivamente.

En base a los datos del arreglo de configuración At, un *script* de Matlab, mostrado en el Anexo 5 – *Script* para generar tráfico para WiMAX, genera tres vectores con longitudes definidas en el *script*, con secuencias aleatorias (en realidad son pseudo aleatorias) de los atributos IPSource, IPDest y DSCP respectivamente, los cuales etiquetan cada paquete a medida que son generados. Ver Fig. 7-6.

Si bien el tráfico UGS se genera en otro bloque del modelo de Simulink con tiempo interarribo constante, se ha dejado la opción UGS para tiempos interarribo aleatorios, por si se desea simular variaciones en el *bitrate* de UGS, dado que las mismas son esperables debido a lo inesperado en los cambios de las características de propagación, propias de un enlace de RF en contraposición con la estabilidad de un enlace de fibra o cobre los cuales son mucho más estables.

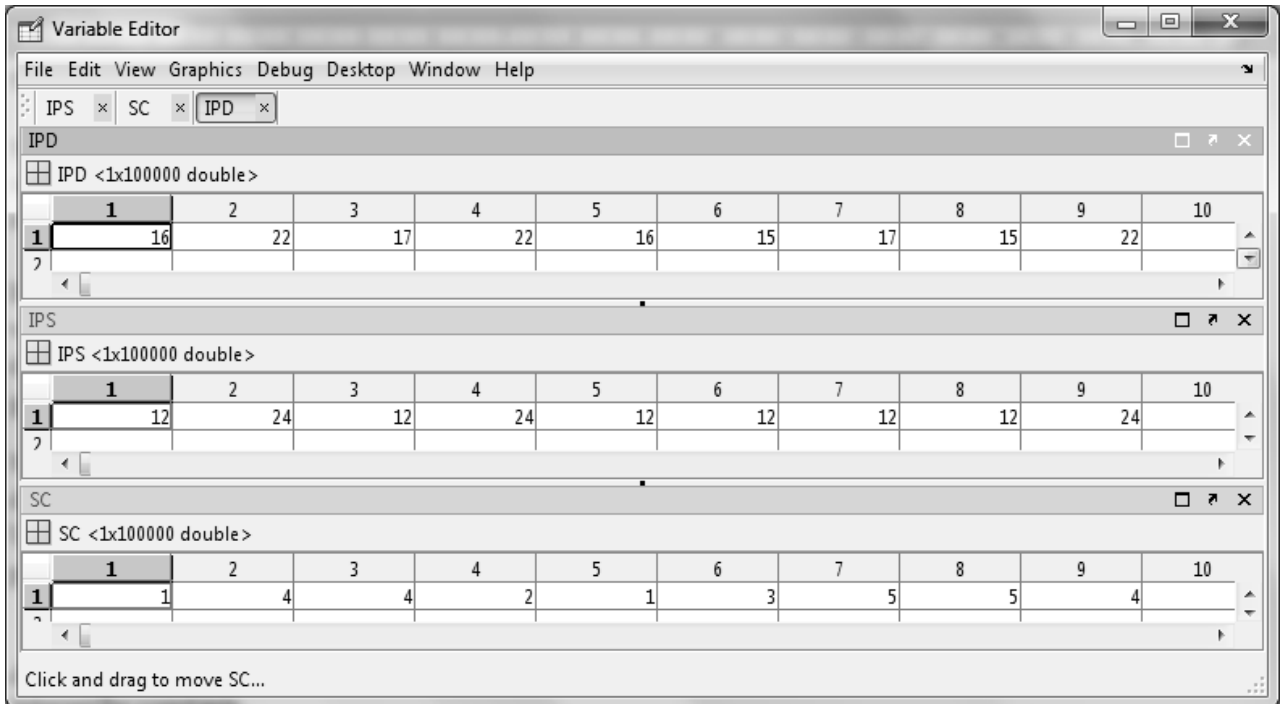


Fig. 7-6: Arreglos de los atributos IPDest (IPD), IPSource (IPS) y DSCP (SC), con las secuencias pseudo aleatorias generadas por un *script*, conforme a la configuración especificada en el arreglo At.

7.3.3 Arreglo de configuración para la distribución estadística de atributos correspondiente a tráfico con tiempos interarribo constante

El tráfico UGS, cuyo tiempo interarribo es aproximadamente constante, se genera en el mismo modelo de Simulink, pero del mismo modo que el tráfico con tiempos interarribo variables, las secuencias de atributos se importan del *workspace* de Matlab. Un *script* genera los atributos conforme a la configuración definida en un arreglo AtCte.

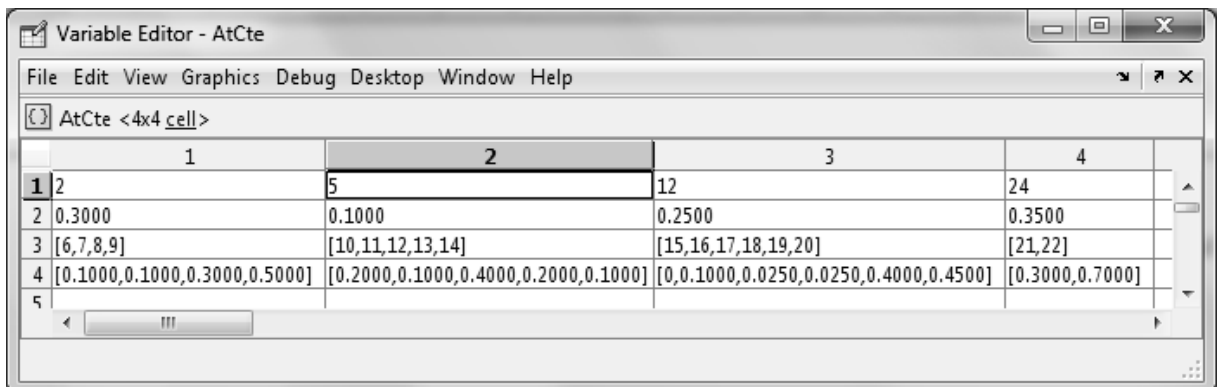


Fig. 7-7: Arreglo AtCte para introducción de atributos para el tráfico con tiempos interarribo constante.

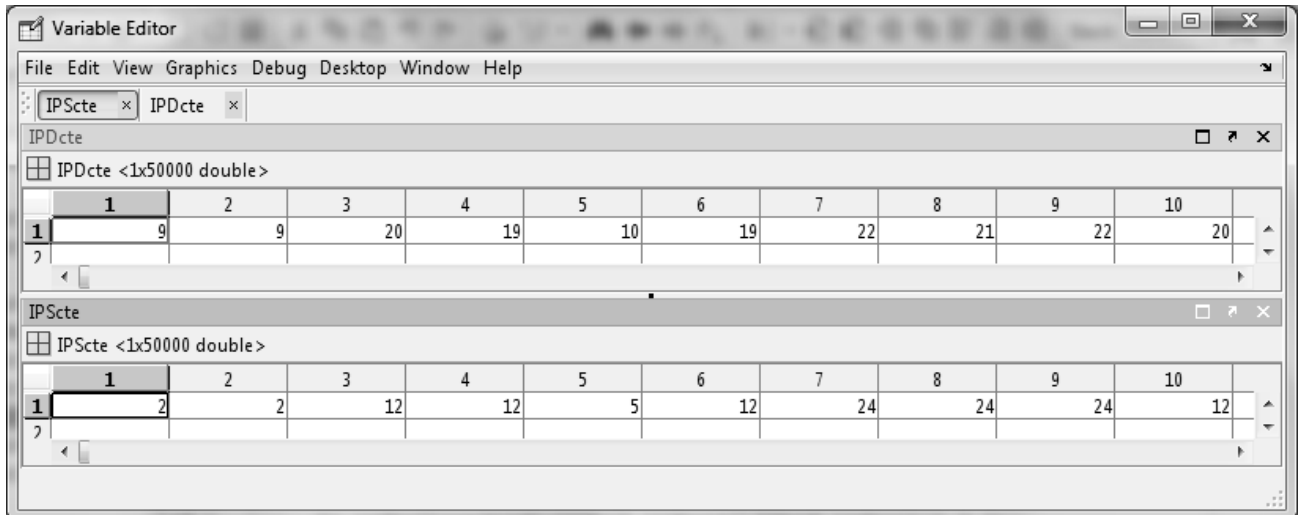


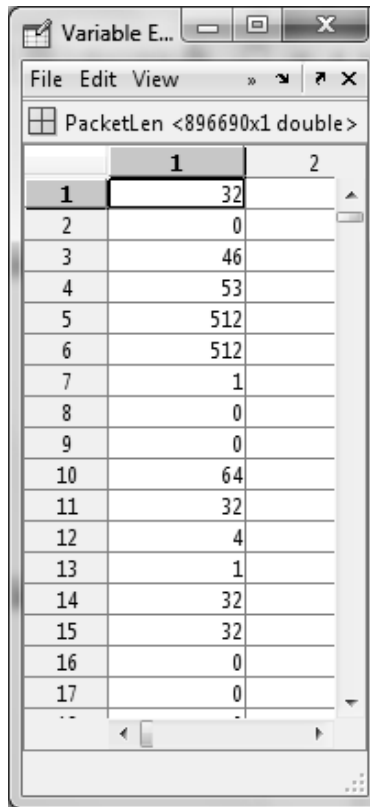
Fig. 7-8: Arreglos de los atributos IPDest (IPDcte) y IPSource (IPScte) con las secuencias pseudo aleatorias generadas por un *script*, conforme a la configuración especificada en el arreglo AtCte. El atributo DSCP es siempre 1.

El arreglo de celdas AtCte para la introducción y modificación de los valores que definen la generación de los atributos IPSource, IPDest se muestra en la Fig. 7-7. Igualmente que en el arreglo At descrito anteriormente, en la primera fila están la IPSource con denominaciones 2, 5, 12 y 24 para el ejemplo mostrado en la figura. En la fila inmediatamente inferior están los valores correspondientes a sus probabilidades.

Si un paquete es marcado con IPSource=5, el atributo IPDest puede ser cualquiera del vector [10, 11, 12, 13, 14] de la tercera fila de la misma columna con probabilidad dada por el vector [0.2000, 0.1000, 0.4000, 0.2000, 0.1000] respectivamente.

Con los datos del arreglo de configuración AtCte, el *script* de Matlab genera dos vectores, con longitudes definidas en el *script*, conteniendo secuencias aleatorias de los atributos IPSource y IPDest los cuales etiquetan cada paquete a medida que son generados en el modelo de Simulink. Ver Fig. 7-8.

7.3.4 Arreglos de configuración para la distribución estadística de longitudes de paquetes



	1	2
1	32	
2	0	
3	46	
4	53	
5	512	
6	512	
7	1	
8	0	
9	0	
10	64	
11	32	
12	4	
13	1	
14	32	
15	32	
16	0	
17	0	

Fig. 7-9: En el vector PacketLen están las longitudes de la carga útil TCP/UDP de cada paquete de la muestra de tráfico LBL-PKT4.

En este caso, se tomó la decisión de obtener la distribución de longitudes de paquetes directamente de la muestra LBL-PKT4 almacenadas en el vector PacketLen mostrado en la en Fig. 7-9 en vez de generar una.

A partir de esta información, el *script* genera el arreglo de configuración Histo compuesto por dos columnas, el que se muestra en la figura Fig. 7-10. La columna de la izquierda contiene las longitudes de paquetes que tienen probabilidad mayor que cero, y la de la derecha donde se especifica el valor de dicha probabilidad.

Si las longitudes de paquetes con probabilidad no nula son pocas, se puede llenar a mano en forma directa el arreglo Histo, verificando que la suma de todas las probabilidades sea uno. En la Fig. 4-4 se muestra el histograma correspondiente a la muestra LBL-PKT4, la información es la misma que la de la Fig. 7-10, pero sin normalizar.

Variable Editor - Histo

Histo <767x2 double>

	1	2
1	0	0.345463872687328
2	1	0.145788399558376
3	2	0.014833442995907
4	3	0.010716078020275
5	4	0.005541491485352
6	5	0.023711650626192
7	6	0.004176471244243
8	7	0.001682855836465
9	8	0.008046259019282
10	9	0.001269111956194
11	10	7.918009568524240e-04
12	11	6.423624664042200e-04
13	12	0.002258305545952
14	13	0.005211388551227
15	14	7.594597910091559e-04
16	15	0.006418048600966
17	16	0.001511113093711
18	17	0.001000000000000

Fig. 7-10: El arreglo Histo contiene la distribución de probabilidades para cada longitud de paquetes de la mencionada muestra, en la columna de la izquierda las longitudes de paquete y en la derecha las probabilidades correspondientes, la suma de todos los valores de la segunda columna totaliza uno.

LongPaq

LongPaq <1x100000 double>

	1	2	3	4	5	6	7	8
1	0	1	13	32	0	0	2	512

Fig. 7-11: Vector del atributo Len (LongPaq) con las secuencias pseudo aleatorias generadas por un *script*, conforme a la distribución definida en el arreglo Histo de la Fig. 7-10.

Posteriormente el *script* genera a partir de la distribución definida en el arreglo Histo una secuencia de longitudes aleatorias para el marcado de los paquetes, ver la Fig. 7-11. Este atributo se aplica a todos los paquetes del tráfico sin tener en cuenta si es UGS o no

7.3.5 Modelo de Simulink para la asignación de atributos a los paquetes y generación de tráfico UGS

En la Fig. 7-13 se muestra el modelo de Simulink. Si no se aclara otra cosa, las explicaciones subsiguientes se refieren a esta figura.

El modelo se compone de dos cadenas de entidades, una en la parte superior y otra en la parte inferior. Ambas se inician con un *Entity Generator* (generador de entidades), las que representan paquetes, y terminan en un *Entity Sink* (sumidero de entidades) respectivamente. Si la posición de la llave *Manual Switch* conecta el bloque *Constant* con valor 1 a la entrada P del bloque *Output Switch*, el flujo de entidades de la cadena inferior se combina con el de la cadena superior, adicionando tráfico con tiempo interarribo constante al tráfico generado por la cadena superior.

Las entidades de ambas cadenas, a medida que se generan son marcadas con atributos tomados de vectores desde el *workspace* desde Matlab, previamente generados por un *script*, el cual se transcribe en el Anexo 5 – *Script* para generar tráfico para WiMAX.

Finalmente, los paquetes antes de finalizar su existencia en el bloque sumidero, se recupera de los mismos sus tiempos de arribo y el valor correspondiente de sus atributos, los cuales son exportados al *workspace* de Matlab para su combinación en un arreglo único el que finalmente se almacena en un archivo.

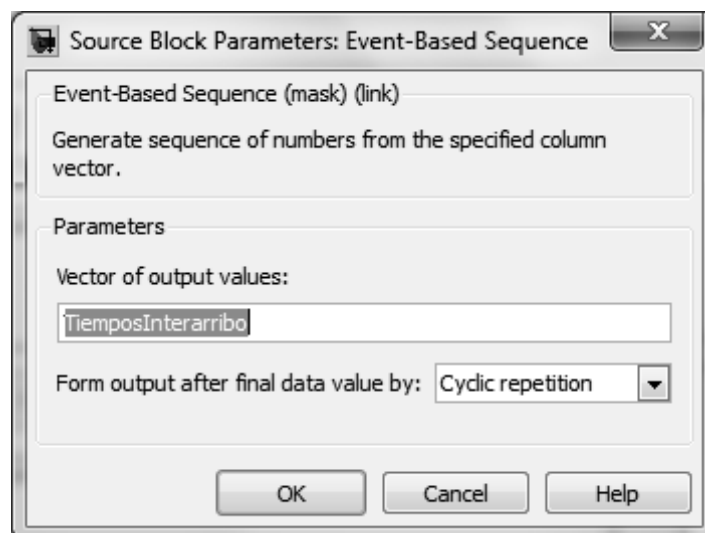


Fig. 7-12: Ventana de diálogo del bloque *Event-Based Sequence* que genera una secuencia de números tomados del vector *TiemposInterarribo* cargado desde el *workspace* de Matlab. Los bloques *Event-Based Sequence1*, *Event-Based Sequence 2* y *Event-Based Sequence3*, poseen la misma ventana de diálogo salvo que el nombre de los vectores son *IPS*, *IPD*, y *SC* respectivamente.

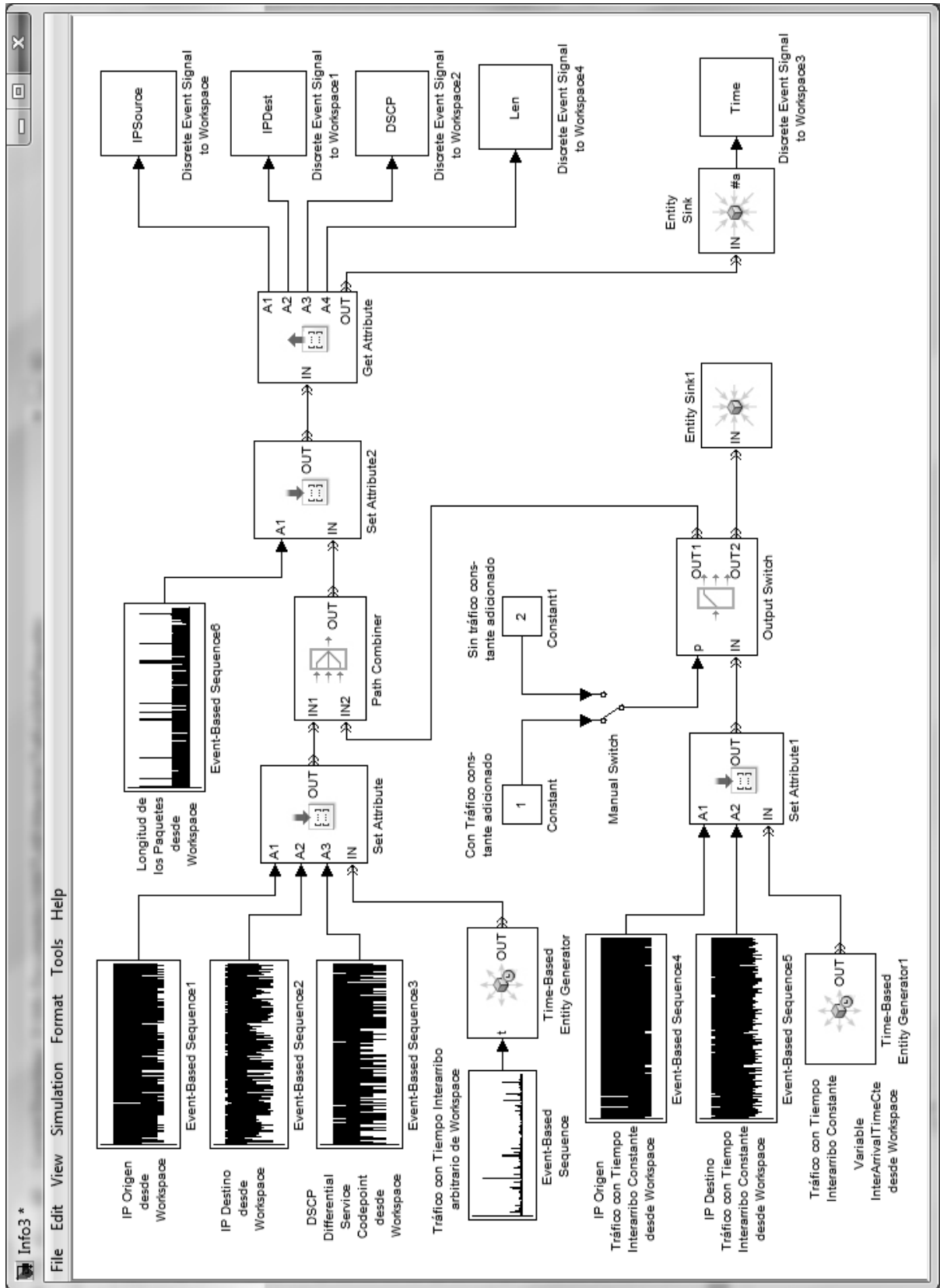


Fig. 7-13: Modelo de Simulink para el marcado de paquetes con atributos y generación de tráfico con tiempo interarribo constante.

7.3.5.1 Cadena superior

La cadena superior genera las entidades en el bloque *Time-Based Entity Generator*, con tiempos interarribo producidos por el bloque *Event-Based Sequence*, cuya ventana de diálogo se muestra en la Fig. 7-12, el cual carga del *workspace* de Matlab el vector *TiemposInterarribo*, en el que se almacena la secuencia de intervalos de generación entre paquete y paquete.

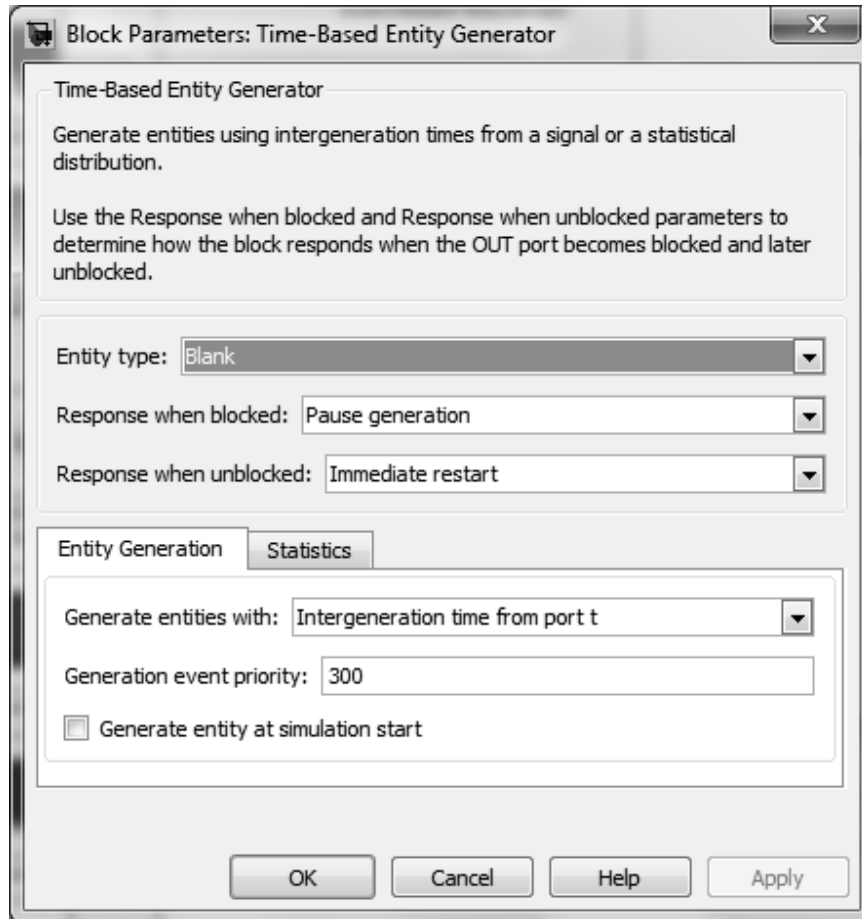


Fig. 7-14: Ventana de diálogo del bloque *Time-Based Entity Generator* configurado para generar entidades con intervalos entre paquetes tomados del puerto *t*.

