



## MODELO DE PÉRDIDA DE PAQUETES PARA REDES HÍBRIDAS

Memoria del Trabajo Final de Carrera  
de Ingeniería Técnica de Telecomunicación,  
especialidad Sistemas Electrónicos  
realizado por  
Jose Antonio Del Peral Rosado  
y dirigido por  
Maria Ángeles Vázquez Castro  
Bellaterra, 9 de julio de 2007



Universitat  
Autònoma  
de Barcelona

**etse**

Escola Tècnica Superior d'Enginyeria

*Agradecimientos a María Ángeles Vázquez Castro, directora de este proyecto, por el apoyo moral y la ayuda ofrecida en cada tutoría. Gracias también a David Pradas, profesor de la titulación, por los recursos facilitados. También a los amigos que incluso han ayudado a realizar medidas. Y, especialmente agradezco a mis padres y familiares el aguante que han tenido durante todo el proyecto.*

# Contenidos

<b>Contenidos .....</b>	<b>1</b>
<b>Índice de figuras .....</b>	<b>3</b>
<b>Índice de tablas .....</b>	<b>6</b>
<b>1. Introducción .....</b>	<b>7</b>
1.1    Principios de funcionamiento en una red IEEE 802.11 .....	7
1.1.1    Capa física .....	7
1.1.1.1.    Especificaciones .....	8
1.1.1.2.    Topologías de la arquitectura de la red.....	8
1.1.2    Capa de enlace de datos.....	10
1.2    Estado del arte de los algoritmos.....	11
1.2.1    Basados en datos de la capa de enlace.....	11
1.2.1.1.    Adaptive ARF (AARF) .....	12
1.2.1.2.    SampleRate.....	13
1.2.1.3.    Hybrid Rate Control (HRC) .....	13
1.2.1.4.    Robust Rate Adaptation Algorithm (RRAA) .....	14
1.2.2    Basados en datos de la capa física .....	14
1.2.2.1.    Receiver-Based AutoRate (RBAR).....	15
1.2.2.2.    Opportunistic Auto-Rate (OAR) .....	15
1.3    Motivación y objetivos .....	16
1.4    Elección de los algoritmos.....	16
1.5    Estructura de la memoria.....	17
<b>2. Modelado del canal para la simulación .....</b>	<b>18</b>
<b>3. Algoritmo ARF .....</b>	<b>20</b>
3.1    Definición del algoritmo.....	20
3.2    Programa de simulación .....	21
3.3    Resultados obtenidos en la simulación .....	26
3.3.1    Simulación 1 .....	26
3.3.2    Simulación 2 .....	28
3.3.3    Simulación 3 .....	30
3.4    Conclusiones.....	31
<b>4. Algoritmo LD-ARF .....</b>	<b>33</b>
4.1    Principal problema del algoritmo arf.....	33
4.2    Definición del algoritmo.....	34
4.2.1    Funcionamiento del estándar para la diferenciación de pérdidas .....	35
4.2.2    Funcionamiento del algoritmo LD-ARF .....	37
4.3    Programa de simulación .....	38
4.4    Resultados obtenidos en la simulación .....	39
4.4.1    Simulación 1: una estación transmisora .....	39
4.4.2    Simulación 2: más de una estación transmisora .....	41

4.4.3	Simulación 3: colisiones en la recepción.....	44
4.5	Conclusiones.....	47
<b>5.</b>	<b>Modelado del canal basado en medidas experimentales.....</b>	<b>48</b>
5.1	Equipo utilizado.....	48
5.2	Estrategia 1: Proceso completo de intercambio de paquetes.....	49
5.3	Estrategia 2: Escucha del canal .....	50
5.4	Estrategia 3: Modo monitor.....	51
5.5	Modelado del canal físico.....	51
5.5.1	Prueba 1 .....	52
5.5.2	Prueba 2 .....	54
5.5.3	Prueba 3 .....	56
5.5.4	Prueba 4 .....	57
<b>6.</b>	<b>Comparación entre algoritmos.....</b>	<b>60</b>
6.1	Comparación con modelo de canal simulado .....	60
6.2	Comparación con modelo del canal físico.....	62
<b>7.</b>	<b>Canal híbrido .....</b>	<b>67</b>
<b>8.</b>	<b>Conclusiones.....</b>	<b>69</b>
<b>Apéndice .....</b>		<b>70</b>
<b>Resumen .....</b>		<b>75</b>
<b>Bibliografía.....</b>		<b>77</b>

# Índice de figuras

<i>Figura 1. Arquitectura a nivel de capas en una red IEEE 802.11 [4].</i>	8
<i>Figura 2. Relación de alcance y velocidad en el 802.11b [3].</i>	8
<i>Figura 3. Conjunto de servicio extendido (ESS) [2].</i>	9
<i>Figura 4. Diagrama del modelo de Gilbert-Elliot con las transiciones de los estados [15].</i>	19
<i>Figura 5. Protocolo simple de parada y espera.</i>	22
<i>Figura 6. Efecto de time-out corto.</i>	22
<i>Figura 7. Llegadas según el proceso de Poisson con <math>\lambda=10</math> eventos/s.</i>	23
<i>Figura 8. Distribución de Poisson según su densidad de probabilidad.</i>	23
<i>Figura 9. Diagrama de flujo del funcionamiento del algoritmo ARF.</i>	25
<i>Figura 10. Comparación del throughput.</i>	26
<i>Figura 11. Diagrama de barras donde se muestra la sucesión de los paquetes enviados y recibidos por el emisor, y la velocidad de transmisión.</i>	27
<i>Figura 12. Comparación entre la velocidad de transmisión y el número de tramas de reconocimiento consecutivas, según el tiempo de aparición.</i>	28
<i>Figura 13. Comparación del throughput.</i>	29
<i>Figura 14. Tráfico de la segunda simulación.</i>	29
<i>Figura 15. Comparación entre velocidad de transmisión y ACK consecutivas en función del tiempo, para la segunda simulación.</i>	30
<i>Figura 16. Comparación del throughput.</i>	30
<i>Figura 17. Comparación entre la velocidad y las ACK consecutivas en función del tiempo, para la tercera simulación.</i>	31
<i>Figura 18. Ejemplo del problema del nodo oculto con tres estaciones.</i>	34
<i>Figura 19. Acceso básico y RTS/CTS en el estándar 802.11 DCF [6].</i>	35
<i>Figura 20. Tráfico de la estación transmisora según el modelo del canal de la simulación 1.</i>	39
<i>Figura 21. Comparación entre velocidad en función del número de paquete enviado (izquierda parte superior), y la velocidad en función del tiempo de llegada de cada paquete (derecha parte superior). En la parte inferior izquierda se muestra el modelo del canal en función de los paquetes enviados, y en la parte inferior derecha el modelo del canal en función del tiempo.</i>	40
<i>Figura 22. Comparación entre el throughput sin aplicar LD-ARF (barra azul) y aplicándolo (barra roja).</i>	40
<i>Figura 23. Ventana de comandos de Matlab, donde aparece la información sobre las transmisiones, proporcionada por el programa de simulación.</i>	41

<i>Figura 24. Tráfico de las tres estaciones. De izquierda a derecha las estaciones son: la 1, la 2 y la 3.</i>	41
<i>Figura 25. Velocidades de las diferentes estaciones en función del tiempo.</i>	42
<i>Figura 26. Throughput de las diferentes estaciones sin y con el algoritmo LD-ARF.</i>	42
<i>Figura 27. Velocidades de las cinco estaciones en función de los paquetes enviados (izquierda) y en función del tiempo (derecha).</i>	43
<i>Figura 28. Throughput de las diferentes estaciones.</i>	43
<i>Figura 29. Tráfico en la estación 1 habiendo diferenciado las pérdidas y detectando que se han producido colisiones en la estación receptora.</i>	44
<i>Figura 30. Tráfico en la estación 2 habiendo diferenciado las pérdidas y detectando que se han producido colisiones en la estación receptora.</i>	45
<i>Figura 31. Comparación entre los periodos de transmisión de las dos estaciones.</i>	45
<i>Figura 32. Comparación entre los throughput con y sin algoritmo en cada una de las estaciones.</i>	46
<i>Figura 33. Estructura gráfica con los datos de la simulación.</i>	46
<i>Figura 34. Dispositivos utilizados para realizar las medidas. En a) vemos uno de los portátiles, y en b) el yellowjacket.</i>	48
<i>Figura 35. Procesado de los datos de las llegadas de los paquetes.</i>	49
<i>Figura 36. Distribución en función de la frecuencia las diferentes señales de potencia.</i>	50
<i>Figura 37. Nivel de potencia de la señal recibida en función del número de paquetes para la prueba 1.</i>	52
<i>Figura 38. Estados de markov en función del tiempo para la primera prueba.</i>	53
<i>Figura 39. Diagrama de los subestados que se encuentran en cada estado de Markov.</i>	53
<i>Figura 40. Nivel de potencia de la señal recibida en función del número de paquetes para la prueba 2.</i>	54
<i>Figura 41. Estados de Markov en función del tiempo para la segunda prueba.</i>	54
<i>Figura 42. Plano de situación de los estados para la prueba 1 y 2.</i>	55
<i>Figura 43. Nivel de potencia de la señal recibida en función del número de paquetes para la prueba 3.</i>	56
<i>Figura 44. Estados de Markov en función del tiempo para la tercera prueba.</i>	56
<i>Figura 45. Plano de situación de los estados para la prueba 3.</i>	57
<i>Figura 46. Nivel de potencia de la señal recibida en función del número de paquetes para la prueba 4.</i>	57
<i>Figura 47. Estados de Markov en función del tiempo para la cuarta prueba.</i>	58
<i>Figura 48. Detalle de los estados del proceso determinístico.</i>	59
<i>Figura 49. Plano de situación de los estados para la prueba 4.</i>	59
<i>Figura 50. Comparación en la estación 1 aplicando el ARF y el LD-ARF.</i>	60
<i>Figura 51. Comparación en la estación 2 aplicando el ARF y el LD-ARF.</i>	61

<i>Figura 52. Comparación de las velocidades de cada estación aplicando los dos algoritmos.</i>	61
<i>Figura 53. Comparación en los niveles de throughput en los dos casos.</i>	62
<i>Figura 54. Tráfico de la estación 1 con el modelo de canal físico y aplicando el ARF.</i>	63
<i>Figura 55. Tráfico de la estación 2 con el modelo de canal físico y aplicando el ARF.</i>	63
<i>Figura 56. Velocidades de las dos estaciones en función del tiempo.</i>	64
<i>Figura 57. Throughput de las dos estaciones cuando se aplica el ARF.</i>	64
<i>Figura 58. Tráfico de la estación 1 con el modelo de canal físico y aplicando el LD-ARF.</i>	65
<i>Figura 59. Tráfico de la estación 2 con el modelo de canal físico y aplicando el LD-ARF.</i>	65
<i>Figura 60. Velocidades de las dos estaciones en función del tiempo.</i>	66
<i>Figura 61. Throughput de las dos estaciones cuando se aplica el LD-ARF.</i>	66
<i>Figura 62. Diferentes modelos de atenuación por lluvia.</i>	68
<i>Figura 63. Diferentes modelos de lluvia SNIR para un usuario específico.</i>	68

# Índice de tablas

<i>Tabla 1. Valores de las transiciones de los estados internos en los estados del proceso determinista.</i>	19
<i>Tabla 2. Especificaciones de las tasas de envío de datos para 802.11b [6].</i>	22
<i>Tabla 3. Resultados obtenidos en el estudio basado en cien realizaciones por cada prueba estadística.</i>	24
<i>Tabla 4. Características de la primera simulación.</i>	26
<i>Tabla 5. Características de la segunda simulación.</i>	29
<i>Tabla 6. Características de la tercera simulación.</i>	30
<i>Tabla 7. Estructura del algoritmo LD-ARF.</i>	37
<i>Tabla 8. Probabilidades de los diferentes estados de Markov.</i>	53
<i>Tabla 9. Probabilidades de los diferentes estados de Markov para la simulación.</i>	55
<i>Tabla 10. Probabilidades de los diferentes estados de Markov para la simulación.</i>	57
<i>Tabla 11. Probabilidades de los diferentes estados de Markov para la simulación.</i>	58
<i>Tabla 12. Resultados del throughput en los diferentes casos.</i>	62

# Capítulo 1

## Introducción

En este apartado se introducirán los principales temas que se estudiarán a lo largo de este proyecto. La primera parte se basará en las redes locales inalámbricas (WLAN, de las siglas en inglés) y ocupará la mayor parte del trabajo. Finalmente se desarrollará teóricamente el canal híbrido, formado por el canal satélite y el canal terrestre.

### 1.1 PRINCIPIOS DE FUNCIONAMIENTO EN UNA RED IEEE 802.11

Este primer apartado se centrará en la WLAN que se rige por la norma IEEE 802.11. A continuación, se explicará la versión más utilizada del estándar, la IEEE 802.11b, comercialmente conocida como Wi-Fi (Wireless Fidelity). El hecho por el cual se haya extendido tanto este modelo radica en la portabilidad del sistema y el bajo coste, manteniendo unas velocidades de transmisión razonables.

El estándar IEEE 802.11 es el protocolo de comunicaciones que define el uso de los dos niveles más bajos de la arquitectura OSI<sup>1</sup> (capa física y enlace de datos), especificando sus normas de funcionamiento en una WLAN. Así explicaremos el funcionamiento de esta red centrándonos en cada una de estas dos capas que se rigen por el estándar. Se puede observar la arquitectura a nivel de capas en la *Figura 1*.

#### 1.1.1 CAPA FÍSICA

La capa física del modelo de referencia OSI es la encargada de las conexiones físicas que se establecen entre el ordenador y la red. Así, se explicarán a continuación las especificaciones del medio en el que se transmitirá la información, los componentes que formarán la red y la topología de la misma.

---

<sup>1</sup> El modelo de referencia OSI (Open System Interconnection) describe como debe transmitirse información desde una aplicación de software de un ordenador hasta otra aplicación en otro ordenador a través de una red.

### 1.1.1.1. Especificaciones

Si nos centramos en la versión del estándar IEEE 802.11b, cualquier dispositivo podrá utilizar este sistema, ya que trabaja a la frecuencia libre de 2,4GHz, es decir, que no precisa de autorización o algún permiso para utilizarlas. Aunque hay que tener en cuenta, que la normativa puede variar según el país. En cuanto a las ondas, estas se transmiten vía radio utilizando una modulación de espectro ensanchado (DSSS, *Direct Sequence Spread Spectrum*). Mediante DSSS, el modelo puede soportar cuatro velocidades de transmisión, 1 Mbps, 2 Mbps, 5.5 Mbps y 11 Mbps. Pero estas tasas se verán limitadas según la distancia a la cual se quieran enviar los datos. Como se observa en la *Figura 2*, el estándar puede proporcionar una velocidad de 11 Mbps como máximo a 30 m., y si nos alejamos hasta 100 m., la velocidad no excederá de 1 Mbps.

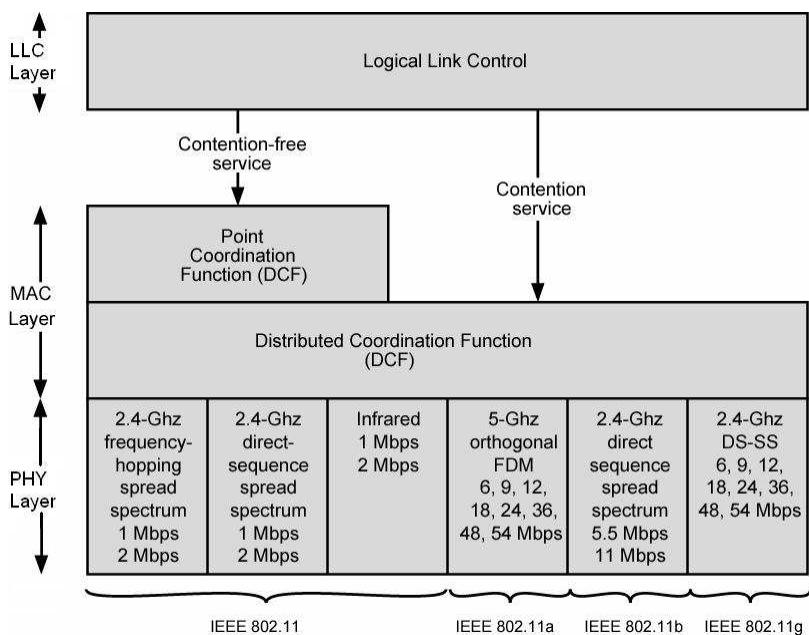


Figura 1. Arquitectura a nivel de capas en una red IEEE 802.11 [4].

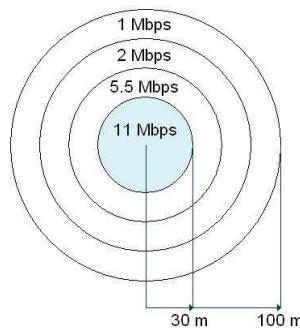


Figura 2. Relación de alcance y velocidad en el 802.11b [3].

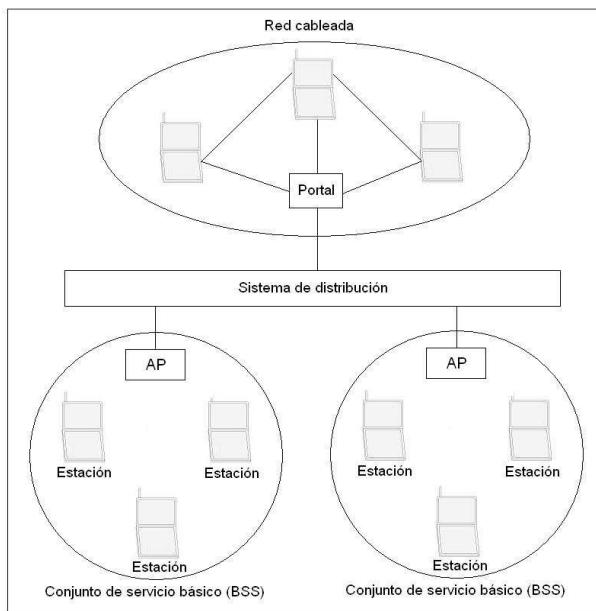
### 1.1.1.2. Topologías de la arquitectura de la red

Las redes de área local se pueden clasificar en dos tipos, las redes de infraestructura y las redes independientes (Ad Hoc LANs).

## 1. Introducción

---

Las redes de infraestructura contienen nodos especiales llamados puntos de acceso (AP), que están conectados a otras redes. Normalmente, establecen un enlace físico entre la red inalámbrica y la red cableada, haciendo las funciones de puente entre ambas y realizando las conversiones de las tramas pertinentes. En las redes de infraestructura diferentes estaciones se comunican entre ellas a través de este punto de acceso. Cuando hay un conjunto de estaciones que se comunican entre ellas se denomina conjunto de servicio básico (BSS). En la *Figura 3* podemos observar este tipo de red en un conjunto de servicio extendido (ESS), es decir, la unión de varias BSS.



*Figura 3. Conjunto de servicio extendido (ESS) [2].*

En la *Figura 3*, también se puede observar como hay una parte del esquema llamada sistema de distribución. Este es el mecanismo encargado de controlar la situación de las estaciones para enviarles las tramas. Además este sistema está adjunto a un portal, que es una puerta de enlace para el acceso a Internet o a otro conjunto de servicio extendido (ESS).

Por otra parte, las redes independientes o “ad hoc” no necesitan una infraestructura fija. Este tipo de redes permiten la conexión directa de dispositivos entre sí, sin necesidad de disponer de un punto de acceso que centralice la gestión de las comunicaciones inalámbricas de la red. El hecho por el cual los nodos estén conectados entre sí, permite que la información pueda viajar por diferentes caminos para llegar a un mismo destino. De esta forma, cuando se saturan algunos terminales, la red no queda colapsada completamente. En cambio, si se envían muchas peticiones a un punto de acceso, este tiene que realizar una cola de espera, ralentizando así la red. Para más información sobre este tipo de redes consultar [2].

### **1.1.2 CAPA DE ENLACE DE DATOS**

En el ámbito de los estándares IEEE 802, la capa de enlace de datos está formada por dos sub-capas, el control del enlace lógico (*Logical Link Control, LLC*) y el control de acceso al medio (*Medium Access Control, MAC*).

La capa LLC está definida en el estándar IEEE 802.2 y sirve para toda la familia de redes IEEE 802. El control del enlace lógico especifica los mecanismos para el direccionamiento de estaciones conectadas al medio y para controlar el intercambio de datos entre usuarios de la red. Se basa en el protocolo de comunicaciones de datos HDLC. Este protocolo proporciona la recuperación de errores en caso de pérdida de paquetes, o fallos en la secuencia, entre otros. Así ofrece una comunicación fiable punto a punto entre el transmisor y el receptor.

En cuanto a la capa MAC, ésta permite el acceso de diferentes dispositivos a la red compartiendo la capacidad de transmisión. Esta será la capa en la que se centre el resto de este trabajo, ya que es la encargada de coordinar la transferencia de paquetes.

La norma IEEE 802.11b [1] marca las especificaciones de esta capa. Según el estándar, se define un protocolo de control de acceso al medio llamado MAC inalámbrico de principio distribuido (DFWMAC). En este protocolo se divide la capa MAC en dos sub-capas que ofrecen dos modos de transmisión para los paquetes de datos: la Función de Coordinación Distribuida (*Distributed Coordination Function, DCF*), y la Función de Coordinación Puntual (*Point Coordination Function, PCF*).

En primer lugar, la sub-capa inferior, la DCF, gestiona el control de acceso utilizando un mecanismo de contienda o proceso de contención llamado CSMA/CA<sup>2</sup>. Este consiste en comprobar si el canal está libre antes de transmitir. Si el canal está desocupado se empleará un algoritmo exponencial de retracción o *backoff* para evadir las colisiones, con el cual se esperará un tiempo aleatorio por si intentan transmitir las otras estaciones. De esta forma se distribuye el acceso, pero las estaciones siguen compitiendo para transmitir. Esta gestión de control deberá ser soportada por todas las implementaciones.

Por otro lado, la PCF coordina el acceso al medio mediante un mecanismo centralizado en el punto de acceso. Este mecanismo se basa en el sondeo o *polling*, y consiste en preguntar a cada estación si desea transmitir. Si diferentes estaciones quieren acceder al medio, el AP dará permiso a una única estación en cada momento. El PCF es un método de acceso opcional, pero es muy útil para dar servicios de prioridad a estaciones que lo necesiten. En cambio, este mecanismo se limita a la implementación

---

<sup>2</sup> CSMA/CA significa *Carrier Sense Multiple Access With Collision Avoidance*, y se define como un protocolo que mediante la detección de la portadora puede evitar colisiones entre los paquetes de datos.

en redes de infraestructura, ya que se basa en los puntos de acceso, y por este motivo no se puede utilizar en redes *ad hoc*.

### 1.2 ESTADO DEL ARTE DE LOS ALGORITMOS

En los últimos años, se han propuesto un gran número de algoritmos para la adaptación de la tasa de transmisión, pero solo algunos han sido implementados en productos reales. La mayoría se basan en estimar la calidad del canal y ajustar la tasa según los resultados obtenidos en la transmisión.

Sin embargo, la adaptación de la tasa no se especifica en el estándar 802.11 para redes WLAN. Esto es debido a la naturaleza dinámica y cambiante del canal inalámbrico. Los errores producidos en el canal pueden ser tan diversos como el origen de los mismos. El desplazamiento del transmisor, la variabilidad del canal o la aparición de nodos ocultos pueden ser algunas de sus causas. Por este motivo, gran parte de los algoritmos implementados se verán limitados según el escenario en el que nos encontremos. Así el fabricante de la tarjeta de red inalámbrica, deberá escoger entre alguno de los algoritmos existentes o crear su propia implementación, dependiendo de sus necesidades.

A continuación, se clasificarán algunos de los algoritmos según el método que utilizan para realizar las estimaciones del canal y como actúan en consecuencia con estas. Así, podemos encontrar algoritmos que se basan en estadísticas obtenidas de la capa de enlace como pueden ser el número de transmisiones exitosas, o a partir de medidas del canal físico como puede ser la relación señal a ruido (SNR) del canal.

#### 1.2.1 BASADOS EN DATOS DE LA CAPA DE ENLACE

Los algoritmos que se explicarán en este apartado, utilizan de manera sencilla la información que reciben de la comunicación. Para obtener las condiciones del enlace y poder adaptar la tasa con mayor eficiencia, solo es necesario evaluar estadísticamente los datos transmitidos con parámetros de la capa de enlace. Así podremos adaptar la tasa de envío mediante el *frame error rate* (FER) o tasa de error de trama, las transmisiones reconocidas, o el *throughput*<sup>3</sup> alcanzado. Como estos datos estadísticos están relacionados con el nivel de *throughput* del usuario, se garantiza el máximo *throughput* a largo plazo. Estos factores (sencillez y *throughput*) explican el predominio de estos algoritmos en los actuales productos 802.11, según [9].

Una vez son obtenidos los datos, se deben tratar para poder decidir si se va a cambiar la tasa. Para ello existen varios métodos que se explicarán a continuación.

---

<sup>3</sup> **Throughput.** Es la capacidad de un enlace de transportar información útil. Representa la cantidad de información útil que puede transmitirse por unidad de tiempo. También se le puede llamar tasa de datos correctos recibidos o velocidad de datos recibidos correctamente.

En primer lugar, se puede decrementar la velocidad de transmisión cuando se produzcan graves pérdidas de paquetes. Estas pérdidas se detectan a partir de un gran número retransmisiones, un límite de trama pérdidas, o transmisiones consecutivas fallidas. Con esto se pretende conseguir, que un transmisor adapte su tasa a las peores condiciones del canal, cuando sufra una significativa pérdida de paquetes. El problema de este método es que solo se corresponde con determinados escenarios previstos en el diseño, y puede darse un pésimo *throughput* en otras condiciones, como la del terminal oculto. Este problema se puede solucionar diferenciando las causas por las que se producen errores en los envíos, y así poder actuar correctamente.

El segundo método, se basa en enviar paquetes de sondeo a una velocidad diferente a la actual, para determinar si tiene un rendimiento mejor. Sin embargo, el reducido número de paquetes que se utiliza, puede producir una adaptación imprecisa de la tasa. Puede ser demasiado optimista respecto a un buen resultado, o demasiado pesimista respecto a uno malo.

Finalmente, el tercer método consiste en incrementar la tasa después de un cierto número de transmisiones consecutivas con éxito, y recuperar la tasa anterior si la transmisión es fallida. La ventaja de este método es la rápida respuesta (se utilizan pocas tramas), y el gran inconveniente es que la mayoría de escenarios reales tiene una probabilidad aleatoria de pérdidas. Así, las ráfagas de errores relativamente cortas causan grandes caídas en el *throughput*. Esto ocurre, ya que para incrementar la tasa se necesitan más transmisiones consecutivas que para decrementar, debido a la necesidad de acumular un FER más significativo y prevenir oscilación.

Los anteriores métodos se utilizan en muchos algoritmos como el ARF [5], el AARF [7], el SampleRate [8] o el HRC [9]. Aunque también hay novedosos algoritmos como el RRAA [10] que se desmarcan de los métodos anteriores pero que se encuentran en la misma capa de enlace de datos. Respecto al algoritmo ARF, se tratará a fondo en el *Capítulo 3*, y los otros algoritmos se explicaran brevemente a continuación.

### 1.2.1.1. Adaptive ARF (AARF)

El algoritmo AARF [7] está basado en el ARF [5], que requiere un mínimo de 10 paquetes reconocidos para cambiar a una tasa superior cuando las condiciones de la transmisión sean favorables. La mejora que se implementa con el AARF es que el límite de paquetes reconocidos se puede variar, para reflejar mejor las condiciones del canal.

Cuando la transmisión de paquetes de sondeo es fallida, se vuelve inmediatamente a la anterior tasa inferior (como en ARF), pero también se multiplica por dos el número de transmisiones consecutivas (con un máximo de 50) necesarias para aumentar la tasa.