Este bloque saca un nuevo valor del vector *TiemposInterarribo* por su salida, solo cuando el puerto del bloque siguiente, en este caso el puerto *t* del bloque *Time-Based Entity Generator*, Fig. 7-14, le notifica del egreso o paso de una entidad.

Estas entidades, al pasar por el bloque *Set Attribute*, cuya ventana de diálogo se muestra en la Fig. 7-15, son marcados por los atributos generados por los bloques *Event-Based Sequence1*, *Event-Based Sequence2* y *Event-Based Sequence3* los que cargan desde el *workspace* de Matlab los vectores de nombre *IPS*, *IPD*, y *SC* generados oportunamente por el

script. Las ventanas de diálogo de estos bloques son similares a la de la Fig. 7-12, excepto por el nombre de los vectores.

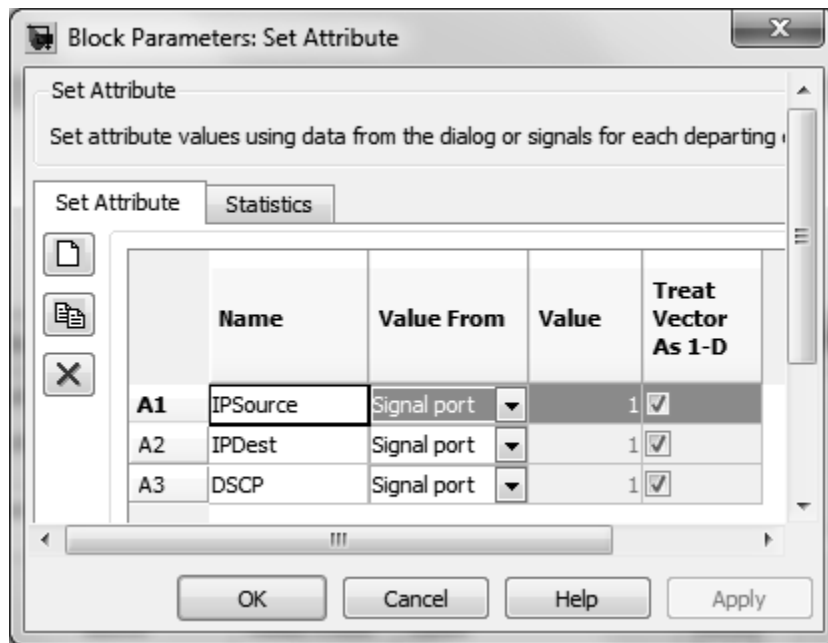


Fig. 7-15: Ventanas de diálogo del bloque Set Attribute el cual marca los paquetes con las secuencias aleatorias disponible en sus *notifying ports* A1, A2 y A3.

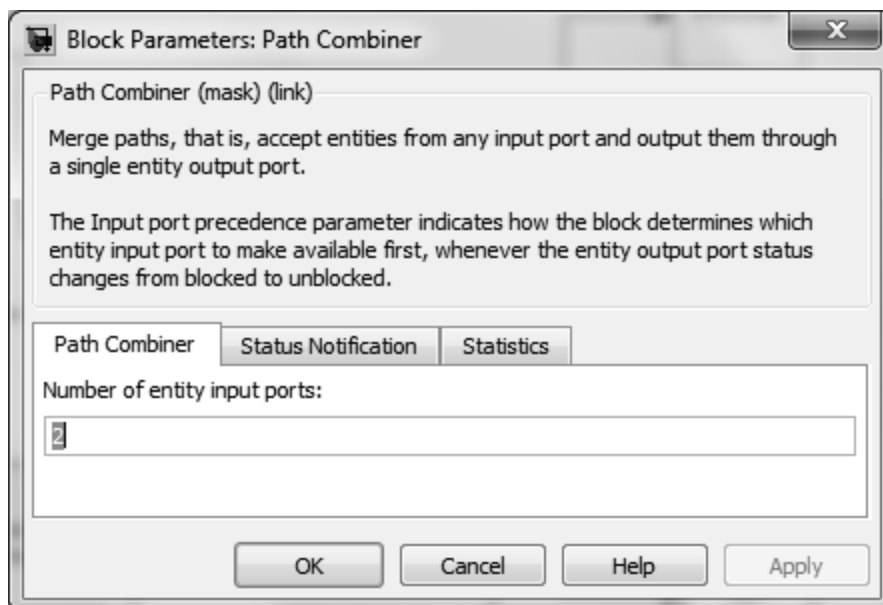


Fig. 7-16: Ventana de diálogo del bloque *Path Combiner*. Este bloque combina flujos de entidades.

A continuación, las entidades atraviesan el bloque *Path Combiner*, cuya ventana de diálogo se muestra en la Fig. 7-16. Este bloque acepta entidades por ambos puertos de entrada, combinando el flujo de las mismas en una única salida. Por el puerto IN1 recepta los paquetes

de la cadena superior y por el puerto IN2 los de la cadena inferior los que poseen tiempos interarribo constantes.

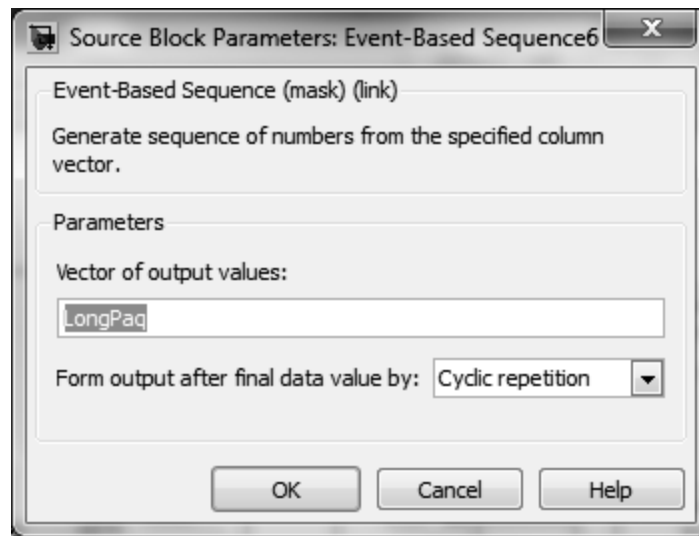


Fig. 7-17: Ventana de diálogo del bloque *Event-Based Sequence6* que genera una secuencia de números tomados del vector LongPaq cargado desde el *worksapce* de Matlab.

El bloque siguiente Set Attribute2, cuya ventana de diálogo se muestra en la Fig. 7-18, marca los paquetes con el atributo PktSize, provisto por el bloque *Event-Based Sequence6*, Fig. 7-17, el cual carga del *workspace* de Matlab el vector LongPaq que contiene una secuencia de tamaños de paquetes previamente generados por el *script*.

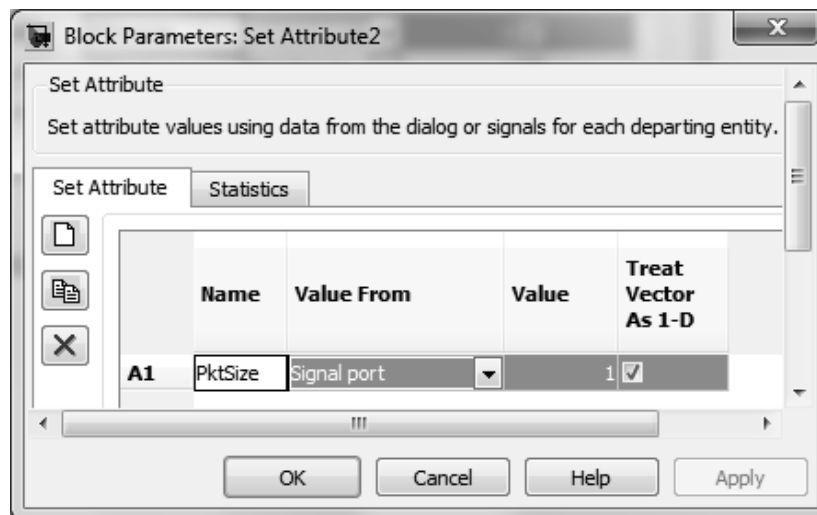


Fig. 7-18: Ventanas de diálogo del bloque *Set Attribute2* el cual marca los paquetes con los secuencias aleatorias disponibles en su *notifying port* A1 provista por el bloque *Event-Based Sequence6*.

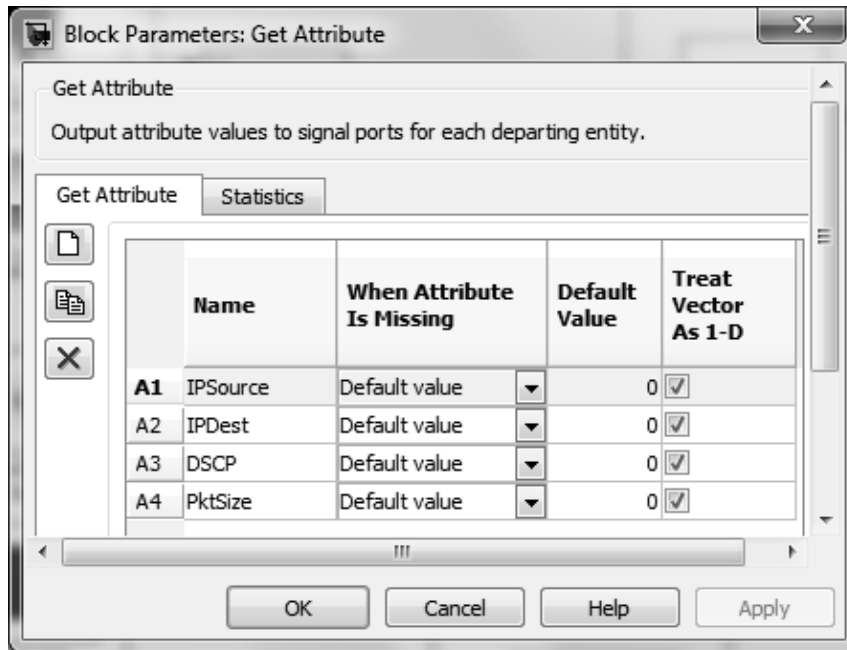


Fig. 7-19: Ventana de diálogo del bloque *Get Attribute*. Este bloque recupera los atributos de cada paquete y los saca por sus puertos A1, A2, A3 y A4.

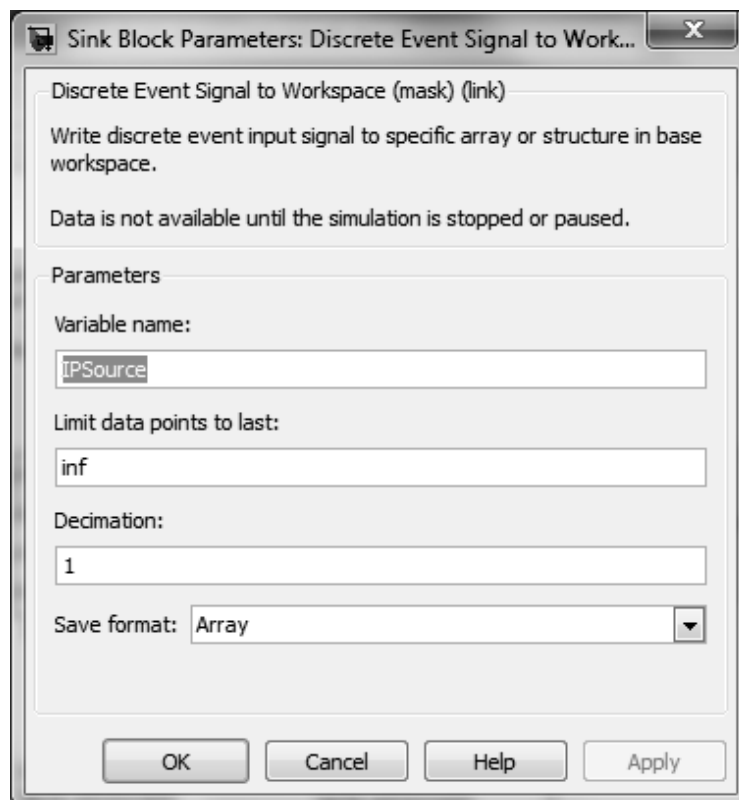


Fig. 7-20: Ventana de diálogo del bloque *Discrete Event Signal to Workspace*. El valor del campo *Variable name* es el nombre con que Simulink exportará al *workspace* el vector con el atributo IPSource. Las instancias 1,2 y 4 del mismo bloque cumplen idéntica función para los atributos IPDest, DSCP y Pkt Size.

A continuación el bloque *Get Attribute* extrae de cada entidad los atributos con los cuales se las marcó previamente, en la Fig. 7-19 se muestra la ventana de diálogo correspondiente.

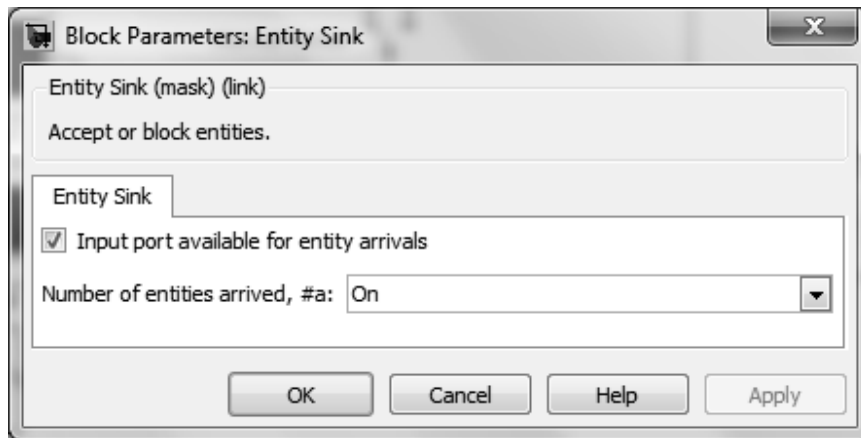


Fig. 7-21: Ventana de diálogo del bloque *Entity Sink* con la salida número de entidades arribadas habilitada.

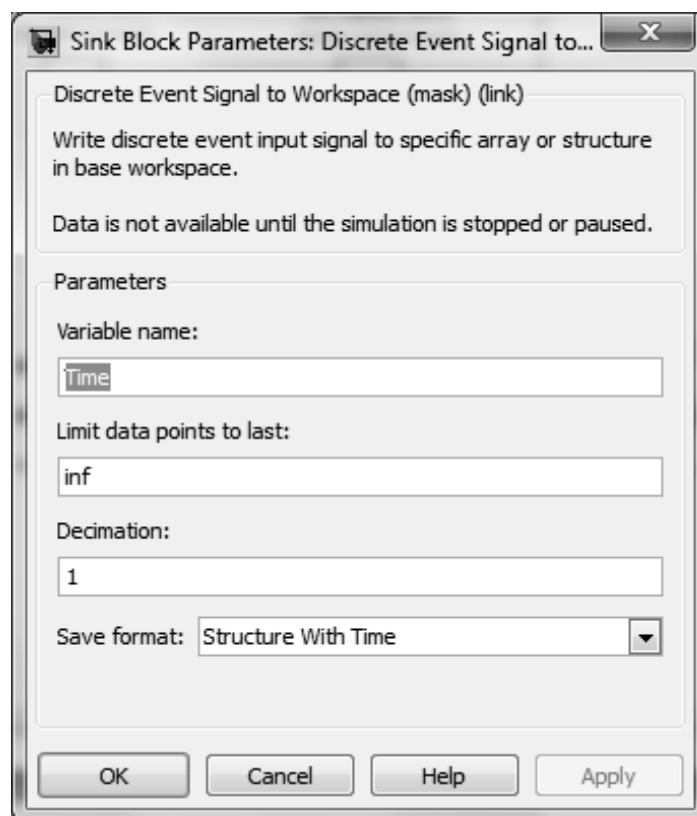


Fig. 7-22: Ventana de diálogo del bloque *Discrete Event Signal to Workspace3*. El valor del campo nombre de variable es el nombre con que Simulink exporta al *workspace* de Matlab una estructura cuyo elemento *Time.time* es un vector que almacena los tiempos de arribo de cada paquete al bloque *Entity Sink*.

La salidas del bloque *Get Attribute* se conectan a los bloques *Discrete Event Signal to Workspace* los que envían al *workspace* en vectores las secuencias de cada atributo recuperado de los paquetes. En la Fig. 7-20 se muestra la ventana de diálogo del bloque

Discrete Event Signal to Workspace que envía el atributo IPSource al *workspace* en un vector del mismo nombre.

Las salidas de los atributos IPDest, DSCP, y PktSize se conectan respectivamente a los bloques *Discrete Event Signal to Workspace1*, *Discrete Event Signal to Workspace2*, y *Discrete Event Signal to Workspace4*, los cuales envían al *workspace* los atributos con nombres IPDest, DSCP, y Len respectivamente.

Las entidades terminan su periplo en el bloque *Entity Sink*, cuya ventana de diálogo se muestra en la Fig. 7-21, el cual tiene la salida con el número de entidades arribadas habilitada.

Esta salida se conecta al bloque *Discrete Event Signal to Workspace3*, cuya ventana de diálogo se muestra en la Fig. 7-22, en el que se ha configurado el formato del archivo de salida como una estructura que incluye el tiempo de arribo al bloque *Entity Sink*.

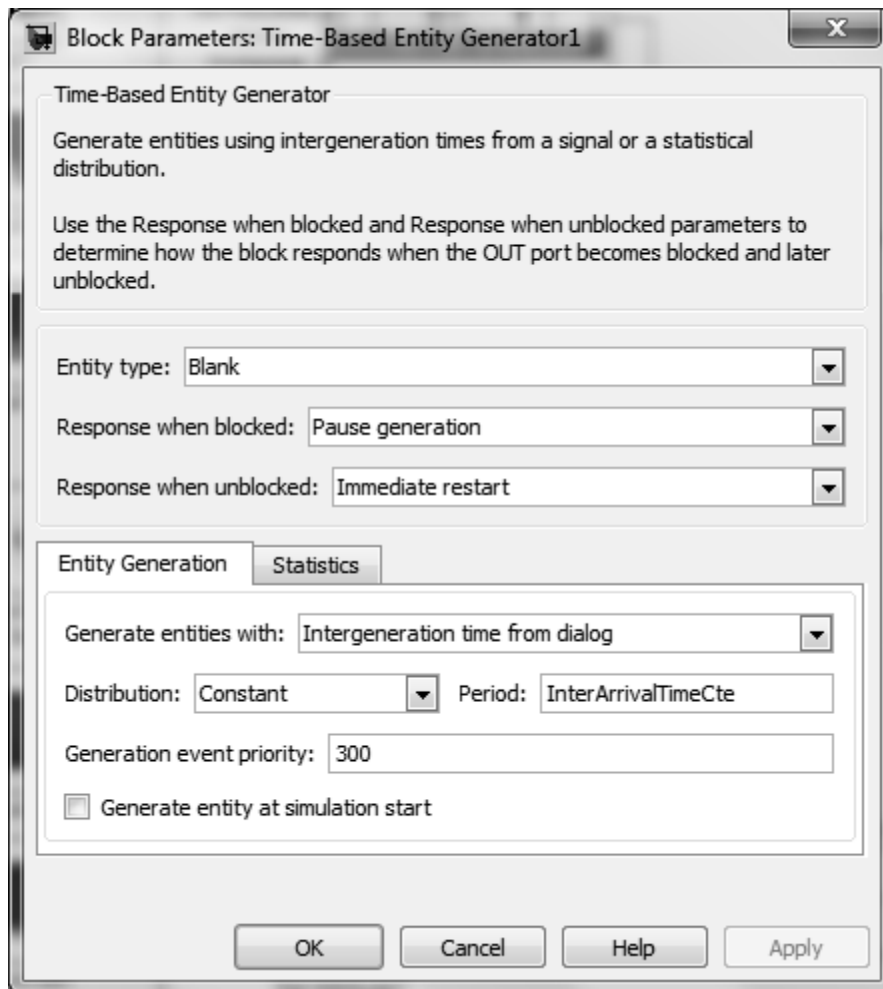


Fig. 7-23: Ventana de diálogo del bloque *Time-Based Entity Generator1* configurado para generar entidades a intervalos constantes, cargando el valor de dicho período de la variable *InterArrivalTimeCte* del *workspace*.

Una vez finalizado el tiempo de ejecución del modelo de Simulink info3, el *script* toma nuevamente el control agrupando los cinco vectores Time.time, IPSource, IPDest, DSCP, y Len en un único arreglo que se almacena como un archivo de Matlab. Este arreglo almacena el el tráfico con atributos para ensayo de los *schedulers* de WiMAX.

7.3.5.2 Cadena inferior

La cadena inferior genera las entidades en el bloque *Time-Based Entity Generator1*, con tiempos interarribo constantes, configurados en su ventana de diálogo, Fig. 7-23. Al abrirse el modelo, Simulink lee el período interarribo de la variable InterArrivalTimeCte, generado previamente por el *script* en el *workspace*.

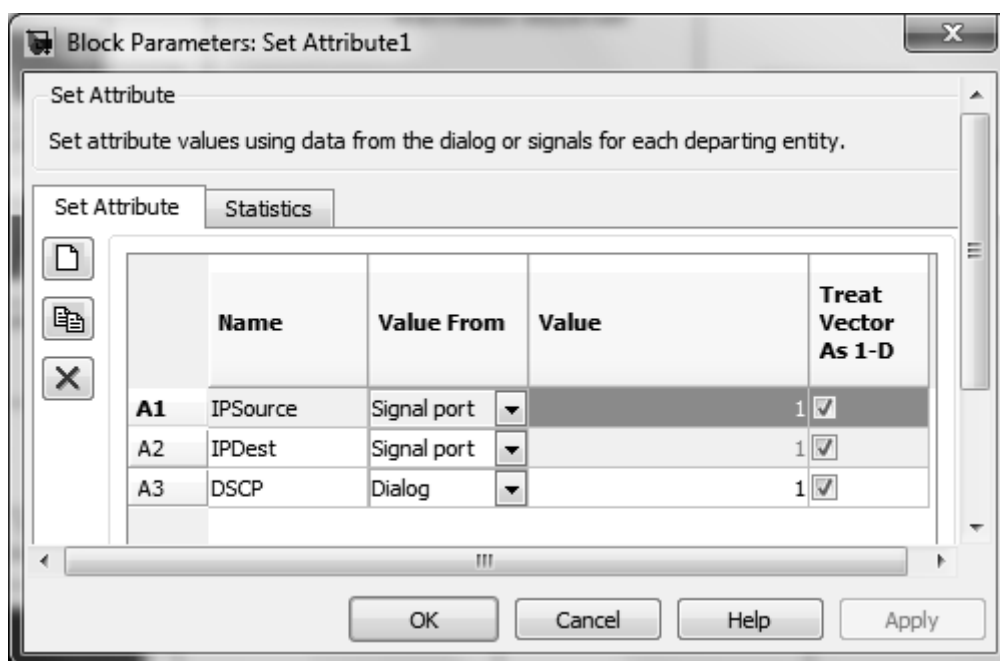


Fig. 7-24: Ventana de diálogo del bloque *Set Attribute1*. Los atributos IPSource e IPDest son señales generadas por los bloques *Event-Based Sequence4*, y *Event-Based Sequence5* respectivamente, mientras que el atributo DSCP se fija en 1 ya que todas las entidades de esta cadena pertenecen a *flows* con QoS UGS.