Este límite o umbral se resetea a un valor inicial de 10 cuando la tasa se decrementa por el fallo de dos transmisiones consecutivas.

El efecto de este mecanismo de adaptación, es aumentar el tiempo transcurrido entre un intento fallido de subir la tasa, y el siguiente consecutivo. En producirse menos transmisiones fallidas y retransmitidas, se mejora el *throughput* total.

### 1.2.1.2. SampleRate

Cuando se empieza a enviar paquetes en un enlace, el algoritmo SampleRate [8] envía datos a la mayor velocidad posible. SampleRate deja de enviar a una determinada tasa si se detectan cuatro fallos consecutivos, de manera que se decrementará la velocidad de transmisión hasta que una sea capaz de enviar paquetes correctamente. Cada diez paquetes de datos, el algoritmo prueba aleatoriamente una tasa entre las posibles para comprobar si hay alguna que ofrezca mayor rendimiento. No se utilizará una tasa si los cuatro paquetes consecutivos más recientes no han sido reconocidos, o si el tiempo de transmisión sin pérdidas (sin retransmisiones) ha superado al tiempo medio de transmisión específico para la tasa actual.

Para calcular cada tiempo medio de transmisión de la velocidad correspondiente, SampleRate utiliza información de la tarjeta inalámbrica para determinar cuánto tiempo se necesita para transmitir un paquete. Por cada paquete transmitido, las tarjetas inalámbricas 802.11 indican si el paquete fue reconocido con éxito y cuando fue enviado por la tarjeta. Este algoritmo calcula el tiempo de transmisión por cada paquete utilizando la longitud del paquete, la velocidad de envío, y el número de retransmisiones (incluyendo el mecanismo de contención o *back-off* del 802.11). Así, el tiempo medio de transmisión para una velocidad determinada, es calculado mediante la media de tiempos de transmisión de los anteriores paquetes con esa tasa.

### 1.2.1.3. Hybrid Rate Control (HRC)

El algoritmo HRC [9] combina las ventajas de los algoritmos basados en la capa de enlace y los basados en la capa física. El objetivo es soportar aplicaciones de *streaming*<sup>4</sup> limitando el retardo y el *jitter*<sup>5</sup> de cada paquete tanto como sea posible, incluso empeorando el *throughput* cuando sea necesario.

A pesar de ser un algoritmo híbrido, su controlador principal de la tasa de transmisión se basa en datos estadísticos obtenidos a partir de la capa de enlace. En concreto, utiliza el segundo método que consistía en ir variando la tasa de transmisión para comprobar cuál se adaptaba mejor a las condiciones del canal.

---

<sup>4</sup> **Streaming.** Emisión en tiempo real de algún tipo de medio multimedia (audio o video).

<sup>5</sup> **Jitter.** Es el efecto por el cual el retardo entre paquetes no es constante. Se trata de una latencia variable producida por la congestión de tráfico en la red.

Sin embargo, las decisiones del controlador principal pueden ser anuladas a partir de un controlador secundario basado en los datos obtenidos de la capa física. Este controlador limita el rango aceptable de nivel de señal de las tramas reconocidas (SSIA, *Signal Strength Indication of the Acknowledged*) por cada tasa de transmisión, basado en la curva *Throughput-vs-SNR* mostrada en la figura 1 de [9]. Los límites para la máxima tasa son ajustados cuando las condiciones del canal cambian rápidamente. Así, las medidas del canal físico como el SNR nos permiten evitar retransmisiones innecesarias disminuyendo la velocidad de transmisión cuando las condiciones del canal varían muy rápidamente, como cuando el usuario se está moviendo,

Este esquema se basa en escoger una tasa más baja para minimizar el *jitter* del paquetem debido a la reducción de la probabilidad de error de trama. Las rápidas variaciones del canal son detectados por las lecturas consecutivas del SSIA.

En [9] se concluye que la información del SNR puede ser utilizada a pesar sus problemas prácticos asociados, como el ruido o las derivas. Así se emplea este parámetro para limitar el rango de los ajustes realizables que puede determinar el controlador principal. Esto evita que se utilice una alta velocidad de transmisión cuando las condiciones del canal de repente empeoren.

### 1.2.1.4. Robust Rate Adaptation Algorithm (RRAA)

El algoritmo RRAA [10] intenta ser robusto en frente de las pérdidas aleatorias, y sensible respecto a los cambios drásticos del canal.

En primer lugar, debería mantener la misma tasa de transmisión y el mismo *throughput*, con la presencia de variaciones aleatorias y poco severas en el canal. En segundo lugar, debe responder rápidamente a cambios significativos del enlace. Especialmente, el algoritmo debe asociar un decremento o incremento de la tasa con un cambio en el canal, cuando la calidad de este empeore o mejore, como un móvil que se aleja o se acerca al punto de acceso (AP). También debe responder frente a degradaciones severas del canal producidas por fuentes interferentes como estaciones ocultas, o dispositivos que operen en la misma banda frecuencial.

El diseño del algoritmo se basa en dos ideas novedosas. La primera consiste en utilizar la relación de pérdidas a corto plazo para evaluar el canal, y oportunamente adaptar la tasa a sus variaciones dinámicas. La segunda, es utilizar la opción RTS de forma adaptativa para filtrar pérdidas provocadas por colisiones.

## 1.2.2 BASADOS EN DATOS DE LA CAPA FÍSICA

Algunos algoritmos utilizan las medidas de la capa física como el SNR para realizar la estimación de la tasa. Como ya hemos visto, este parámetro es utilizado en la presencia

de clientes móviles en HRC [9]. También han sido aplicados en algoritmos como RBAR [11] y OAR [12], aunque estos no cumplen estrictamente con el estándar 802.11. En teoría, las medidas de la capa física deberían producir estimaciones precisas.

Sin embargo, el canal inalámbrico ofrece dificultades respecto a este método. En la realidad, para ciertas condiciones en el enlace la relación entre la tasa óptima y el SNR es muy variable. Esto es debido a las imperfecciones de los modelos que describen el canal radio, y también porque la calidad del enlace depende hasta cierto punto de otros parámetros. Tampoco es fácil obtener una estimación fiable del SNR de un enlace. Las variaciones en el SNR pueden producir estimaciones imprecisas de la tasa.

### 1.2.2.1. Receiver-Based AutoRate (RBAR)

El algoritmo RBAR [11] escoge la tasa basándose en las medidas del SNR obtenidas en el receptor. Básicamente se centra en la medida de la calidad del canal desde el receptor para determinar si un paquete puede ser recibido. Cuando a un receptor le llega un paquete RTS (Request To Send), este calcula la máxima velocidad basándose en el nivel de señal medido y en rangos de SNR establecidos *a priori* con un modelo de canal. El receptor responde con un paquete CTS a la tasa calculada, y el emisor utilizará esta tasa para enviar los paquetes de datos.

RBAR fue diseñado para trabajar bien en entornos móviles donde las condiciones del canal cambiaban en el periodo de tiempo de envío de un paquete. RBAR supone que el dispositivo del receptor está preparado para poder calcular la mejor tasa de transmisión a partir del SNR de un paquete RTS.

Con este sistema surgen algunos inconvenientes que se resumirán a continuación. En primer lugar, se requiere un conocimiento previo del modelo del canal para calcular los valores del SNR, que no es trivial. En segundo lugar, se necesitan receptores que puedan medir el SNR, requerimiento difícil si queremos que los dispositivos de una WLAN sean de bajo coste. Finalmente, el algoritmo de adaptación de la tasa, no contempla los errores producidos por colisiones.

### 1.2.2.2. Opportunistic Auto-Rate (OAR)

El algoritmo OAR [12] consigue las tasas más altas cuando la calidad del canal cambia rápidamente. Los terminales de este algoritmo envían seguidos los paquetes de datos siempre y cuando la calidad del canal sea buena. Esto permite incrementar el *throughput* del canal debido al menor tiempo utilizado para las transmisiones.

La idea de OAR es aprovechar la buena calidad de un enlace cuando esta aparezca, de tal manera que se incremente el *throughput* del canal. OAR utiliza el intercambio RTS/CTS para el control de la tasa (como en RBAR), pero otorga a cada emisor el

mismo tiempo para enviar una trama CTS como para transmitir un paquete a la velocidad base. Los otros terminales que escuchen el CTS no enviarán durante este tiempo, y si está bien modelado el enlace, el emisor podrá enviar múltiples paquetes a una tasa más alta en el mismo tiempo que una transmisión a una tasa más baja.

### 1.3 MOTIVACIÓN Y OBJETIVOS

La tecnología de red inalámbrica está siendo un revulsivo en el marco de las telecomunicaciones. Cada vez hay más usuarios que desean obtener un servicio determinado en cualquier momento y desde cualquier punto. Por este motivo siguen aumentando los sistemas de este tipo. Así se crea la necesidad de mejorar la comunicación en todos los niveles.

Con este proyecto se pretende modelar el canal inalámbrico y poder garantizar cierta velocidad de recepción de datos correctos. Para ello se deberá crear un programa capaz de simular el tráfico producido en el canal físico, a partir de su modelado, para finalmente obtener una visión sobre el posible *throughput* que se podrá conseguir en ese medio.

Con este resultado podremos, enlazar nuestro canal terrestre con el canal satélite ya que el *throughput* alcanzado determinará su velocidad de transmisión. Así podríamos crear un modelo de pérdida de paquetes en canal terrestre de una red híbrida.

Como motivación personal, cabe destacar la ambición por crear un programa de estas características, y el interés por las medidas de campo a la hora de realizar el modelado del canal físico.

### 1.4 ELECCIÓN DE LOS ALGORITMOS

Una vez se han analizado los diferentes tipos de algoritmos que existen, se ha decidido que en el presente proyecto solo se estudiará el *Automatic Rate Fallback* (ARF) [5] y el *Loss Differentiating ARF* (LD-ARF) [6].

El ARF fue el primer algoritmo de adaptación de la tasa de transmisión que se implementó en productos WLAN, al que siguieron otros como RBAR. Este último, utiliza un sistema complejo basado en medidas físicas, el cual necesita dispositivos especializados para poder obtener una adaptación correcta de la tasa. Sin embargo, el algoritmo ARF utiliza pocos recursos para poder ser implementado, procesa exclusivamente los datos que obtiene de la capa de enlace. Además es el algoritmo más utilizado en este campo. Así, se ha elegido el ARF por su sencillez y bajo coste de aplicación.

Por otra parte, el ARF tiene sus limitaciones ya que en determinados escenarios no obtiene resultados favorables. Por lo tanto, se ha buscado un nuevo algoritmo que pueda hacer frente a estas limitaciones. Este algoritmo se llama LD-ARF, e intenta paliar las deficiencias del anterior para poder obtener una mayor eficiencia. Pese a los cambios que se han realizado, se sigue manteniendo la sencillez del sistema ya que se actúa en la capa MAC sin afectar en la configuración del estándar 802.11.

## 1.5 ESTRUCTURA DE LA MEMORIA

Esta memoria estará formada por nueve capítulos, incluyendo la bibliografía como último. A continuación de esta introducción, se explicarán diversas técnicas para el modelado de un canal para la simulación. Esta simulación se detallará en el capítulo tres, y tratará de mostrar el funcionamiento del algoritmo de adaptación de la tasa frente a la pérdida de paquetes del canal. Seguidamente, y relacionado con el anterior, en el cuarto capítulo se explicará la evolución del algoritmo ARF, llamada LD-ARF, y el cual se basará en la diferenciación de pérdidas producidas por colisiones o por errores en el enlace. Así se creará otro programa para la simulación de colisiones entre tres estaciones. En el quinto capítulo, se expondrán las medidas de campo utilizadas para modelar el canal físico. A continuación, en el capítulo 6 se compararán los algoritmos estudiados en los capítulos anteriores. En el antepenúltimo capítulo, se analizará teóricamente el canal híbrido. Finalmente, en el capítulo ocho se argumentarán las conclusiones.

## Capítulo 2

# Modelado del canal para la simulación

Antes de realizar las medidas y modelar nuestro canal real, crearemos un canal propio para simular los dos algoritmos a estudiar. Mediante este modelado, en los siguientes dos capítulos comprobaremos el funcionamiento de una comunicación aplicando o no, el ARF o el LD-ARF.

El modelado del canal consiste en poder prevenir que sucedería si un paquete se enviara en un escenario concreto. En nuestro caso, esta predicción nos proporciona la información necesaria para saber si un paquete de datos podrá llegar a su destinatario o se perderá por efectos del canal.

En primer lugar, se utilizará el modelo de Gilbert-Elliot ya que destaca por su sencillez. En 1960, Gilbert [13] propuso un modelo de dos estados donde el estado “bueno” no tenía errores, y el estado “malo” representaba un estado con ráfagas de errores. Tres años después, Elliot [14] propuso que el estado “bueno” también pudiera tener una probabilidad de error diferente a cero, pero menor a la del estado “malo”.

Como se muestra en la *Figura 4*, el modelo de Gilbert-Elliot está formado por dos estados a los que llamamos G (estado bueno) y B (estado malo). Las probabilidad de las transiciones entre estados cumplen con  $\Pr(B|G) = b$  (probabilidad de pasar al estado malo estando en el bueno), y con  $\Pr(G|B) = g$  (probabilidad de pasar al estado bueno estando en el “malo”). Si nos fijamos, cada estado del modelo actúa exactamente como un canal binario simétrico (BSC, de sus siglas en inglés). Las probabilidades de cambio, es decir, las probabilidades de error en el estado bueno y en el estado malo son nombradas como  $P_G$  y  $P_B$ , respectivamente. Por consiguiente, el modelo de Gilbert-Elliot es un modelo de cuatro parámetros, que son:  $g$ ,  $b$ ,  $P_G$  y  $P_B$ .

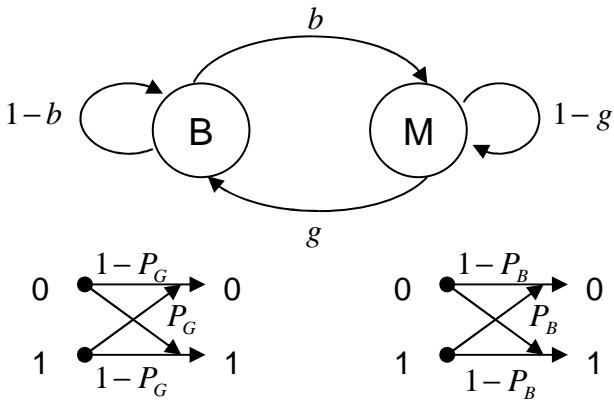


Figura 4. Diagrama del modelo de Gilbert-Elliott con las transiciones de los estados [15].

Normalmente,  $p_G$  suele ser relativamente menor a  $p_B$  ya que se asume que el estado G es un estado bueno y favorable donde no debería haber muchos errores. También,  $g$  acostumbra a ser pequeña, por lo que el estado malo tiende a persistir, emulando así un estado con ráfagas de errores en un sistema práctico. El proceso de estados para el modelo de Gilbert-Elliott puede hacerse estacionario asumiendo que la probabilidad de estado inicial se define como en (1), según [16].

$$\Pr(S_0 = G) = \frac{g}{g + b} \quad ; \quad \Pr(S_0 = B) = \frac{b}{g + b} \quad (1)$$

Una vez ya hemos programado este modelo simple de dos estados, ya podremos obtener la primera impresión del funcionamiento del algoritmo a aplicar, especificando únicamente los cuatro parámetros del modelo.

Por otra parte, este modelo solo nos permite simular una situación concreta de un canal. A continuación, se intentará mejorar el anterior modelo para poder acercarnos a la realidad. Para ello, se realizará un proceso determinista que emulará un modelo de diez estados de markov. Con este proceso no se utilizarán las probabilidades de cambio entre estos estados, sino que estos cambios vendrán dados por el usuario o un programa generador de procesos. Cada estado emulado de markov define cierto rango SNR en el nivel de señal de recepción. Además dentro de cada uno de los diez estados habrá un modelo de Gilbert-Elliott específico por cada supuesto rango de SNR. Así los parámetros de cada modelo de dos estados serán especificados al inicio, aunque se define que en canal bueno siempre llegan los paquetes ( $P_G = 0$ ), y en el malo siempre hay errores ( $P_B = 1$ ). En la *Tabla 1* se muestran los valores definidos de todo el modelo.

Estado	1	2	3	4	5	6	7	8	9	10
$\Pr(B G)$	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
$\Pr(G B)$	1	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1

Tabla 1. Valores de las transiciones de los estados internos en los estados del proceso determinista.

# Capítulo 3

## Algoritmo ARF

En este capítulo estudiaremos el algoritmo *Automatic Rate Fallback* (ARF) implementado en los productos WaveLan II de Lucent [5]. También crearemos en Matlab un programa de simulación con el cual comprobaremos su funcionamiento. Finalmente, analizaremos los resultados obtenidos.

### 3.1 DEFINICIÓN DEL ALGORITMO

Las diferentes técnicas de modulación utilizadas para las diferentes velocidades de transmisión de WaveLan II se caracterizan por crear comunicaciones más robustas a menores velocidades. Esto se traduce en diferentes rangos de cobertura con comunicación fiable para las diferentes tasas, dando 1 Mbps el mayor alcance. Cuando se diseña un sistema WaveLan II buscando el mayor tamaño de celda de cobertura, se escoge el alcance asociado a 1 Mbps. Pero las estaciones como se pueden mover por toda la celda deben ser capaces de operar a la máxima velocidad según el alcance. Para poder cumplir esta expectativa se incluye en los dispositivos el algoritmo ARF.

Además del amplio alcance, las bajas velocidades también son más robustas contra condiciones interferentes como las altas pérdidas en la trayectoria de la señal, el alto ruido en el canal, y los desvanecimientos profundos producidos por la multirayectoria de la señal. Así, el ARF también actúa cuando estas condiciones aparecen.

En cuanto al funcionamiento del algoritmo, este se basa en detectar las condiciones del medio donde se produce la comunicación, a partir de las transmisiones que se han completado con éxito y las que han fallado. Así, según la cantidad de mensajes de reconocimiento (ACK) que recibe, puede determinar si se debe aumentar o disminuir la tasa de transferencia de paquetes.

A continuación, se explicarán los diferentes casos que pueden surgir en una comunicación y como interviene el algoritmo en cada uno de ellos. En primer lugar, cuando se inicia la transferencia de datos, como no se tiene conocimiento previo del canal se asigna por defecto una velocidad de transferencia de 2 Mbps. Cuando se pierde un ACK por primera vez después de transmisiones correctas, se retransmitirá el paquete

### 3. Algoritmo ARF

---

perdido a la misma velocidad (2 Mbps). Si se pierde un ACK otra vez, la segunda retransmisión y posteriores transmisiones se enviarán a la tasa inferior (en el primer caso, 1 Mbps). También se activará un temporizador para que el sistema actúe correctamente en situaciones en las que las transmisiones llegadas con éxito y las perdidas se alternen. Así, cuando el temporizador expire o lleguen 10 ACK consecutivas, el dispositivo intentará aumentar la tasa al nivel superior. En cuanto a los primeros paquetes que se envían después de aumentar la velocidad, se les llama de sondeo y sirven para comprobar la calidad del canal. Si no llega un ACK de los paquetes de sondeo, el sistema inmediatamente reestablecerá la tasa anterior.

## 3.2 PROGRAMA DE SIMULACIÓN

En primer lugar, se intentará crear una simulación de un enlace de comunicación. Para ello, se especificará un escenario simple y se optará por el modelo de canal de Gilbert-Elliott. Así una vez funcione el modelo básico se podrá proceder a otros más complejos.

El escenario en el cual se aplicará el programa consiste en un enlace de datos entre dos estaciones. Una de ellas será la transmisora y la otra será la receptora. Además, este enlace será *half-dúplex*, por lo que sólo podrá enviar información una de las dos estaciones en cada instante.

El modelo de dos estados de Gilbert-Elliott se aplicará sobre la capa de enlace de datos. Así, no se deberá determinar a nivel físico si cada bit se recibe o se pierde, sino que se centrará a nivel de paquete. Esta consideración viene dada por la naturaleza del algoritmo ARF. Con esto evitamos la codificación de la información que será especificada por el dispositivo que se emplee.

Una vez ya podemos predecir mediante el modelo cuando se perderá un paquete, debemos actuar en consecuencia. Por lo tanto, aplicaremos un mecanismo de corrección de errores mediante retransmisión. Esta técnica consiste en que el receptor informe al transmisor del éxito o fracaso de la transmisión. Los protocolos en los cuales el receptor envía una trama de reconocimiento positivo, suelen llamarse PAR (*Positive Acknowledgment with Retransmission*) o ARQ (*Automatic Repeat reQuest*).

El protocolo que se ha escogido para realizar la simulación es el de parada y espera o *Stop&Wait*. Este consiste en que el transmisor después de enviar un paquete, espere a que el receptor le responda antes de enviar el siguiente. En caso contrario, volverá a emitir el mismo paquete ya que probablemente se haya perdido, como se puede observar en la *Figura 5*. Para ello, se necesitarán temporizadores que expiren con tiempo mayor al de ida y vuelta del paquete. Si se escoge un *time-out* menor, puede que llegue una

### 3. Algoritmo ARF

trama de reconocimiento después de la retransmisión del paquete, lo que supone un envío y gasto de recursos innecesario, como demuestra la *Figura 6*.

En definitiva, con este protocolo se facilitará la programación de la simulación debido a la simple arquitectura que lo compone. Sin embargo, esta ventaja se consigue a costa de una baja utilización del canal, ya que dependiendo de los tiempos de ida y de vuelta el transmisor se pasará la mayor parte del tiempo esperando respuesta.

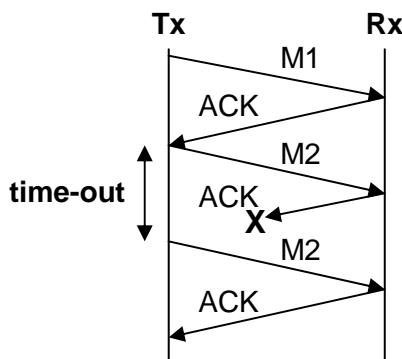
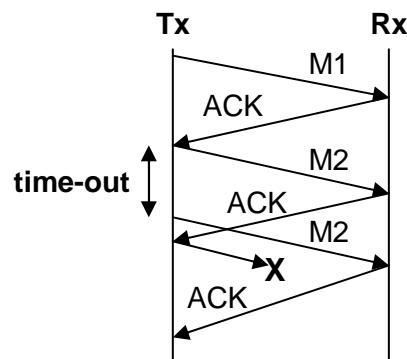


Figura 5. Protocolo simple de parada y espera.



*Figura 6. Efecto de time-out corto.*

Para el programa de simulación se establecerá que el *time-out* tenga el tiempo equivalente al utilizado para el envío de un paquete de datos y de dos recepciones de ACK. Con este valor para el temporizador evitaremos el reenvío innecesario de paquetes. Además hay que considerar que los ACK son tramas de longitud mucho menor a la de los datos, por lo que tardarán menos tiempo en llegar. Así es preferible aumentar un poco el *time-out* que realizar una nueva retransmisión. Este aumento apenas disminuye el *throughput* en un 3% o en un 10%, en los casos más desfavorables.

Por otro lado, el algoritmo ARF se implementará en este programa operando a las velocidades de transmisión especificadas por el estándar 802.11b [1]: 1 Mbps, 2 Mbps, 5.5 Mbps y 11 Mbps. De nuevo, como el algoritmo actuará en la capa de control de acceso al medio (MAC), no se deberán definir ni simular los procesos que se llevan a cabo en el cambio de la tasa, como pueden ser la modulación utilizada o la longitud de código como se muestra en la *Tabla 2*.

Tasa de envío de datos	Longitud de código	Modulación	Tasa de símbolo	Bits / Símbolo
1 Mbps	11 (Secuencia Barker)	BPSK	1 MSps	1
2 Mbps	11 (Secuencia Barker)	QPSK	1 MSps	2
5.5 Mbps	8 (CCK)	QPSK	1.375 MSps	4
11 Mbps	8 (CCK)	QPSK	1.375 MSps	8

Tabla 2. Especificaciones de las tasas de envío de datos para 802.11b [6].

### 3. Algoritmo ARF

Para poder implementar estas velocidades en el programa, los paquetes llegarán según un proceso aleatorio. El proceso de llegada se modelará como un proceso de Poisson. En el modelo de probabilidad de Poisson los paquetes llegarán con una tasa media  $\lambda$  constante. Así,  $\lambda$  se puede definir como el número medio de repeticiones por unidad de tiempo. Por lo tanto, para poder determinar los tiempos de llegada de los paquetes también se deberá establecer el intervalo de tiempo en el cual se producirán los eventos. En la *Figura 7*, se puede observar un proceso de llegadas de Poisson con una tasa de diez eventos por segundo durante tres segundos.

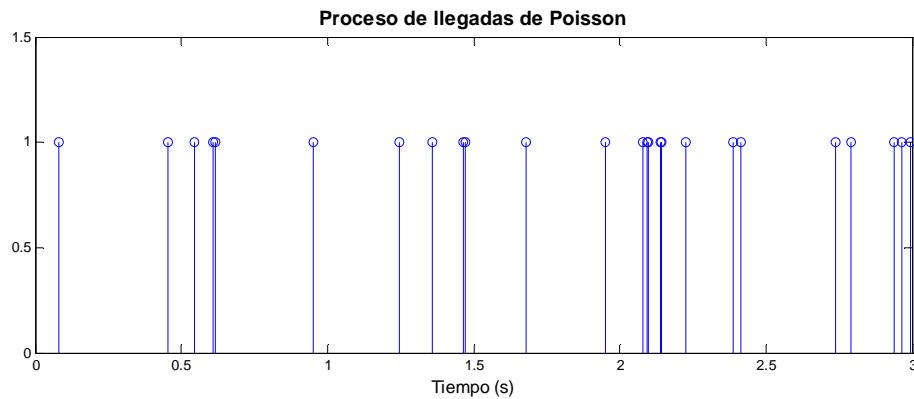


Figura 7. Llegadas según el proceso de Poisson con  $\lambda=10$  eventos/s.

Mediante el proceso de Poisson, obtenemos una variable aleatoria  $N(t)$  que representa el número de llegadas independientes que ocurren a una velocidad constante en el tiempo. Esta variable aleatoria obedece a la distribución de Poisson de (2).

$$P\{N(t) = n\} = \frac{(\lambda \cdot t)^n}{n!} e^{-\lambda \cdot t} \quad (2)$$

Si representamos gráficamente la anterior función de distribución, obtendremos la densidad de probabilidad del modelo de Poisson. Podemos observar en la *Figura 8* como en los valores próximos a 4, que es la media de la función, se concentra los valores de máxima probabilidad de la variable.

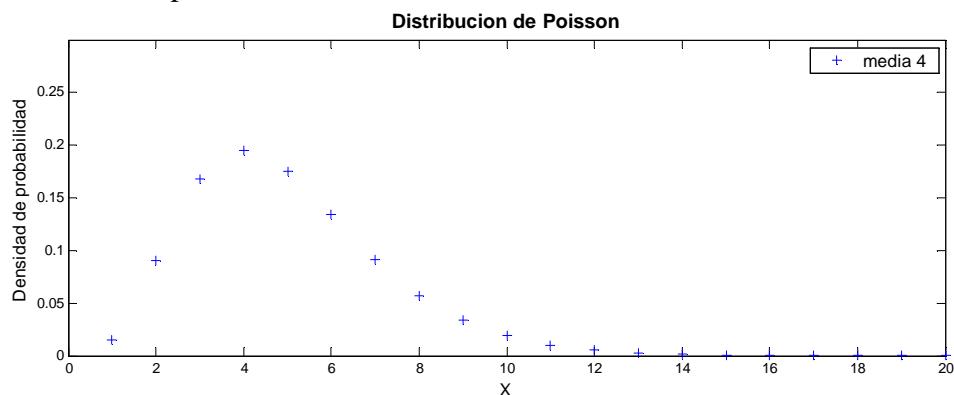


Figura 8. Distribución de Poisson según su densidad de probabilidad.

### 3. Algoritmo ARF

---

Si aplicamos el anterior modelo de Poisson, conseguiremos un proceso con los tiempos de las llegadas de los paquetes, sucediendo a una velocidad determinada por cada tasa. A continuación, mediante los valores de todas las realizaciones podremos calcular el *throughput*. Sin embargo, surge un problema relacionado con la tasa media de las llegadas. Este se produce cuando intentamos realizar una simulación con un número reducido de paquetes, ya que la media puede variar debido a la aleatoriedad de los eventos. Si ponemos como ejemplo el proceso de la *Figura 7*, podemos comprobar en (3) como si calculamos la tasa media de eventos, esta es diferente a la especificada inicialmente.

$$\left. \begin{array}{l} \text{Tiempo último evento} = 3.0941 \text{ s} \\ \text{nº de eventos} = 26 \end{array} \right\} \rightarrow \lambda = \frac{\text{eventos}}{\text{tiempo}} = \frac{26}{3.0941} = 8.4032 \neq 10 \quad (3)$$

Así, dependiendo de los valores introducidos en el modelo de Poisson, el resultado obtenido podrá diferir del correcto. Por este motivo, se ha realizado un estudio sobre la relación entre el número de paquetes simulados y la desviación del resultado. Este estudio se ha llevado a cabo, mediante pruebas estadísticas de la tasa media obtenida respecto a un valor de referencia en cien realizaciones diferentes. El valor de referencia era una  $\lambda$  igual a 2 Mbps, y se han dado como correctas las tasas medias con una desviación inferior o igual a 0.2 Mbps. En la *Tabla 3* se observan los resultados obtenidos en el estudio.

Numero de paquetes enviados	Media de error con la tasa de referencia (Mbps)	Porcentaje de simulaciones correctas (%)
1	9.7968	10
2	2.7472	13
5	0.86758	25
10	0.53707	26
50	0.24073	49
100	0.15451	69
200	0.11497	84
1000	0.055504	100
2000	0.033117	100
2375	0.032016	100

*Tabla 3. Resultados obtenidos en el estudio basado en cien realizaciones por cada prueba estadística.*

Según los resultados obtenidos, se puede concluir que el valor del *throughput* simulado por el programa será fiable cuando se superen los 50 paquetes enviados. Si se desea realizar simulaciones con valores menores, se deberá asumir que el resultado puede ser muy dispar.

Finalmente, se ha escogido una longitud de trama de datos igual a 800 bits. En cuanto, a la longitud del ACK es de 112 bits como se especifica en [6], y se envía a 1 Mbps.

### 3. Algoritmo ARF

El programa de simulación se resume en la *Figura 9*, que muestra en un diagrama de flujo de manera simple el funcionamiento del sistema.

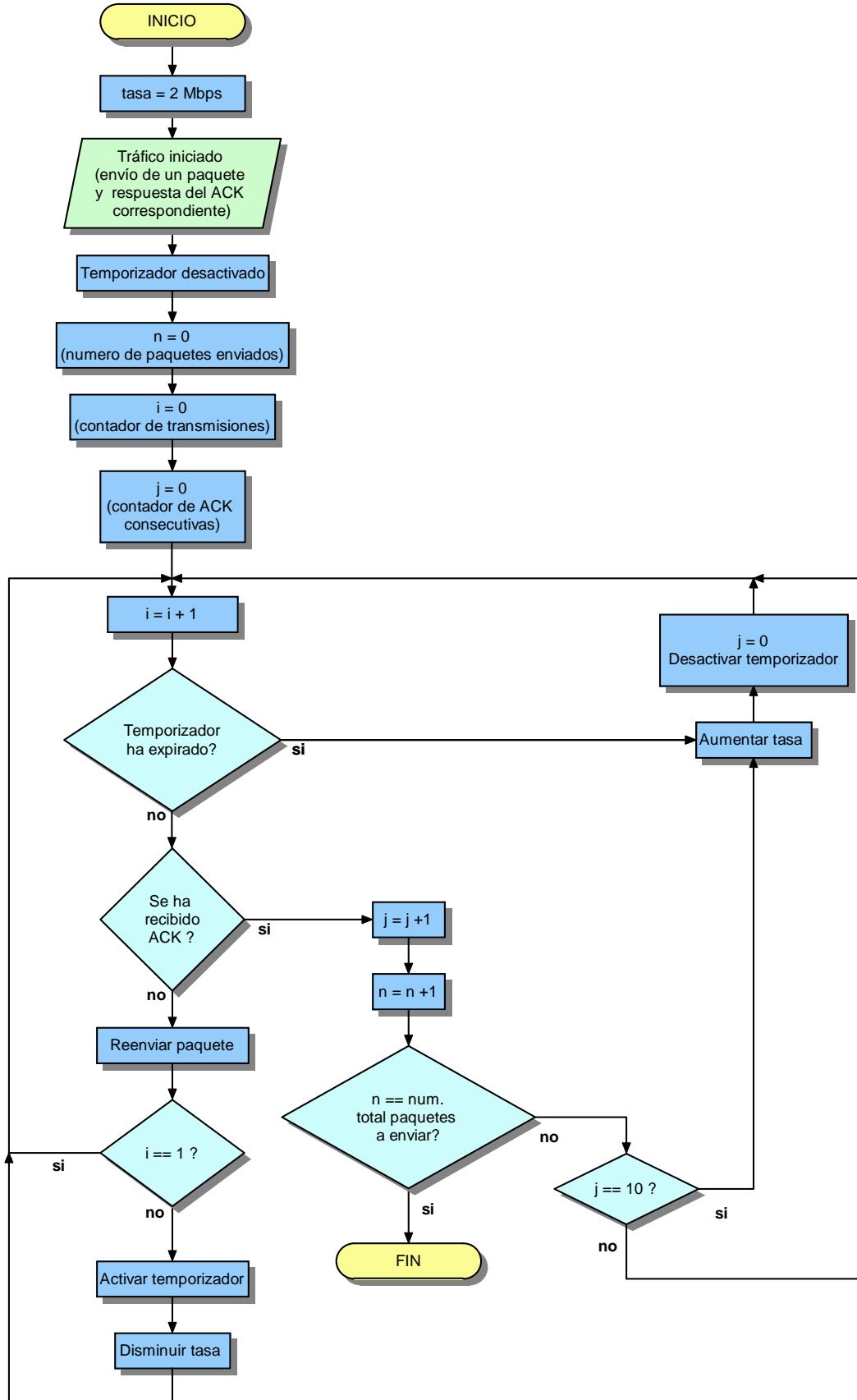


Figura 9. Diagrama de flujo del funcionamiento del algoritmo ARF.

### 3.3 RESULTADOS OBTENIDOS EN LA SIMULACIÓN

En el último apartado de este capítulo, se podrán observar los resultados obtenidos en la simulación del enlace, aplicando principalmente el algoritmo ARF. Además se analizarán sus ventajas e inconvenientes respecto a una simulación en la cual no se implemente el algoritmo.

Cuando no aplicamos algoritmo alguno, la estación emisora tendrá que transmitir a la velocidad más baja posible, ya que no se conocen las condiciones del canal. Por lo tanto, se enviarán los paquetes a una tasa de 1 Mbps.

#### 3.3.1 SIMULACIÓN 1

Una vez ya se ha creado el programa según las especificaciones del apartado anterior, deberemos introducir los parámetros necesarios para obtener la simulación. Estos parámetros son: numero de paquetes a enviar; el tiempo entre tramas correctas y fallidas equivalente al tiempo del *timer*; la probabilidad de pasar al estado malo estando en el bueno ( $Pr(G/B)$ ); la probabilidad de pasar al estado bueno estando en el malo ( $Pr(B/G)$ ); la probabilidad de error en el estado bueno; y la probabilidad de error en el estado malo. Las dos últimas, normalmente se definirán como 0 y 1, respectivamente. Esto es debido, a que se simulará que en el estado bueno siempre llega un paquete, y en el malo siempre se pierde. En la *Tabla 4* encontramos los valores de los otros parámetros, y el resultado del *throughput* aplicando o no el algoritmo ARF, para la primera simulación.

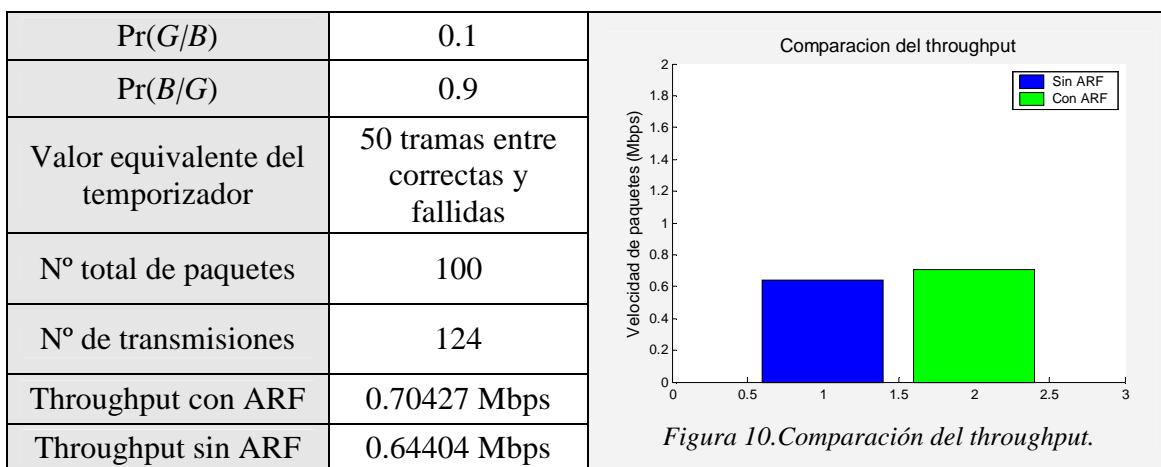


Tabla 4. Características de la primera simulación.

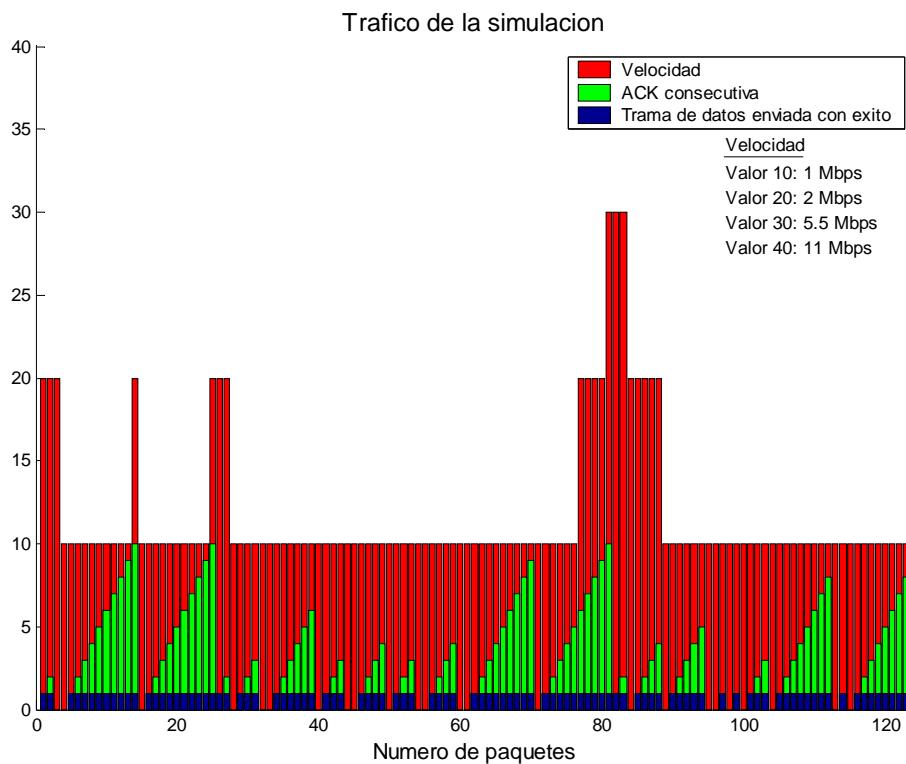
En la *Figura 10* se puede observar como en este caso la diferencia entre los dos *throughput* no es considerable, apenas llega a 50 Kbps. Las diversas las causas de este inconveniente se enumeraran a continuación.

### 3. Algoritmo ARF

---

En primer lugar, si definimos un valor alto para el temporizador, este no tendrá mucho efecto sobre la tasa de envío, a menos que coincida con una ráfaga de transmisiones correctas. Al esperar mucho tiempo para realizar una trama de sondeo, lo más probable es que antes lleguen diez ACK consecutivas.

En segundo lugar, la arquitectura conservadora del algoritmo ARF también juega un papel importante. Para reforzar esta explicación nos centraremos en la *Figura 11*, donde se puede observar el tráfico de paquetes de la estación transmisora y la tasa de envío utilizada. Si nos fijamos en esta figura podemos ver como un simple error puede evitar el incremento de la tasa cuando las condiciones del canal son favorables. Un ejemplo de ello se encuentra alrededor del paquete 70, donde el algoritmo estaba a punto de aumentar el valor de la velocidad (le habían llegado nueve tramas de reconocimiento consecutivas) cuando hubo un error en la transmisión de un paquete. Además si seguimos, observamos como al final se aumenta la tasa pero debido al temporizador. Según este resultado podemos argumentar que el algoritmo solo cambiará la tasa, cuando se produzcan paquetes de sondeo o se asegure de que las condiciones del canal han mejorado.



*Figura 11. Diagrama de barras donde se muestra la sucesión de los paquetes enviados y recibidos por el emisor, y la velocidad de transmisión.*

A continuación, si estudiamos el tráfico de la simulación según su tiempo de aparición podremos analizar otras características del algoritmo. Para ello observaremos la *Figura 12*, que muestra tanto la velocidad como las tramas de reconocimiento recibidas consecutivamente en función del tiempo.

### 3. Algoritmo ARF

---

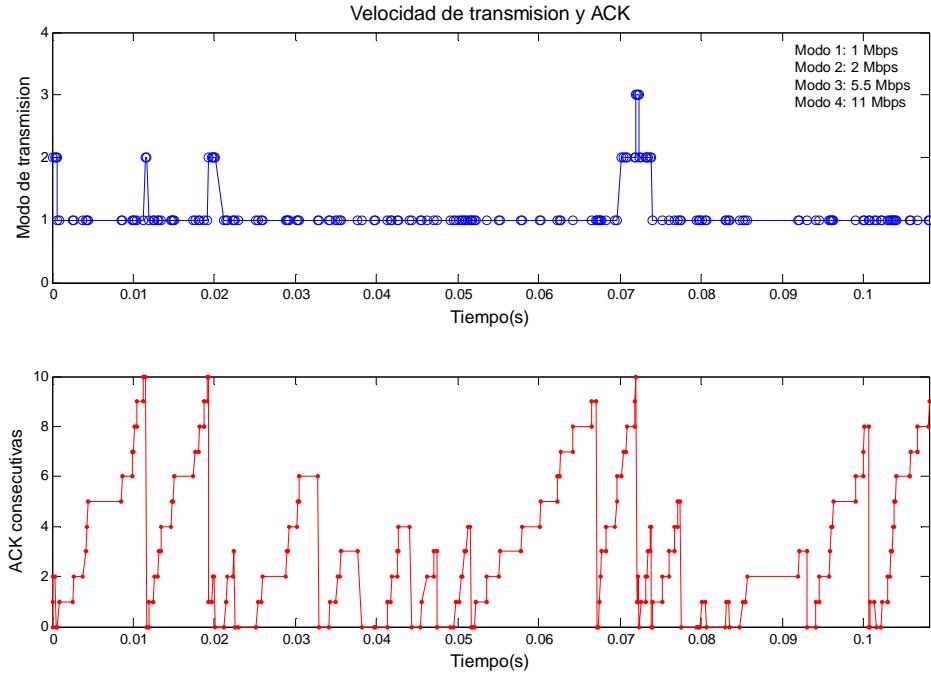


Figura 12. Comparación entre la velocidad de transmisión y el número de tramas de reconocimiento consecutivas, según el tiempo de aparición.

Si nos centramos en la gráfica de las ACK consecutivas de la *Figura 12*, podemos intuir el proceso de llegadas de Poisson con el que se ha realizado. Notamos como algunas de las llegadas de las tramas consecutivas en una determinada tasa, se producen con mayor rapidez que otras, siguiendo con cierta concordancia esta distribución.

Por otro lado, con las gráficas de la *Figura 12* se ve claramente como el algoritmo tarda en reaccionar frente a la mejoría de las condiciones del canal. Esto produce una reducción considerable del *throughput* resultante. Sin embargo, en ciertos casos se asegura una transmisión constante de la información evitando oscilaciones innecesarias de la tasa.

#### 3.3.2 SIMULACIÓN 2

En esta segunda simulación, se intentará obtener un mejor *throughput* aplicando el algoritmo en un canal favorable. Así podremos verificar la ventaja de tener un algoritmo de adaptación en nuestra implementación.

Para simular este canal favorable definiremos unas probabilidades de transición entre estados, que den como resultado una mayor presencia en el estado bueno. Así, sucederán muy pocos reenvíos que impidan que el algoritmo incremente la tasa de transmisión. Además, se especificará un *timer* relativamente corto, para producir más paquetes de sondeo.

### 3. Algoritmo ARF

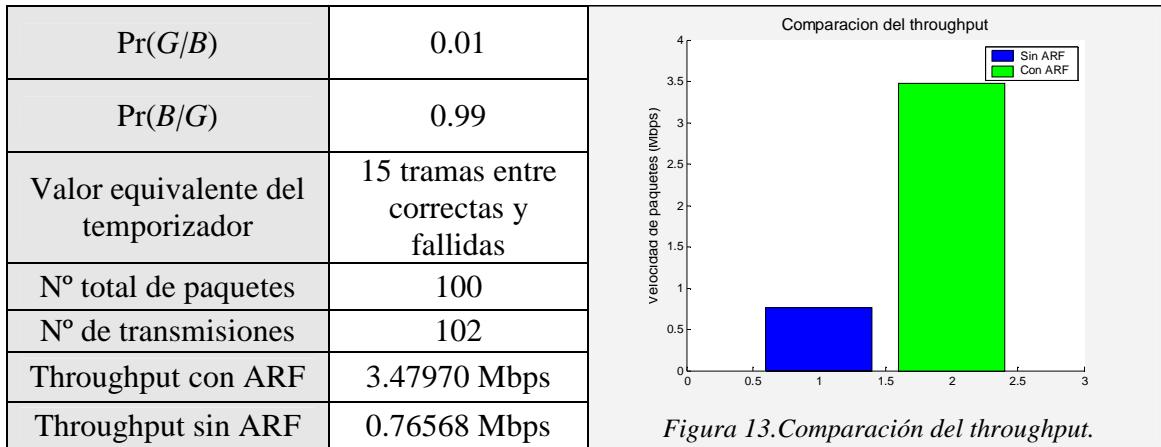


Tabla 5. Características de la segunda simulación.

En la *Tabla 5* se muestran los parámetros introducidos en el programa. Se ha sido bastante restrictivo respecto a la transición al estado malo, y bastante benevolente respecto a la transición al estado bueno. Mediante estas probabilidades se han conseguido unas estupendas condiciones del canal. En esta situación el algoritmo funciona de maravilla ya que reconoce el buen estado del canal. Prueba de ello es el magnífico *throughput* que resulta de aplicar el ARF. Ahora si que se nota una gran diferencia entre aplicar algoritmo o no, como se muestra en la *Figura 13*.

Sin embargo, pese a las buenas condiciones del canal, un error en una transmisión produce una reducción de la tasa, como se observa en la *Figura 14*. El algoritmo ARF elude el primer error que aparece, pero en detectar el segundo decrementa la velocidad. Esto supone una disminución del *throughput* resultante, ya que podría haber sido mayor.

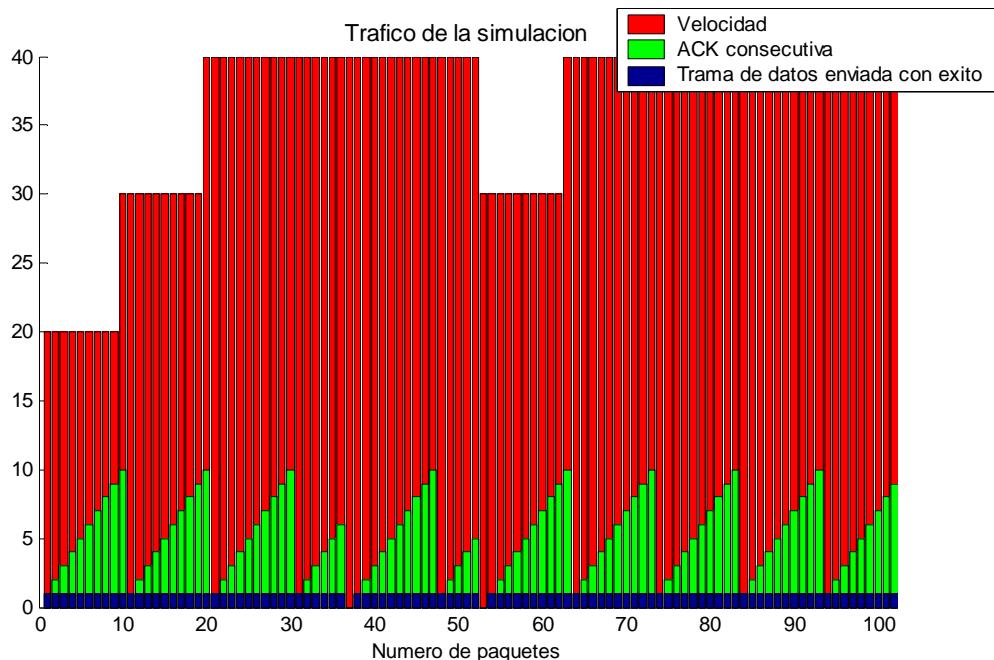
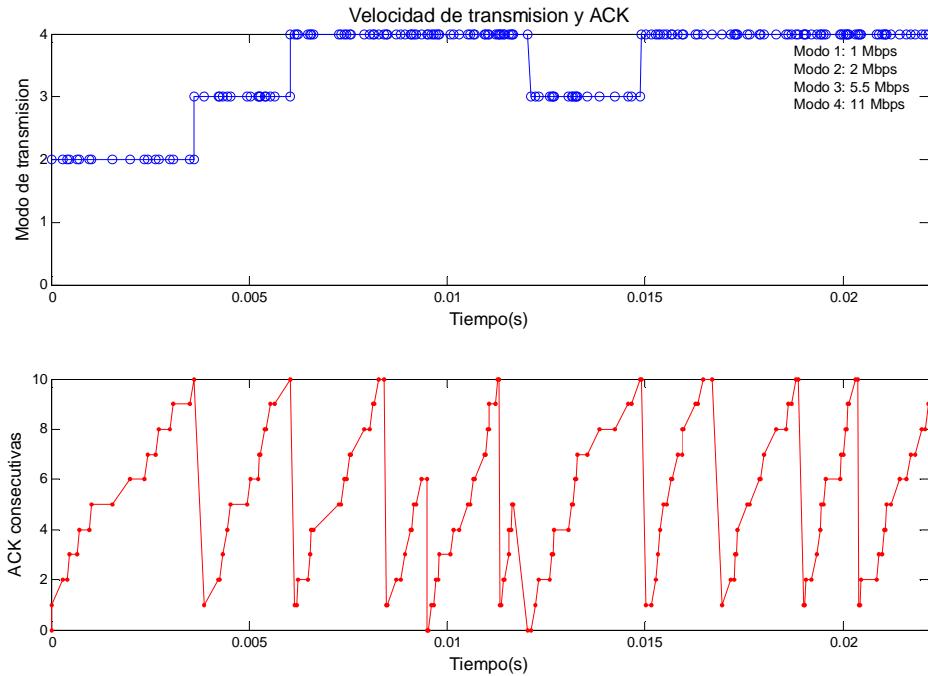


Figura 14. Tráfico de la segunda simulación.

### 3. Algoritmo ARF

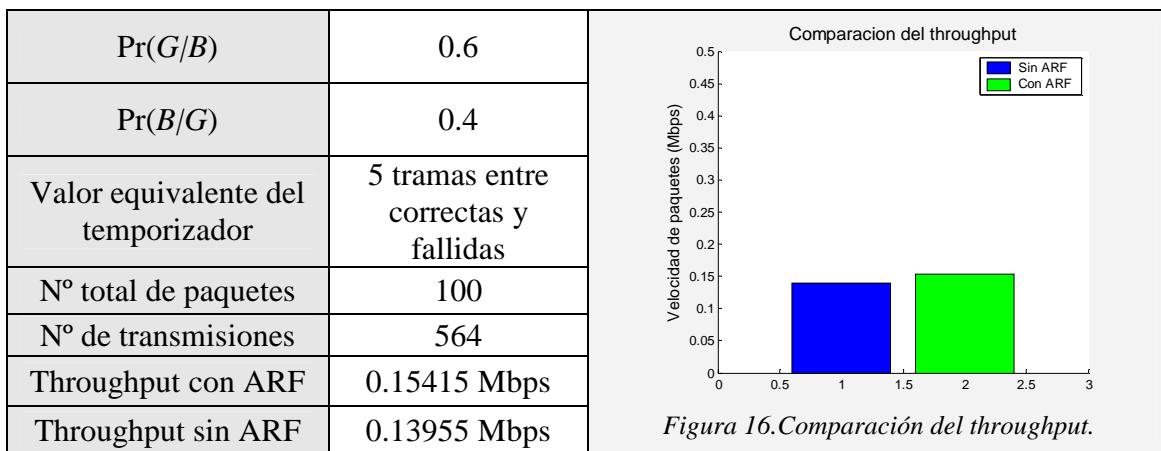
En la *Figura 15*, se comprueba como el segundo error se produce alrededor de los 0.012 segundos. Este error provoca que la tasa esté 2 milisegundos por debajo de lo que debería estar, si se hubieran verificado bien las condiciones del canal. Este intervalo de tiempo compone el 10% del total, y si se hubiera enviado a la tasa superior se habría conseguido un *throughput* muy próximo a los 4 Mbps, 0.5 Mbps de diferencia.



*Figura 15. Comparación entre velocidad de transmisión y ACK consecutivas en función del tiempo, para la segunda simulación.*

#### 3.3.3 SIMULACIÓN 3

En la última simulación, se buscará conocer el rendimiento que supone aplicar un temporizador corto en un canal que produce muchos errores. Así se analizaran los resultados y se destacaran los límites de este algoritmo. Dado que habrá muchas retransmisiones, no se restringe el número máximo de reenvíos, para no perder información.



*Tabla 6. Características de la tercera simulación.*

### 3. Algoritmo ARF

En la *Tabla 6* vemos como las probabilidades de transición que hemos introducido en el programa son muy parecidas, pero es un poco mayor la probabilidad de la transición del estado bueno al malo. Podemos destacar de la información que se nos ofrece, el gran número de retransmisiones producidas, 464 reenvíos. Con este valor, el *throughput* resultante es muy bajo respecto a las anteriores simulaciones.

Sin embargo, el *throughput* conseguido por el ARF es un poco superior al otro. Esto se produce debido a los numerosos paquetes de sondeo enviados en eximir el reducido tiempo introducido para el temporizador. En la *Figura 17* se observa como a pesar de los muchos errores que se producen en el canal, constantemente se intenta mejorar la tasa de transmisión. Pero el incremento del rendimiento, no se ve compensado con la oscilación que sufre la tasa en condiciones estables. Esta oscilación producirá un cambio continuo de modulación, aumentando el gasto de energía para transmitir la información.