Estas entidades, al pasar por el bloque *Set Attribute1*, cuya ventana de diálogo se muestra en la Fig. 7-24, son marcadas por los atributos generados por los bloques *Event-Based Sequence4*, y *Event-Based Sequence5* los cuales toman desde el *workspace* de Matlab secuencias de atributos almacenados en los vectores IPScte y IPDcte, generados previamente por el *script*. El atributo de nombre DSCP se especifica un 1 directamente en el campo valor.

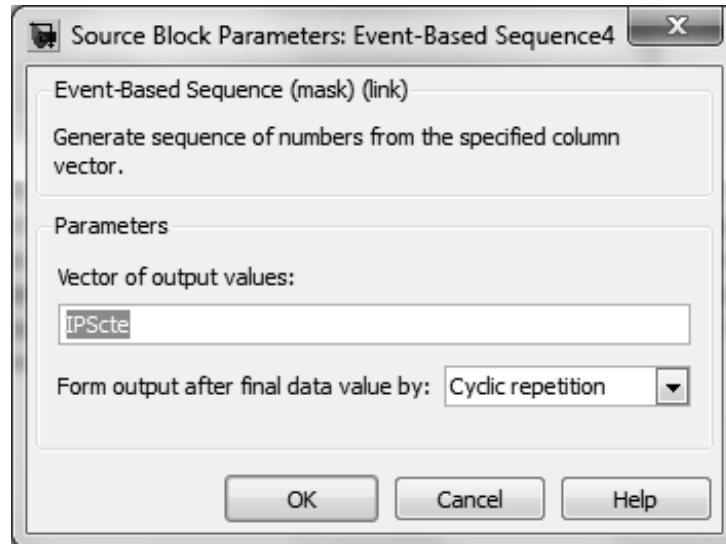


Fig. 7-25: Ventana de diálogo del bloque *Event-Based Sequence4* que genera una secuencia de números tomados del vector *IPScete* cargado desde el workspace de Matlab. El bloque *Event-Based Sequence5*, posee la misma ventana de diálogo salvo que el nombre del vector de donde se carga la secuencia de números es *IPDcte*

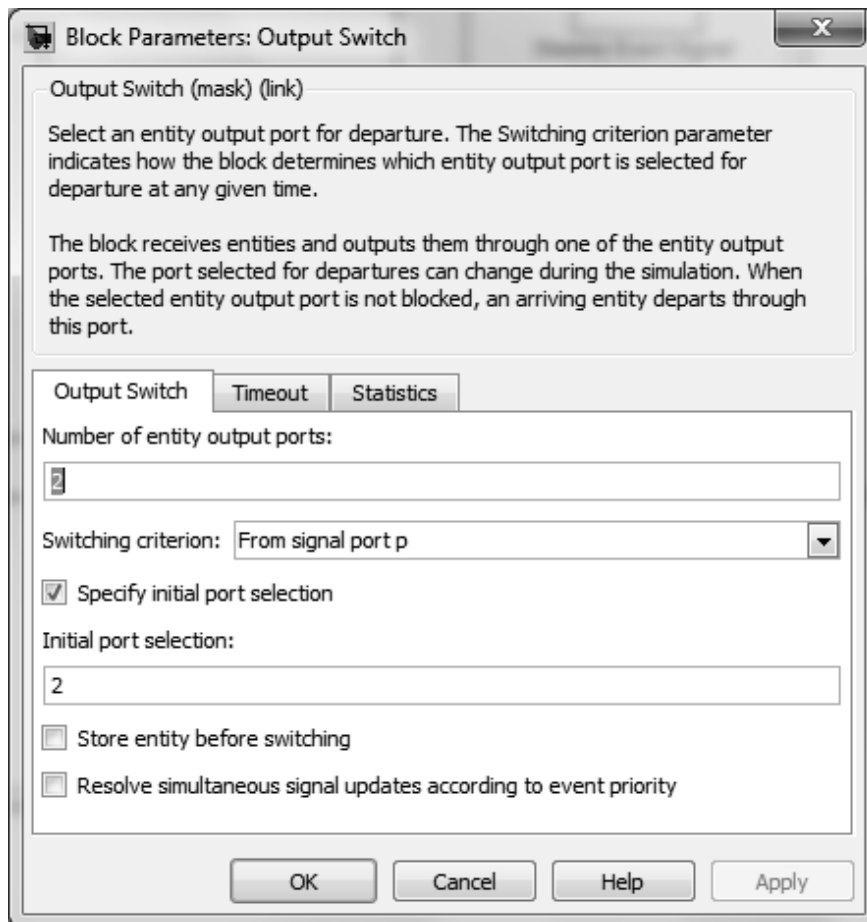


Fig. 7-26: Ventana de diálogo del bloque *Output Switch*. Este bloque conduce las entidades por el puerto de salida *OUT1* hacia el *Path Combiner* de la cadena superior si el puerto de entrada *P* tiene valor 1, si *P* tiene valor 2 las entidades son derivadas al puerto de salida *OUT2*.

En la Fig. 7-25 se muestra la ventana de diálogo del bloque *Event-Based Sequence4*. La ventana de diálogo del bloque *Event-Based Sequence5* es idéntica salvo que en el campo *Vector of output values* está escrito el nombre correspondiente al vector IPDcte.

El siguiente bloque que atraviesan las entidades de la cadena inferior, es el bloque *Output Switch*, cuya ventana de diálogo se muestra en la Fig. 7-26. Su función consiste en derivar todo el flujo de entidades al *Path Combiner* de la cadena superior, sumando al tráfico de la misma los paquetes con tiempo interarribo constante generados en la cadena inferior, si el valor de la entrada P vale 1. Si P vale 2 los paquetes con tiempo interarribo constante generados en la cadena inferior finalizan su existencia en el bloque *Entity Sink1* sin ninguna trascendencia.

El bloque *Manual Switch* conmuta de estado cuando se clikea sobre su ícono. Sus entradas están conectadas a los bloques *Constant* y *Constant1* que presentan en forma permanente los valores 1 y 2 respectivamente. La salida del bloque *Manual Switch* se conecta a la entrada P del bloque *Output Switch*.

7.3.6 Efecto del tráfico UGS en el comportamiento de la SSQ FIFO

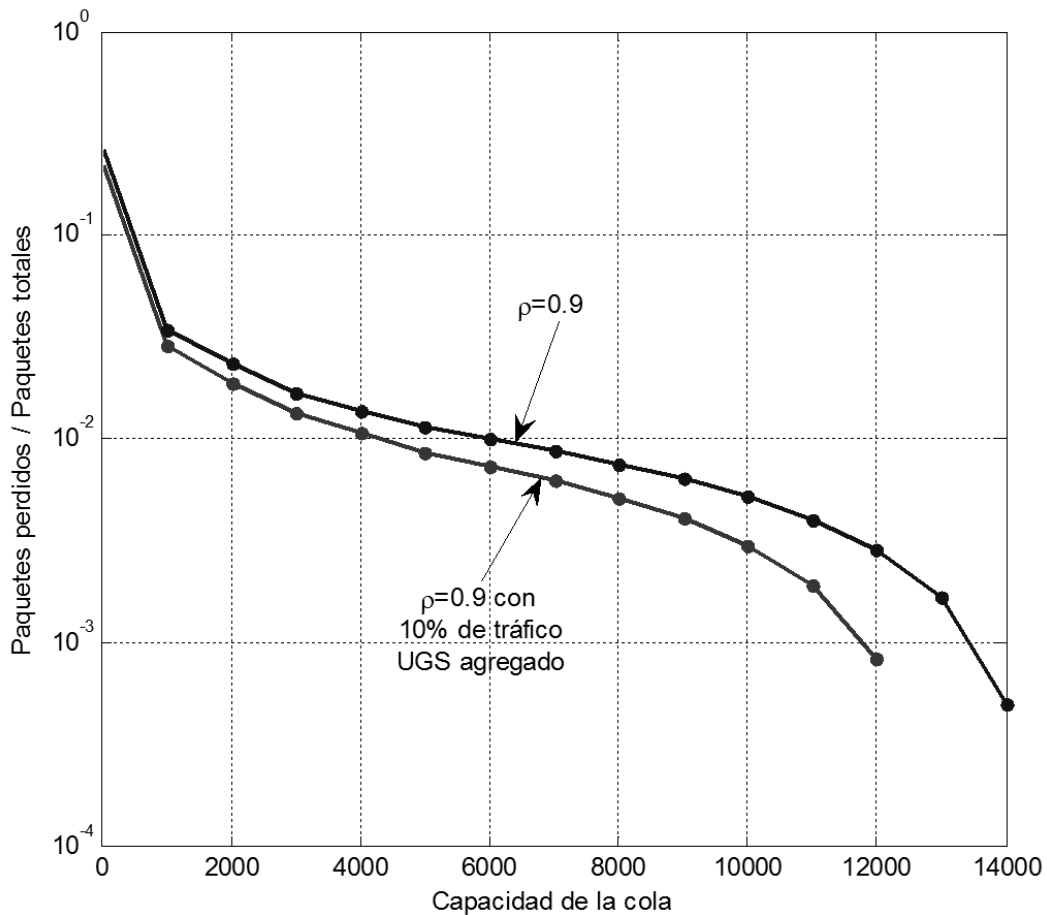


Fig. 7-27: Curvas de Paquetes perdidos/Paquetes arribados versus la Capacidad de la cola

Como parte de las pruebas para verificar que el modelo de Simulink para la asignación de atributos a los paquetes y generación de tráfico UGS funcione correctamente, se genera tráfico sin UGS agregado y con 10% de UGS agregado a partir de tráfico sin atributos generado por el método PMPP con valores $\alpha= 1,38$, $\mu= 0,18$, $\lambda_2/\lambda_1= 7,6$. Estos parámetros son uno de los que mejor aproximan el comportamiento del tráfico LBL-PKT4 al ser aplicado a una cola SSQ FIFO, Fig. 6-9.

En Fig. 7-27 se muestra el comportamiento de la SSQ FIFO del tráfico generado sin UGS y con el 10% de UGS, ambos para $\rho=0.9$. La curva superior es la misma que la correspondiente de la Fig. 6-9 como no podía ser de otra manera ya que el simple marcado de los paquetes con atributos no modifica el tráfico.

La curva inferior muestra el efecto de agregar 10% de tráfico UGS. Se aprecia que esto alivia los requerimientos de capacidad de la cola ya que la pérdida de paquetes es sensiblemente menor. Para una capacidad de la cola de 1000 paquetes, la pérdida es menor en un 24%, esta diferencia disminuye para capacidades de cola menores y aumenta para capacidades de cola mayores.

8. Conclusiones

8.1 Relativas a los objetivos planteados

El presente trabajo muestra que con herramientas simples, gráficas e intuitivas como Simulink de Matlab y en especial su *toolbox* SimEvents, es posible generar tráfico autosimilar, con características tales que reproduzcan el comportamiento de la cola de un sistema SSQ FIFO al ser alimentada por una muestra de tráfico real dado.

De los dos algoritmos generadores de tráfico analizados, el M/Pareto (Markov/Pareto) desarrollado por Addie, Neame y Zukerman (18) y el Pareto-*modulated* Poisson process (PMPP) desarrollado por Le-Ngoc y Subramanian (13), el segundo resultó ser más exacto en la reproducción del comportamiento de la cola, esto es la relación paquetes perdidos a paquetes totales versus capacidad de la cola, comparado con el del tráfico real que se pretende recrear. Además en el tráfico generado por el modelo PMPP el coeficiente de Hurst presenta un valor mucho más parecido al del tráfico de referencia.

Los modelos en Simulink para generar tráfico sintético, y el modelo para agregar atributos y tráfico UGS, para que el tráfico pueda ser útil para la prueba y desarrollo de *schedulers* de WiMAX, resultaron ser herramientas comprensibles, aún para el no especialista por su alto contenido gráfico, y lo suficientemente flexibles para permitir la realización de rápidas modificaciones o agregados, si así se requiriera.

8.2 Otros subproductos

Además del desarrollo de las herramientas nombradas, el desarrollo de la tesis permitió elaborar interesantes conclusiones conceptuales respecto al tráfico de red resultado de un análisis minucioso de la muestra de tráfico autosimilar de referencia LBL-PKT4.

8.2.1 Respecto a la autosimilaridad del tráfico

El análisis del tráfico de referencia mediante los *logscale diagrams* (LDs) en 2.2 y las curvas de las funciones de autocorrelación (ACFs) del tráfico en 2.3, muestra que los tráficos de la mayoría de las estaciones, considerados individualmente, sea enviado o recibido, presentan un comportamiento no autosimilar.

Se corroboró que la autosimilaridad en el tráfico de red es un fenómeno robusto, natural y persistente. La superposición del tráfico de pocas estaciones, cuyos tráficos no son autosimilares, construye uno que si lo es.

La regulación del tráfico ya sea por fenómenos como la congestión, el paso del tráfico por un cuello de botella como un enlace inalámbrico, o cualquier otro fenómeno saturante, destruye la autosimilaridad en las escalas pequeñas de tiempo, no así el comportamiento a largo plazo. Fig. 2-1.

8.2.2 Respetto a la *heavy-tailness* de la CCDF

La *heavy-tailness* o poca pendiente de la asíntota de la cola de la función de distribución acumulativa complementaria (CCDF), como ha sido demostrado por varios autores (12) (13) (14), es el ingrediente necesario que debe estar presente en las distribuciones de los tiempos interarribo de los tráficos individuales de las estaciones, para que la superposición de los mismos componga un tráfico total con características autosimilares.

El análisis de la muestra de referencia en 2.4.2, puso de manifiesto que el tráfico de la mayoría de las estaciones es *heavy-tailed*, independientemente si es enviado o recibido. También se observó que la distribución de los tiempos interarribo producto de la superposición de los tráficos de cada estación no es *heavy-tailed*, y menos *heavy* aún, cuanto mayor es el número de estaciones que lo componen.

De lo expresado en el párrafo anterior y el 8.2.1, se puede arriesgar a proponer la siguiente simplificación, que es al menos válida para la muestra de referencia: el tráfico total de muchas estaciones es autosimilar pero no es *heavy-tailed* mientras que el tráfico individual de cada estación es *heavy-tailed* pero no es autosimilar.

8.2.3 Efecto del agregado de un tráfico con tiempo interarribo constante a un tráfico autosimilar

El agregado de tráficos con tiempo interarribo constante entre paquetes, el que equivale salvando algunas diferencias a un CBR de ATM o a un UGS de WiMAX, a los tráficos individuales de las estaciones, que componen un tráfico autosimilar como el de la muestra LBL-PKT4, permite realizar las siguientes observaciones.

No modifica la pendiente de las LDs, ni la pendiente de las curvas variancia-agregación que son indicadores del nivel de autosimilaridad dados por el coeficiente de Hurst. Ver 3.5. Esta observación echa por tierra el concepto de que el dimensionamiento de una cola, para una probabilidad de pérdida porcentual de paquetes especificada, pueda realizarse considerando únicamente el *throughput*, el coeficiente de Hurst.

Se exageró el efecto agregando a la muestra de referencia un tráfico con tiempos interarribo que multiplicó por más de 13 veces la cantidad de paquetes, quedando

prácticamente el tráfico total reducido a uno con tiempos interarribo constante, sin embargo el coeficiente de Hurst del tráfico así compuesto no varió significativamente en ninguna escala. Sin embargo como se muestra en la Fig. 7-27 a igual intensidad de tráfico ρ , en el que tiene UGS agregado las pérdidas por desborde de la cola son significativamente menores.

Esto indica a las claras que para que un tráfico sintético produzca un determinado comportamiento en la curva de pérdida porcentual de paquetes versus capacidad de la cola, el porcentual de tráfico con tiempo interarribo constante debe ser considerado como un parámetro más para el dimensionamiento de la cola. Estas circunstancias fueron ya advertidas, siguiendo un camino diferente por Neame, Zukerman, etc. (14) y (18).

9. Trabajos a futuro

9.1 Análisis de la dispersión entre muestras de tráfico

Para poder construir modelos para la generación de tráfico no solo se debe tener en cuenta una única muestra como en este caso, que se empleó la LBL-PKT4. Ya que de una sola muestra no se puede inferir la variabilidad entre muestras tomadas repetidamente en condiciones similares. Este tema surge, cuando al correr los modelos aquí propuestos para la generación de tráfico, cada corrida brinda resultados diferentes, sección 6.3.3. Estas diferencias no están controladas ya que no se ha incluido en los modelos la información sobre la dispersión esperada entre las mismas.

Un trabajo a futuro relacionado con estas observaciones, sería la toma de muestras repetidas de tráfico en condiciones similares, el análisis de la dispersión entre ellas, y la inclusión en los modelos desarrollados de los parámetros que describen la dispersión entre las muestras.

9.2 Extracción del tráfico tipo CBR de una muestra

Este trabajo puso en evidencia la necesidad de manejar separadamente el tráfico con tiempos interarribo constante.

Un trabajo a futuro consiste en diseñar un programa para la extracción de tráfico CBR de una muestra real de tráfico. Disponiendo de esta información, se podrá modelar el tráfico para simulación como compuesto por un tráfico autosimilar más un tráfico con tiempos interarribo constante.

La palabra constante no debe ser interpretada literalmente, ya que un tráfico TCP/IP es esencialmente no constante, sin contar las alteraciones sufridas en las capas inferiores.

9.3 Modelos para sintetizar tráfico por aplicación

Diseño de generadores de tráfico por aplicación con características y porcentajes arbitrarios, por ejemplo, video online, bajadas de archivos, chat, *surfing*, etc.

Para la realización de este trabajo, primero es necesario la determinación, mediante el análisis de muestras, de la composición porcentual del tráfico discriminado por aplicación y por ámbito, universitario, hogareño, oficinas, etc.

10. Anexo 1 - Modelo de Simulink para analizar el comportamiento de una SSQ FIFO sometida a un tráfico dado

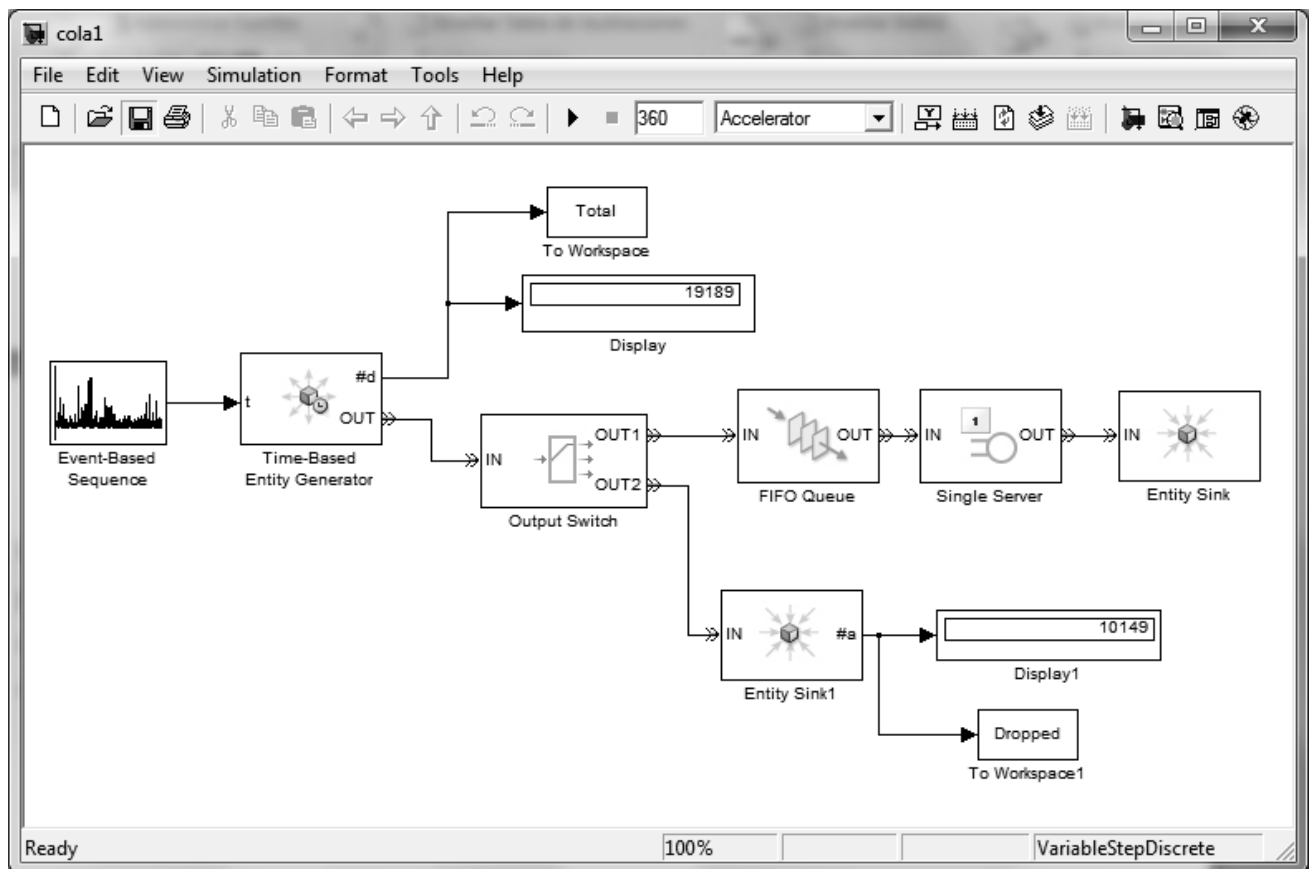


Fig. 10-1: Modelo en Simulink de un sistema SSQ FIFO.

En la Fig. 10-1 se muestra el modelo SSQ FIFO donde la mayoría de los bloques pertenecen al *toolbox* SimEvents de Simulink.

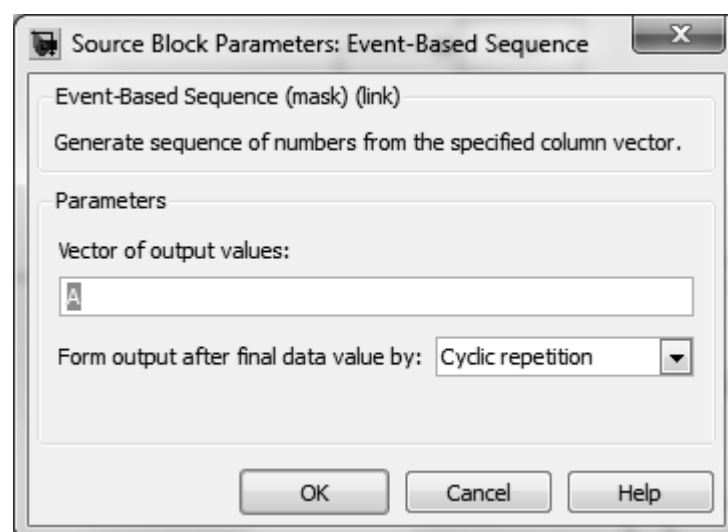


Fig. 10-2: Ventana de diálogo del bloque *Event-Based Sequence*.

El bloque *Event-Based Sequence* cuya ventana de diálogo se muestra en la Fig. 10-2 carga del *workspace* de Matlab un arreglo denominado A, correspondiente a una secuencia de tiempos interarribo de un tráfico real o sintético. El arreglo debe estar presente en el *workspace* antes de ejecutarse el modelo.