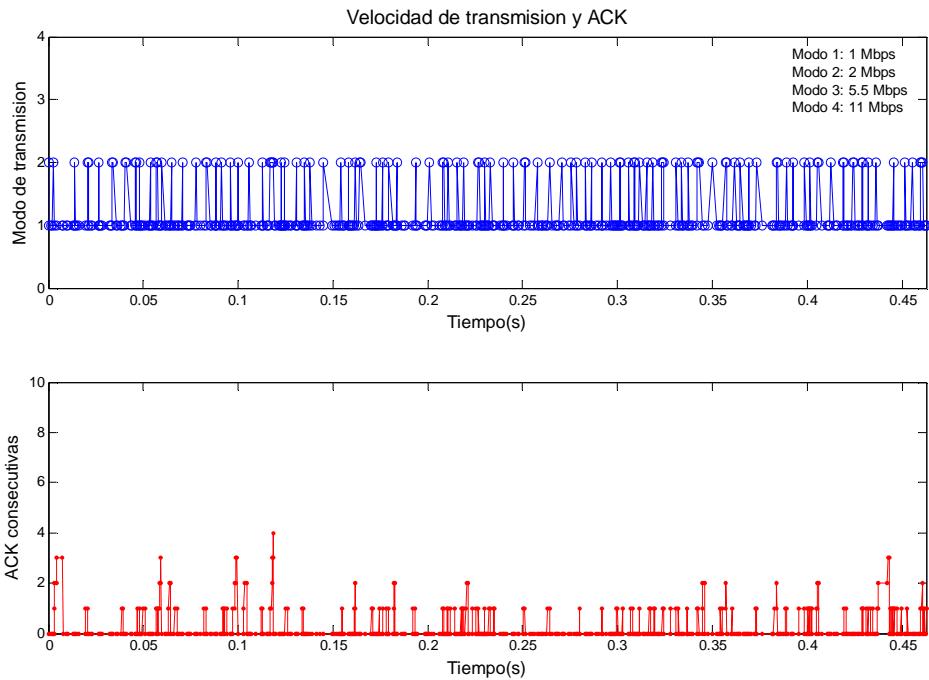


Figura 17. Comparación entre la velocidad y las ACK consecutivas en función del tiempo, para la tercera simulación.

## 3.4 CONCLUSIONES

De forma muy básica, en centrarse en los paquetes transmitidos con éxito, el algoritmo puede aumentar la velocidad a la que se envían, para poder ofrecer al receptor un mayor número de paquetes en menor tiempo. Si se detecta que están habiendo errores en el canal (no llegan ACK), se decrementará la tasa para que el nombre de paquetes perdidos no se incremente. Está sencillez en la aplicación le ha hecho ser al ARF uno de los algoritmos más implementado en los productos WLAN.

### 3. Algoritmo ARF

---

Sin embargo, tiene ciertas carencias cuando las condiciones del canal varían muy rápidamente. Aunque mejore el canal en un intervalo de tiempo muy corto, el algoritmo no actuará en consecuencia hasta que no se reciban diez tramas de reconocimiento consecutivas. Esto puede desembocar en una adaptación incorrecta de la tasa. De la misma manera que mejoran, pueden empeorar las condiciones rápidamente, por lo que el algoritmo podría desaprovechar una oportunidad para haber mejorado el *throughput*.

Por otra parte, si el canal tiene unas buenas condiciones, un único error puede ser malinterpretado por el algoritmo como un indicio del posible empeoramiento del canal. Esto reducirá la velocidad de datos correctos recibidos en la estación receptora. Pese a este inconveniente, el algoritmo ARF funciona bastante bien en estas situaciones, consiguiendo un elevado rendimiento.

Finalmente, si se producen muchos errores en el canal el valor del temporizador mejorará muy levemente el *throughput*, y por contra puede llegar a producir oscilaciones si su valor es reducido.

En conclusión, a pesar de no ser un algoritmo óptimo para un modelo de canal específico, el ARF es muy estable frente a las variaciones del canal, y puede alcanzar un rendimiento próximo al óptimo en la mayoría de los modelos.

# Capítulo 4

## Algoritmo LD-ARF

En este capítulo se estudiará el algoritmo LD-ARF. En primer lugar, explicaremos el problema principal por el cual se intenta mejorar el ARF con este nuevo algoritmo. A continuación, se plantearán las modificaciones que ha sufrido el programa de simulación. Finalmente, se analizarán los resultados y se expondrán las conclusiones.

### 4.1 PRINCIPAL PROBLEMA DEL ALGORITMO ARF

El rendimiento del algoritmo ARF dependía de las condiciones del canal y de los errores que se producían en este. Ahora bien, en este algoritmo no se especifica cuál ha sido el origen de estos fallos. A simple vista, parece que este factor no vaya a influir en la adaptación. Sin embargo, puede ser determinante para obtener un resultado razonable en la transmisión.

El ARF supone que todos los errores son producidos por circunstancias relacionadas con el canal, como puede ser el ruido o la atenuación por la distancia. Pero esto es una suposición errónea cuando hay diferentes estaciones emitiendo en el mismo canal, y hacia el mismo receptor. Entonces se producen las llamadas colisiones, que consisten en la corrupción de los datos recibidos debido a la llegada simultánea de dos paquetes diferentes.

Entonces, cuando hay colisiones en el receptor, el ARF las malinterpretará considerándolas como simples errores en el canal. Así reducirá la tasa innecesariamente, ya que puede que las condiciones del medio sean favorables. Esto sucederá normalmente con la afluencia de estaciones emitiendo a la vez, y debido a la imperfecta sincronización de los envíos (no se enviarán todos los paquetes cada cierto tiempo constante). Pero la consecuencia de esta incorrecta interpretación es que en disminuir la velocidad de envío, el tiempo de transmisión se prolongará por cada paquete. Por lo tanto, aumentará la probabilidad de que se produzcan colisiones ya que las estaciones ocuparán más tiempo el canal. En conclusión, la adaptación de la tasa por parte del ARF en este escenario ocasionará más errores en los envíos y degradará su rendimiento.

Como ya vimos en el primer capítulo, el estándar 802.11 contempla dos mecanismos de gestión de paquetes de la capa de acceso al medio (MAC) para poder evitar las colisiones y solucionar el problema anterior. Estos son el mecanismo de contienda CSMA/CA por parte de la sub-capa DCF, y la gestión de todas las transmisiones en la sub-capa PCF. Sin embargo, contra más estaciones se encuentren en la red, más probabilidad habrá de que sucedan colisiones. Sobretodo, cuando haya saturación de la red, es decir, cuando todas las estaciones intenten transmitir.

Por otra parte, estos mecanismos no se podrán aplicar cuando se produzca el problema del nodo oculto o estación oculta. Este aparece cuando las estaciones transmisoras detectan el canal libre ya que no conocen de la existencia de las otras estaciones. Esto es debido a que el alcance de la señal no llega a todas las estaciones. Así, las transmisoras comenzaran a enviar la información a la estación receptora, y probablemente habrán colisiones. Si ponemos el ejemplo de tres estaciones, tal y como se muestra en la *Figura 18*, se puede observar como la estación A y la estación C solo detectan a B, ya que su alcance no es tan grande. Así las estaciones A y C escuchan el canal, y en comprobar que nadie está transmitiendo, emiten su señal. Entonces se produce una colisión en la estación B y se pierde la información ya que se corrompe.

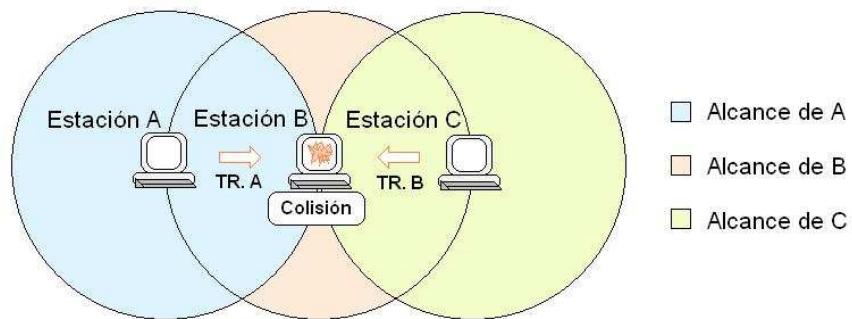


Figura 18. Ejemplo del problema del nodo oculto con tres estaciones.

Según las anteriores técnicas se logra evitar gran parte de las colisiones, pero aún existe la posibilidad de que se produzcan. Por lo tanto, dado que el estándar de la capa MAC no proporciona ningún mecanismo para la diferenciación entre los dos tipos de pérdidas, se deberá modificar el protocolo de esta capa y el algoritmo ARF, para mejorar su rendimiento.

## 4.2 DEFINICIÓN DEL ALGORITMO

Como evolución de ARF, surge el llamado algoritmo de adaptación de tasa con diferenciación de pérdidas o *loss-differentiating-ARF* (LD-ARF). Con este algoritmo pueden coexistir las pérdidas por errores producidos por el canal y las colisiones, sin perder rendimiento en el sistema. Además con la diferenciación de pérdidas se evita la variación innecesaria de la tasa de transmisión.

Como ya hemos visto, muchos de los algoritmos que se utilizan (entre ellos el ARF), reducen su funcionamiento eficiente a determinadas situaciones ya que no diferencian las pérdidas. Por este motivo, en [6] se plantea la mejora del estándar IEEE 802.11 en las especificaciones de la capa MAC, y la implementación del algoritmo alternativo de adaptación de tasas llamado LD-ARF, que combinará el algoritmo ARF y estas especificaciones de diferenciación de pérdidas de la capa MAC.

Con la solución que se mostrará a continuación, se pretende llegar al objetivo final de disminuir la pérdida de paquetes. Con esta disminución de errores, se intentará que no se reenvíen tantos paquetes y que no se llegue a cancelar la transmisión debido a las supuestas condiciones irregulares del canal, como pasaba con los anteriores métodos. Así se conseguirá aumentar el caudal eficaz (*throughput*).

#### 4.2.1 FUNCIONAMIENTO DEL ESTÁNDAR PARA LA DIFERENCIACIÓN DE PÉRDIDAS

El protocolo de la capa MAC de diferenciación de pérdidas se prevé como una mejora del IEEE 802.11 DCF. Este procedimiento de acceso se puede realizar con el método básico (se envía directamente y se espera una trama de reconocimiento) o con RTS/CTS. Estas dos opciones están incluidas en este nuevo protocolo. En concreto, el procedimiento de acceso por RTS/CTS es útil para tramas de datos que tienen un gran tamaño, cuando hay muchas estaciones, o terminales ocultos. Los dos métodos se pueden observar en la *Figura 19*.

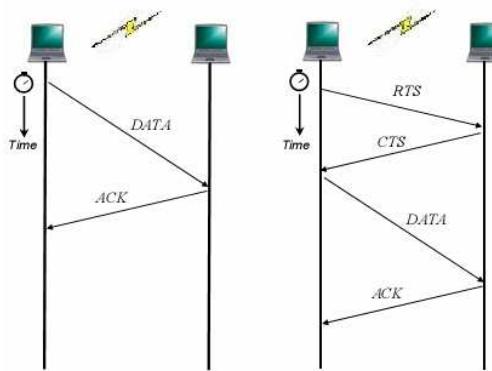


Figura 19. Acceso básico y RTS/CTS en el estándar 802.11 DCF [6].

El método de diferenciación de pérdidas con RTS/CTS es sencillo, y se basa en cuatro caminos de secuencia de intercambio que no cambian: (i) Si se recibe tanto el CTS como el ACK se interpreta que la transmisión se ha completado con éxito. (ii) Si la trama CTS se ha recibido pero no se ha recibido la de la ACK, probablemente haya habido un error en el enlace. (iii) Si no se recibe ni la trama CTS ni la ACK, probablemente se haya producido una colisión. Ya que RTS y CTS son tramas cortas y normalmente se transmiten a una baja tasa, la diferenciación de pérdidas es bastante eficiente. Además con este mecanismo se soluciona el problema del nodo oculto, ya que

#### 4. Algoritmo LD-ARF

---

en mandar el receptor una trama CTS, las otras estaciones se dan cuenta que otra estación esta intentando transmitir, y ya no emiten evitando una colisión en el receptor. Pasado cierto tiempo, la estación que no ha podido enviar, volverá mandar una trama RTS esperando su contestación.

El acceso básico es el acceso por defecto en el estándar 802.11 DCF. En muchos casos, como por ejemplo cuando no hay terminal oculto, el acceso básico es más eficiente que el acceso con RTS/CTS. En el procedimiento original de acceso básico, solo se envía una trama de reconocimiento cuando la trama de datos se recibe correctamente. En cambio, cuando la trama recibida es errónea, el receptor no emite ninguna respuesta. Como ya se ha explicado anteriormente, una colisión o un error en el enlace pueden ser las causas del fallo en la trama de datos, y el transmisor no puede distinguir entre estos mecanismos de pérdidas ni puede tomar una medida correcta. Por lo tanto, se debe incorporar un mecanismo que determine la causa del fallo y que se lo comunique al transmisor.

El método de diferenciación de pérdidas está basado en la siguiente observación. La trama de datos MAC se puede dividir en dos bloques funcionales: la cabecera MAC y el cuerpo de trama. La cabecera puede contener información como el tipo de trama, la dirección de origen y la dirección de destino. En el cuerpo se incluye la información que se quiere enviar. Si en una red WLAN todas las estaciones se pueden escuchar (no hay terminal oculto), solo podrá haber una colisión cuando más de una estación envíe una trama de datos en el mismo intervalo de tiempo. En ese caso, se perderán tanto la cabecera como el cuerpo del paquete, y el receptor no podrá recibir nada.

Sin embargo, si solo hay una estación enviando un paquete de datos y este se pierde debido a un error en el enlace, hay una probabilidad muy alta que el receptor pueda recibir la cabecera correctamente. Esto es así porque la cabecera es mucho más corta que la trama entera. Analizando el contenido de la cabecera se puede obtener información sobre el emisor que envió el paquete. Para comprobar la fiabilidad de la cabecera recibida se deberá incluir un pequeño campo de *checksum*<sup>6</sup>.

Cuando ocurre un error en el enlace, una trama de control es enviada por la estación receptora hacia la transmisora. Esta trama de control se llama NACK, y ha sido añadida al 802.11 DCF. La NACK es utilizada para comunicar al emisor de que la transmisión fue fallida debido a un error en enlace. La estructura de esta trama es la misma que la de la trama ACK, exceptuando un bit en la cabecera. También se envía a la misma velocidad que una ACK. Así, si una estación transmisora recibe una NACK sabrá muy

---

<sup>6</sup> **Checksum.** Método de control de redundancia muy simple, utilizado para proteger la integridad de datos, verificando que no hayan sido corrompidos.

pronto que el envío ha fallado. Así, la estación transmisora no tendrá que esperarse un *time-out* para saber si se ha producido una colisión o un error en el enlace, y por lo tanto retransmitir el paquete a una velocidad adecuada.

Con este nuevo protocolo se han reducido los cambios en el estándar DCF, y no se ha necesitado ninguna modificación de la capa física. Por lo tanto es fácil de implementar.

### 4.2.2 FUNCIONAMIENTO DEL ALGORITMO LD-ARF

Este algoritmo, se basará en la tramas ACK, NACK y en el temporizador para proceder a incrementar la tasa (*rate-up timer*). Este tiempo se activará justo cuando se acabe de bajar el nivel de transmisión. Cuando expire el tiempo especificado, se seguirá el mecanismo para aumentar el ritmo de envío de datos. A continuación, se explicará el procedimiento que sigue este algoritmo para realizar la variación de la tasa, *Tabla 7*.

En una primera situación, nos podemos encontrar que recibimos una ACK, con lo cual podemos interpretar que la transmisión habrá sido completada con éxito. Entonces se incrementará el contador de la tasa de subida y se pondrá a cero el de bajada. Cuando el contador de subida sea igual o mayor al umbral predeterminado para elevar la tasa, se procederá a enviar los datos más deprisa. También, cuando el *rate-up timer* llegue a su valor predeterminado se realizaran los mismos pasos que cuando se reciba una ACK.

En cambio, cuando recibimos una NACK, se asumirá que se ha detectado una pérdida por un error en el canal. Al revés que en el caso anterior, se incrementará el contador de la tasa de bajada y se pondrá a cero el de subida. Cuando el contador de bajada sea igual o mayor al umbral para decrementar la tasa, se bajará al nivel inferior de transmisión. También se bajará, si se recibe una NACK justo después de subir la tasa de transmisión.

Como ya se ha explicado anteriormente, cuando ocurra una colisión la tasa de envío no se reducirá. Así, solo se disminuirá la velocidad de transmisión, cuando se produzca una pérdida de paquete debida a un error en el enlace, es decir, que llegue una NACK o que expire el tiempo de espera de una respuesta.

▪ <b>Si</b> se recibe una ACK (transmisión ha tenido éxito) o expire el <i>rate-up timer</i> , <b>entonces</b> $contador\_bajada = 0;$ $contador\_subida ++;$ <b>Si</b> ( $contador\_subida \geq N_{subida}$ ) { incremento de la tasa física; $contador\_subida = 0;$ parar <i>rate-up timer</i> ; }	▪ <b>Si</b> se recibe una NACK (se detecta un error en el enlace), <b>entonces</b> $contador\_subida = 0;$ $contador\_bajada ++;$ <b>Si</b> ( $contador\_bajada \geq N_{bajada}$ ) o la tasa se acaba de incrementar { incremento de la tasa física; $contador\_bajada = 0;$ activar <i>rate-up timer</i> ; }
---	---

Tabla 7. Estructura del algoritmo LD-ARF.

## 4.3 PROGRAMA DE SIMULACIÓN

Este programa de simulación se basará en el anterior, pero se introducirán modificaciones importantes. Mediante estos cambios se conseguirán unas mejores características de simulación, aunque tendrá una mayor complejidad a nivel de código.

La primera adaptación que se realiza es en el modelo del canal. Para este programa se utiliza el proceso determinista que emula un modelo de diez estados de Markov. Como se explicó en el capítulo 2, dentro de cada estado emulado de Markov habrá un modelo de dos estados de Gilbert-Elliot que definirá si se pierde o no el paquete. Las probabilidades de transición para el modelo de Gilbert-Elliot se pueden encontrar en la *Tabla 1*. Respecto a los procesos de llegada según la velocidad de transmisión, se continuará utilizando un proceso de Poisson. Además, las tasas de transmisión serán las mismas ya que seguimos con la misma versión del estándar. En cuanto al método que se utilizará para la diferenciación de pérdidas será el *básico*, ya que el RTS/CTS no es tan eficiente, y además también lo puede utilizar el ARF.

Por otra parte, en el programa se añade la trama de control NACK. Esta trama nos permitirá posteriormente diferenciar las pérdidas de los paquetes. Si recordamos la consideración del apartado 4.2.1, donde se argumentaba que si se producía un error debido al enlace, la cabecera muy probablemente llegaría. Según esto, en el programa de simulación se asumirá que aunque las ACK o las NACK sufran errores en el canal, el transmisor podrá saber que el receptor le ha intentado enviar una trama de control debido a la llegada de la cabecera. Así, la estación emisora siempre sabrá si un paquete fallido ha sufrido un error en enlace (se reciben datos por parte de la estación receptora), o una colisión (no se recibe nada). Una vez haya esperado un *time-out* y se determine que ha habido una colisión, se retransmitirá el paquete a la misma velocidad. El valor de este temporizador será equivalente al tiempo de ida de una trama de datos y de dos llegadas de tramas de control.

Los creadores del algoritmo LD-ARF [6] dejan como parámetros a determinar, los límites de ACK y NACK consecutivas para aumentar y disminuir la tasa, respectivamente. También se deberá escoger el valor del temporizador para enviar un paquete de sondeo a una velocidad superior, pero solo en las simulaciones en las que no se asumen colisiones. Estos parámetros se escogerán en realizarse la simulación.

Una vez ya se han especificado las características técnicas, se han creado dos versiones del programa para poder simular diferentes escenarios. En la primera versión, se ha simulado, sin contemplar colisiones, la transmisión de paquetes a una única estación receptora por parte de una o más estaciones emisoras. En la segunda, se limita a dos el número de estaciones emisoras y se muestran las colisiones producidas.

## 4.4 RESULTADOS OBTENIDOS EN LA SIMULACIÓN

Mediante las características del programa descrito en el anterior apartado, se obtendrán simulaciones concluyentes sobre el algoritmo LD-ARF. En primer lugar, nos situaremos en un escenario con solo una estación transmisora. Después se harán dos simulaciones con más de dos estaciones enviando. Finalmente, se mostrará una simulación entre tres estaciones (dos emisoras y una receptora) donde ocurrirán colisiones.

### 4.4.1 SIMULACIÓN 1: UNA ESTACIÓN TRANSMISORA

Esta primera simulación nos servirá para comprobar la funcionalidad del algoritmo. Se ha buscado por encima de todo la sencillez del escenario. Así, solo hay una estación transmisora y una estación receptora.

En la *Figura 20*, se puede apreciar claramente como actúa el algoritmo de adaptación según las condiciones del canal. En la sub-figura inferior se muestra el modelo de canal utilizado, con el proceso determinístico de estados entre el 1 y el 6. Como ya se enunció en el capítulo 2, cada estado estará formado por un modelo de Gilbert-Elliott propio. A medida que el número de estado vaya subiendo, las condiciones del canal empeorarán. Por lo tanto, si nos fijamos en el diagrama de barras superior de la *Figura 20*, podemos observar como en los estados 1, 2 y 3, recibimos muchas tramas de reconocimiento (barras de color verde), y la tasa aumenta (modos de transmisión en color amarillo). Por otro lado, cuando empeora el canal (estados 4, 5 y 6), se producen más pérdidas de paquetes por lo que se reduce la velocidad.

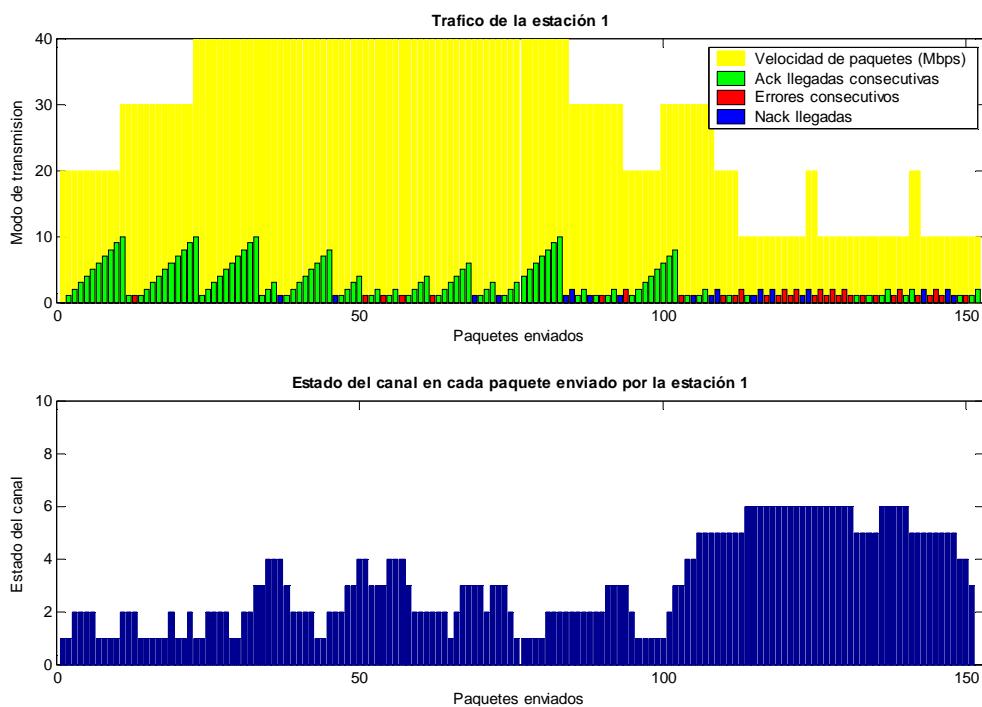
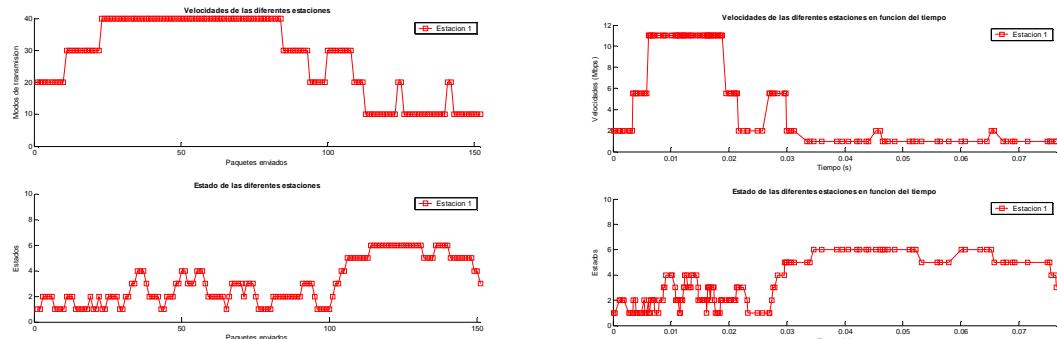


Figura 20. Tráfico de la estación transmisora según el modelo del canal de la simulación 1.

#### 4. Algoritmo LD-ARF

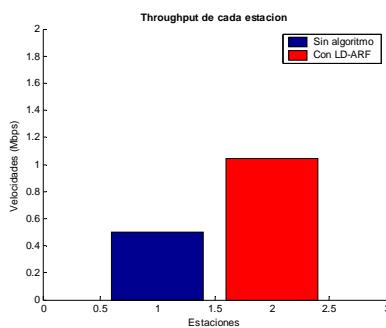
Para poder realizar esta simulación se ha introducido como umbral para incrementar la tasa un valor de 10 ACK consecutivas, y para decrementarla un valor de dos errores consecutivos en el canal. Además se han enviado 100 paquetes y se han retransmitido 50, sobre todo en la última parte. En la *Figura 20*, también podemos observar como actúa el temporizador, ya que envían paquetes de sondeo después de haber pasado cierto tiempo desde el primer error que hizo decrecer la tasa. En este caso, este tiempo equivale a 15 tramas correctas o fallidas. En cuanto a la información restante que se proporciona en este diagrama, podemos resaltar la diferencia entre los errores consecutivos (barras de color rojo) y las NACK llegadas (barras de color azul). Realmente tienen la misma función, que es comunicar al transmisor de que ha habido un error en el enlace. Sin embargo, una trama llega correctamente (NACK), y la otra llega defectuosamente, aunque el transmisor puede detectar que es una trama de control (ACK o NACK).

A continuación, se muestra en la *Figura 21* de manera clara el tiempo empleado para enviar cierta cantidad de paquetes a una determinada tasa. Como es lógico a una mayor velocidad, se transmitirán un mayor número de paquetes en menor tiempo. Pero esta trivialidad nos sirve para comprobar el correcto funcionamiento del proceso de Poisson.



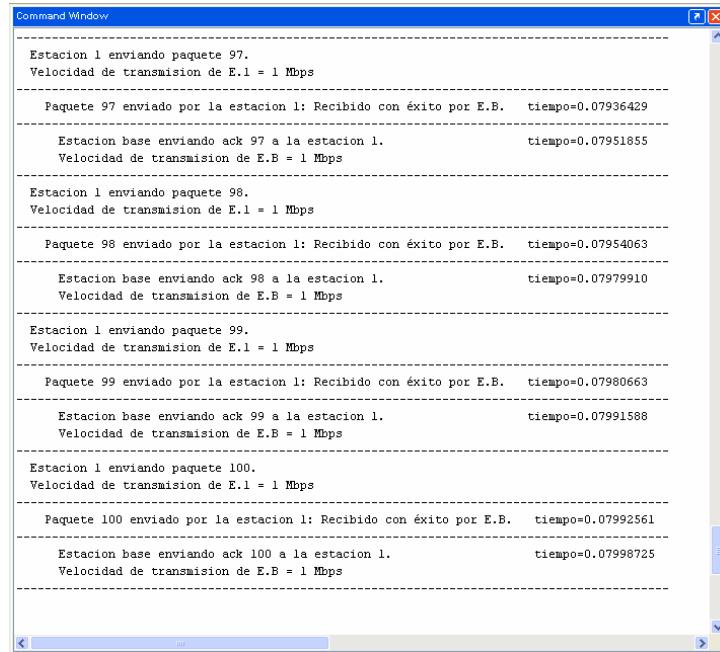
*Figura 21. Comparación entre velocidad en función del número de paquete enviado (izquierda parte superior), y la velocidad en función del tiempo de llegada de cada paquete (derecha parte superior). En la parte inferior izquierda se muestra el modelo del canal en función de los paquetes enviados, y en la parte inferior derecha el modelo del canal en función del tiempo.*

También, podemos corroborar como el algoritmo de adaptación realiza su función correctamente, aumentando el *throughput* hasta 1 Mbps, pese a los errores del canal.



#### 4. Algoritmo LD-ARF

Por último, el programa de simulación proporciona toda la información necesaria para saber en todo momento que ocurrió en el canal durante la transmisión. Esta información se muestra en la ventana de comandos de Matlab, y también queda guardada en el disco en un archivo de texto. En la *Figura 23* se muestra un trozo de los datos del resultado de la transmisión.



```

Command Window

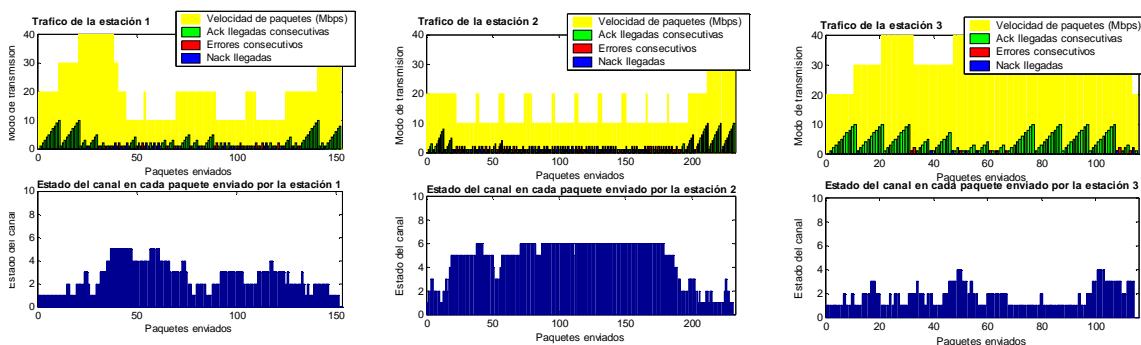
Estacion 1 enviando paquete 97.
Velocidad de transmision de E.1 = 1 Mbps
-----
Paquete 97 enviado por la estacion 1: Recibido con éxito por E.B. tiempo=0.07936429
-----
Estacion base enviando ack 97 a la estacion 1. tiempo=0.07951855
Velocidad de transmision de E.B = 1 Mbps
-----
Estacion 1 enviando paquete 98.
Velocidad de transmision de E.1 = 1 Mbps
-----
Paquete 98 enviado por la estacion 1: Recibido con éxito por E.B. tiempo=0.07954063
-----
Estacion base enviando ack 98 a la estacion 1. tiempo=0.07979910
Velocidad de transmision de E.B = 1 Mbps
-----
Estacion 1 enviando paquete 99.
Velocidad de transmision de E.1 = 1 Mbps
-----
Paquete 99 enviado por la estacion 1: Recibido con éxito por E.B. tiempo=0.07980663
-----
Estacion base enviando ack 99 a la estacion 1. tiempo=0.07991588
Velocidad de transmision de E.B = 1 Mbps
-----
Estacion 1 enviando paquete 100.
Velocidad de transmision de E.1 = 1 Mbps
-----
Paquete 100 enviado por la estacion 1: Recibido con éxito por E.B. tiempo=0.07992561
-----
Estacion base enviando ack 100 a la estacion 1. tiempo=0.07998725
Velocidad de transmision de E.B = 1 Mbps
-----
```

*Figura 23.* Ventana de comandos de Matlab, donde aparece la información sobre las transmisiones, proporcionada por el programa de simulación.

#### 4.4.2 SIMULACIÓN 2: MÁS DE UNA ESTACIÓN TRANSMISORA

Mediante las simulaciones de este apartado, podremos obtener la máxima velocidad recepción de datos correctos que se podría alcanzar si no hubiera colisiones. Así, habrá más de una estación transmisora y una sola receptora. Respecto a los parámetros que introduciremos en el programa, estos serán los mismos que en el apartado anterior.

En primer lugar, comenzaremos simulando con tres estaciones emisoras. Si analizamos las gráficas mostradas en la *Figura 24* y sin centrarnos a nivel de paquete, observamos como según los diferentes modelos de canal el LD-ARF adapta la tasa.



*Figura 24.* Tráfico de las tres estaciones. De izquierda a derecha las estaciones son: la 1, la 2 y la 3.

#### 4. Algoritmo LD-ARF

Si ahora nos fijamos en el resultado de las velocidades de las tres estaciones con el tiempo (*Figura 25*), podemos ver claramente lo que ha sucedido en el canal durante la simulación. Se puede observar como la estación 3 es primera en finalizar su transmisión, a esta le siguen la 1 y la 2, respectivamente.

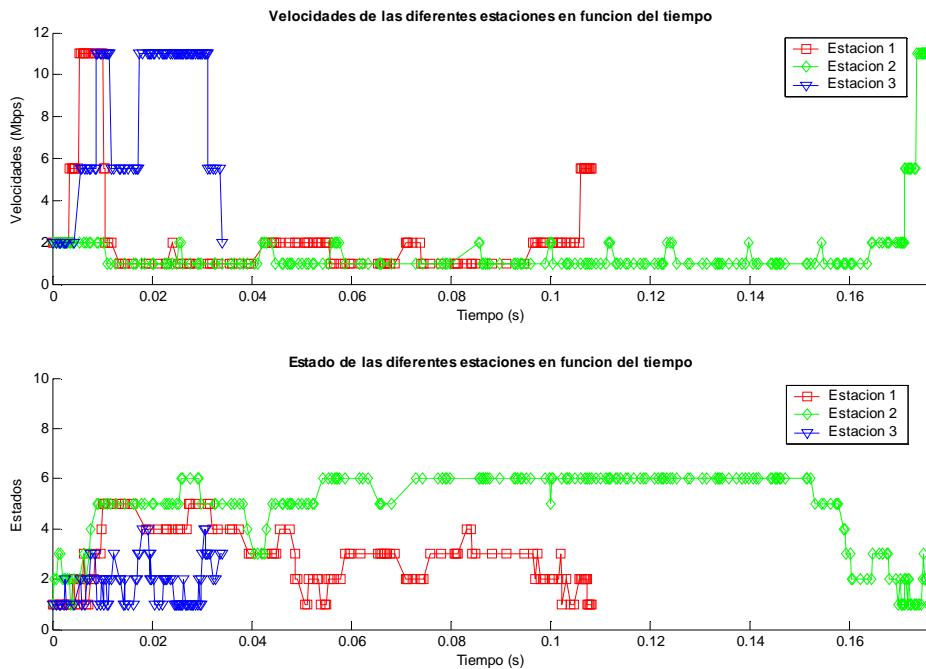


Figura 25. Velocidades de las diferentes estaciones en función del tiempo.

Mediante los resultados anteriores podemos argumentar, que las condiciones del canal determinan el tiempo de transmisión. Por lo tanto, este tiempo no solo depende de la tasa a la que se envíen los datos. No obstante, no se debe obviar que si coinciden estos dos factores el rendimiento aumentará. Es decir, si mejora el medio en el cual se transmiten los datos, y esta mejora se aprovecha aumentando la tasa de envío, probablemente el *throughput* crecerá considerablemente. Como prueba de ello, tenemos en la *Figura 26* un gran rendimiento de la estación 3 (buenas condiciones del canal), frente a uno pésimo de la estación 2 (condiciones relativamente malas). Todas estaciones ofrecen una mejora respecto a no aplicar ningún tipo de algoritmo.

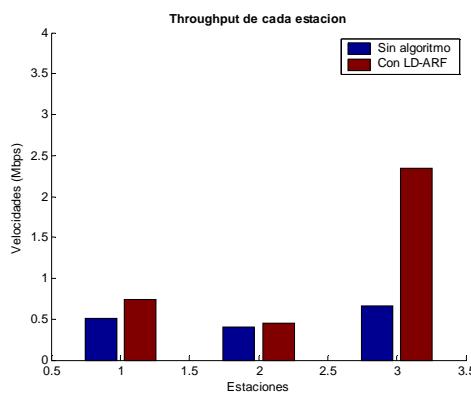


Figura 26. Throughput de las diferentes estaciones sin y con el algoritmo LD-ARF.

#### 4. Algoritmo LD-ARF

Para finalizar este sub-apartado, trataremos de producir unas condiciones del canal muy favorables, para averiguar cual es el throughput máximo que se puede conseguir aplicando el LD-ARF. Para ello, simularemos la transmisión de 100 paquetes por parte de 5 estaciones emisoras, y recibiremos los paquetes en una sola, sin asumir las colisiones que se producirían.

A partir de la *Figura 27* podemos comprobar como debido a las buenas condiciones del canal (solo se encuentra en los estados 1 y 2), las diferentes estaciones solo han tardado 35 milisegundos aproximadamente en transmitir 110 paquetes (entre envíos correctos y retransmisiones). Esto supone una gran utilización del canal por parte de las cinco estaciones.

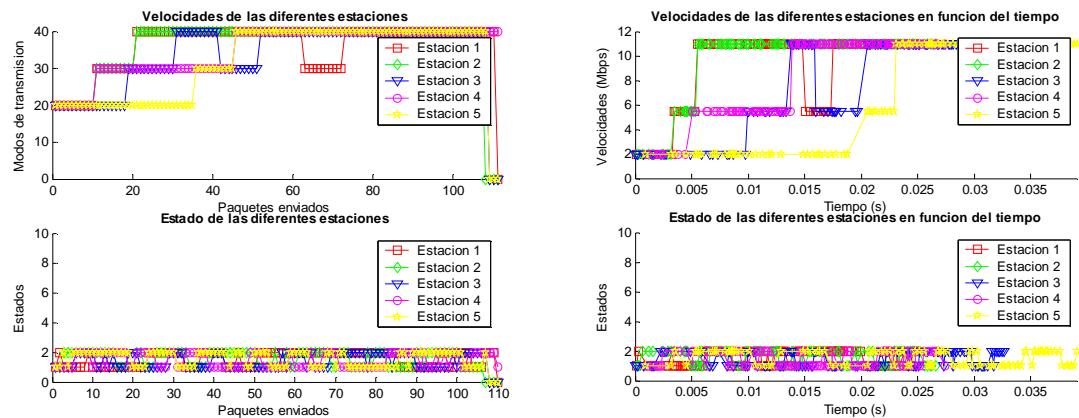


Figura 27. Velocidades de las cinco estaciones en función de los paquetes enviados (izquierda) y en función del tiempo (derecha).

A continuación, se ha obtenido el *throughput* de las respectivas estaciones como muestra la *Figura 28*. Mientras que, si no aplicamos el algoritmo obtendremos una media de 0.75 Mbps por cada estación aproximadamente. En cambio, si actúa el algoritmo el resultado ascenderá cerca de los 3 Mbps. Hay que considerar que no se ha simulado un canal perfecto libre de errores, por lo que se producían reenvíos.

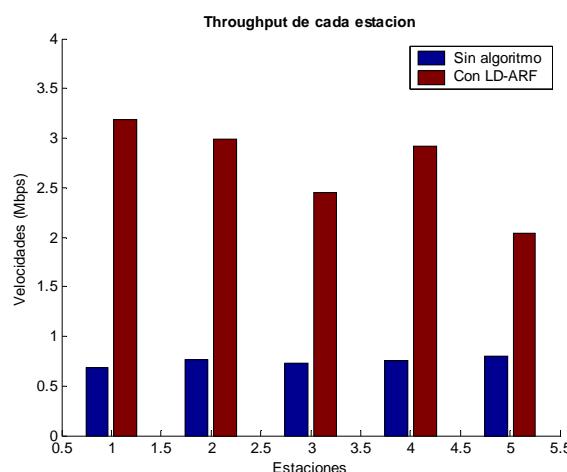


Figura 28. Throughput de las diferentes estaciones.

Finalmente, el *throughput* total que se ha obtenido ha sido de 10.2 Mbps. Este resultado se podría aplicar a un caso real si la estación receptora tuviera la suficiente de velocidad de recepción como capturar todos los paquetes enviados por todas las estaciones. Así, si no cumple con esta condición se producirán numerosas colisiones.

#### 4.4.3 SIMULACIÓN 3: COLISIONES EN LA RECEPCIÓN

En este último sub-apartado, se tratará el caso más real de todas las simulaciones. En este se tendrán en cuenta las colisiones que se producen en el envío de paquetes de dos estaciones emisoras a una receptora.

Esto conlleva una mayor complejidad a nivel de programación respecto a las anteriores simulaciones, ya que se comparan los tiempos de llegada de los paquetes de cada estación, y si algunos de ellos coinciden se determinará que ha habido una colisión. Por otra parte, se continuarán introduciendo los mismos parámetros que en las simulaciones anteriores, pero sin utilizar el temporizador.

En la *Figura 29* queda de manifiesto como el algoritmo LD-ARF en detectar que no se envió ninguna trama de control (ACK o NACK), asume que ocurrió una colisión en el receptor. Entonces, actúa en consecuencia, reenviando el paquete perdido y continuando con las condiciones anteriores a la colisión, como si no hubiera ocurrido nada en este intervalo de tiempo.

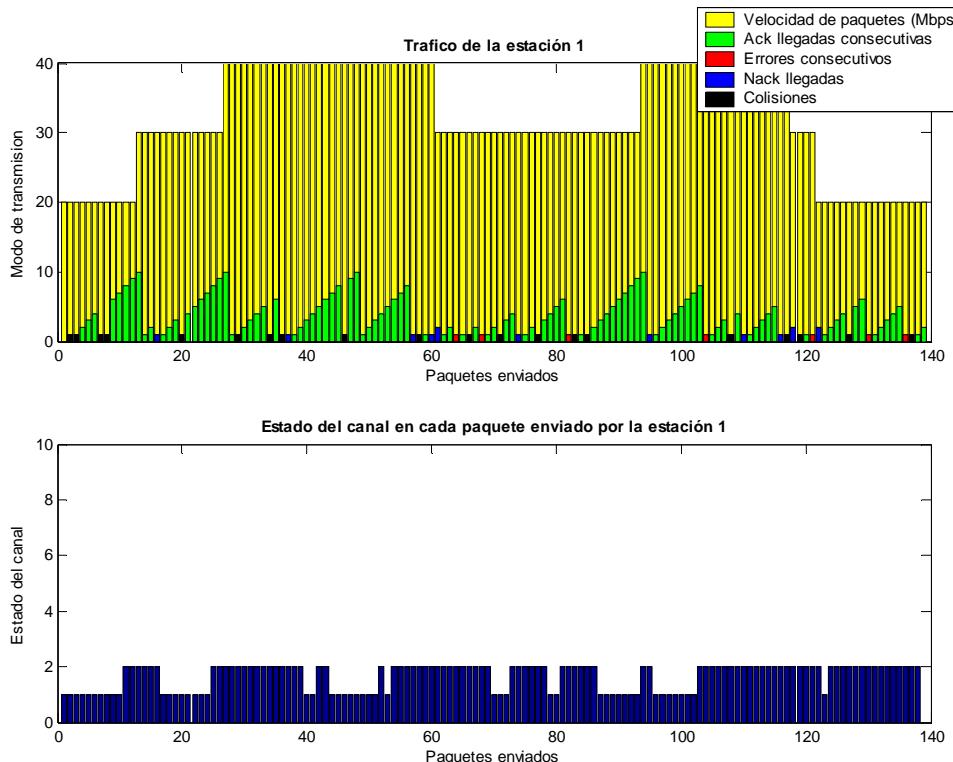
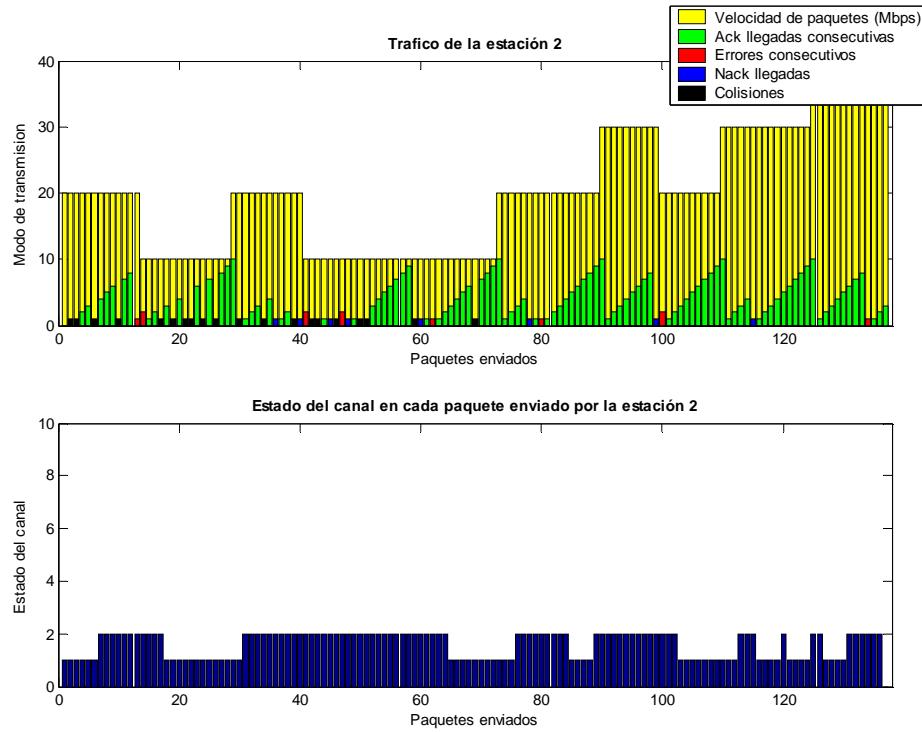


Figura 29. Tráfico en la estación 1 habiendo diferenciado las pérdidas y detectando que se han producido colisiones en la estación receptora.

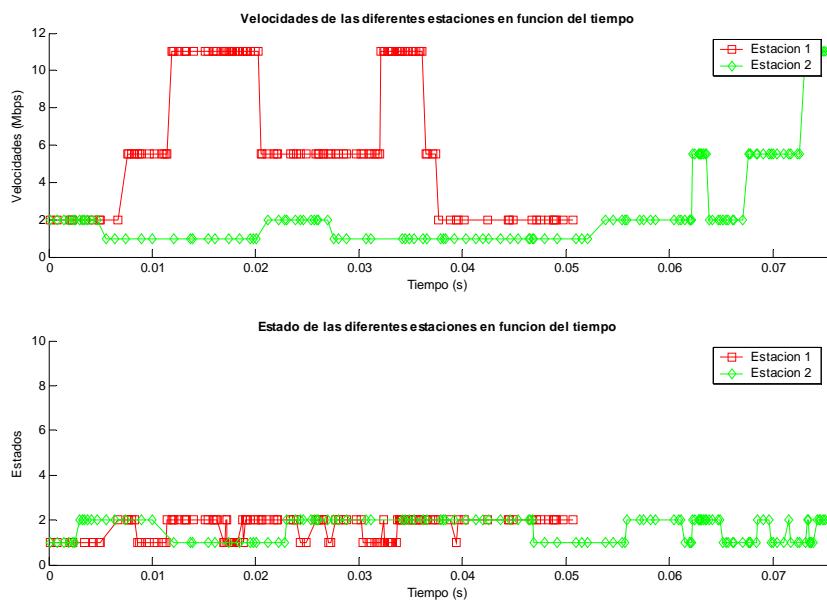
#### 4. Algoritmo LD-ARF

Si nos fijamos en la estación 2 (*Figura 30*), pasa exactamente lo mismo que en la otra estación. A parte de reenviar el paquete perdido, actúa de manera indiferente frente a la colisión.



*Figura 30. Tráfico en la estación 2 habiendo diferenciado las pérdidas y detectando que se han producido colisiones en la estación receptora.*

Por otra parte, observaremos las velocidades en función del tiempo en la en la *Figura 31*, centrándonos en el periodo de colisiones. En la estación 1, se producían colisiones durante toda la transmisión. En cambio, en la estación 2, se producían hasta los 70 paquetes enviados, que era cuando la estación 1 terminaba su transmisión.



*Figura 31. Comparación entre los períodos de transmisión de las dos estaciones.*

#### 4. Algoritmo LD-ARF

Como resultado final de esta simulación, aplicando el LD-ARF hemos obtenido un *throughput* bastante bueno como vemos en la *Figura 32*, ya que hay que considerar que se han realizado muchos reenvíos producidos por las colisiones. El *throughput* total en la recepción incluyendo las colisiones ha sido de 2.12 Mbps. Además, al no reducir la tasa de transmisión cuando ocurría cada colisión, se evitaba que transcurriera un período más largo en la transmisión, y por consiguiente que se produjeran más colisiones.

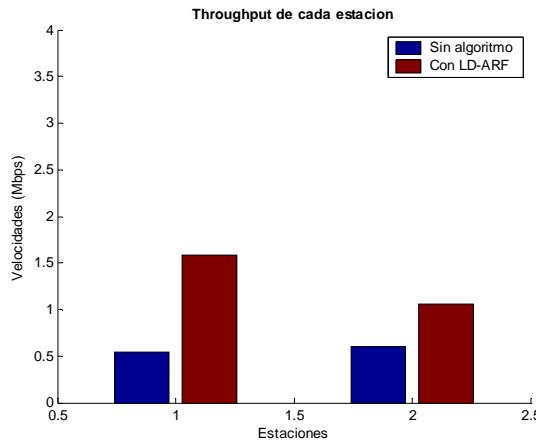


Figura 32. Comparación entre los throughput con y sin algoritmo en cada una de las estaciones.

Finalmente, en este programa se ofrece en una estructura gráfica todos los datos de la simulación. Así, a simple vista se puede saber lo que ha sucedido en la transmisión con información complementaria muy útil. En la *Figura 33* se expone un recorte del fichero generado por el programa. En la parte izquierda se presenta el esquema básico de envío y respuesta, y en la parte derecha se muestran diferentes características como el número de secuencia del paquete, la velocidad a la que se envía o el tiempo de llegada. Se contempla más caso en el *Apéndice*.

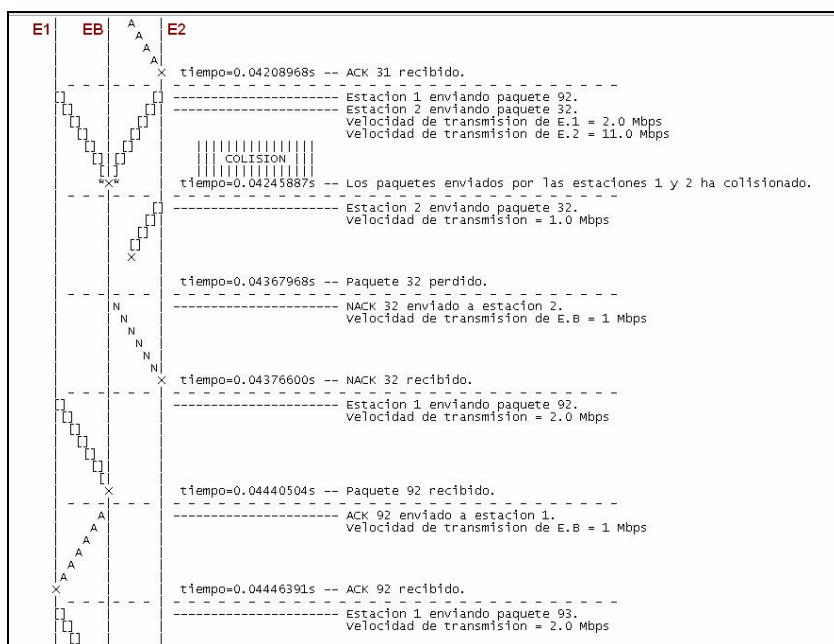


Figura 33. Estructura gráfica con los datos de la simulación.

## 4.5 CONCLUSIONES

Como hemos visto, la adaptación de la tasa tiene una gran importancia en los medios cambiantes. Cualquier oportunidad de mejora que ofrezca el canal se debe aprovechar, pues puede que no se vuelva a repetir.

En la primera simulación, pudimos apreciar como el algoritmo LD-ARF tenía un funcionamiento casi idéntico en condiciones favorables y con solo una única estación transmisora. También comprobamos la efectividad del temporizador para aumentar relativamente el *throughput*.

Después, estudiamos una situación muy favorable del canal en la cual se consideraba un receptor muy rápido en la captura de datos, y se asumía que ningún paquete llegaría en el mismo instante de tiempo que otro. Así, pudimos comprobar como con 5 estaciones transmisoras, se podía llegar a obtener una velocidad de recepción total de datos correctos de aproximadamente 10 Mbps.

Finalmente, dado que en la realidad ocurren colisiones, se estudiaron las posibilidades del algoritmo LD-ARF en estas situaciones. Con los resultados obtenidos se ha comprobado como el LD-ARF es capaz de detectar las colisiones (diferenciando las pérdidas) y no reducir la tasa cuando las condiciones del canal son buenas.

## Capítulo 5

# Modelado del canal basado en medidas experimentales

Para poder realizar el modelado del canal basado en medidas experimentales se contemplaron diferentes opciones para realizar las capturas. A continuación, se comentaran las ventajas y los inconvenientes de las estrategias utilizadas.

## 5.1 EQUIPO UTILIZADO

Antes de explicar los pasos que se realizaron para tomar la medidas, se explicará brevemente el material utilizado.

Los dispositivos principales para esta tarea fueron los dos portátiles idénticos que sirvieron para transmitir y recibir los paquetes. En concreto, son modelos “*Benq joybook S52*” (*Figura 34.a*) en los cuales se había instalado el sistema operativo Ubuntu Linux. Además llevaba incorporada una tarjeta de red inalámbrica interna del modelo “*Intel Pro Wireless 2200BG*”. También se disponía de una tarjeta PCMCIA “*D-link DWL-G650*”, aunque esta no se utilizó tanto como la interna.

Finalmente, el último dispositivo que se pudo utilizar fue el sistema de análisis de redes 802.11b/g llamado *Yellowjacket*, el cual vemos en la *Figura 34.b*. Este consistía en una *pocket PC* a la cual iba adaptado un hardware específico, que nos mostraba diferentes características de la red inalámbrica en la que nos encontrábamos.



a)

b)

Figura 34. Dispositivos utilizados para realizar las medidas. En a) vemos uno de los portátiles, y en b) el yellowjacket.

## 5.2 ESTRATEGIA 1: PROCESO COMPLETO DE INTERCAMBIO DE PAQUETES

La primera estrategia que se pensó, se basaba en obtener mediante medidas el proceso completo de una transmisión. Es decir, se buscaba averiguar los tiempos de llegada de las tramas de datos enviadas, y los tiempos de las tramas de reconocimiento recibidas. Así se buscaba conseguir directamente las diferentes situaciones que se podrían dar en el canal. Así, el algoritmo se aplicaría sobre las medidas con una adaptación del programa de simulación. De esta manera no se debería predecir la pérdida de paquetes ya que se utilizaría una secuencia fija determinada por las muestras. Sin embargo, este método conllevaba una serie de limitaciones y consideraciones que se explicaran a continuación.

El primer inconveniente que se encontró, es que no se pudo capturar ni se detectó ninguna trama de reconocimiento enviada por la estación receptora. Al no enviar tramas ACK, no se podía medir un tráfico normal basado en el protocolo de parada y espera o de *Stop&Wait* que vimos en el tercer capítulo. Así que se optó por variar un poco el modelo. En vez de enviar un paquete y esperar una respuesta, se enviaron ráfagas de paquetes, y acto seguido se intercambiaban los portátiles para medir las tramas ACK de respuesta. Esto provocó que las medidas no fueran muy correctas ya que en el tiempo de intercambio el canal podía haber cambiado.