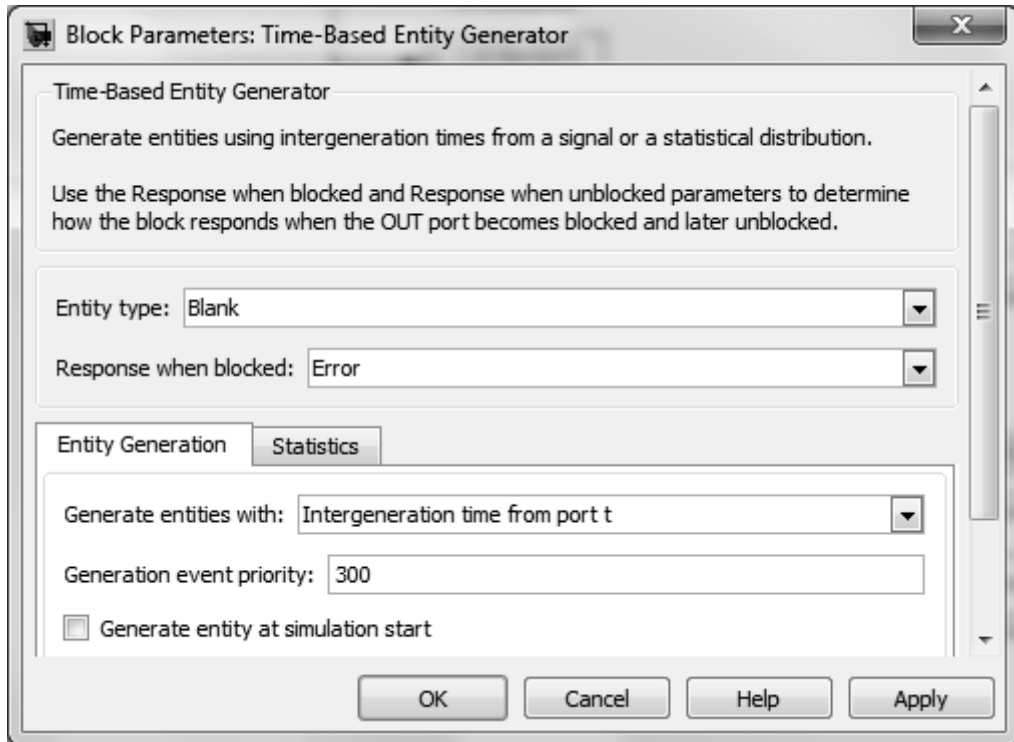


Fig. 10-3: Ventana de diálogo del bloque *Time-Based Entity Generator*.

Estos valores de intervalos de tiempo son puestos en la salida del bloque, la cual está conectada al *notifying port t* del bloque *Time-Based Entity Generator*, Fig. 10-3. Este último genera una nueva entidad cuando transcurre el tiempo indicado en *t* contado desde el instante de la generación de la entidad anterior, notificando al bloque predecesor *Event-Based Sequence* que le ofrezca un nuevo valor de tiempo *t*.

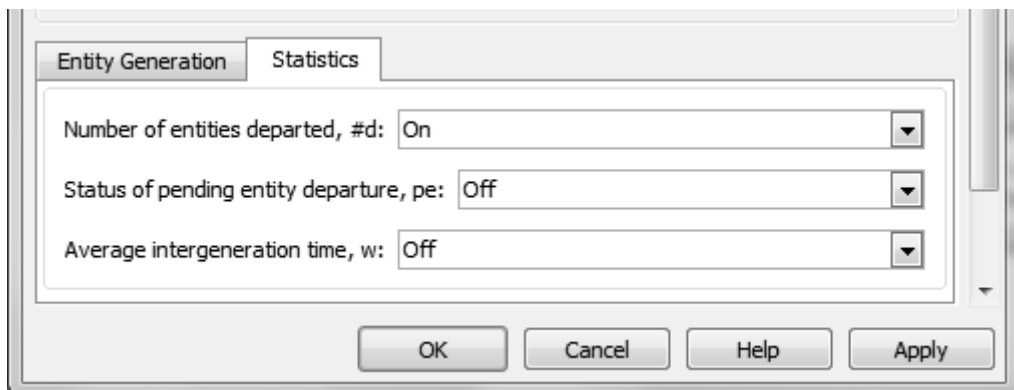


Fig. 10-4: Ventana de diálogo de la solapa *Statistics* del bloque *Time-Based Entity Generator*.

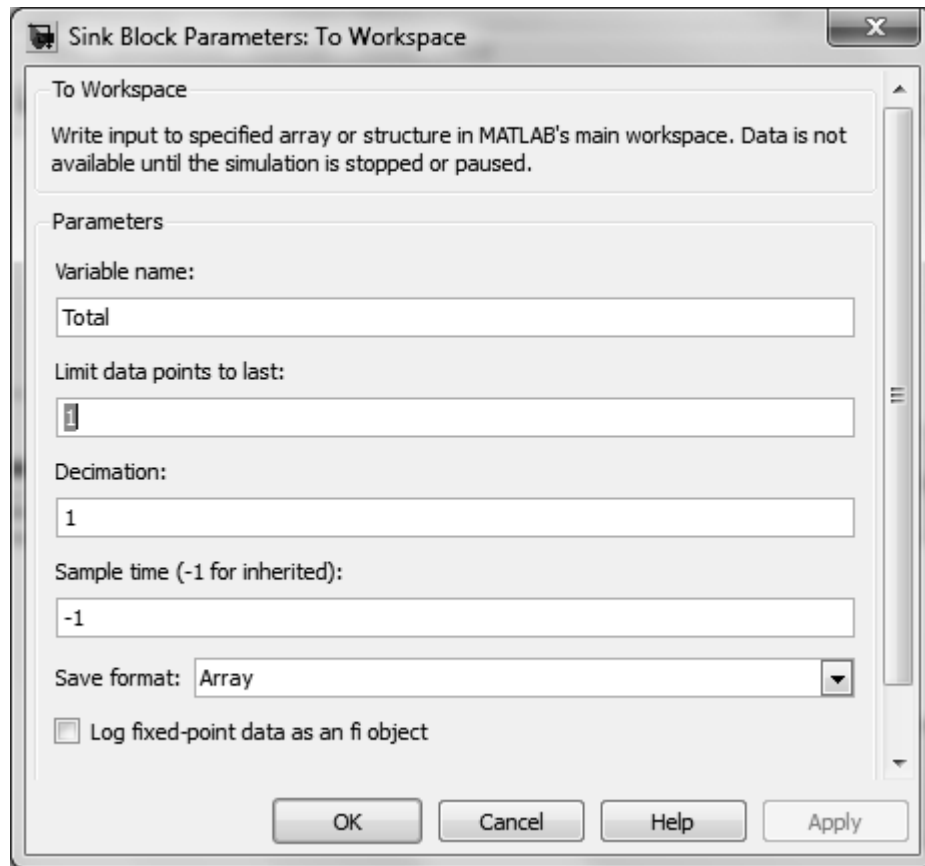


Fig. 10-5: Ventana de diálogo del bloque *To Workspace*.

En la solapa *Statistics* del bloque *Time-Based Entity Generator*, Fig. 10-4, se ha habilitado un puerto que informa el número de entidades egresadas, llamado #d, esta señal está conectada a dos bloques, el bloque *Display* que monitorea la cantidad de entidades mientras se corre el modelo y el bloque *To Workspace*, Fig. 10-5, que exporta al *workspace* en una variable denominada *Total*, la cantidad total de entidades generadas.

A continuación el flujo de entidades arriba al bloque *Output Switch*, Fig. 10-6. Si el bloque *Queue*, Fig. 10-7, que está aguas abajo en el puerto *OUT1*, no tiene llena su cola la entidad que arriba al *switch* es derivada hacia este último. Pero si la cola del bloque *Queue* está llena, la entidad es descartada hacia el puerto *OUT2* conectado al sumidero *Sink1* para su deposición final.

El bloque *Sink1* cuya ventana de diálogo se muestra en la Fig. 10-8 tiene habilitado el puerto #a, que informa la cantidad de entidades arribadas, esta salida está conectada al bloque *Display* para monitoreo visual cuando se ejecuta el modelo, y el bloque *To Workspace1* que almacena en la variable *Dropped* en el *workspace* de Matlab la misma cantidad, que queda disponible para ser usada para futuros cálculos.

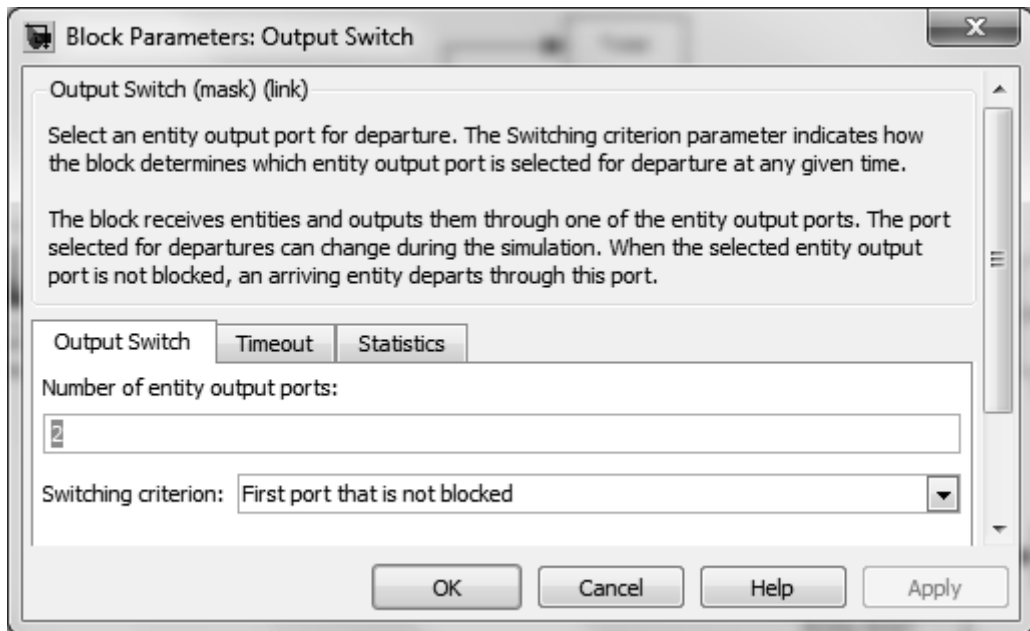


Fig. 10-6: Ventana de diálogo del bloque Output Switch

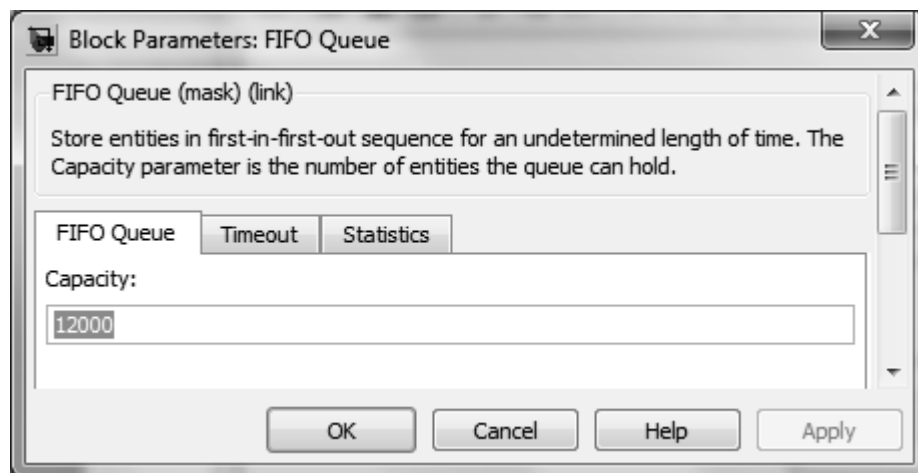


Fig. 10-7: Ventana de diálogo del bloque FIFO Queue.

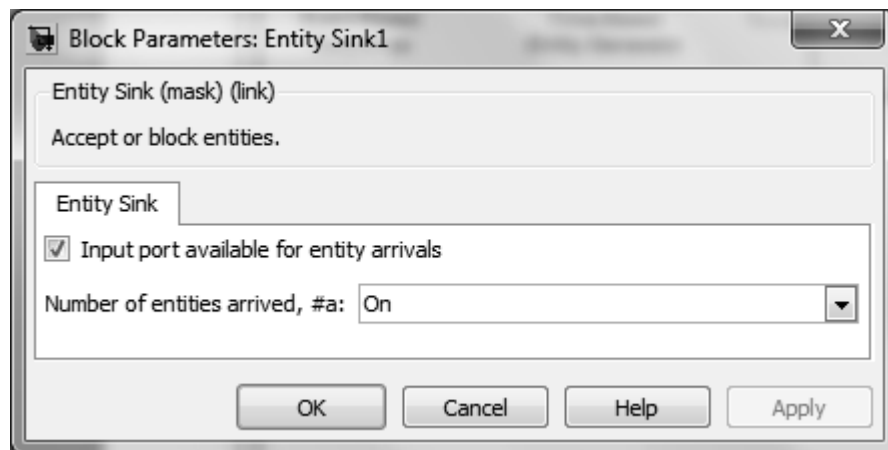


Fig. 10-8: Ventana de diálogo del bloque Sink1.

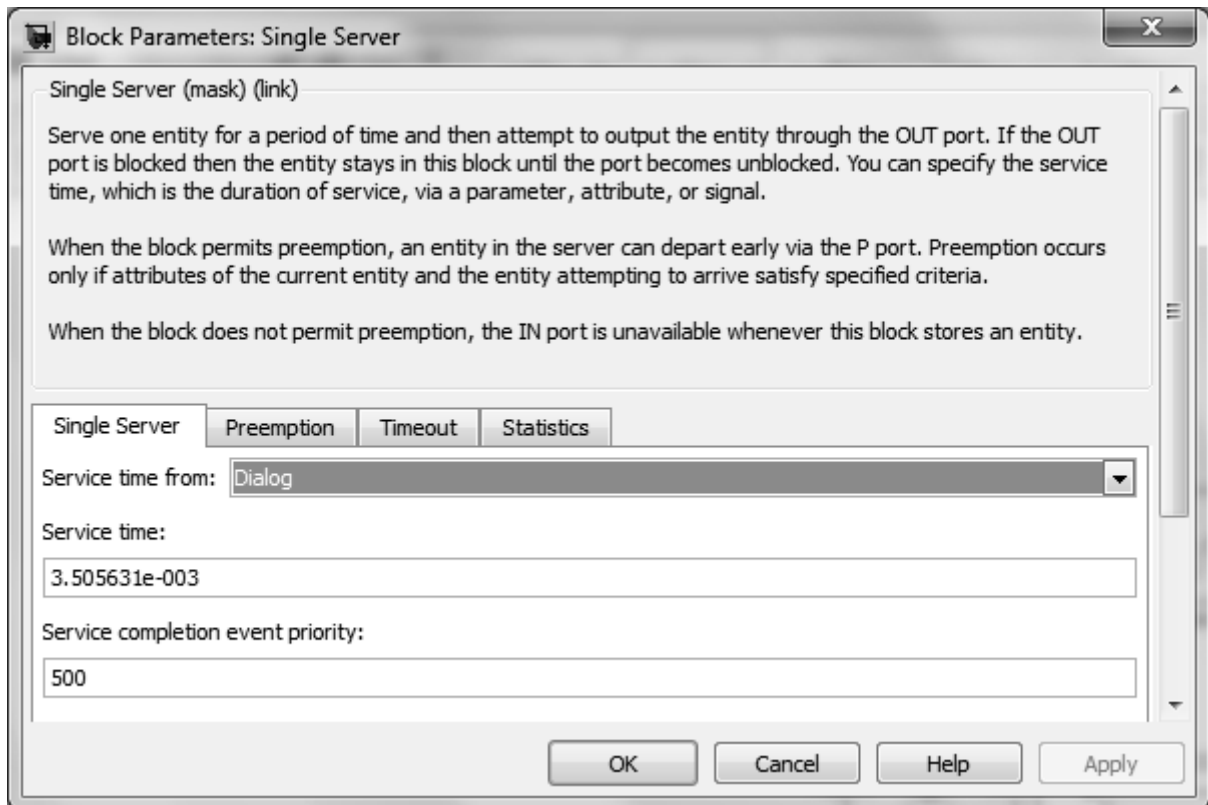


Fig. 10-9: Ventana de diálogo del bloque *Single Server*.

Continuando con el cauce principal de las entidades, si la cola del bloque *Queue* no está llena las entidades son derivadas al mismo por el bloque *Output Switch*. La salida del bloque *Queue* está conectada al bloque *Single Server*, donde el tiempo de servicio es establecido en el campo *Service time* de la ventana de diálogo del bloque, Fig. 10-9.

Finalmente una vez servidas las entidades terminan en el bloque sumidero *Sink*.

Este modelo, tal como fue usado en este trabajo, es configurado y ejecutado por un *script* de Matlab quien provee al mismo con un arreglo de los tiempos interarribo del tráfico a analizar que es tomado por el bloque *Event-Based Sequence*, y el tiempo de servicio por el bloque *Single Server*.

Posteriormente el *script* lo ejecuta repetidamente para una secuencia de valores de capacidad de la cola, la que está parametrizada en el bloque FIFO *Queue*. Finalmente grafica la relación Dropped/Total versus la capacidad de la cola, de la forma mostrada por ejemplo en la Fig. 7-27.

11. Anexo 2 – Programas utilitarios en Matlab

En este anexo se transcriben los programas utilitarios más relevantes realizados en Matlab.

11.1 Curva variancia-agregación

En esta sección se transcribe el *script* VarianciaAgregacionScript.m y sus funciones invocadas ArrivalTimeToAgregate.m, MAgregate.m, VarianciaAgregacion.m, y Pendientes.m. Permite graficar la curva variancia-agregación, de un tráfico dado descrito por un arreglo unidimensional de sus tiempos de arribo, en este caso simouttime.

La función Pendientes.m, transcripta en último lugar solicita al usuario que fije dos puntos con el *mouse* y traza una recta indicando su pendiente. Con el valor de la pendiente se puede calcular el coeficiente de Hurst H con la relación $H = 1 + \frac{Pen}{2}$. Ver Fig. 3-5 y el texto que la referencia en la sección 3.5.

```
%VarianciaAgregacionScript.m
```

```
%se pone en cuentas por intervalo de 50ms
%en simouttime, arreglo con tiempos de arribo del tráfico
ATTA=ArrivalTimeToAgregate(simouttime,0.050);
```

```
%Genera un Cell donde la primera columna es Agregate, k es fila,
%cada columna es Xk=(1/m)*(Xkm-m+ ....+Xkm), para k>=1
%los valores de m van por octavas, 1,2,4,8,16,
mAggregate=MAgregate(ATTA);
```

```
%cálculo de la variancia
VarianciaAgregacion(mAggregate);
```

```
Pendientes;
%H=1+Pend/2
```

```
function [Agregate] = ArrivalTimeToAgregate(Array,TimeInterval)
%Use: function [Agregate] = ArrivalTimeToAgregate(Array,TimeInterval)
%Tráfico descrito por tiempos interarribo los convierte
%a cantidad de arribos por intervalos Time
```

```
n=ceil(max(Array)/TimeInterval); tic;
for j=1:n
    y=find((Array >= ((j-1)*TimeInterval)) & (Array < (j*TimeInterval)));
    B(j,1)=length(y);
    %j
end
Agregate=B;
toc;

end
```

```

function [ MA ] = MAgregate(Agregate)
%Genera un Cell donde la primera columna es Agregate, k es fila,
%cada columna es  $X_k=(1/m)*(X_{k-m}+ \dots +X_k)$ , para  $k \geq 1$ 
%los valores de m van por octavas, 1, 2, 4, 8, 16,..
len=length(Agregate);
octavas=floor(log2(len));

for j=0:octavas
    m=2^j;
    k=1;
    for k=1:(floor(len/m))
        X{1,j+1}(k,1)=sum(Agregate(m*(k-1)+1:m*k))/m;
    end
    MA=X;
end

```

```

function []=VarianciaAgregacion(mAgregate)
%function []=VarianciaAgregacion(mAgregate)

```

```

%grafica variancia-agregación
len=length(mAgregate);
for j=1:len-3
    Var(j)=var(mAgregate{1,j});
end
m=2.^(0:len-4);
V=log10(Var);
M=log10(m);
plot(M,V);
xlabel('log(m)');
ylabel('log(variancias)');
grid on;

```

```

function []= Pendientes()
hold on;
X=[]; Y=[];
while true
    [x,y,button]=ginput(1);
    if button==3
        break;
    end
    plot(x,y,'x');
    X=[X;x]
    Y=[Y;y]
    if length(X)==2
        plot(X,Y);
        gtext(sprintf(strcat('Pen\n',num2str((Y(2)-Y(1))/(X(2)-
X(1)),'%4.1f'))));
        X=[]; Y=[];
    end
end
end
hold off;

```

11.2 IDC (*Index of Dispersion for Counts*)

En esta sección se transcribe el *script* IDCScript.m y la función IDC.m. Las funciones ArrivalTimeToAgregate.m, MAgregate.m, y Pendientes.m invocadas en el *script* están transcritas en 11.1.

Permite graficar la curva IDC (índice de dispersión por cuentas), de un tráfico dado descrito por un arreglo unidimensional de sus tiempos de arribo, en este caso simouttime. Ver sección 6.2.3.

```
%IDCScript.m
```

```
%se pone en cuentas por intervalo de 50ms
%en simouttime, arreglo con tiempos de arribo del tráfico
ATTA=ArrivalTimeToAgregate(simouttime,0.050);
```

```
%Genera un Cell donde la primera columna es Agregate, k es fila,
%cada columna es  $X_k=(1/m)*(X_{k-m}+ \dots +X_k)$ , para  $k \geq 1$ 
%los valores de m van por octavas, 1,2,4,8,16,
mAgregate=MAgregate(ATTA);
```

```
%grafica la curva IDC
IDC(mAgregate);
```

```
%permite aproximar la pendiente con una recta y muestra su valor
Pendientes;
```

```
function []=IDC(mAgregate)
%function []=IDC(mAgregate)
```

```
%cálculo de IDC
len=length(mAgregate);
m=2.^(0:len-4);
for j=1:len-3
    Var(j)=var(mAgregate{1,j});
    Mean(j)=mean(mAgregate{1,j});
    IxDC(j)=m(j)*Var(j)/Mean(j);
end
V=log10(IxDC);
M=log10(m);
plot(M,V);
xlabel('log(m)');
ylabel('log(IDC)');
grid on;
```


12. Anexo 3 – *Scripts* para simulación con el modelo M/Pareto

Se transcribe acá un *script* de Matlab denominado automata2.m, el cual manipula dos modelos de Simulink, MPareto3.mdl y cola1.mdl. Este *script* invoca la función paretogen3.m necesaria para generar la secuencia Pareto que requiere el modelo MPareto3.mdl. La función paretogen3.m se transcribe a continuación de automata2.m

automata2.m genera entre otros datos, la matriz Drop que almacena la información para que el *script* figuras.m, transcrito después de paretogen3.m, construya el diagrama paquetes perdidos a paquetes arribados versus la capacidad de la cola. En particular los *scripts* transcritos de automata2.m y figuras.m fueron configurados para generar las curvas de la Fig. 6-3. Generar la matriz Drop para construir esta figura insumió aproximadamente tres días completos de ejecución de automata2.m.

```
%automata2.m

%Script para la generación de tráfico sintético mediante el modelo M/Pareto
%Coordinadamente los tráficos sintetizados se aplican a una cola SSQ FIFO
%para estimar la cantidad de paquetes descartados para su posterior
%graficación.

%Las características del generador M/Pareto se ingresa como arreglos en
%las variables de la distribución
%Pareto: alfa y media
%Exponencial: lambda

%Los parámetros para el modelo de la cola que permite verificar el tráfico
%generado son
%Tiempo de verificación: StopTime
%Capacidades de la cola a evaluar: Capacity
%Intensidad de tráfico: Rho

%inicializaciones
clear; close; clc;
Pareto=sprintf('MPareto3');
Cola=sprintf('cola1');
open_system({Pareto,Cola});
StopTime=3600; %tiempo de ejecución del modelo cola1
Capacity=0:1000:15000; %Capacidades de la cola

alfa=[1.3 1.4]; %pareto
media=[4.3 4.5 4.7]; %pareto
lambda=[2 2.1 2.2]; %exponencial

Ro=[0.7 0.8 0.9 1]; %intensidad de tráfico
N=50000; %cantidad valores pareto a generar
n=896690; %cantidad de paquetes a generar

for ii=1:length(alfa)
```

```

for jj=1:length(media)
for kk=1:length(lambda)
    %se calcula cantidad media de paquetes por ráfaga
    r=n/3600/media(jj)/lambda(kk);

    %se genera eventos con distribución Pareto
    VectorPareto=paretogen3(alfa(ii),media(jj),N);

    %se abre y corre el modelo en Simulink
    set_param(Pareto,'StopTime',sprintf('%d',StopTime));
    set_param(Pareto,'SimulationCommand','start');

    %lazo para detectar si terminó Simulink
    Stop=false;
    while ~Stop
        pause(10);
        S=get_param(Pareto,'SimulationStatus');
        Stop=strcmp(S,'stopped');
    end

    %salva en archivo el tráfico generado
    nombreadchivo=sprintf('alfa%4.2fmedia%4.2f
        lambda%4.2f.mat',alfa(ii),media(jj),lambda(kk));
    simouttime=simout.time;
    save (nombreadchivo,'simouttime'); clear simouttime;
    fprintf('    %s\n',nombreadchivo);

    %Comportamiento de la cola con el tráfico generado
    A=TimeToInterarrival(simout.time);
    TiempoInterarriboPromedio=mean(A);
    set_param(Cola,'StopTime',sprintf('%d',StopTime));
    ServiceTime=Ro*TiempoInterarriboPromedio;
    for ll=1:length(ServiceTime)
        set_param(strcat(Cola,'/SingleServer'),
            'ServiceTime',sprintf('%d',ServiceTime(ll)));
        for mm=1:length(Capacity)
            tic;
            set_param(strcat(Cola,'/FIFO Queue'),'Capacity',
                sprintf('%d',Capacity(mm)));
            set_param(Cola,'SimulationCommand','stop');
            set_param(Cola,'SimulationCommand','start');

            %este lazo es para detectar si terminó Simulink
            Stop=false;
            while ~Stop
                pause(10);
                S=get_param(Cola,'SimulationStatus');
                Stop=strcmp(S,'stopped');
            end

            %muestra en pantalla
            fprintf('Tot %d, Drop %d, ServTime %1.5f,Cap %d,
                tic-toc %3.1f\n',...
                    Total,Dropped,ServiceTime(ll),Capacity(mm),toc);

            %calcula en tanto por uno los paquetes descartados
            Drop(ii,jj,kk,ll,mm)=Dropped/Total;

            %evita ejecutar el modelo si Dropped comenzó a ser 0

```

```

        if Dropped==0
            Drop(ii,jj,kk,ll,mm:length(Capacity))=0;
            mm=length(Capacity);
            break;
        end
    end
end
end
%salva en archivo el tanto por uno de los paquetes perdidos
save('Drop1.mat','Drop');

    end %lambda
end %media
end %alfa

close_system(Pareto,0);
close_system(Cola,0);
save Workspacel;

```

```

function [ R ] = paretogen3( alfa, media, N )
%function [ R ] = paretogen3( alfa, media, N)
% genera un vector fila de longitud: N
% ccdf con pendiente: -alfa
% valor medio: media
theta=media*((alfa-1)/alfa);
k=1/alfa;
sigma=k*theta;
R=gprnd(k, sigma, theta, N,1);
end

```

```

%figuras.m

```

```

%Script para graficar el comportamiento de una colas SSQ FIFO
%En ordenadas: Paquetes Perdidos / Paquetes totales
%En abcisa: Capacidad de la cola

```

```

%Drop(alfa,media,lambda,ServiceTime,Capacity)
%Drop(ii,jj,kk,ll,mm)

```

```

alfa=[1.3 1.4]; %pareto
media=[4.3 4.5 4.7]; %pareto
lambda=[2 2.1 2.2]; %exponencial
ServiceTime=0.00372895579514714; %Ro=0.9
Capacity=0:1000:15000; %Capacidades de la cola

```

```

for ii=1:length(alfa)
    for jj=1:length(media)
        for kk=1:length(lambda)
            for ll=1:length(ServiceTime)
                A=Squeeze(Drop(ii,jj,kk,ll,:));
                semilogy(Capacity,A);
                S=sprintf('%1.1f\\%1.1f\\%1.1f',...
                    alfa(ii),media(jj),lambda(kk));
                gtext(S);
            end
        end
    end
end

```

```
                hold on;
                end %ServiceTime
            end %lambda
        end %media
    end %alfa

xlabel('Capacidad de la cola');
ylabel('Paquetes perdidos / Paquetes totales');
grid on;
```

13. Anexo 4 – *Scripts* para simulación con el modelo PMPP

Se transcribe acá un *script* de Matlab denominado autómeta PMPP3.m, el cual manipula dos modelos de Simulink, BatchPMPP2.mdl y cola1.mdl. Este *script* invoca la función pareto4.m, transcripto a continuación, la cual difiere de la usada para la simulación con el modelo MPareto.mdl en que cada vez que es invocada reinicializa el generador pseudo aleatorio de Matlab. Como se comentó en 6.3.3 y en 9.1, al ser este modelo notablemente más sensible a la dispersión por repeticiones que el M/Pareto, esta acción evita el efecto.

autómataPMPP3.m genera entre otros datos, la matriz Drop que almacena la información para el *script* figuras2.m, transcripto después de pareto4.m, construya el diagrama paquetes perdidos a paquetes arribados versus la capacidad de la cola. En particular los *scripts* transcriptos de automataPMPPP3.m y figuras2.m fueron configurados para generar las curvas de la Fig. 6-8.

```
%automataPMPP3.m

%Script para la generación de tráfico sintético mediante el modelo PMPP
%Coordinadamente los tráficos sintetizados se aplican a una cola SSQ FIFO
%para estimar la cantidad de paquetes descartados para su posterior
%graficación.

%Las características del generador PMPP se ingresa como arreglos en
%las variables de la distribución
%Pareto: alfa y media
%Exponencial: lambdaR (lambda2/lambda1), lambdaProm se calcula de la
%cantidad de paquetes a generar

%Los parámetros para el modelo de la cola que permite verificar el tráfico
%generado son
%Tiempo de verificación: StopTime
%Capacidades de la cola a evaluar: Capacity
%Intensidad de tráfico: Ro

%version3, se incluye repeticiones nn, para analizar la dispersión entre
%corridas.

%inicializaciones
clear, close, clc;
Pareto=sprintf('BatchPMPP2');
Cola=sprintf('cola1');
open_system({Pareto,Cola});

StopTime=3600;
Capacity=0:1000:15000; %Capacidades de la cola
alfa=[1.34 1.38 1.42]; %pareto
mu=[0.16 0.18 0.20]; %pareto media
Ro=[0.9];
Repeticiones=1;
```

```

lambdaR=[7.2 7.4 7.6]; %lambda2/lambda1
N=20000; %cantidad valores pareto a generar
n=896690; %cantidad de paquetes a generar
lambdaProm=n/StopTime;

for nn=1:Repeticiones
    for ii=1:length(alfa)
        for jj=1:length(mu)
            VectorPareto=paretogen4(alfa(ii),mu(jj),N);

            for kk=1:length(lambdaR)
                %se calculan los lambda de la distribución exponencial
                lambda1=2*lambdaProm./(1+lambdaR(kk));
                lambda2=lambda1.*lambdaR(kk);

                %se abre y corre el modelo en Simulink
                set_param(Pareto,'StopTime',sprintf('%d',StopTime));
                set_param(Pareto,'SimulationCommand','start');

                %lazo para detectar si terminó Simulink
                Stop=false;
                while ~Stop
                    pause(10);
                    S=get_param(Pareto,'SimulationStatus');
                    Stop=strcmp(S,'stopped');
                end

                %salva en archivo el tráfico generado
                nombreadchivo=sprintf('Rep%dalfa%4.2fmu%4.2f
                    lambdaR%4.2f.mat',nn,alfa(ii),mu(jj),lambdaR(kk));
                simouttime=simout.time;
                save (nombreadchivo,'simouttime'); clear simouttime;
                fprintf('    %s\n',nombreadchivo);

                %Comportamiento de la cola con el tráfico generado
                A=TimeToInterarrival(simout.time);
                TiempoInterarriboPromedio=mean(A);
                set_param(Cola,'StopTime',sprintf('%d',StopTime));
                ServiceTime=Ro*TiempoInterarriboPromedio;
                for ll=1:length(ServiceTime)
                    set_param(strcat(Cola,'/SingleServer'),
                        'ServiceTime',sprintf('%d',ServiceTime(ll)));
                    for mm=1:length(Capacity)
                        tic;
                        set_param(strcat(Cola,'/FIFOQueue'),
                            'Capacity',sprintf('%d',Capacity(mm)));
                        set_param(Cola,'SimulationCommand','stop');
                        set_param(Cola,'SimulationCommand','start');

                        %este lazo es para detectar si terminó Simulink
                        Stop=false;
                        while ~Stop
                            pause(10);
                            S=get_param(Cola,'SimulationStatus');
                            Stop=strcmp(S,'stopped');
                        end

                        %muestra en pantalla
                        fprintf('Tot %d, Drop %d, ServTime %1.5f, Cap %d,
                            Rep %d,tic-toc %3.1f\n',Total,Dropped,

```

```

        ServiceTime(ll),Capacity(mm),nn,toc);
%calcula en tanto por uno los paquetes descartados
Drop(nn,ii,jj,kk,ll,mm)=Dropped/Total;

%evita ejecutar el modelo si Dropped vale 0
if Dropped==0
    Drop(nn,ii,jj,kk,ll,mm:length(Capacity))=0;
    mm=length(Capacity);
    break;
end
end
end
%salva en archivo el tanto por uno de los paquetes perdidos
save('Drop1.mat','Drop');

end %lambda
end %mu
end %alfa
end %repeticiones

close_system(Pareto,0);
close_system(Cola,0);
save Workspace1;

```

```

function [ R ] = paretogen4( alfa, mu, N )
%function [ R ] = paretogen3( alfa, mu, N)
% genera un vector fila de longitud: N
% ccdf con pendiente: -alfa
% valor medio: mu
% con reset del generador
s=RandStream('mt19937ar');
RandStream.setDefaultStream(s);
theta=mu*((alfa-1)/alfa);
k=1/alfa;
sigma=k*theta;
R=gprnd(k, sigma, theta, N,1);
end

```

```

%figuras2.m

```

```

%Script para graficar el comportamiento de una cola SSQ FIFO
%En ordenadas: Paquetes Perdidos / Paquetes totales
%En abcisa: Capacidad de la cola

```

```

%Drop(repeticion,alfa,media,lambda,ServiceTime,Capacity)
%Drop(nn,ii,jj,kk,ll,mm)

```

```

Repeticiones=1;
alfa=[1.34 1.38 1.42]; %pareto
mu=[0.16 0.18 0.20]; %media pareto
lambdaR=[7.2 7.4 7.6]; %exponencial
Rho=[0.9];
%Capacity=0:1000:15000; %Capacidades de la cola

```

```

Capacity=[0:1000:15000];
for nn=1:Repeticiones
    for ii=1:length(alfa)
        for jj=1:length(mu)
            for kk=1:length(lambdaR)
                for ll=1:length(Rho)
                    A=Squeeze(Drop(nn,ii,jj,kk,ll,:));
                    semilogy(Capacity,A);
                    S=sprintf('%0.3g\\%0.3g\\%0.3g\\%0.3g',...
                        alfa(ii),mu(jj),lambdaR(kk),Rho(ll));
                    gtext(S);
                    hold on;
                end %ServiceTime
            end %lambdaR
        end %media
    end %alfa
end %repeticiones

xlabel('Capacidad de la cola');
ylabel('Paquetes perdidos / Paquetes totales');
grid on;

```


14. Anexo 5 – Script para generar tráfico para WiMAX

Como se detalló en 7, el modelo de Simulink para la asignación de atributos a los paquetes y generación de tráfico UGS, carga las secuencias de atributos que han sido previamente generadas por un *script* en el *workspace* de Matlab

Este *script* se denomina *script1.m*. El mismo invoca los programas *Atributos.m*, *AtributosCte.m*, y *AtributosLongPaq.m* los cuales generan arreglos unidimensionales con secuencias de atributos, basados en los valores consignados por el usuario en los arreglos de configuración *At*, y *AtCte*. Tabla 14-1.

Tabla 14-1: Arreglos con secuencia de atributos generados por programas invocados por *script1.m*

Programa	Atributo	Arreglo	Atributo	Arreglo	Atributo	Arreglo
<i>Atributos.m</i>	IP origen	IPS	IP destino	IPD	DSCP	SC
<i>AtributosCte.m</i>	IP origen	IPScte	IP destino	IPDcte		
<i>AtributosLongPaq.m</i>	Longitud de paquetes	LongPaq				

En *script1.m* también se define el intervalo interarribo entre paquetes del tráfico UGS, generando la variable *InterArrivalTimeCte* y el tiempo de simulación del modelo en la variable *StopTime*.

El *script1.m* se encarga de abrir, correr y cerrar el modelo de Simulink *Info3.mdl*. Finalmente con los datos generados por el modelo, disponibles en el *workspace*, *script1.m* los combina en un único arreglo.

A continuación se transcriben *script1.m*, *Atributos.m*, *AtributosCte.m* y *AtributosLongPaq.m*. Al final se agrega la función *anyrnd1.m* modificación de la original *anyrnd1.m* (24).

```
%script1.m
```

```
%script generador de atributos para
% Trafico con tiempo interarribo arbitrario
% Trafico con tiempo interarribo constante
```

```
%porcentaje trafico interarribo constante
PTIC=10; %editar porcentaje
```

```
clear, close, clc;
```

```
%para obtener los tiempos interarribo del tráfico generado por el
%simulador PMPP con valores alfa= 1.38, mu= 0.18, y lambdaR= 7.60
load Replalfa1.38mu0.18lambdaR7.60.mat
TiemposInterarribo=TimeToInterarrival(simouttime);
```

```

display('Generando Atributos para trafico con tiempo interarribo
arbitrario');
Atributos;
display('Generando Atributos para tráfico con tiempo interarribo
constante');
AtributosCte;
display('Generando Atributos Tamaño de paquete (carga util TCP)');
AtributosLongPaq;

%paquetes para tráfico con tiempo interarribo arbitrario
PaquetesCant=length(TiemposInterarribo);
%paquetes para tráfico con tiempo interarribo constante
PaquetesCteCant=PTIC/100*PaquetesCant; % 10% de trafico con interarribo
cte.
InterArrivalTimeCte=3600/PaquetesCteCant; %para Simulink
StopTime=3600; %Tiempo de ejecución del modelo
Info=sprintf('Info3');

open_system(Info);
set_param(Info,'StopTime',sprintf('%d',StopTime));
set_param(Info,'SimulationCommand','start');

%este lazo es para detectar si terminó Simulink
Stop=false;
while ~Stop
    pause(10);
    S=get_param(Info,'SimulationStatus');
    Stop=strcmp(S,'stopped');
end
close_system(Info,0); %cierra Simulink

%ensambla archivo de tráfico
TraficoWiMAX=[Time.time ,IPSource, IPDest,DSCP,Len];
save TraficoWiMAX TraficoWiMAX; %guarda trafico sintético en un archivo
save Workspace; %guarda workspace en un archivo

```

```

%Atributos

```

```

%Generador de atributos para tráfico con tiempos interarribo arbitrarios
%Entrada: cell de configuración At
%Salida:
%   IP origen: IPS
%   IP destino: IPD
%   Differential Service Code Point (o TOS): DSCP

tic;
load At;
N=100000;

for ii=1:N;
    %IP origen
    [r,index]=anyrnd1([[At{1,:}]],[At{2,:}]'],1,1);
    IPS(ii)=r;
    %IP destino
    [r,index1]=anyrnd1([[At{3,index}],[At{4,index}]]',1,1);
    IPD(ii)=r;

```

```

    %clase de servicio
    r=anyrnd1([[1 2 3 4 5]','At{index1+4,index}''],1,1);
    SC(ii)=r;
end
toc;
gong;

```

```

%AtributosCte.m

```

```

%Generador de atributos para tráfico con tiempo interarribo constante
%Entrada: cell de configuración AtCte
%Salida:
%   IP origen: IPScte
%   IP destino: IPDcte

tic;
load AtCte;
N=50000;

for ii=1:N;
    %IP origen
    [r,index]=anyrnd1([[AtCte{1,:}'],'[AtCte{2,:}']'],1,1);
    IPScte(ii)=r;
    %IP destino
    [r,index1]=anyrnd1([[AtCte{3,index}'],'[AtCte{4,index}']'],1,1);
    IPDcte(ii)=r;
end
toc;
gong;

```

```

%AtributosLongPaq.m

```

```

%Generador de atributos longitud de paquete
%Entrada: Array AtPacketSize
%Salida: secuencia LongPaq

tic;
%Histograma 1ra columna: long en bytes, 2da columna: ocurrencias
load PacketLen; %Campo len de los paquetes de LBL-PKT4
x = min(PacketLen):1:max(PacketLen);
[a,b]=hist(PacketLen,x);
Histo=[b','a'];

% %Se muestra el histograma
% stairs(Histo(:,1),Histo(:,2));

%se genera una secuencia randómica arbitraria discreta
%en array columna LongPaq se almacenan la longitud de los paquetes
N=100000;
[LongPaq,index]=anyrnd1([Histo(:,1),Histo(:,2)],1,N);
toc;
gong;

```

```

function [r,index]= anyrnd1(f,M,N)
%[r,index]= ANYRND(F,M,N) returns a matrix r of random numbers of size M by
N
% generated from an user-defined frequency distribution F. F can be
% specified as a 2-column vector where first column are the centers
% of bins and second column are frequencies. F can also be a 1-column
% vector of values, in which case a frequency distribution is generated
% with HIST using 100 bins. It works better the more detailed is the
% distribution (obviously).
%
% Example:
% x= linspace(0,10*6.28,1000)';
% y= 10*sin(x);
% [n,v]= hist(y,100); %Generate freq. dist.
% f= [v',n']; %Put it in f
% r= anyrnd(f,1000,1); %Call anyrnd
% hist(y,50);hold on; %Plot both dist.
% hist(r,50);
%
% Author: F. de Castro
% Modificado por D. Rabinovich 26/8/2010
% returns a matrix index with the corresponding file of f (26/8/10)

%-- Generate freq. dist. if needed
if size(f,2) == 2
    v= f(:,1); n= f(:,2);
else
    [n,v]= hist(f(:,1),100);
end

%-- Initialize
r= zeros(M,N);
fcum= zeros(length(f),1);
s= sum(n);
maxf= max(f);

%-- Cummulative frequencies
fcum(1)= n(1)/s;
for j= 2:length(n)
    fcum(j)= fcum(j-1)+ n(j)/s;
end

%-- Random picking
for k= 1:N
    for j= 1:M
        loe= find(fcum >= rand);
        r(j,k)= v(loe(1));
        index(j,k)=loe(1);
    end
end
end

```

15. Anexo 6 – Reseña de WiMAX

15.1 Introducción

La siguiente reseña intenta ser de carácter ilustrativo, ya que un ataque con mayor profundidad llevaría mucho espacio. Se basa en un corto pero efectivo tutorial de la revista *Teletronik* (25) escrita al comienzo del año 2010 que ya contempla la unificación de las normas de IEEE 802.16 realizada a fines del año 2009 en la IEEE 802.16-2009 (1). No se incluye las referencias bibliográficas del tutorial.

WiMAX incorpora casi todos los nuevos avances en telecomunicaciones, por lo cual su capa física y parte de la capa MAC (*Media Access Control layer*) son sumamente complejas, por lo cual un análisis detallado está totalmente fuera del alcance de esta tesis. De hecho solo el estándar principal del grupo IEEE 802.16, un documento franciscano por su austeridad, tiene más de 2000 páginas, y a eso hay que sumarle un voluminoso corpus de especificaciones técnicas aportado por el *WiMAX Forum* que comprende más de 20 documentos activos.

WiMAX es una tecnología de acceso inalámbrico para estaciones móviles y fijas, basado en el estándar IEEE 802.16. Tiene su origen en la industria de la computación y es una alternativa a tecnologías 3GPP (*3rd Generation Partnership Project*) tales como la HSPA (*High Speed Packet Access*) y LTE (*Long Term Evolution*).

15.2 Historia

En 1999 el IEEE *Standards Board* constituye un grupo de trabajo llamado IEEE 802.16 con el fin de desarrollar estándares para la implementación global de Redes de Área Metropolitana Inalámbricas. El primer estándar fue publicado en 2001 y en el mismo se especificó la MAC y la PHY (*Physical Layer*) de un sistema enfocado en las bandas de frecuencia de 10 a 66 GHz, bautizada IEEE 802.16-2001.

Era un sistema punto multipunto (PMP) con soporte opcional para tecnología MESH. Se requería línea de vista (LOS, *Line of Sight*) o casi LOS para la comunicación entre la estación base y las antenas de las terminales de usuarios, además las antenas de los usuarios debían ser direccionales. La PHY usaba portadora única y el ancho de banda del canal era típicamente 28MHz.

Mucha de la gente que trabajó en la fase inicial del estándar provenía de la industria de la televisión por cable. El más avanzado estándar de cable en ese momento era DOCSIS (*Data Over Cable Service Interface Specification*), y DOCSIS MAC fue elegida como modelo base para la MAC del sistema inalámbrico.

En 2001, el grupo IEEE 802.16 comenzó a desarrollar enmiendas a la norma que incluían modificaciones a la MAC y especificaciones adicionales a la PHY para la operación en las bandas de frecuencias entre 2 y 11 GHz. La enmienda fue aprobada en enero de 2003 y dándosele el nombre de IEEE 802.16a-2003.