El segundo inconveniente fueron los intervalos de tiempo que espera el transmisor en enviar cada paquete. Así se capturaban tiempos de llegada muy superiores a los que se deberían dar según la tasa de envío, dando como resultado un *throughput erróneo*. Por la tanto, se intentó comparar los tiempos de llegada de las diferentes realizaciones. En todas las mediciones se estimo que entre el primer y segundo paquete había un tiempo aproximado de 50ms a 100ms. A continuación, se observó como aproximadamente 100ms llegaba entre 4 y 5, como se muestra en la *Figura 35.a*. Por desgracia, estas medidas reales están expuestas a fluctuaciones del canal y a diferentes factores que pueden alterar su tiempo de llegada, dando un procesado incorrecto. También se intento aumentar el intervalo de tiempo, para que fuera conocido, pero no se consiguió.

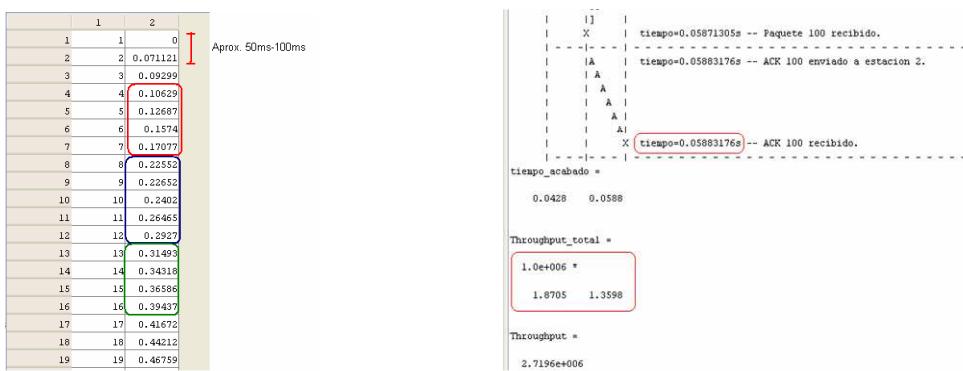


Figura 35. Procesado de los datos de las llegadas de los paquetes.

En la *Figura 35.a* y en la *Figura 35.b* se puede observar como los tiempos medidos difieren mucho de los obtenidos por un proceso de Poisson como el utilizado para el programa de simulación.

Finalmente, se intentaron plantear ciertas soluciones para este método, pero no fueron muy concluyentes. Por las razones anteriores, no se contempla esta estrategia para realizar el modelado del canal.

### 5.3 ESTRATEGIA 2: ESCUCHA DEL CANAL

Después de haber tenido el anterior contratiempo, se pensó en modelar el canal a partir de diferentes rangos de SNR. En estos rangos se comprobaría los paquetes perdidos que se habían producido en cierto instante.

Para ello se utilizó el dispositivo Yellowjacket. Con este se analizó el tráfico que se transmitía por la red. Además, se utilizó la opción de detectar todos los paquetes que se dirigen a una red. Sin embargo, con esta opción se captura una gran cantidad de paquetes por lo que no se puede distinguir los datos enviados.

Por otra parte, aunque no se modele el canal con este método, se podrá obtener una idea de la señal que recibe el receptor. Además podremos saber el grado de utilización del canal, en el cual se pueda transmitir los datos sin que haya interferencias por otros dispositivos.

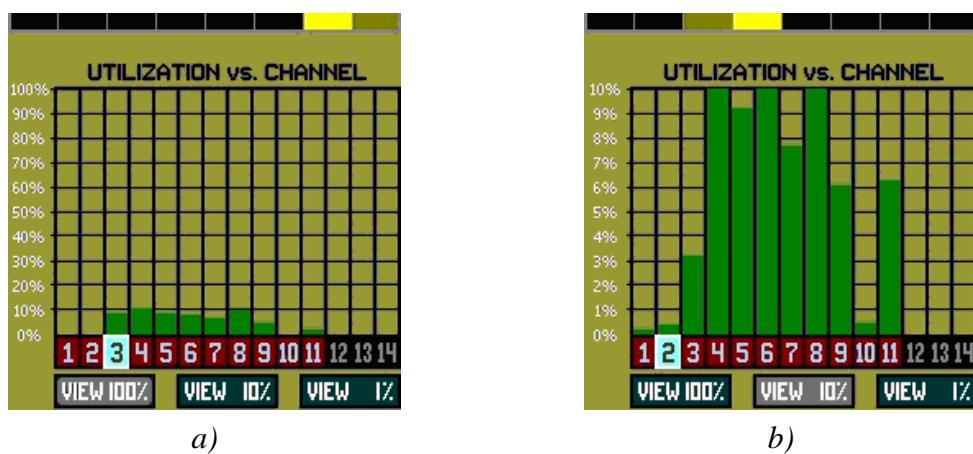


Figura 36. Distribución en función de la frecuencia las diferentes señales de potencia.

En la *Figura 36*, en el punto donde se tomaron las medidas la red no estaba nada saturada, ya que la utilización estaba por debajo del 10%. Ampliando la *Figura 36.a* obtenemos la *Figura 36.b*), donde se puede apreciar que los canales 1 y 2 no tiene mucha utilización. Así la medidas se realizaran en estos canal para evitar interferencia.

## 5.4 ESTRATEGIA 3: MODO MONITOR

Después de haber planteado dos métodos, se intentará establecer una nueva estrategia para obtener el modelo. Se partirá de la idea básica del anterior apartado, donde se distinguían diferentes rangos de potencia de señal recibida.

En primer lugar, se configurará en modo *ad-hoc* un portátil, y el otro en modo *monitor* para que pueda capturar todos los paquetes que recibe. Al poner en modo monitor, las tramas vendrán ampliadas por información complementaria referente a la capa física, a la capa MAC o a la capa de enlace lógico. En las características que se pueden obtener encontramos el RSSI (*Received Signal Strength Indicator*) que nos dará el nivel de señal recibida en dBm. También podemos encontrar otros parámetros como la tasa de transmisión, o la frecuencia del canal que se está utilizando.

Una vez ya sabemos los datos que podemos utilizar, la estrategia se centrará en establecer tres rangos de nivel de señal, y en cada uno de estos se definirá un modelo de Gilbert-Elliot.

Pero antes de crear el modelo del canal, se deberán tomar las medidas. Mediante las tarjetas internas, los datos se enviarán desde un portátil y se recibirán en el otro. Estas medidas se capturarán utilizando el programa *tcpdump*, donde se filtrarán los paquetes recibidos que se correspondan con las direcciones de origen y destino especificadas. A continuación, se analizarán las medidas en el programa *ethereal* y se exportarán a un fichero de texto. Finalmente, se leerán los datos y se procesarán en Matlab.

## 5.5 MODELADO DEL CANAL FÍSICO

El modelado del canal puede ser correcto si se analiza el nivel de señal recibido (RSS) medido según las tramas enviadas por el transmisor, y tomando la siguiente consideración. Si es fijo o varía poco el nivel de potencia transmitida por la estación emisora, entonces los cambios en el RSS probablemente serán indicativos de las condiciones del canal.

En cuanto a la toma de medidas, el receptor se quedará fijo en una situación determinada, y el transmisor se irá moviendo. Este transmisor caminará a una velocidad aproximada de 0.5 m/s.

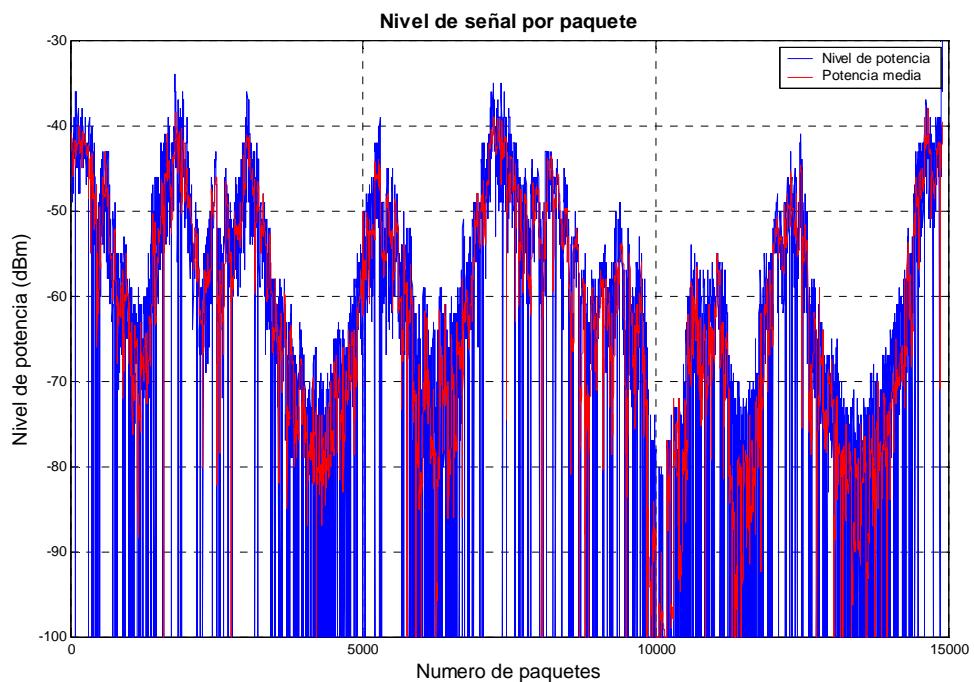
Finalmente, se establecerán 3 estados que emularán a un proceso de Markov. Los rangos de potencia que separarán los estados 1, 2 y 3 son de 0 a -30 dBm, de -30 dBm a -60 dBm y menor a -60 dBm, respectivamente.

### 5.5.1 PRUEBA 1

Mediante las siguientes pruebas se intentará modelar el canal del interior del edificio de la *Escola Técnica Superior d'Enginyeria* (ETSE). Para la primera prueba el receptor se ha situado en el vestíbulo principal al lado de la espina Q1. En cambio el transmisor se ha ido moviendo entre los diferentes pasadizos o escaleras.

Una vez tenemos las medidas procesadas, podremos visualizarlas en una gráfica como la de la *Figura 37*. En esta gráfica se encuentran todos los valores calculados a partir de las tramas recibidas, y además se ha añadido su potencia media intermedia.

Por otra parte, si un paquete no se ha recibido o se ha recibido con un nivel de señal muy bajo, inferior a los -85 dBm aproximadamente, se considerará que su nivel de señal es de -100 dBm. Esto es debido a que con estas medidas no se puede determinar el nivel de señal de los paquetes perdidos. Por otra parte, esto nos sirve para ver directamente cuando hay una mayor perdida de paquetes.



*Figura 37. Nivel de potencia de la señal recibida en función del número de paquetes para la prueba 1.*

En la *Figura 38* se muestran los estados de Markov en función del tiempo. En esta prueba se han detectado 15000 paquetes enviados en un tiempo de aproximadamente 457 segundos.

Dado que el emisor no ha estado lo suficientemente cerca como para que se reciba la señal por encima de -30dBm, no se ha pasado por el primer estado de Markov. Aproximadamente el emisor y el receptor deben estar situados a una distancia de entre

## 5. Modelado del canal basado en medidas experimentales

medio metro y un metro, aunque si se enfocan bien las antenas de los portátiles pueden alcanzar un poco más.

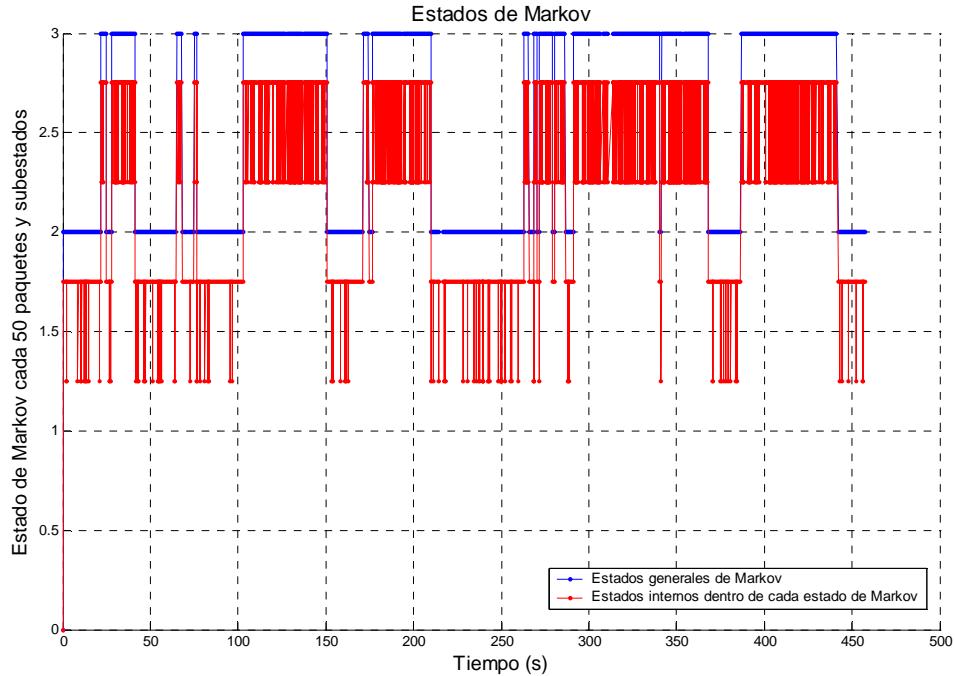


Figura 38. Estados de markov en función del tiempo para la primera prueba.

Para poder realizar el modelado del canal, se aplicará el proceso determinístico anterior de tres estados, emulando así un proceso de Markov. Pero como ya hemos dicho anteriormente, en cada uno de los tres estados tendremos un modelo de dos estados como se muestra en color rojo en la *Figura 38*. Este modelo de dos estados se expone en el diagrama de la *Figura 39*.

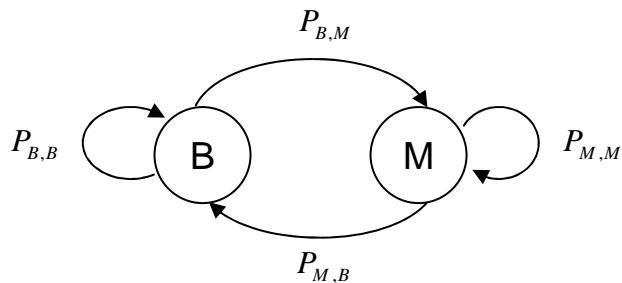


Figura 39. Diagrama de los subestados que se encuentran en cada estado de Markov.

A partir de [17], hemos calculado las probabilidades de transición del modelo de Gilbert-Elliott para cada estado, como se muestra en la *Tabla 8*. Puesto que no se ha pasado por el estado 1, no se han podido calcular las probabilidades correspondientes.

PROBABILIDAD	ESTADO 1	ESTADO 2	ESTADO 3
Prob. de bueno a bueno	NaN	0.9686	0.7049
Prob. de bueno a malo	NaN	0.0314	0.2951
Prob. de malo a malo	NaN	0.0314	0.2951
Prob. de malo a bueno	NaN	0.9686	0.7049

Tabla 8. Probabilidades de los diferentes estados de Markov.

### 5.5.2 PRUEBA 2

Esta segunda prueba servirá para corroborar los resultados obtenidos en la primera prueba. Esto es así ya que se han tomado las medidas en los mismos lugares y en tiempos similares. De nuevo recorreremos caminando las espinas Q1, Q2, Q3 y hasta Q4 de planta baja. También volveremos a subir las escaleras hacia la primera planta y bajaremos por escaleras principales. Segundo este recorrido obtenemos el nivel de señal de la *Figura 40*.

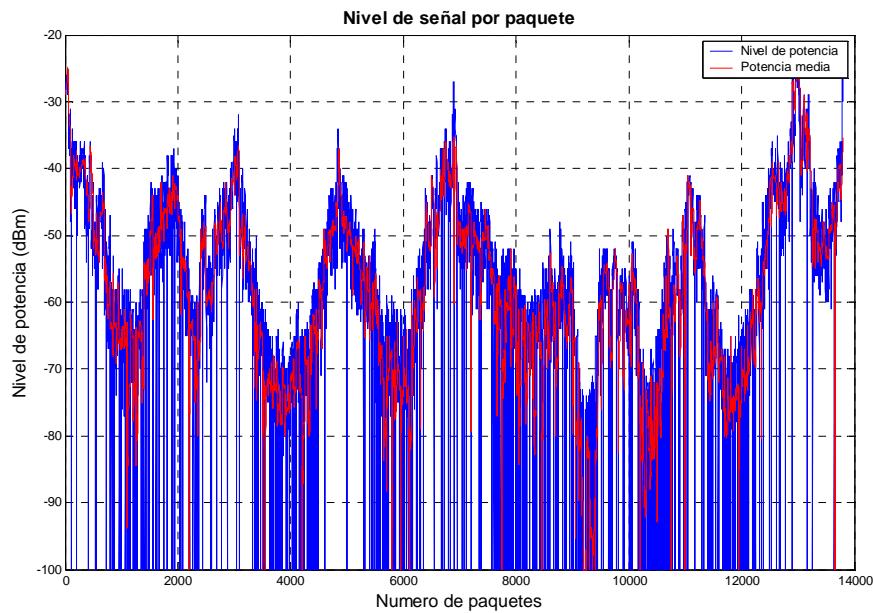


Figura 40. Nivel de potencia de la señal recibida en función del número de paquetes para la prueba 2.

En la *Figura 41*, se muestran los estados de Markov en función del tiempo. En esta prueba se han detectado 14000 paquetes enviados en un tiempo de aproximadamente 420 segundos.

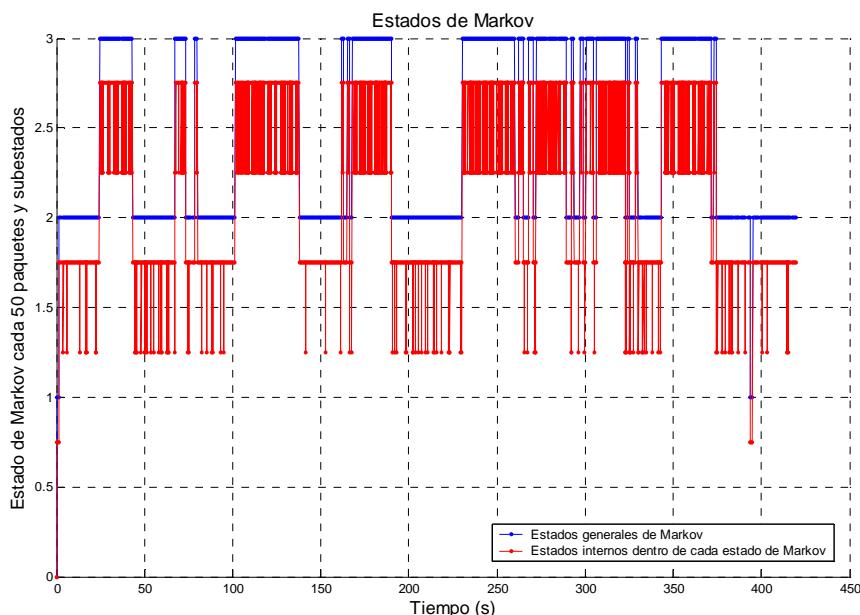


Figura 41. Estados de Markov en función del tiempo para la segunda prueba.

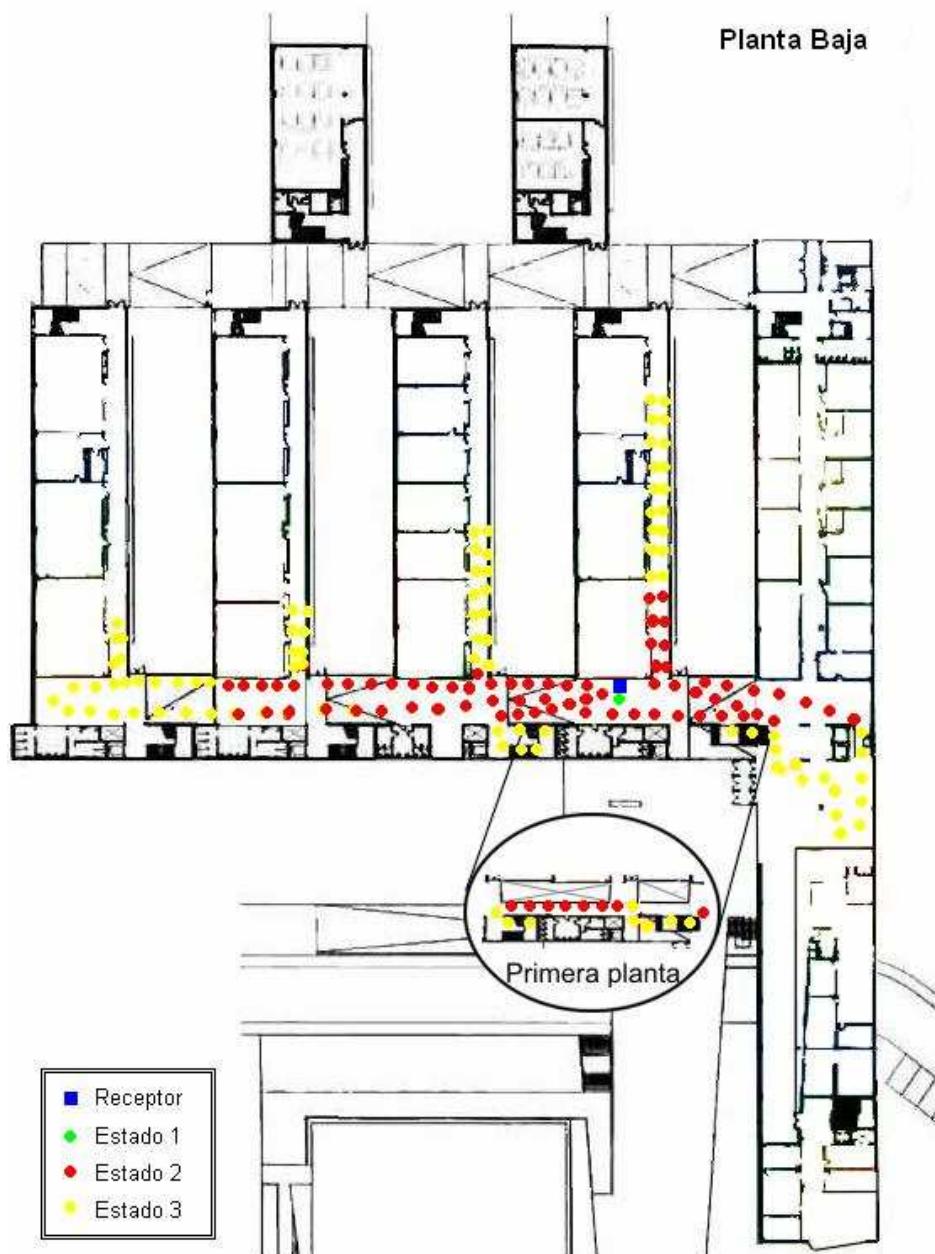
## 5. Modelado del canal basado en medidas experimentales

En la *Tabla 9* se muestran los resultados de las probabilidades para esta prueba.

PROBABILIDAD	ESTADO 1	ESTADO 2	ESTADO 3
Prob. de bueno a bueno	1.0000	0.9663	0.7629
Prob. de bueno a malo	0	0.0337	0.2371
Prob. de malo a malo	0	0.0337	0.2371
Prob. de malo a bueno	1.0000	0.9663	0.7629

*Tabla 9. Probabilidades de los diferentes estados de Markov para la simulación.*

Respecto a las pruebas realizadas, pese a haber enviado 1000 paquetes menos en la segunda prueba que en la primera, se puede comprobar en las *Tabla 8* y *Tabla 9* como las probabilidades son muy parecidas. Con los resultados obtenidos se puede concluir que el modelado del canal es correcto, para el escenario que se muestra en *Figura 42*.



*Figura 42. Plano de situación de los estados para la prueba 1 y 2.*

### 5.5.3 PRUEBA 3

En este caso, el receptor esta situado encima de las escaleras principales de delante del vestíbulo principal. Se comprueba como en pasar por debajo de estas escaleras el nivel de señal empeora mucho (alrededor de la muestra 5000). Aunque las mayores atenuaciones se consiguen por efecto de la distancia (alrededor de muestras 10000 y 12000).

Por otra parte, la recepción de la señal no ha sido mayor a -30 dBm, incluso a menos de un metro de distancia entre el receptor y el emisor, como vemos en la *Figura 43*.

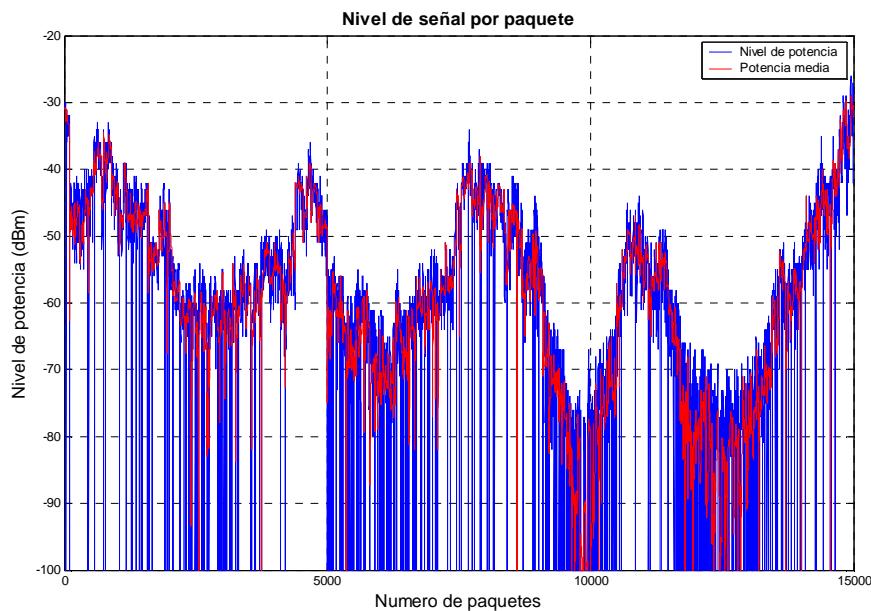


Figura 43. Nivel de potencia de la señal recibida en función del número de paquetes para la prueba 3.