La enmienda especificaba tres PHY adicionales, una con portadora única (SCa, *Single Carrier*), y dos alternativas multiportadoras denominadas OFDM (*Orthogonal Frequency Division Multiplexing*) y OFDMA (*Orthogonal Frequency Division Multiple Access*). Los sistemas basados en la enmienda 16a podrían operar en bandas de frecuencia licenciadas y no licenciadas.

El *WiMAX Forum*, se enfocó en las frecuencias altas especificadas de la PHY del estándar original, pero no tuvo el crecimiento y éxito que las empresas fundadoras habían esperado debido al escaso interés del mercado por equipos destinados a operar en frecuencias tan elevadas.

En Abril de 2003, Intel junto con otras empresas de la industria WLAN como Proxim, Fujitsu Microelectronics y Wi-LAN se unió al *WiMAX Forum* y se hicieron cargo de la dirección de la organización. Con la adquisición o el reinicio del foro, se modificó el enfoque desde las elevadas frecuencias originales de 10 a 66GHz a las nuevas opciones para la PHY de 2 a 11GHz.

Reorientando WiMAX a bandas de frecuencia más baja, los costos de los equipos resultarían substancialmente menores y los sistemas podían operar en condiciones de no LOS. Pero todavía era un sistema inalámbrico fijo, cuya aplicación principal era ser una alternativa o complemento de ADSL.

En 2003, IEEE inició una revisión completa de la norma, que finalizó en el siguiente año dando lugar a una nueva versión de la norma denominada IEEE 802.16-2004. Uno de los cambios fue que el rango operativo de frecuencias para la PHY que se modificó para incluir bandas de frecuencias por debajo de 11GHz.

En diciembre de 2005, se aprobó una nueva enmienda a la norma denominada IEEE 802.16e-2005. Incluía mejoras en la MAC y OFDMA en la PHY para soportar operación móvil. Los sistemas basados en esta enmienda y se denominaron *Mobile WiMAX* por el *WiMAX Forum*, ya que estos sistemas fueron dirigidos principalmente a aplicaciones nómadas y móviles como una alternativa, por ejemplo, para UMTS/HSPA (*Universal Mobile Telecommunications System/High Speed Packet Access*).

En 2009 se publicó una nueva revisión de la norma denominada IEEE 802.16-2009 que incluyó la enmienda 16e y algunas otras modificaciones y correcciones. Así que desde 1999

y hasta hoy, WiMAX ha sufrido grandes mutaciones. Desarrollada a partir de una red inalámbrica de cable a un sistema móvil de banda ancha. Pero aunque la aplicación ha cambiado dramáticamente, muchas de las soluciones utilizadas en WiMAX tienen su origen en el primer estándar de WiMAX, donde la capa MAC sigue siendo muy similar a la DOCSIS.

15.2.1 El WiMAX Forum

El WiMAX *Forum* es una organización industrial formada para promover la adopción de productos y servicios WiMAX compatibles. Los miembros incluyen a fabricantes de chips, proveedores de equipos, operadores y otros.

El WiMAX *Forum* desempeña un papel similar para el desarrollo de sistemas basados en el estándar IEEE 802.16 al que la *Wi-Fi Alliance* juega para los sistemas basados en el estándar IEEE 802.11.

Una de las tareas más importantes que realiza el WiMAX *Forum* es el control de un acuerdo de certificación que garantiza que los equipos WiMAX de diferentes proveedores puedan interactuar y que los equipos WiMAX satisfagan los presupuestos mínimos de rendimiento y funcionalidad.

Cada equipo es certificado conforme a uno o más perfiles, que representan distintas opciones del estándar IEEE 802.16. La certificación se realiza por los laboratorios que son designados por el WiMAX *Forum*.

Como ayuda a los vendedores antes de la certificación, el WiMAX *Forum* organiza *PlugFests* (fiestas de interconexión) que son reuniones informales donde los vendedores y fabricantes pueden traer sus productos y verificar la interoperabilidad contra los productos de otros fabricantes. Los *PlugFests* son también una buena oportunidad de poner al descubierto ambigüedades o defectos de la norma.

El WiMAX *Forum* ha desarrollado y continúa enriqueciendo una especificación de red extremo a extremo, para los sistemas WiMAX.

Inicialmente, se consideró que esta normalización no debía ser una tarea que el Foro debiera realizar. Pero los operadores y otros actores consideraron que era necesario para la certificación de los componentes de red, y ya que la IEEE 802.16 solo especifica la funcionalidad de las capas 1 y 2, el WiMAX *Forum* decidió desarrollar una especificación de red.

Una actividad muy importante del WiMAX *Forum* es participar e influir en los organismos reguladores para garantizar la operación de WiMAX en bandas de frecuencias

adecuadas. Por ejemplo, el WiMAX *Forum* desempeñó un papel muy importante para que *Mobile WiMAX* fuera aprobada por ITU como un sistema IMT-2000 (*International Mobile Telecommunications-2000*, mejor conocida como 3G) cuando *Mobile WiMAX* se propuso como un sistema candidato para las IMT-Advanced (4G).

El WiMAX *Forum* es una de las organizaciones consultadas frecuentemente por los organismos reguladores nacionales sobre cuestiones relacionadas con el espectro electromagnético para los sistemas inalámbricos fijos y móviles, por ejemplo, antes de las subastas de espectro.

El WiMAX *Forum* participa activamente en la promoción de WiMAX. Organiza su propia serie de conferencias y participa en muchas otras. Publica *white papers* y otros materiales de difusión. En general, juega un papel central para racionalizar y coordinar la información sobre WiMAX que se envía al mercado.

15.3 Las dos variantes de WiMAX

WiMAX está disponible en dos variantes denominadas *Fixed WiMAX* y *Mobile WiMAX*.

Fixed WiMAX se basa en OFDM de la PHY del estándar IEEE 802.16. Como su nombre indica, *Fixed WiMAX* fue diseñado para proveer servicios fijos y nómadas. *Fixed WiMAX* suele ser implementado como una alternativa o complemento de ADSL. *Fixed WiMAX* está orientado para la operación en bandas de frecuencias inferiores a 11 GHz, con licencia y exentas de licencia. Actualmente, existen equipos certificados disponibles para la banda de 3,5 GHz, tanto en la versión *Time Division Duplexing* (TDD) como en la *Frequency Division Duplexing* (FDD).

Mobile WiMAX se basa en OFDMA de la PHY del estándar IEEE 802.16. Esta PHY es más flexible y poderosa que la OFDM, pero también mucho más compleja. *Mobile WiMAX* está principalmente orientado para proveer servicios nómadas y móviles, pero también puede ser utilizado para la prestación de servicios fijos. Su uso está dirigido para la utilización de bandas de frecuencia por debajo de 6 GHz con licencia. En la actualidad hay equipos disponibles con certificación para las bandas de 2,3GHz, 2,5GHz y 3,5 GHz, pero únicamente en las versiones TDD.

Se espera que más tarde aparezcan las versiones FDD de *Mobile WiMAX*.

En 2007, la versión TDD de *Mobile WiMAX* fue aprobada por la ITU como IMT-2000 denominándola IMT-2000 OFDMA TDD WMAN

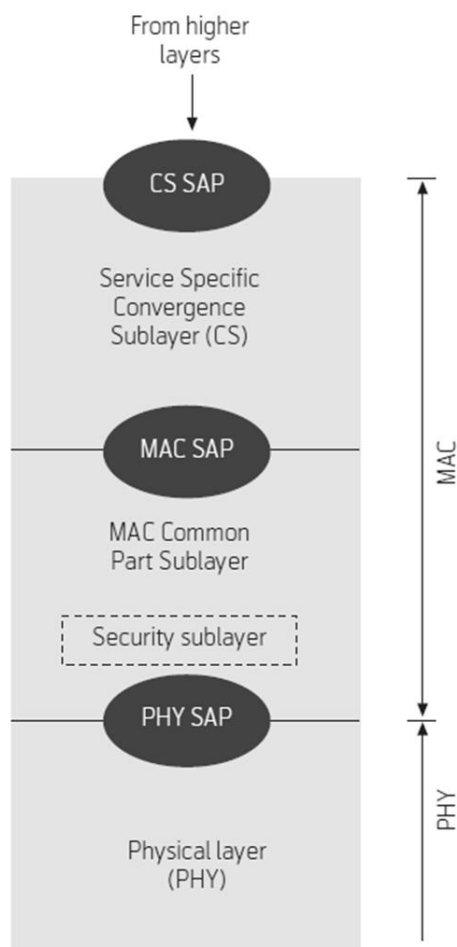


Fig. 15-1: Modelo de referencia del estándar IEEE 802.16. Tomada de (25).

15.4 MAC de WiMAX

La MAC de WiMAX consta de tres subcapas: La subcapa de convergencia específica de servicios (CS, *service specific Convergence Sublayer*), la subcapa parte común (MAC CPS, *Common Part Sublayer*) y la subcapa de seguridad, Fig. 15-1.

Las unidades de datos se comunican entre las subcapas a través de los puntos de acceso de servicio (SAPs, *Service Access Points*).

15.4.1 CS de la MAC

La CS tiene las siguientes funciones:

- La aceptación de unidades de datos de protocolos de mayor nivel (SDU, *Service Data Units*) de la capa superior
- Clasificación de las SDU de la capa superior
- Procesamiento (si es necesario) de las SDU de la capa superior basado en la clasificación
- Entrega de CS PDUs (PDU, *Protocol Data Units*) a la correspondiente MAC SAP
- Recepción de CS PDUs de la entidad par.

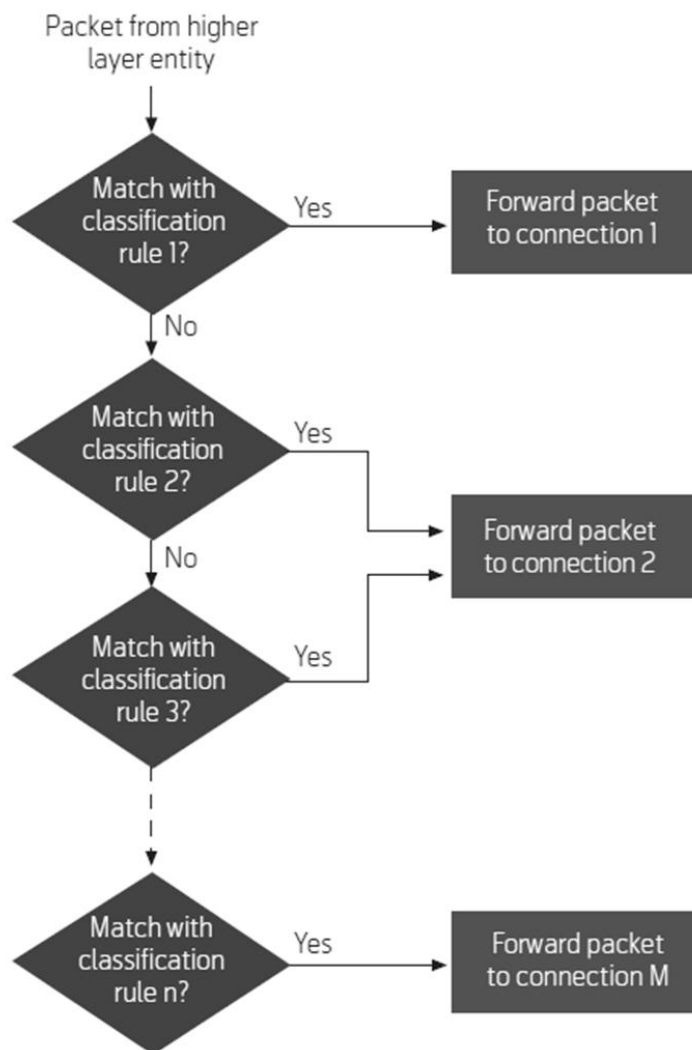


Fig. 15-2 Clasificación de paquetes realizada por la *Convergence Sublayer*. Tomada de (25).

La función principal de la CS es clasificar las SDUs que provienen de la capa superior y enviarlas a las adecuadas conexiones de transporte. La clasificación se realiza con un conjunto de reglas, Fig. 15-2.

Una regla de clasificación consiste en algún criterio de selección de paquetes por campos específicos del protocolo (por ejemplo, dirección IP de destino), clasificación por prioridad, por referencia a una identificación de conexión de transporte (CID, *Connection ID*).

La regla de clasificación de más alta prioridad a la que un paquete corresponde determina a que conexión de transporte se reenvía el paquete.

Las reglas de clasificación del *downlink*, flujo desde la BS hacia las SS (*Subscribe Station*), son aplicadas por la BS, mientras que en el *uplink*, flujo desde las SS hacia la BS, las reglas las aplica la terminal de usuario.

El soporte por parte de la CS para paquetes IPv4 e IPv6 es mandatorio para los sistemas *Mobile WiMAX*. El Soporte para Ethernet IPv4 sobre Ethernet y IPv6 sobre Ethernet es mandatorio para las estaciones base pero opcional para las terminales de los usuarios.

15.4.2 CPS de la MAC

La capa MAC de 802.16 está orientada a la conexión. Todos los servicios, incluidos los servicios inherentemente sin conexión se asignan a una conexión. Las conexiones son unidireccionales y se referencian con identificadores de 16 bits (CID).

Una conexión está asociada a un conjunto de requerimientos de QoS. Normalmente hay varias conexiones entre una BS y un terminal, donde las diferentes conexiones transportan datos con requerimientos de QoS diferentes.

A los paquetes de las diferentes conexiones se les asigna ranuras (*slots*) de transmisión en el mapa tiempo-frecuencia de espacio de modulación por el *scheduler*. Una de las principales tareas del *scheduler* es asignar las ranuras de transmisión de tal manera que se satisfagan los requerimientos de QoS para cada conexión.

El *scheduler* de la BS se encarga de asignar ranuras en el *uplink* a las estaciones subordinadas para proveer ancho de banda (*grants*) u oportunidades para que lo soliciten (*polls*).

Al conocer el estado de las conexiones y sus parámetros de QoS asociados, el *scheduler* de la BS puede prever el *throughput* (tasa de transferencia) y latencia necesarios para el tráfico *uplink*, y proveer *polls* y *grants* en los tiempos apropiados.

Para realizar el *polling* de una terminal, la BS asigna ranuras de tiempo-frecuencia suficientes para que la terminal envíe en ellas una solicitud de ancho de banda (BR, *Bandwidth Request*). La terminal se informará que le ha sido asignado ranuras de tiempo-frecuencia en el mensaje UL-MAP enviado por el BS.

El *polling* se realiza por terminal. El ancho de banda siempre es requerido por CID pero es asignado por terminal.

Fixed WiMAX y *Mobile WiMAX* soportan cuatro y cinco clases de servicios de *scheduling* respectivamente.

15.4.2.1 UGS (*Unsolicited Grant Service*).

No hay *polling*, el ancho de banda ya está preasignado. La clase UGS está diseñada para soportar en el *uplink*, flujos de paquetes de tamaño fijo, sobre una base periódica, para

servicio en tiempo real tales como conexiones T1/E1 y VoIP (*Voice over IP*) sin supresión de silencios.

El servicio ofrece periódicamente *grants* (ancho de banda para transmitir datos en el UL) de tamaño fijo, lo que elimina la sobrecarga y latencias debidas a las solicitudes de ancho de banda de las SS, asegurando que los *grants* estén disponibles para satisfacer las necesidades del flujo de tiempo real.

Entre los parámetros de QoS obligatorios están *Minimum Reserved Traffic Rate*, *Maximum Latency and Tolerated Jitter*.

15.4.2.2 rtPS (*real-time Polling Service*)

El rtPS está diseñado para soportar en el *uplink* flujos de transporte de datos en tiempo real consistente en paquetes de tamaño variable sobre una base periódica, como el vídeo MPEG. El servicio ofrece oportunidades de requerimiento periódicas, que responden a las necesidades del flujo de tiempo real y permiten a la terminal especificar el tamaño del *grant* deseado.

Este servicio tiene más sobrecarga que UGS por la solicitud, pero admite *grants* de tamaño variable. Los parámetros obligatorios de QoS incluyen *Minimum Reserved Traffic Rate*, *Maximum Sustained Traffic Rate*, *Maximum Latency and Unsolicited Polling Interval*.

15.4.2.3 ertPS (*extended real-time Polling Service*) (solo para *Mobile WiMAX*)

El rtPS extendido es un mecanismo de *scheduling* que se basa en las eficiencias de UGS y rtPS. La BS proporciona *unicast grants* sin solicitudes, como en UGS, evitando así la latencia producida por un mensaje de solicitud de ancho de banda.

Sin embargo, mientras que en UGS los *grants* tienen un tamaño fijo, en ertPS los tamaños de los *grants* son dinámicos. La BS puede proporcionar asignaciones periódicas en el *uplink* que pueden ser utilizadas para que las SS tanto para requerir el ancho de banda como para la transferencia de datos. Existe un tamaño de asignación por defecto que puede ser cambiado por la terminal.

Los parámetros obligatorios de QoS incluyen *Maximum Sustained Traffic Rate*, *Minimum Reserved Traffic Rate*, *Maximum Latency and Unsolicited Grant Interval*.

15.4.2.4 nrtPS (*non real-time Polling Service*)

El servicio nrtPS realiza *unicast polls* sobre una base regular, lo que asegura que el flujo de servicios de *uplink* recibe oportunidades de solicitar ancho de banda, aún durante una congestión de la red. La BS normalmente sondea cada conexión nrtPS con un intervalo del

orden de un segundo o menos. Los parámetros de QoS mandatorios para el servicio de *scheduling* son, *Minimum Reserved Traffic Rate*, *Maximum Sustained Traffic Rate* y *Traffic Priority*.

15.4.2.5 BE (*Best Effort*)

El servicio de mejor esfuerzo, BE, no ofrece ninguna garantía de QoS. El servicio es adecuado para flujos de datos que no requieren ningún nivel mínimo de servicio, por lo que puede ser manejado en función del ancho de banda disponible.

15.4.3 Subcapa de Seguridad

La subcapa de seguridad proporciona a los suscriptores de privacidad, autenticación, y confidencialidad a través de las redes de banda ancha inalámbrica. Esto se realiza mediante la aplicación de transformaciones criptográficas a los datos transportados a través de las conexiones entre las terminales y la BS. Además, la subcapa de seguridad provee a los operadores una fuerte protección contra el robo de servicio.

La subcapa de seguridad emplea un protocolo de gestión de claves autenticado por el cliente/servidor, en el que la BS, el servidor, controla la distribución del material de claves a las terminales de los clientes. Además, los mecanismos de seguridad básica son reforzados mediante la adición de certificados digitales, para la autenticación de los dispositivos de las terminales, al protocolo de gestión de llaves.

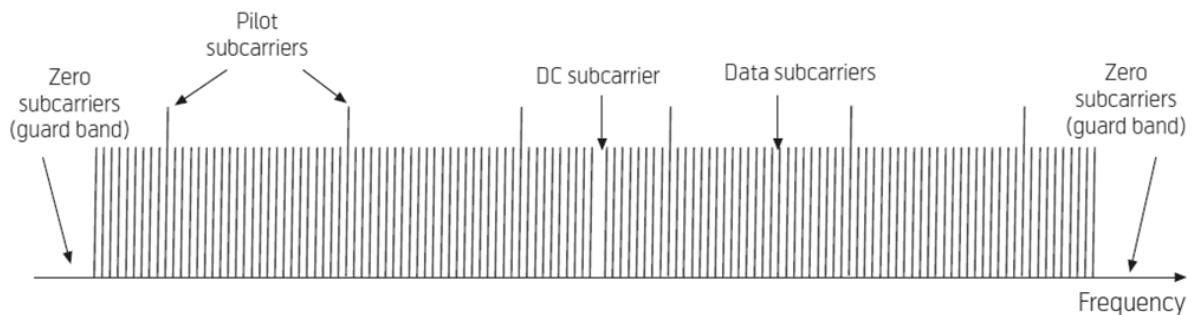


Fig. 15-3: Descripción de subportadoras de OFDM. Tomada de (25).

15.5 PHY de *Fixed WiMAX*

La capa física de *Fixed WiMAX* está basada en un esquema *Orthogonal Frequency Division Multiplexing* (OFDM) con 200 subportadoras. Como para implementar la modulación y demodulación de la señal se usa la FFT (*Fast Fourier Transform*) de 256 puntos, la señal es

frecuentemente mal interpretada como constituida por 256 portadoras, ya que solo 200 de las 256 son no nulas, Fig. 15-3.

OFDM es un sistema de modulación muy robusto contra condiciones de propagación *multipath* (multitrayecto). La señal recibida por una terminal de usuario o una estación base puede ser interpretada como la suma de varias copias de la misma señal que viajan por diferentes rutas desde el transmisor al receptor. Puesto que las señales han viajado por caminos diferentes, presentan diferentes demoras y atenuaciones. Esto provoca un desvanecimiento selectivo en frecuencia (*fading*) e interferencia entre símbolos (ISI, *Inter Symbol Interference*).

El desvanecimiento selectivo en frecuencia provoca niveles de atenuación diferentes para cada subportadora del canal. Sin embargo, mediante el uso de separaciones de frecuencias suficientemente pequeñas entre las subportadoras, cada subportadora experimentará prácticamente la misma respuesta en frecuencia. Por lo tanto, las subportadoras serán atenuadas pero virtualmente no distorsionadas.

Para reducir el ISI, a cada símbolo OFDM se le anexa un prefijo cíclico (conocido como intervalo de guarda). La longitud del prefijo se elige de forma que "absorba" la mayor parte del ISI. El receptor puede ignorar la señal durante el tiempo que corresponde al prefijo cíclico, por lo tanto, la parte de la señal utilizada para descifrar los datos será prácticamente libre de ISI. Obtener un canal casi libre de ISI produce un pequeño desperdicio de tiempo y potencia libre, pero bien vale la pena. En *Fixed WiMAX* el prefijo cíclico puede ser de 1/4, 1/8, 1/16 o 1/32 de la longitud del símbolo (útil). La longitud de símbolo (útil) y la separación entre subportadoras no puede ser elegido de forma independiente, su producto debe ser igual a 1.

Por lo general, cada símbolo OFDM lleva los datos hacia o desde una terminal única de usuario. Sin embargo, está contemplado como opción la división de las subportadoras en 2, 4, 8 o 16 grupos, donde los grupos pueden transportar datos desde o hacia terminales diferentes. Esto se denomina subcanalización.

Las subportadoras que pertenecen a un subcanal, se distribuyen en una forma no regular en todo el ancho de banda del canal. La principal ventaja de utilizar subcanalización es disminuir los requerimientos en la relación potencia pico a potencia promedio para los transmisores de las terminales de usuario para una cobertura dada, o aumento de la cobertura para una potencia de transmisión terminal dado pico.

15.6 PHY de *Mobile WiMAX*

La capa física de *Mobile WiMAX* se llama OFDMA, que es esencialmente una versión más flexible de OFDM donde la subcanalización se utiliza en forma generalizada. Es decir, un símbolo OFDMA por lo general lleva los datos hacia o desde varios usuarios.

En la Fig. 15-4 se pone de manifiesto la diferencia entre OFDM de *Fixed WiMAX* y OFDMA de *Mobile WiMAX*. En OFDMA cada columna representa un símbolo y cada fila un subcanal.

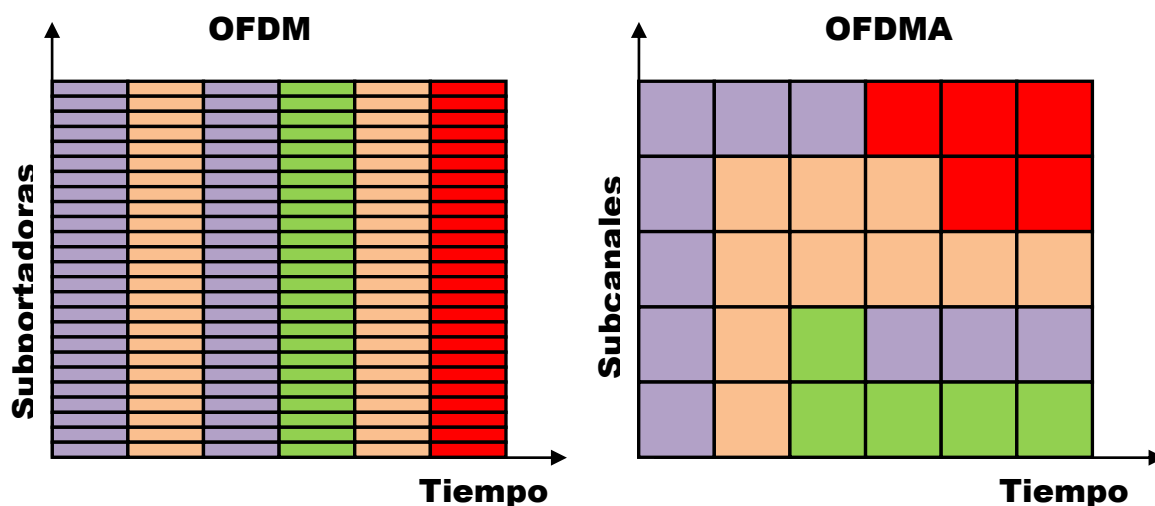


Fig. 15-4: Diferencia entre OFDM y OFDMA. Cada color representa datos de un usuario.

Una de las ventajas de la OFDMA PHY sobre OFDM PHY es que se puede utilizar un número diferente de subportadoras. De esta manera es posible adaptar mejor el OFDMA PHY al ancho de banda del canal.

La modulación y demodulación OFDMA se basan en la FFT de 128, 512, 1024 o 2048 puntos, para adaptarla a los diferentes anchos de banda que puede tener un canal. Cabe señalar que la opción de 256 puntos de la FFT ha sido excluida, lo que parece muy extraño desde una perspectiva técnica. La razón de esto es que el modo FFT de 256 puntos quedó reservado para el OFDM PHY, resultado de un compromiso político en el grupo de trabajo IEEE 802.16.

Tabla 15-1: Tamaños de la FFT para distintos anchos de banda del canal. Tomada de (25).

Ancho de banda del canal en MHz	1,25	3,5	5	7	8,75	10	20
Tamaño de la FFT	128	512	512	1024	1024	1024	2048

La Tabla 15-1 lista los diferentes tamaños de la FFT usada para diferentes anchos de banda de canal.

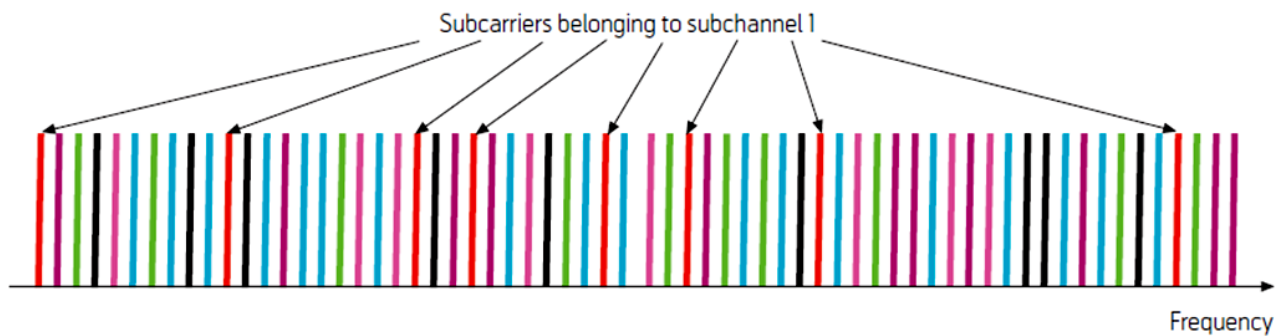


Fig. 15-5: Descripción en frecuencia de OFDMA. Subportadoras con el mismo color pertenecen al mismo subcanal. Tomada de (25).

Un símbolo OFDMA puede ser considerado como un conjunto compuesto de tres tipos de subportadoras, de datos, piloto y subportadoras nulas. Como su nombre lo indica, las subportadoras de datos llevan los datos de la información. Las subportadoras piloto se utilizan para la realización de diversas tareas de estimación y sincronización.

Mediante el uso del concepto de subportadoras nulas, un símbolo OFDMA se puede considerar como un conjunto con el mismo número de subportadoras que el número de puntos de la FFT, pero con algunas de las subportadoras no utilizadas (es decir, las subportadoras nulas).

El subportadoras de un símbolo OFDMA se reparten en subconjuntos de subportadoras llamados subcanales. La subportadoras que componen un subcanal, pueden pero no necesitan ser adyacentes. Fig. 15-5.

Hay diferentes formas de asignación de subportadoras a los subcanales. Los modos de asignación de subportadoras reciben nombres como PUSC (*Partial Usage of Subchannels*), FUSC (*Full Usage of Subchannels*), AMC (*Adaptive Modulation and Coding*) etc. Una descripción detallada de estos modos está fuera del alcance de esta introducción al WiMAX.

Los esquemas de asignación de subcanal se pueden agrupar en dos categorías principales: asignaciones distribuidas y asignaciones adyacentes. Para las asignaciones distribuidas, las subportadoras pertenecientes a un subcanal se distribuyen pseudo aleatoriamente sobre el ancho de banda de total del canal. Esto es ventajoso en los escenarios móviles donde las características del canal cambian rápidamente a medida que la terminal se mueve.

Al utilizar los subcanales portadoras distribuidas en todo el ancho de banda del canal, el sistema se vuelve resistente contra el desvanecimiento selectivo en frecuencia. El mismo puede causar una gran atenuación en partes de la banda de frecuencia pero difícilmente pueda afectar a todas las subportadoras de un subcanal. Otra ventaja de distribuir la asignación de

subportadoras se da en los sistemas celulares ya resulta promediada la interferencia entre celdas.

En el modo de asignación de subportadoras adyacentes, las subportadoras de un subcanal son contiguas y están confinadas a sólo una parte del ancho de banda del canal. Este modo es ventajoso cuando las terminales están fijas o se mueven lentamente. Para una terminal dada, se usan los subcanales cuyas subportadoras están situadas en la parte del canal en que la atenuación del canal es más pequeña. Se trata de que en conjunto las terminales transmitan y reciban en los subcanales que les son más favorables, incrementándose de esta forma la capacidad del sistema.

Los símbolos OFDMA se agrupan en los *frames*. La estructura de un *frame* depende si se utiliza dúplex por división de tiempo (TDD) o por división de frecuencia (FDD). En el modo TDD se utiliza la misma banda de frecuencias para el *downlink* y *uplink*. Mientras que en modo FDD se utilizan diferentes bandas de frecuencias. Para permitir la existencia de terminales de menor complejidad y por ende menor costo, el modo *half duplex* FDD también es compatible con el modo FDD.

La siguiente descripción se aplica a los sistemas basados en TDD de *Mobile WiMAX*. Todos los equipos certificados hasta el momento de escribir este resumen (Febrero de 2010) son de este tipo.

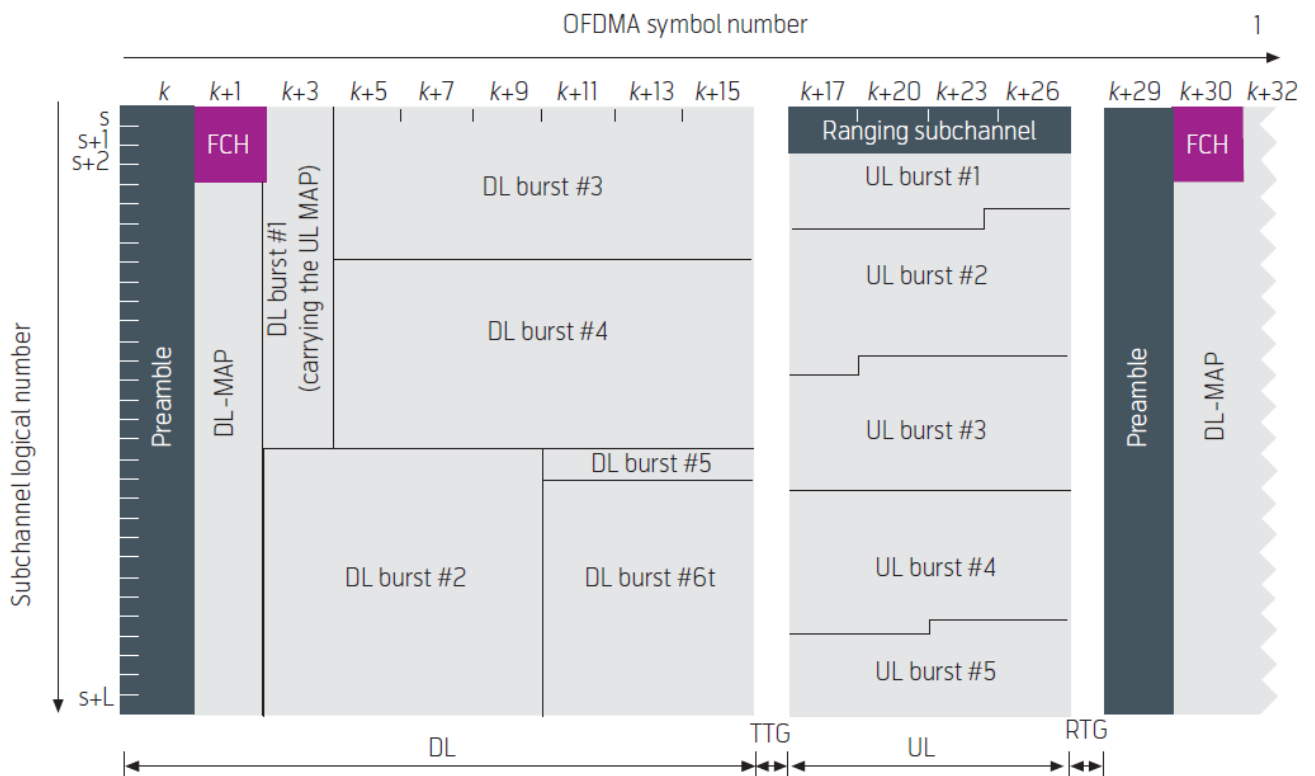


Fig. 15-6: Estructura general de un *frame* TDD de WiMAX. Tomada de (25).

En la Fig. 15-6 se muestra la estructura general de un *frame* TDD de WiMAX. Consiste en un intervalo para el *downlink* y un intervalo para el *uplink* separados por pequeñas transiciones vacías, la TTG (*Transmit/Receive Transition Gap*) y la RTG (*Receive/Transmit Transition Gap*). La TTG brinda un respiro a la BS para cambiar de modo transmisión a modo de recepción. Durante este pequeño intervalo la BS cesa la transmisión de RF dando tiempo para actuar al *switch* de antenas. El RTG brinda un tiempo para que la BS cambie de modo recepción a modo de transmisión y vuelva a actuar el *switch* de antenas.

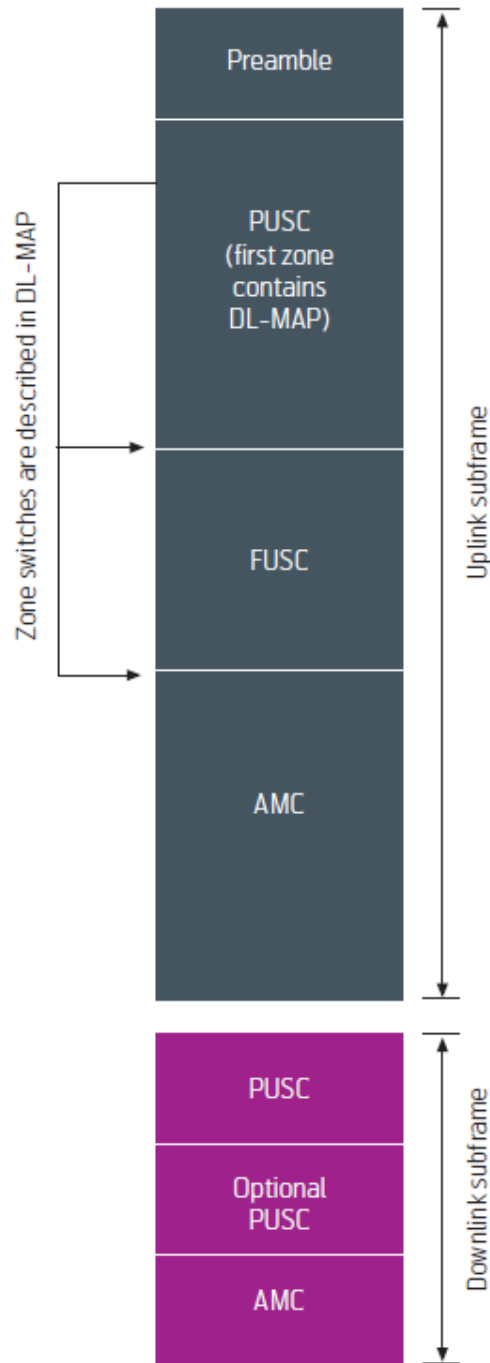


Fig. 15-7: *Frame* TDD de OFDMA con múltiples zonas. Tomada de (25).

Las longitudes de los *gaps* de transición determinan el máximo tamaño de una celda del sistema. Por ejemplo, el TTG debe ser mayor que el tiempo que toma la señal para ir desde la BS a la estación terminal más lejana y volver. Para un valor dado del TTG, el tamaño de la celda se puede aumentar realizando un uso inteligente en el *scheduling* de las oportunidades de transmisión para las estaciones de usuario más lejanas.

Cada intervalo descendente comienza con un preámbulo seguido por el *Frame Control Header* (FCH), el DL-MAP y el UL-MAP. El FCH especifica el perfil del *burst* (ráfaga) y la longitud del DL-MAP. Los mensajes DL-MAP y UL-MAP definen la utilización de los intervalos *downlink* y el *uplink* respectivamente.

Para obtener una óptima eficiencia en el *downlink* y *uplink* de diferentes tipos de terminales, cada *frame* puede ser dividido en múltiples zonas de permutación. Una zona de permutación es un número de símbolos contiguos OFDMA, en el *downlink* o el *uplink* que utilizan el mismo modo de asignación de subportadoras. El *downlink subframe* y el *uplink subframe* pueden contener más de una zona de permutación de subportadoras tal como se ilustra en la Fig. 15-7.

Los *downlink* y *uplink subframes* comienzan en el modo PUSC. El esquema de asignación de subportadoras puede ser cambiado a una zona con diferente esquema de asignación de subportadoras. Este cambio está descrito en el mensaje DL-MAP. Por ejemplo cambiando a una zona AMC, donde las subportadoras que componen cada subcanal son adyacentes, las terminales fijas y nómadas serán servidas óptimamente.

WiMAX soporta una amplia variedad de implementaciones multiantena de avanzada tecnología. El término MIMO (*Multiple Input Multiple Output*) se utiliza frecuentemente en relación con las implementaciones de antenas múltiples, pero a menudo con definiciones ligeramente diferentes, y por lo tanto de las implementaciones incluidas. En este artículo MIMO simplemente se usa para referirse a cualquier tecnología multiantena. Un importante parámetro de un sistema MIMO es el número de antenas utilizadas en el transmisor y el receptor. Un sistema con N_t antenas transmisoras y N_r antenas de receptoras se denomina sistema MIMO ($N_t \times N_r$).

Las tecnologías MIMO se pueden dividir en dos grupos: MIMO de lazo abierto y MIMO de lazo cerrado. En MIMO de lazo abierto el transmisor no utiliza ningún conocimiento de las características de propagación del canal, mientras que en MIMO de lazo cerrado el transmisor recolecta información sobre el canal de propagación y optimiza la señal transmitida.

WiMAX soporta varios tipos de MIMO de lazo abierto: *Space Time Coding* (STC), *Spatial Multiplexing* (SM) y *collaborative SM*.

Con STC se emplean antenas múltiples en el transmisor mientras que el receptor puede tener una o más antenas. El mismo flujo de datos se codifica con diferentes versiones las que alimentan a cada antena transmisora, de esta forma se crea una diversidad espacial. Esto hace que la transmisión sea mucho más robusta ante fenómenos de propagación *multipath*.

La capacidad no aumenta directamente como lo es para SM, pero puede aumentar indirectamente ya que una recepción más robusta permite la utilización de esquemas de modulación y de codificación de orden superior. STC se puede aplicar para mejorar la cobertura de una red. STC es referido frecuentemente como esquema *Matrix-A* en WiMAX.

Para SM se usan múltiples antenas tanto en el transmisor y como en el receptor. Un flujo de datos se divide en una serie de subflujos que se transmiten desde diferentes antenas con las mismas asignaciones de tiempo y frecuencia. En un entorno *multipath* suficientemente variado las múltiples señales llegarán a las antenas receptoras con firmas espaciales suficientemente diferentes como para permitir al receptor discernir los múltiples flujos de datos. El resultado es un aumento de la capacidad.

La capacidad se puede aumentar por un factor igual a el número mínimo de antenas en el transmisor y el receptor, por ejemplo, un sistema MIMO 4 x 4 puede aumentar la capacidad en un factor de 4, mientras que un sistema MIMO 4 x 2 puede aumentar la capacidad en un factor de 2. SM es referido como *Matrix-B scheme* en WiMAX.

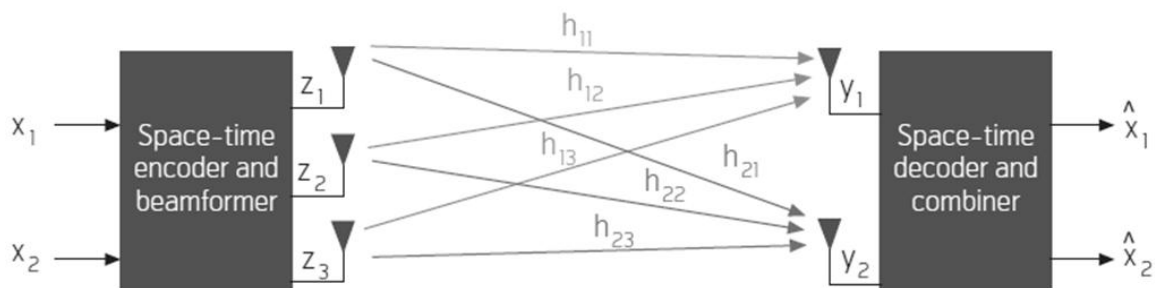


Fig. 15-8: Sistema MIMO 3x2. Tomada de (25).

Con el fin de operar siempre de la manera más eficiente, un sistema de WiMAX que utiliza MIMO debe soportar tanto STC como SM. STC se comporta mejor que SM cuando la SNR (*Signal to Noise Ratio*) es baja, por ejemplo lo que típicamente suele producirse cerca del borde de la celda. SM se comporta mejor que en la STC situaciones de alta SNR. Por lo tanto, para operar en la forma más eficiente todo momento, el sistema debe ser capaz de cambiar entre STC y SM.

Collaborative SM es una técnica en la que dos terminales de usuario, cada uno con una única antena pueden utilizar el mismo tiempo y asignación de frecuencias para transmitir

hacia un BS con dos (o más) antenas. Esta técnica puede efectivamente duplicar la capacidad del *uplink* del sistema.

Las técnicas MIMO de lazo cerrado pueden permitir un mejor rendimiento que aquellas MIMO de lazo abierto mediante la explotación de la información de estado del canal en el lado del transmisor. Las técnicas MIMO de lazo cerrado en WiMAX incluyen MIMO de precodificación y MIMO multiusuario *downlink* (MU-MIMO).

MIMO de precodificación se puede interpretar como un *beamforming* generalizado. Considérese por ejemplo un sistema MIMO 3 x 2 que se muestra en la Fig. 15-8. Las señales y_1 e y_2 recibidas por la antena 1 y 2, respectivamente, están dadas por:

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \end{bmatrix} \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \\ w_{31} & w_{32} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \end{bmatrix}$$

o $y = H \cdot W \cdot x + n$

donde h_{ij} describe el canal de transmisión entre la antena transmisora j y la antena receptora i , W es la matriz de precodificación y n es el ruido recibido. La capacidad del sistema puede ser optimizado mediante la elección de la mejor matriz de precodificación W .

Para que el transmisor que se capaz de encontrar la matriz de precodificación óptima, debe conocer el canal (la matriz H del canal). En TDD, en principio es posible tomar ventaja de la reciprocidad del canal para que la BS o la terminal puedan utilizar las señales recibidas para estimar el comportamiento del canal en transmisión.

Sin embargo, en la práctica la retroalimentación se utiliza para evitar la calibración de los componentes de RF. Para los sistemas FDD, usar realimentación es la única opción.

Downlink MU-MIMO, también llamada *Space Division Multiple Access* (SDMA) en el estándar IEEE 802.16, puede ser vista como la superposición de hasta cuatro transmisiones *beamforming* de usuarios únicos en el mismo recurso tiempo-frecuencia.

15.7 Arquitectura de red WiMAX

El estándar IEEE 802.16 sólo especifica las PHY y MAC del sistema. Los operadores y demás integrantes del WiMAX *Forum* vislumbraron que sería muy útil definir una arquitectura de red estándar.

Con una arquitectura de red estándar, los componentes de red pueden ser certificados garantizando una interoperabilidad básica entre equipos de diferentes vendedores, y un nivel mínimo de funcionalidad y rendimiento de los mismos. Finalmente se decidió que el

desarrollo de la arquitectura de red fuera realizado en el ámbito del WiMAX *Forum*, por lo tanto la misma no está incluida en el estándar IEEE 802.16.

También se decidió que la arquitectura de red WiMAX debe ser una arquitectura totalmente IP con el fin de maximizar el uso de estándares abiertos y los protocolos de la IETF.

La Fig. 15-9 muestra el modelo de referencia de red de WiMAX (NRM, *Network Reference Model*), que es una representación lógica de la arquitectura de red WiMAX. El NRM de WiMAX diferencia entre los proveedores de acceso a la red (NAPs, *Network Access Providers*) y los proveedores de servicios de red (NSPs, *Network Service Providers*). La NAP es una entidad comercial que provee la infraestructura de acceso de RF de WiMAX, mientras que el NSP es la entidad de negocios que proporciona conectividad IP y servicios WiMAX a los suscriptores conforme a acuerdos de nivel de servicio con uno o más NAPs.