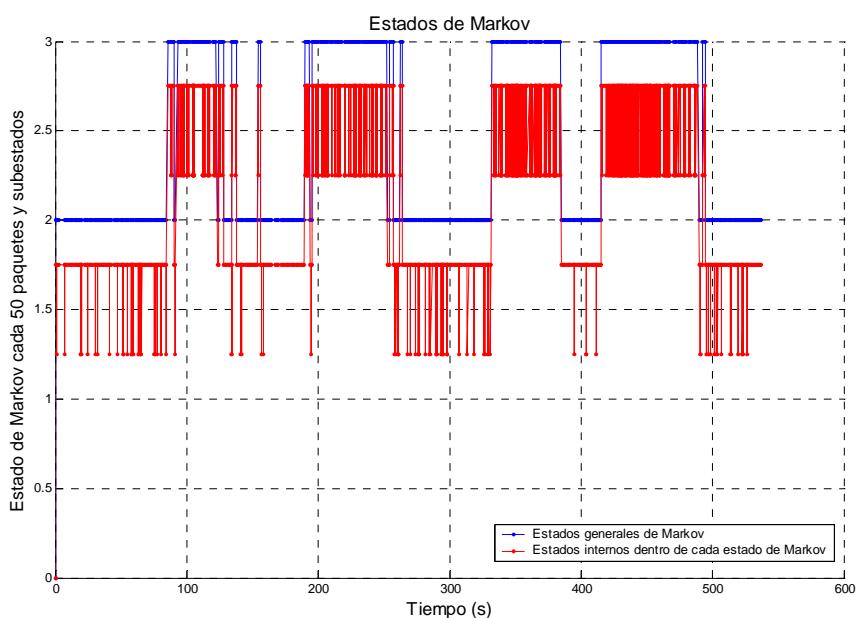


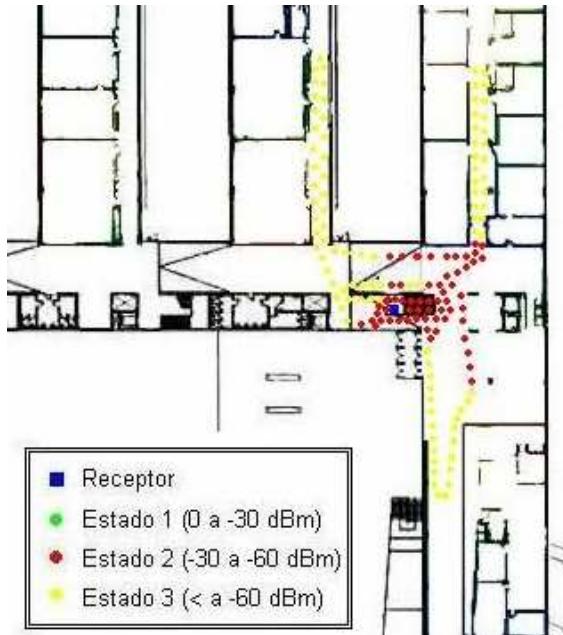
Figura 44. Estados de Markov en función del tiempo para la tercera prueba.

## 5. Modelado del canal basado en medidas experimentales

Las probabilidades de transiciones se muestran en la *Tabla 10*, y el plano con la situación de todos los puntos de envío se encuentra en la *Figura 45*. De nuevo volvemos a tener probabilidades parecidas a las de los anteriores casos.

PROBABILIDAD	ESTADO 1	ESTADO 2	ESTADO 3
Prob. de bueno a bueno	NaN	0.9752	0.7958
Prob. de bueno a malo	NaN	0.0248	0.2042
Prob. de malo a malo	NaN	0.0248	0.2042
Prob. de malo a bueno	NaN	0.9752	0.7958

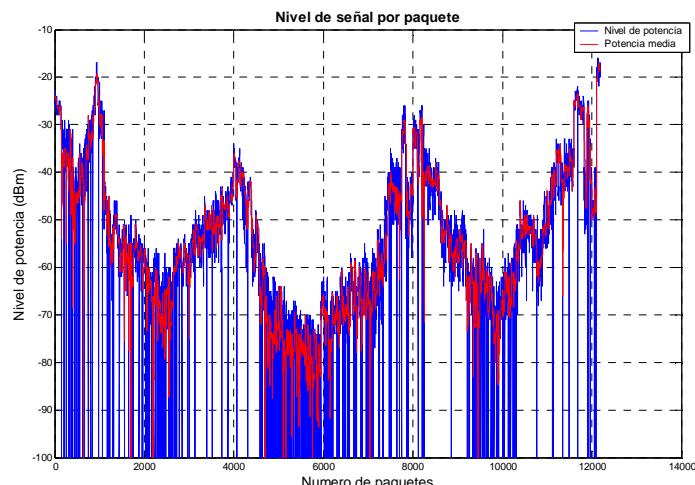
*Tabla 10. Probabilidades de los diferentes estados de Markov para la simulación.*



*Figura 45. Plano de situación de los estados para la prueba 3.*

### 5.5.4 PRUEBA 4

Por último, el receptor está situado a lado de las escaleras de la espina Q2 justo detrás de un muro. Se intenta comprobar como se obtiene mayor nivel de señal por las reflexiones de los muros que rodean al receptor, y como se pierde mucha potencia cuando se está detrás de uno de los muros.



*Figura 46. Nivel de potencia de la señal recibida en función del número de paquetes para la prueba 4.*

El tiempo que se muestra en la *Figura 47*, solo es correcto para los paquetes recibidos, porque en las mediciones solo se captura el tiempo en el que se recibe un paquete. Por lo tanto, no se puede saber el tiempo en el que se perdió uno, como es lógico. Esto es un contratiempo si se quiere visualizar el nivel de señal de los paquetes (tanto los recibidos como los fallidos) ya que solo se podrán visualizar en función del tiempo los recibidos. Por este motivo en la gráfica anterior donde se expone el estado de Markov por cada intervalo de tiempo, los paquetes fallidos tienen el mismo tiempo que el último recibido.

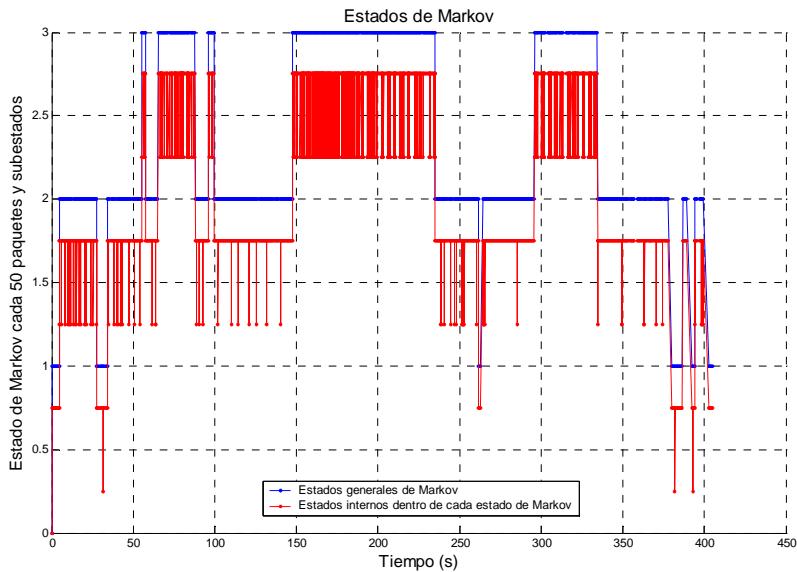


Figura 47. Estados de Markov en función del tiempo para la cuarta prueba.

Las probabilidades de transición del modelo interno de Gilbert-Elliott para cada estación se pueden observar en la *Tabla 11*.

PROBABILIDAD	ESTADO 1	ESTADO 2	ESTADO 3
Prob. de bueno a bueno	0.9951	0.9836	0.8037
Prob. de bueno a malo	0.0049	0.0164	0.1963
Prob. de malo a malo	0.0049	0.0164	0.1963
Prob. de malo a bueno	0.9951	0.9836	0.8037

Tabla 11. Probabilidades de los diferentes estados de Markov para la simulación.

En la *Figura 48*, se ve como en las mediciones no hubo un flujo constante de envío de datos ya que estos intervalos eran utilizados para ejecutar manualmente la orden de enviar paquetes en el portátil. También se demuestra como los tiempos de los datos fallidos son los mismos que el del último paquete recibido. Sin embargo, el modelo sigue siendo correcto ya que los tiempos entre cada envío son despreciables, y para realizar el cálculo de las probabilidades de transición, solo se necesita el intervalo de tiempo en que no llegó un paquete (no cuando se perdió).

Finalmente, en esta última prueba se ha podido observar como el proceso pasa varias veces por el estado 1, debido como ya hemos dicho a las múltiples reflexiones del entorno. Se ha podido comprobar como los muros reflejan casi toda la señal que incide

## 5. Modelado del canal basado en medidas experimentales

sobre ellos, y que incluso llegan a amplificar la señal. En cambio, cuando nos situamos detrás de la pared en la que se encuentra el receptor, y el único camino posible es un cristal, la señal si consigue llegar lo hará con un nivel bajo. Este tipo de materiales dejan pasar gran parte de la señal, y poca la reflejan. La situación de todos los puntos se observa en la *Figura 49*.

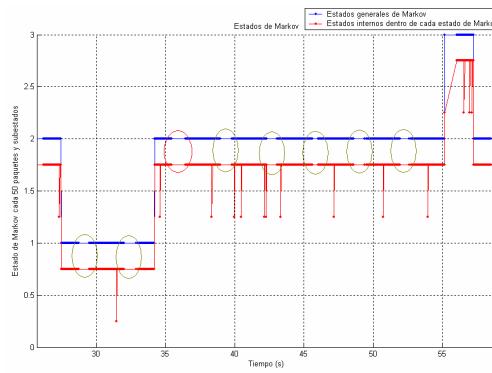


Figura 48. Detalle de los estados del proceso determinístico.

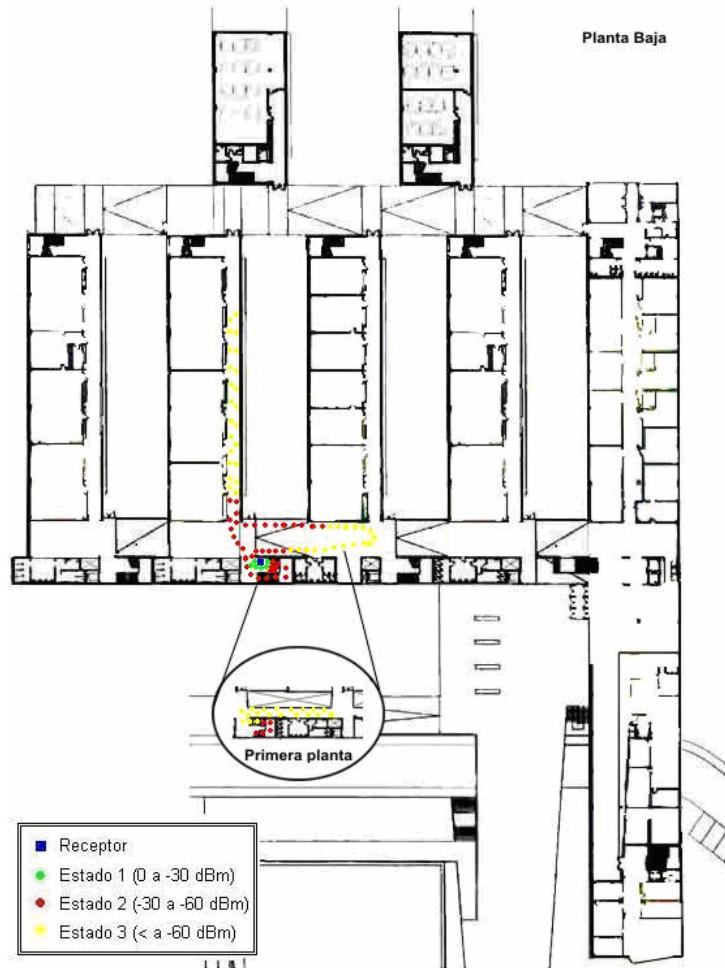


Figura 49. Plano de situación de los estados para la prueba 4.

En el próximo capítulo se utilizarán los modelos anteriores y se aplicarán los diferentes algoritmos para adaptar la tasa a las condiciones del canal modelado.

## Capítulo 6

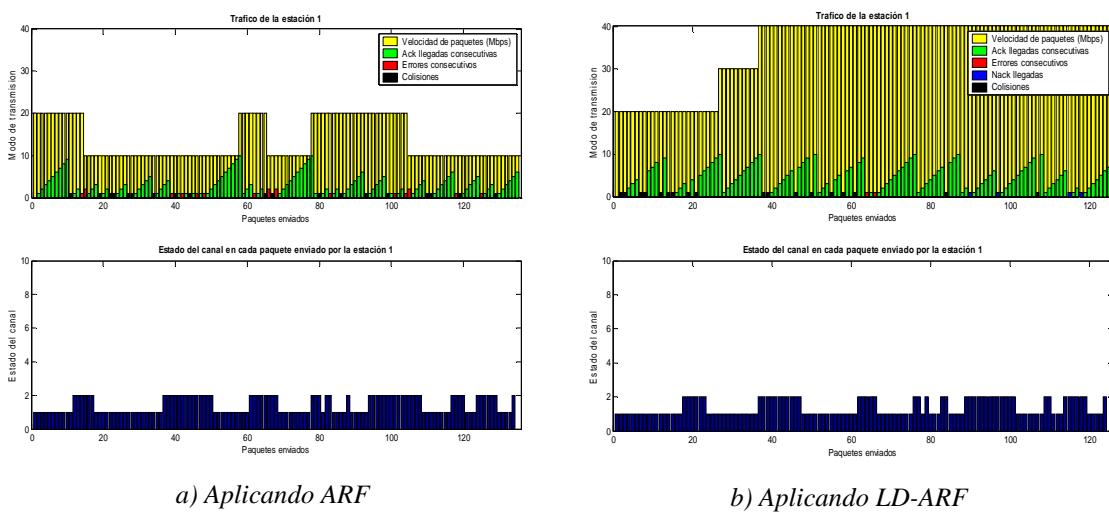
# Comparación entre algoritmos

En este capítulo se comparan los algoritmos ARF y LD-ARF. Estos se aplicarán tanto para el canal simulado como el basado en medidas experimentales.

### 6.1 COMPARACIÓN CON MODELO DE CANAL SIMULADO

Para realizar la comparación del algoritmo, se establecerá un escenario donde se marquen las diferencias. Hasta ahora ya hemos visto como en condiciones normales los dos algoritmos actúan de forma muy parecida (uno está basado en el otro). Sin embargo hay situaciones en las que uno pierde efectividad frente al otro.

Así el escenario que se planteará será el de dos estaciones transmisoras enviando a la misma estación receptora, con lo cual se producirán colisiones. Para que la comparación sea más imparcial, los dos tendrán los mismos parámetros de entrada y tendrán el mismo proceso determinista. En cuanto al temporizador, este no se utilizará ya que funciona de la misma manera en los dos y así se simplificará el programa.



*Figura 50. Comparación en la estación 1 aplicando el ARF y el LD-ARF.*

Una vez ya están los dos algoritmos simulados, comparemos los resultados. En la *Figura 50*, observamos como el algoritmo ARF asume las colisiones como errores

## 6. Comparación entre algoritmos

(porque no los diferencia), y por lo tanto reduce la tasa. En cambio el LD-ARF, detecta las colisiones y lo único que hace es retransmitir las tramas. Esto le supone al LD-ARF tener la oportunidad de seguir sumando tramas ACK consecutivas, y por consiguiente incrementar la tasa.

A continuación, nos fijaremos en la estación 2 que se corresponde con la *Figura 51*, y podremos observar como se obtienen resultados similares. Un caso típico de error del algoritmo ARF, es cuando esta apunto de incrementar la tasa y debe reiniciar el contador porque ha ocurrido una colisión. Así se penaliza dos veces, tanto por la retransmisión del paquete como el no incremento de la tasa.

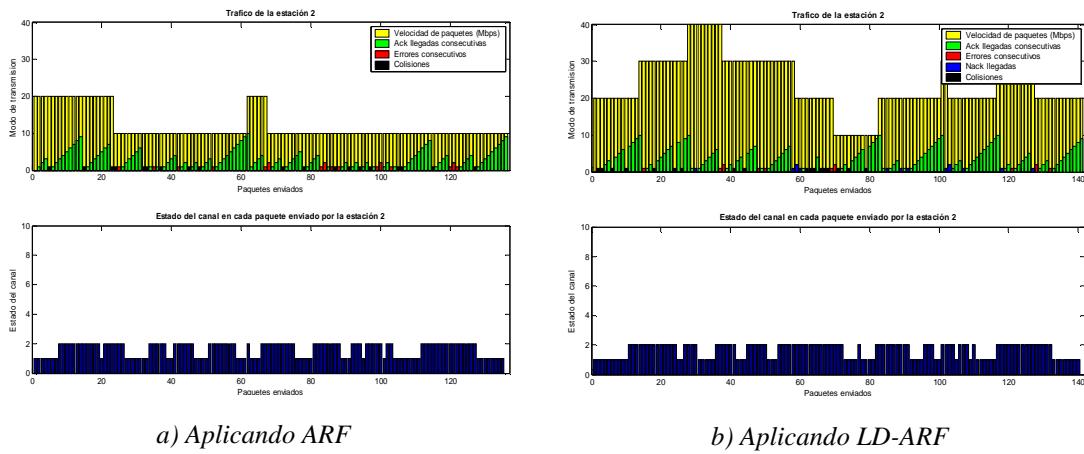


Figura 51. Comparación en la estación 2 aplicando el ARF y el LD-ARF.

Las gráficas de la *Figura 52* son muy interesantes ya que nos muestran las velocidades de cada estación en función del tiempo. En estas se demuestra como la detección de la colisión es una solución mucho más efectiva que la del método RTS/CTS (no hay nodo oculto), y mucho mejor que considerar toda pérdida de paquete como error del enlace. Esto es debido, a que no se prolonga el tiempo de envío en el canal que produciría a su vez más colisiones. Además se puede incluso aumentar la velocidad de transmisión. Con la trama RTS/CTS se intenta evitar toda colisión posible, pero cuando se produce se sigue interpretando como un error del canal.

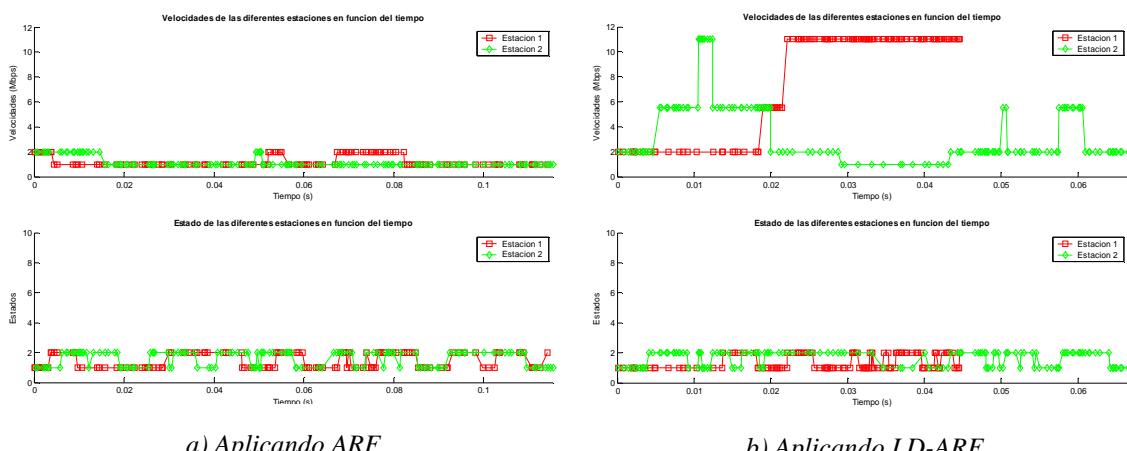


Figura 52. Comparación de las velocidades de cada estación aplicando los dos algoritmos.

## 6. Comparación entre algoritmos

Además, en la *Figura 52* se observaba como las dos estaciones que aplicaron el LD-ARF, acabaron de transmitir 40 milisegundos antes que las estaciones que aplicaban ARF, ya que estas tardaron 100ms.

Finalmente, obtenemos los *throughput* resultantes de las estaciones en las diferentes situaciones como se muestra en la *Tabla 12*. Si observamos la *Figura 53*, vemos como hay un aumento de la tasa del LD-ARF respecto al ARF, aunque no es un incremento muy significativo. Esto se debe al gran número de colisiones, que aunque no reduzcan la tasa, pero aumentan el número de retransmisiones.

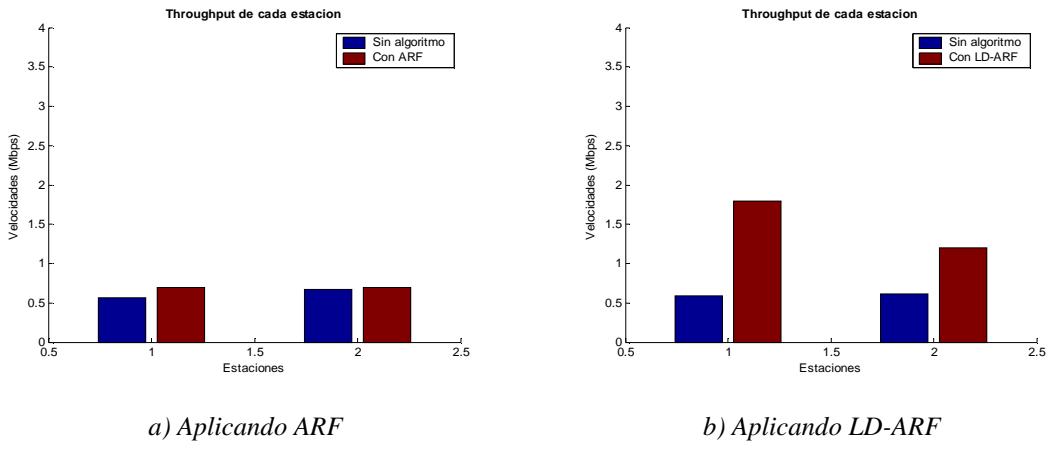


Figura 53. Comparación en los niveles de throughput en los dos casos.

	Aplicando ARQ	Aplicando ED-ARQ
Throughput en la estación 1 (Mbps)	0.70011	1.7902
Throughput en la estación 2 (Mbps)	0.69262	1.2024
Throughput total (Mbps)	1.3852	2.4049

Tabla 12. Resultados del throughput en los diferentes casos.

En conclusión, la diferenciación de la pérdida de paquetes aumenta la eficiencia del algoritmo. Además realiza una adaptación correcta del escenario, de otra manera se estará dando una solución errónea. Cabe destacar que el modelo del canal para el cual se ha simulado los dos algoritmos es muy favorable, y que por lo tanto cualquier trama que no colisione podría llegar perfectamente. Así, el LD-ARF y el mecanismo de diferenciación, son unos buenos avances frente a la mala interpretación de colisiones.

## 6.2 COMPARACIÓN CON MODELO DEL CANAL FÍSICO

Por último, mediante el modelo del canal físico se compararán los dos algoritmos. Esta será la simulación más real entre todas las que se han realizado, ya que se contemplarán las colisiones y el modelo real del canal. Así obtendremos un resultado aplicable en la realidad para todo aquel usuario que cumpla con características similares al canal medido.

## 6. Comparación entre algoritmos

Mediante el modelo del canal físico se podrán predecir si los paquetes llegan satisfactoriamente o se pierden por errores en el enlace. También se prevén un número elevado de colisiones, para poder ver como actúa el algoritmo.

De nuevo aplicaremos los mismo parámetros que en las anteriores simulaciones: umbral de ACK consecutivas igual 10; un umbral de NACK consecutivas igual a 2; y en este caso no aplicaremos el temporizador. Por otra parte, trataremos de enviar 244 paquetes, que es la longitud del proceso determinístico medido.

Una vez ya se han introducido los valores en el algoritmo ARF, observamos los siguientes resultados. En la *Figura 54* y *55* se demuestra como el algoritmo actúa incorrectamente frente a las colisiones, ya que se decrementa la tasa. Aunque como el canal es favorable, aprovecha las circunstancias para aumentar la velocidad.

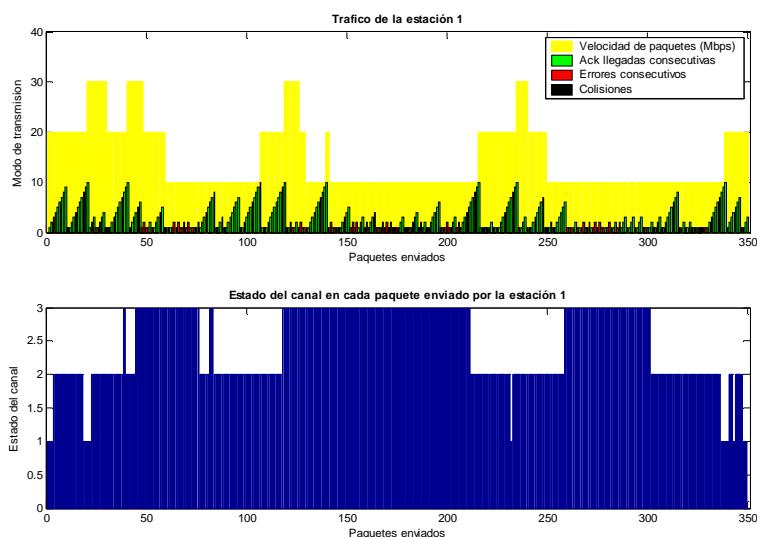


Figura 54. Tráfico de la estación 1 con el modelo de canal físico y aplicando el ARF.

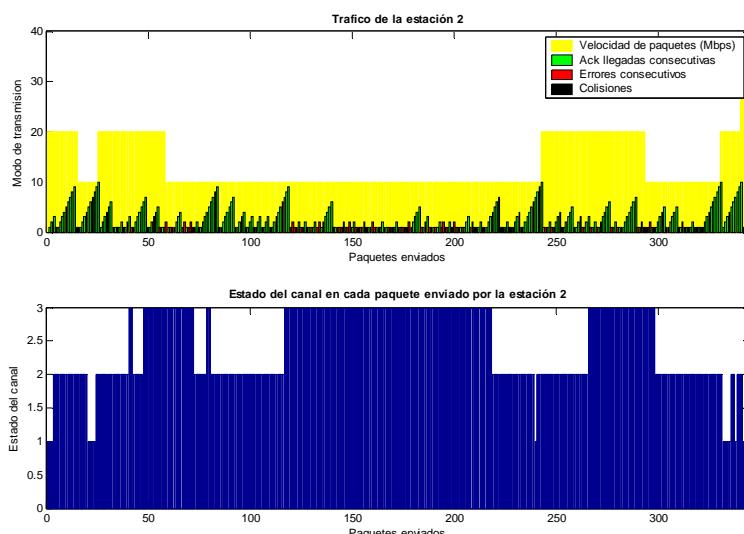


Figura 55. Tráfico de la estación 2 con el modelo de canal físico y aplicando el ARF.

## 6. Comparación entre algoritmos

Si nos fijamos de la *Figura 56* en los tiempos en los que se han producido cambios en la tasa, podemos observar como las dos estaciones están enviando casi en lo mismo periodo. Además no hay ninguna estación que acabe mucho antes que la otra, y por lo tanto se producirán colisiones en todo el tiempo de transmisión.

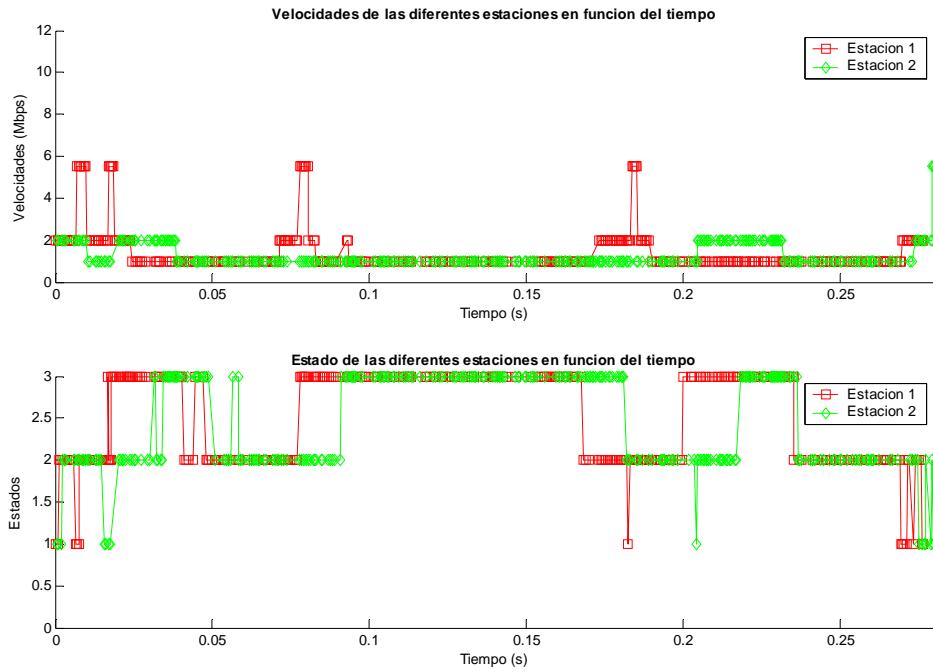


Figura 56. Velocidades de las dos estaciones en función del tiempo.