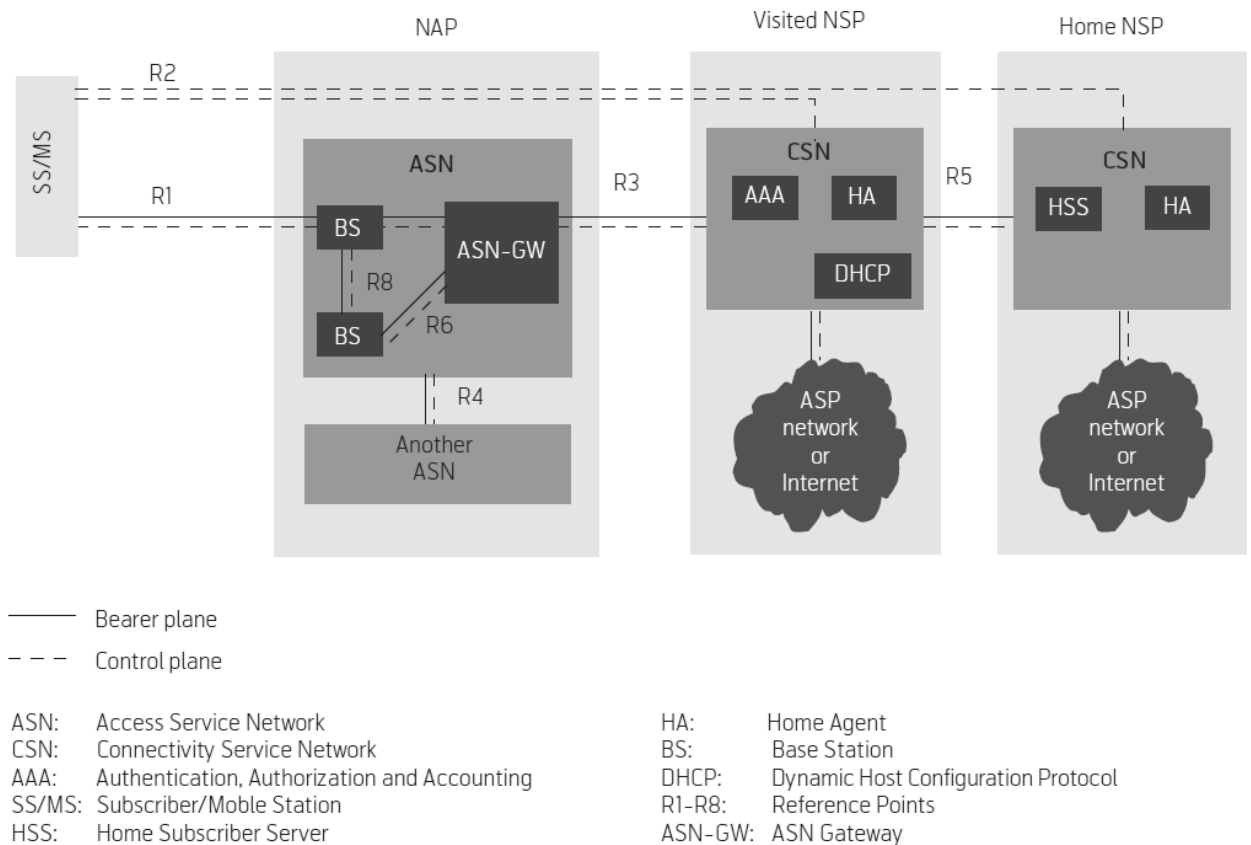


Fig. 15-9: Modelo de referencia de red de WiMAX. Tomada de (25).

Las SS/MS, ASN y CSN representan una agrupación lógica de las funciones.

SS/MS representa la terminal del usuario. SS se refiere a una terminal fija, mientras que MS (*Mobile Station*) se refiere a un terminal móvil. La terminal interactúa con las BSs de una

ASN (*Access Service Network*) a través del punto de referencia R1 que representa la interfaz de radio.

La ASN representa las funciones necesarias para proveer acceso de radio a la terminal de usuario (SS/MS). Esto incluye uno o varios *gateways* ASN. La ASN conforma un agrupamiento de funciones centralizadas relacionadas con la QoS, seguridad y gestión de la movilidad para todas las conexiones de datos atendidos por sus estaciones de base asociadas.

La CSN (*Connectivity Service Network*) representa un conjunto de funciones de red que prestan servicios de conectividad IP a los suscriptores WiMAX. Algunas de las principales funciones del CSN son:

- Gestión de las direcciones IP
- Conectividad a Internet
- Movilidad inter ASN ligada a la CSN
- AAA (*Authentication, Authorization and Accounting*) *proxy* o servidor
- Políticas de QoS y control de admisión
- Facturación y acuerdos entre operadores

La movilidad está soportada por dos niveles de anclaje. Cada *handover* entre BSs implica movilidad pero ligada al mismo ASN, lo cual es completamente transparente a la red principal (CSN). Las funciones de control de ASN modifican las rutas de los datos dentro de un ASN o entre ASNs.

Si el agente externo o el prefijo IP cambia, recién se dispara un *handover* entre CSNs. Este procedimiento de traspaso está basado en el protocolo IP Móvil, cliente o *proxy*, para IPv4 o IPv6.

15.8 Multicast y broadcast

WiMAX soporta servicios de difusión *multicast* y multimedia (MCBCS, *Multicast and Broadcast Multimedia Services*) lo que permite a las redes WiMAX proveer a bajo costo la entrega de grandes volúmenes de contenido populares como IP-TV y los medios de comunicación *peer-to-peer* para compartir información a un grupo de usuarios. Los servicios *unicast* y MCBCS se pueden proporcionar al mismo tiempo mediante la asignación de diferentes intervalos de tiempo en el *downlink subframe* a los mismos.

Comúnmente los flujos de datos MCBCS se transmiten a una zona geográfica MCBCS, que a su vez consiste de una o más zonas MBS (*Multicast and Broadcast Service*). Una zona MBS se define como un conjunto de estaciones base donde se usa el mismo CID y la misma asociación de seguridad para la transmisión del contenido de determinado flujo de servicio.

Dentro de una zona MBS, se puede implementar una MCBCS con la coordinación u opcionalmente, con la sincronización entre las estaciones base de la zona de MBS. Si las BSs se coordinan, y sus *subframes* de transmisión estén alineados, esto permite a una terminal seguir recibiendo transmisiones MBS de cualquier BS de esa zona, incluso cuando la terminal está en modo de reposo o inactiva.

Si todos las BSs en una zona MBS son sincronizadas en tiempo y en frecuencia, pueden formar una red de frecuencia única durante los intervalos MCBCS. Esto mejora la intensidad de la señal recibida, y estadísticamente, resulta en una mejora de la velocidad de transmisión de datos y en los niveles de confiabilidad. La continuidad de servicio, incluso cuando la terminal está en movimiento, y el ahorro de energía se mantienen a través de las celdas de una zona MBS con un mínimo de interacciones con la red.

15.9 Next Generation WiMAX

En octubre de 2007 *Mobile WiMAX* fue aprobado como una nueva tecnología IMT-2000 por la ITU (*International Telecommunications Union*). La versión de *Mobile WiMAX* que se incluye se basa en TDD, y para reflejar esto ITU consigna “IMT-2000 OFDMA TDD WMAN”. Al disponer WiMAX de esta etiqueta puede tener acceso a las frecuencias asignadas para IMT-2000. Actualmente IEEE y el *WiMAX Forum* también están tratando de incluir la versión *Mobile WiMAX FDD* en las IMT-2000.

La ITU también ha definido un conjunto de requisitos para una nueva generación de sistemas móviles para seguir a los sistemas IMT-2000. La familia del nuevo sistema se llama *IMT-Advanced*, también conocido como como la 4^{ta} generación de los sistemas móviles denominado brevemente como 4G.

Como una continuación natural de la aprobación IMT-2000, *WiMAX Forum* y *WiMAX IEEE* desean que *WiMAX* sea aprobado también como una tecnología *IMT-Advanced*. Actualmente la ITU sólo ha recibido dos propuestas para las tecnologías *IMT-Advanced*, una basada en *Mobile WiMAX* y la otra basada en *LTE (Long Term Evolution)*.

El sistema *Mobile WiMAX* propuesto para *IMT-Advanced* se basa en una nueva enmienda a la IEEE 802.16 llamada IEEE 802.16m, que cual aún está siendo desarrollada. La enmienda 16m contiene nuevas funcionalidad y nuevos métodos para mejorar el rendimiento del sistema lo que permitirá satisfacer los requisitos de *IMT-Advanced*.

Básicamente, el objetivo en el desarrollo de la enmienda 16m es el de duplicar el rendimiento del actual sistema *Mobile WiMAX* en varios puntos, estos incluyen velocidad de

transferencia de datos por sector, velocidad promedio por usuario, velocidad pico y velocidad en el borde de la celda.

El sistema *Mobile WiMAX* propuesto para *IMT-Advanced* usa OFDMA como esquema de múltiple acceso tanto en el *downlink* como en el *uplink*. También soporta TDD y FDD como sistemas de duplexión. El ancho de banda es escalable entre 5 y 20 MHz utilizando FFTs de diferentes tamaños. El sistema soporta operación multicanal que permite la operación en anchos de banda tan grandes como 100 MHz, adicionando canales de RF contiguos o no contiguos.

Tabla 15-2: Rendimiento espectral para *Mobile WiMAX* TDD. Los requerimientos del ITU para la *IMT-Advanced* están entre paréntesis. Tomada de (25).

Escenario	Rendimiento espectral promedio, DL [bit/s/Hz/celda]	Rendimiento espectral en el borde de celda, DL [bit/s/Hz/celda]	Rendimiento espectral promedio, UL [bit/s/Hz/celda]	Rendimiento espectral en el borde de celda, DL [bit/s/Hz/celda]
InH	6,93 (3,0)	0,260 (0,1)	5,99 (2,25)	0,426 (0,07)
UMi	3,22 (2,6)	0,092 (0,075)	2,58 (1,8)	0,111 (0,05)
UMa	2,41 (2,2)	0,069 (0,06)	2,57 (1,4)	0,109 (0,03)
Rma	3,23 (1,1)	0,093 (0,04)	2,66 (0,7)	0,119 (0,015)

Tabla 15-3: Rendimiento espectral en escenarios móviles para *Mobile WiMAX* TDD. Tomada de (25).

Escenario de ensayo	Relación Señal/Ruido mediana en dB	Rendimiento espectral para LOS [bit/s/Hz]	Rendimiento espectral para no LOS [bit/s/Hz]	Rendimiento espectral requerido por ITU [bit/s/Hz]
InH (10 km/h)	6,93 (3,0)	0,260 (0,1)	5,99 (2,25)	0,426 (0,07)
UMi (30 km/h)	3,22 (2,6)	0,092 (0,075)	2,58 (1,8)	0,111 (0,05)
UMa (120 km/h)	2,41 (2,2)	0,069 (0,06)	2,57 (1,4)	0,109 (0,03)
Rma (350 km/h)	3,23 (1,1)	0,093 (0,04)	2,66 (0,7)	0,119 (0,015)

Cada tecnología candidata para las *IMT-Advanced* tiene que ser evaluada conforme a los criterios y metodologías propuestos por ITU. La presentación de una tecnología candidata para las *IMT-Advanced* debe incluir una autoevaluación de la tecnología. A esto le seguirá una evaluación realizada por uno o más grupos técnicos independientes. Los resultados de la autoevaluación de *Mobile WiMAX* realizado por IEEE se pueden utilizar como una indicación inicial del rendimiento de *next generation Mobile WiMAX*.

La actuación fue estimada en cuatro escenarios diferentes:

- InH: *hotspot* interior
- UMI: micro celda urbano
- UMa: macro celda urbano
- RMA: escenario macro rural

La eficiencia del espectro estimado para *Mobile WiMAX TDD* para los diferentes escenarios, se dan en la Tabla 15-2. Los requisitos de *IMT-Advanced* se muestran entre paréntesis. Se puede observar que la estimación del rendimiento está por encima de los requisitos de la ITU en todos los casos, y muy por encima en la mayoría de los casos.

La Tabla 15-3 muestra la eficiencia espectral de *Mobile WiMAX TDD* en escenarios en movimiento. Una vez más los resultados están muy por encima de los requisitos de la UIT.

15.10 Equipamiento WiMAX y estado de la implementación

15.10.1 Estado de Implementación

Hasta Octubre de 2010 el *WiMAX Forum* registró alrededor de 587 implementaciones WiMAX distribuidas en 149 países. Se estima que WiMAX provee servicio a más de 621 millones de personas. De acuerdo a Marvedis, el número total de suscriptores de WiMAX para fines de 2009 fue de alrededor de 6,5 millones.

Existen más implementaciones *Fixed WiMAX* que *Mobile WiMAX* en el mundo, pero en el primavera de 2009 el número de abonados de *Mobile WiMAX* sobrepasó al número de suscriptores de *Fixed WiMAX*. La Fig. 15-10 muestra las redes de los *Mobile WiMAX* y *Fixed WiMAX* se encuentren en servicio o en implementación a Octubre de 2010.

Tabla 15-4: Los cinco mayores operadores de WiMAX en el mundo. Tomada de (25).

Operador	País	Banda de frecuencia	Número aproximado de suscriptores
Clearwire	EE.UU.	2,5 GHz	690000
Yota	Rusia	2,5 GHz	350000
Inukshut	Canada	2,5 GHz	300000
Korea Telecom	Corea	2,3 GHz	290000
Telmex	México	3,5 GHz	250000

Los cinco mayores operadores del mundo se enumeran en el Tabla 15-4. La red WiMAX más grande y más importante del mundo es la implementada por Clearwire en los EE.UU. Clearwire es el resultado de una fusión realizada en 2008 entre la original Clearwire y Sprint Xohm WiMAX por la infraestructura de red. Intel, Google y las tres empresas de televisión por cable Comcast, Time Warner Cable y Bright House Networks son también accionistas de

la sociedad. Sprint ha contribuido con todas sus explotaciones en el espectro de 2,5 GHz, brindando a la nueva Clearwire más de 100 MHz de espectro en la mayoría de los mercados en los EE.UU.

Clearwire ofrece WiMAX en 27 mercados de todo el EE.UU. que cubren 34 millones de personas, entre ellas Chicago, Las Vegas, Filadelfia, Seattle y Honolulu. Clearwire planea extender sus operaciones a más mercados en 2010, incluyendo Nueva York, Boston, Washington DC, Houston, la Bahía de San Francisco, Denver, Minneapolis y Kansas City lo que cubriría aproximadamente 120 millones personas a finales de 2010.

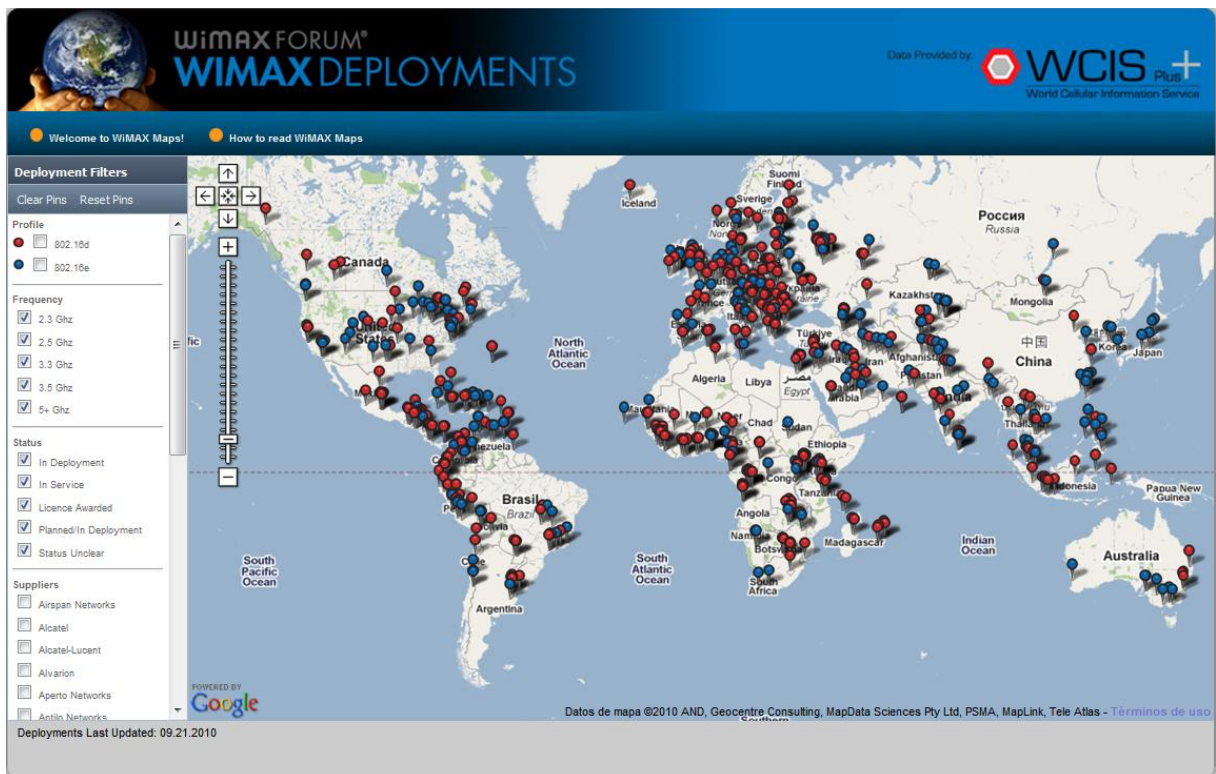


Fig. 15-10: Redes WiMax fijas (en rojo) y móviles (en azul) que están en servicio o en implementación a Octubre de 2010. Fuente <http://wimaxmaps.org/>.

15.10.2 Disponibilidad de equipos

A Octubre de 2010 están disponibles equipos certificados en dos perfiles de *Fixed* WiMAX y cinco de *Mobile* WiMAX. Están disponibles 247 productos diferentes certificados, 62 estaciones base de 26 vendedores, y 185 estaciones de abonado de 50 vendedores. Actualmente, la mayoría de las estaciones de abonado son USB *dongles* y PC *cards*. También están disponibles algunas *notebooks* con WiMAX integrado. Intel ya ha integrado WiMAX en la frecuencia de 2,5MHz en su chipset Centrino y planea incluir todos las demás bandas de

frecuencia de *Mobile* WiMAX. Muy pocos dispositivos WiMAX de mano, tales como teléfonos celulares, están disponibles hasta ahora.

16. Bibliografía

1. **IEEE Computer Society and the IEEE Microwave Theory and Techniques Society.** *IEEE Std 802.16™-2009*, Standard for Local and metropolitan area networks, Part 16: Air Interface for Broadband Wireless Access Systems. Abril 2009.
2. **Paxon, Vern and Sally, Floyd.** *Wide-Area Traffic: The Failure of Poisson Modeling*. Julio 18, 1995.
3. **Lawrence Berkeley Laboratory.** [Online] <http://ita.ee.lbl.gov/html/contrib/LBL-PKT.html>.
4. **Veitch, Darryl and Abry, Patrice.** *A Wavelet Based Joint Estimator of the Parameters of Long-Range Dependence*. s.l. : pp 878-897, Vol. 45(3), IEEE Trans. Info. Theory, special issue "Multiscale Statistical Signal Analysis and its Applications", Abril 1999.
5. **Abry, P., et al.** *Self-similarity and long-range dependence through the wavelet lens*. Enero 2000.
6. **Uhlig, Steve, Bonaventure, Olivier and Rapiet, Chris.** *3D-LD: a Graphical Wavelet-based Method for Analyzing Scaling Processes*. s.l. : Proc. of the 15th Specialist Seminar, 2002.
7. **Uhlig, Steve.** *Non-stationarity and high-order scaling in TCP flow arrivals: a methodological analysis*. s.l. : ACM SIGCOMM Computer Communication Review, Abril 2004.
8. **Comer, Douglas E.** *Internetworking with TCP/IP, Principles, Protocols, and Architecture*. s.l. : Department of Computer Sciences Purdue University, Fourth Edition, Prentice Hall, 2000.
9. **Stevens, W. Richard.** *TCP/IP Illustrated, Volume 1, The Protocols*. s.l. : Addison-Wesley, 16th Printing, Febrero 2000.
10. **Park, Kihong and Willinger, Walter.** *Self-Similar Network Traffic and Performance Evaluation*. s.l. : John Wiley & Sons, Inc., 2000.
11. **Hae-Duck, Joshua Jeong.** *Modeling of Self-Similar Teletraffic for Simulation*. s.l. : Doctoral Thesis in Computer Science, University of Canterbury, Julio 2002.
12. **Taqqu, Murad S., Willinger, Walter and Sherman, Robert.** *Proof of a Fundamental Result in Self-Similar Traffic Modeling*. s.l. : Computer Communication Review 27, Mayo 1997.
13. **Le-Ngoc, T. and Subramnian, S. N.** *A Pareto-modulated Poisson process (PMPP) model for long-range dependent traffic*. s.l. : Computer Communications 23, 2000.

14. **Neame, Timothy D., Zukerman, Moshe and Addie, Ronald G.** *Modeling Broadband Traffic Streams*. s.l. : Proceedings of Globecom '99, Rio de Janeiro, Brazil, 5 – 9 Diciembre 1999.
15. **Leland, Will E., y otros.** *On the Self-similar Nature of Ethernet Traffic (Extended Version)*. Febrero 1994.
16. **Crovella, Mark E. and Bestavros, Azer.** *Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes*. s.l. : IEEE/ACM Transactions On Networking, Vol. 5, NO. 6, Diciembre 1997.
17. **Willinger, W., et al.** *Selfsimilarity through high-variability: Statistical analysis of Ethernet LAN traffic at the source level*. s.l. : IEEE/ACM Trans. Networking, vol. 5, pp. 71–86, Febrero 1997.
18. **Addie, Ronald G., Neame, Timothy D. and Zukerman, Moshe.** *Modeling Superposition of Many Sources Generating Self Similar Traffic*. s.l. : Proceedings of ICC '99, Vancouver, BC, Canada, pp 387 – 391, 6 – 10 Junio 1999.
19. **Neame, Timothy D., Zukerman, Moshe and Addie, Ronald G.** *Applying Multiplexing Characterization to VBR Video Traffic*. s.l. : Proceedings of ITC 16, Edinburgh, Scotland, UK, pp 846-856. , 7-11 Junio 1999.
20. **Muscariello, L., et al.** *Markov models of internet traffic and a new hierarchical MMPP model*. s.l. : Elsevier, Marzo 2005.
21. **Salvador, Paulo, Valadas, Rui and Pacheco, Antonio.** *Multiscale Fitting Procedure using Markov Modulated Poisson Processes*. s.l. : Kluwer Academic Publishers, Enero 2003.
22. **Taqqu, M.S. and Levy, J.B.** *Using renewal processes to generate long range dependence and high variability*. s.l. : E. Eberlein, M.S. Taqqu (Eds.), Dependence in Probability and Statistics, Progress in Probability and Statistics, 11, Birkhauser, Boston, pp. 73-89, 1986.
23. **Chou, J., et al.** *IEEE 802.16 Medium Access Control and Service Provisioning*. s.l. : Intel Technology Journal, Volume 8, Issue 3, Agosto 20, 2004.
24. **de Castro, F.** [Online] <http://www.mathworks.com/matlabcentral/fileexchange/24938-anyrnd>.
25. **Grondalen, Ole.** *A Tutorial Overview of WiMAX*. s.l. : Investigador y científico de Telenor, Telektronikk 1, 2010.