Aplicando el ARF en esta simulación, se han contabilizado 43 colisiones por cada estación, es un poco más de la quinta parte de la transmisión de todos los datos. Pese a este alto número de reenvíos las velocidades en las dos estaciones se aproximan a 1 Mbps, dando una *throughput* total de 1.4 Mbps.

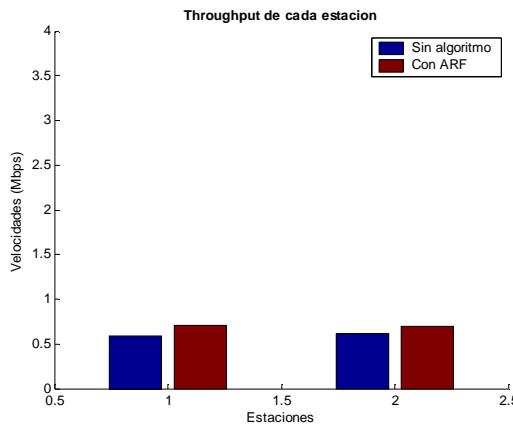
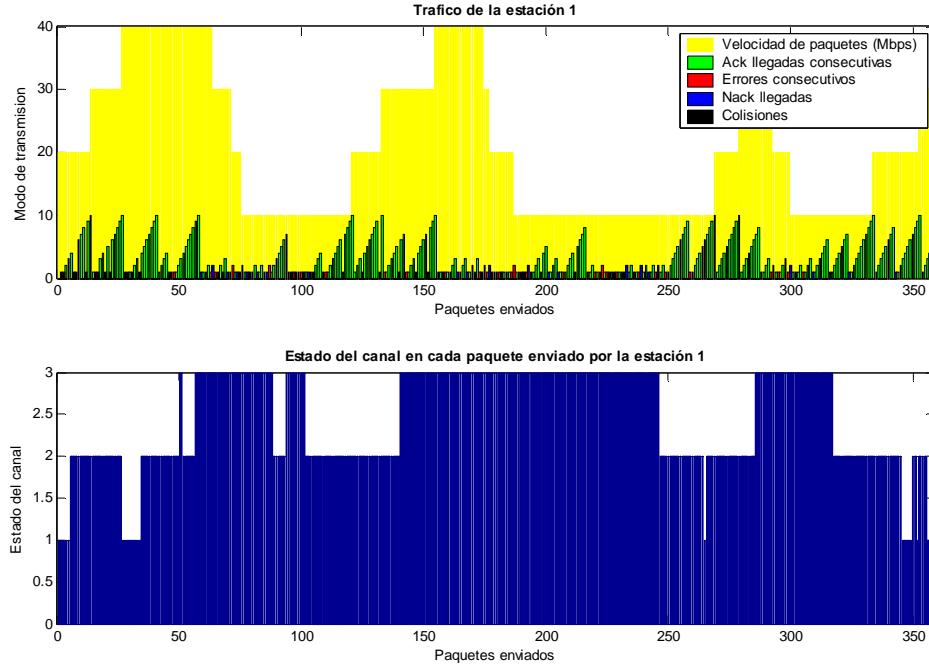


Figura 57. Throughput de las dos estaciones cuando se aplica el ARF.

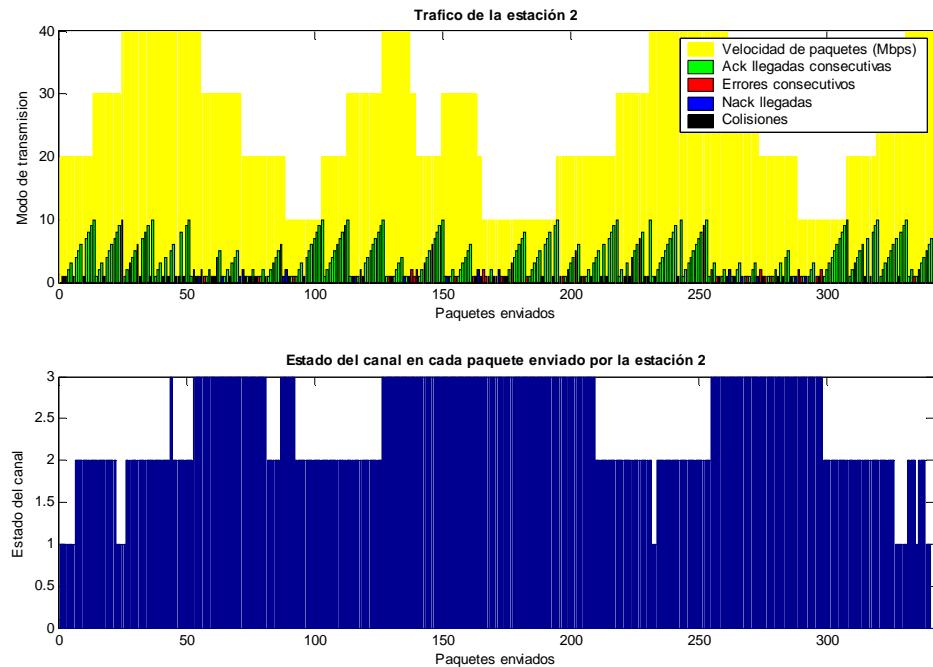
Si ahora pasamos a aplicar el algoritmo LD-ARF, veremos como aumenta el rendimiento de la comunicación.

## 6. Comparación entre algoritmos

Si observamos tanto la *Figura 58* (estación 1) como la *Figura 59* (estación 2), volvemos a comprobar como aunque hayan colisiones entre las tramas consecutivas, estas se detectarán y se aumentará la tasa correctamente. Además a simple vista podemos observar como la velocidad de transmisión mejora cuando nos encontramos en los canales 1 y 2, o empeora cuando nos encontramos en el canal 3.



*Figura 58. Tráfico de la estación 1 con el modelo de canal físico y aplicando el LD-ARF.*



*Figura 59. Tráfico de la estación 2 con el modelo de canal físico y aplicando el LD-ARF.*

## 6. Comparación entre algoritmos

Si nos centramos más detalladamente en la velocidad, vemos como la estación 2 que había tenido más oportunidades de aumentar la tasa, ha podido acabar antes la transmisión.

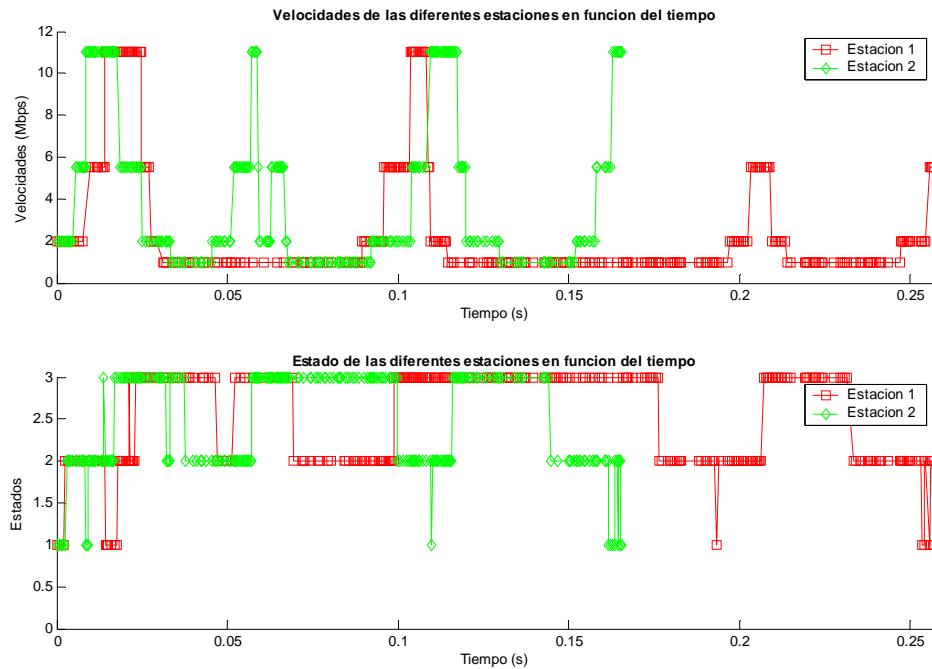


Figura 60. Velocidades de las dos estaciones en función del tiempo.

Finalmente, respecto al throughput resultante obtenemos que con el LD-ARF igualamos o superamos 1 Mbps. Así podemos argumentar que aplicando el algoritmo LD-ARF se garantiza una velocidad de recepción de datos correctos de alrededor de 1 Mbps, como se muestra en la *Figura 61*. Esta conclusión se cumple solo si el usuario se encuentra en unas condiciones similares a las del canal medido.

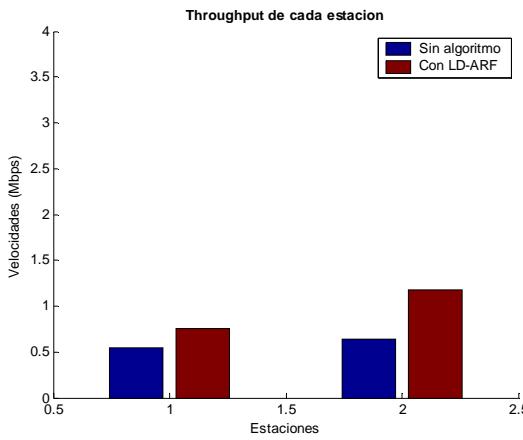


Figura 61. Throughput de las dos estaciones cuando se aplica el LD-ARF.

# Capítulo 7

# Canal híbrido

Un enlace híbrido consta de un canal terrestre y un canal satélite. En este apartado se comentará a nivel teórico la influencia que tiene el canal satélite en nuestro enlace híbrido.

El funcionamiento de esta comunicación consiste en transmitir los paquetes desde la estación transmisora terrestre hasta el punto de acceso satélite. Entonces, este punto de acceso enviará los paquetes al satélite a una frecuencia determinada.

Dependiendo de la banda en la que se envíe la información obtendremos un resultado u otro. Si estamos operando en la banda Ku (de 10.70 a 12.75 GHz.) las condiciones atmosféricas no afectan al enlace, y apenas tiene una probabilidad de error de  $10^{-5}$ . Así se podrá enviar a la máxima velocidad que nos proporcione el *throughput* conseguido por el punto de acceso en el canal terrestre. Como ya hemos estudiado durante casi todo el proyecto, los algoritmos de adaptación de la tasa de transmisión nos garantizan un mínimo *throughput* si las condiciones del canal no son muy malas. Así si aplicamos conjuntamente el algoritmo con esta banda, obtendremos un buen rendimiento en el canal híbrido. Con los resultados obtenidos en el capítulo anterior, para un canal real, se garantiza que le llegarán al punto de acceso del satélite datos correctos a una velocidad de 1 Mbps aproximadamente.

Por lo contrario, si utilizamos la banda Ka (de 20 a 30 GHz), esta si que se verá afectada por las condiciones atmosféricas. Por lo tanto, deberá cambiar de modulación según el estado en el que se encuentre el enlace, para no perder tanta información. Así ya no se podrá enviar a la velocidad que se reciba, sino que dependerá del canal.

Uno de los factores atmosférico que produce más pérdidas en el canal satélite es la lluvia. Según su intensidad aportará mayor o menor atenuación. Como se observa en la *Figura 62*, existen diferentes modelos de atenuación por lluvia en un mismo escenario. Aunque también la atenuación depende de otros factores.

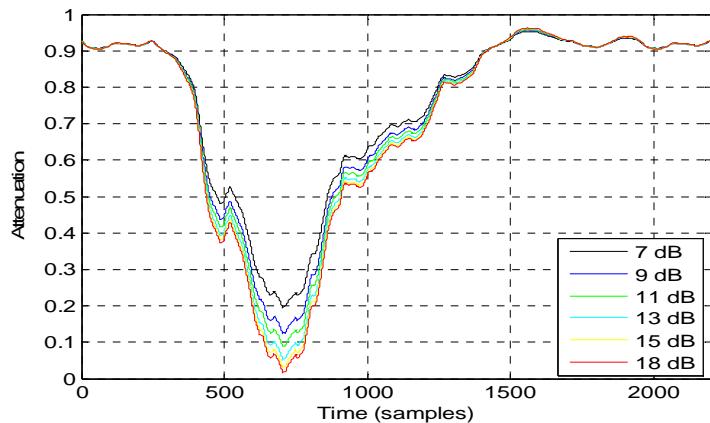


Figura 62. Diferentes modelos de atenuación por lluvia.

Si nos fijamos en la *Figura 62*, podríamos tener el siguiente funcionamiento. Entre 0 y 500 muestras tendremos una modulación que proporcione una tasa elevada de transmisión. Pero a medida que la atenuación vaya aumentando (entre 500 y 1000 muestras) se deberá de cambiar de modulación inmediatamente en cada estado. Esto sucederá durante toda la transmisión y no se podrá garantizar un cierto rendimiento, ya que este dependerá de las condiciones del enlace.

Ahora si nos fijamos en la *Figura 63* podremos ver la relación señal a ruido con la interferencia añadida. Según la intensidad de la lluvia obtendremos un mejor o peor SNIR. Cuando tenemos una de 18 dB la señal decaerá muchísimo, en cambio con 7 dB esta reducción será más suave, aunque también pueden afectar otros parámetros como las pérdidas de propagación.

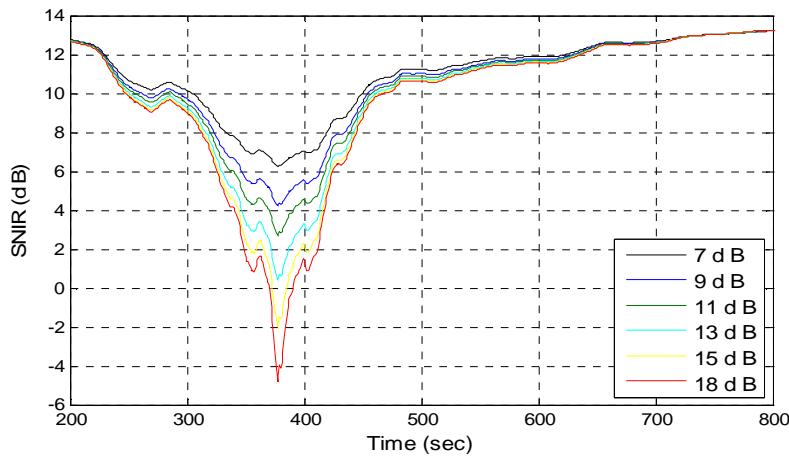


Figura 63. Diferentes modelos de lluvia SNIR para un usuario específico.

En conclusión, si se desea realizar un canal híbrido se recomienda utilizar la banda Ka, ya que de lo contrario la eficiencia producida por el algoritmo LD-ARF o el ARF no servirá de nada, ya que se deberá enviar a una tasa menor.

# Capítulo 8

# Conclusiones

En este proyecto se han visto diferentes puntos fundamentales de las redes inalámbricas. Además se han estudiado los diferentes algoritmos que pueden mejorar la tasa de recepción de datos correctos. Pero principalmente se ha conseguido modelar un canal físico.

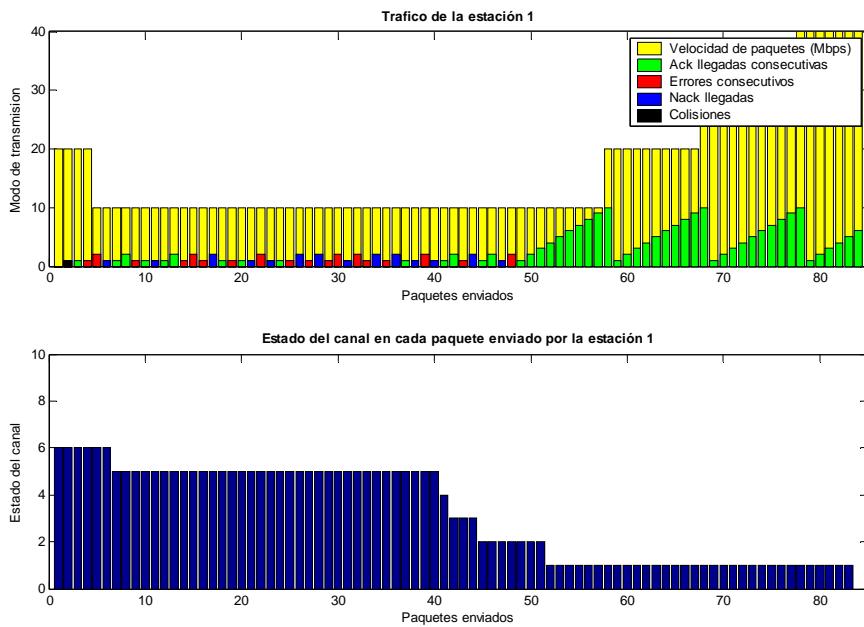
El modelado del canal nos permitirá determinar con cierta precisión el comportamiento de la transmisión de un usuario en las condiciones adecuadas. Como en todas las simulaciones, evitan periodos de estudios de campo para comprobar ese funcionamiento. Así el programa creado nos servirá como una buena herramienta para predecir el comportamiento de una red inalámbrica, con algunas restricciones.

Finalmente, con el estudio de los algoritmos de la adaptación de la tasa, se ha podido garantizar 1 Mbps de *throughput* para el canal modelado. Con este mínimo de velocidad, se puede decidir enlazar con un canal satélite y crear una red híbrida. Así hemos analizado, y concluido que la mejor banda frecuencial en la cual operar en el canal satélite, es la Ku (de 10.70 a 12.75 GHz.) ya que al no influirle las condiciones atmosféricas, no se debe reducir la velocidad conseguida en el canal terrestre.

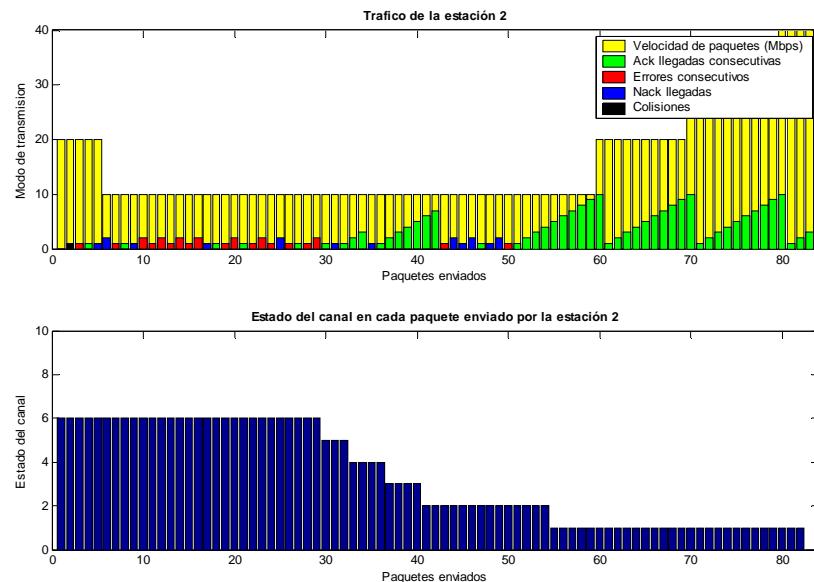
# Apéndice

En este apéndice se mostrará una simulación completa entre dos estaciones transitorias que quieren enviar 50 paquetes a una sola receptora. Así tendremos una idea exacta de lo que proporciona el programa.

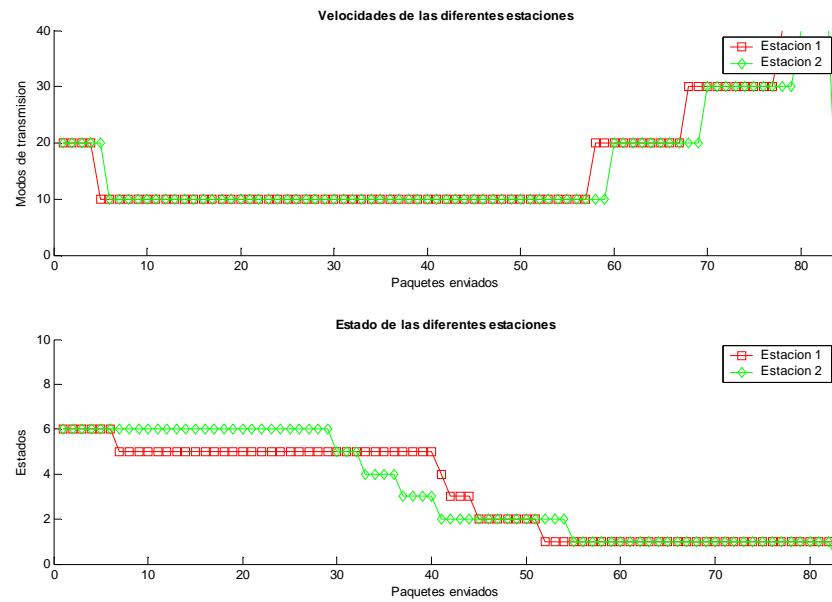
## RESULTADO 1:



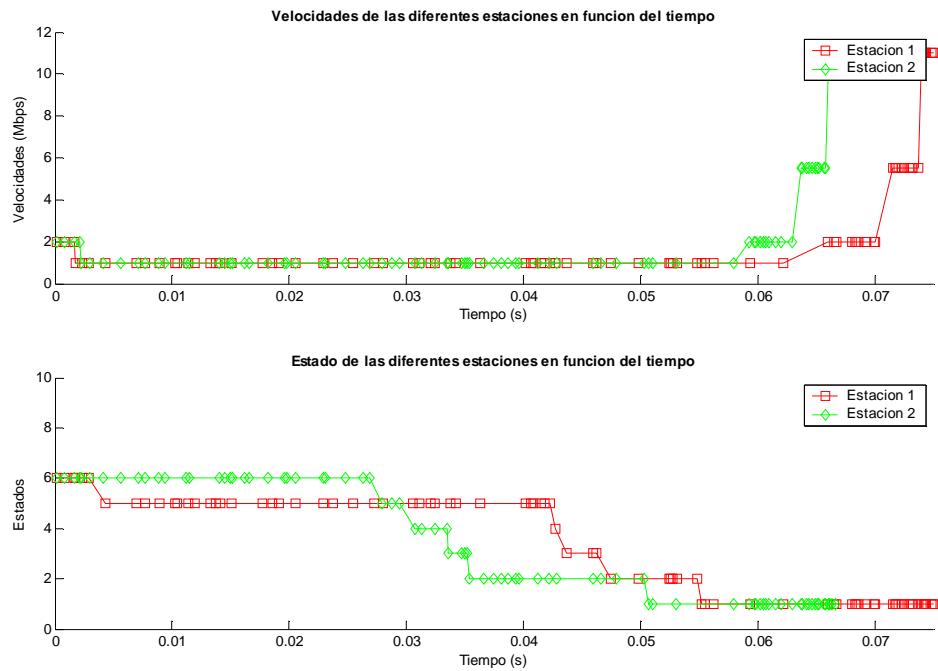
## RESULTADO 2:



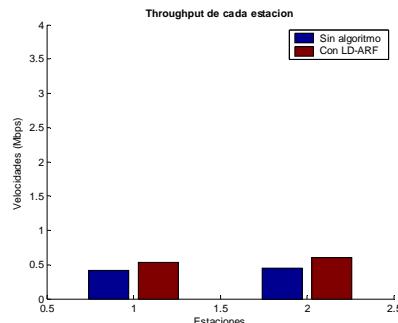
RESULTADO 3:



RESULTADO 4:



RESULTADO 5:



## Apéndice

Representación de los paquetes recibidos en forma de diagrama de los 2 primeros paquetes correctos de las dos estaciones. El resto se encuentran en:

“simulacion\_de\_apendice.txt” del CD adjunto.

```
| | | | tiempo=0.00170038s -- Paquete 2 perdido.  
| - - - - - | -----  
| | A | ----- ACK 1 enviado a estacion 2.  
| | A | Velocidad de transmision de E.B = 1 Mbps  
| | A |  
| | A |  
| | A |  
| | A |  
| | X |  
| | | | tiempo=0.00174488s -- ACK 1 recibido.  
| - - - - - | -----  
| | N | ----- NACK 2 enviado a estacion 1.  
| | N | Velocidad de transmision de E.B = 1 Mbps  
| | N |  
| | X |  
| | | | tiempo=0.00205077s -- NACK 2 perdido.  
| - - - - - | -----  
| | [] | ----- Estacion 2 enviando paquete 2.  
| | [] | Velocidad de transmision = 2.0 Mbps  
| | [] |  
| | [] |  
| | X |  
| | | | tiempo=0.00208269s -- Paquete 2 perdido.  
| - - - - - | -----  
| | N | ----- NACK 2 enviado a estacion 2.  
| | N | Velocidad de transmision de E.B = 1 Mbps  
| | N |  
| | N |  
| | N |  
| | N |  
| | X |  
| | | | tiempo=0.00215597s -- NACK 2 recibido.  
| - - - - - | -----  
| | [] | ----- Estacion 2 enviando paquete 2.  
| | [] | Velocidad de transmision = 2.0 Mbps  
| | [] |  
| | [] |  
| | X |  
| | | | tiempo=0.00218021s -- Paquete 2 perdido.  
| - - - - - | -----  
| | [] | ----- Estacion 1 enviando paquete 2.  
| | [] | Velocidad de transmision = 1.0 Mbps  
| | [] |  
| | [] |  
| | X |  
| | | | tiempo=0.00233715s -- Paquete 2 perdido.  
| - - - - - | -----  
| | N | ----- NACK 2 enviado a estacion 1.  
| | N | Velocidad de transmision de E.B = 1 Mbps  
| | N |  
| | N |  
| | N |  
| | N |  
| | X |  
| | | | tiempo=0.00234787s -- NACK 2 recibido.  
| - - - - - | -----  
| | N | ----- NACK 2 enviado a estacion 2.  
| | N | Velocidad de transmision de E.B = 1 Mbps  
| | N |  
| | N |  
| | N |  
| | N |  
| | X |  
| | | | tiempo=0.00260759s -- NACK 2 recibido.  
| - - - - - | -----  
| | [] | ----- Estacion 1 enviando paquete 2.  
| | [] | Velocidad de transmision = 1.0 Mbps  
| | [] |  
| | [] |  
| | [] |  
| | [] |  
| | X |  
| | | | tiempo=0.00289173s -- Paquete 2 recibido.  
| - - - - - | -----  
| | A | ----- ACK 2 enviado a estacion 1.  
| | A | Velocidad de transmision de E.B = 1 Mbps  
| | A |  
| | A |  
| | A |  
| | X |  
| | | | tiempo=0.00291846s -- ACK 2 recibido.
```

## Apéndice

# Resumen

El objetivo de este proyecto es la predicción de la pérdida de paquete. Para ello necesitaremos el modelado del canal. De esta manera, podremos determinar cuando una transmisión llega con éxito o no.

En primer lugar, se han estudiado los algoritmos de adaptación de la tasa. Estos algoritmos mejoran el rendimiento de la comunicación. Por este motivo, el programa de simulación se basa en algunos de estos algoritmos. En paralelo, se han capturado medidas del canal terrestre para realizar el modelado.

Finalmente, con un programa mucho más completo se ha simulado el comportamiento de una transmisión con el modelado del canal físico, y se han asumido algunas consideraciones, como las colisiones. Por lo tanto, se ha obtenido un resultado más realista, con el cual se ha analizado teóricamente las posibilidades de un enlace entre el canal terrestre y el canal satélite, para crear una red híbrida.

---

L'objectiu d'aquest projecte és la prediccó de la pèrdua de paquet. Per a això necessitarem el modelat del canal. D'aquesta manera, podrem determinar quan una transmissió arriba amb èxit o no.

En primer lloc, s'han estudiat els algoritmes d'adaptació de la taxa. Aquests algoritmes milloren el rendiment de la comunicació. Per aquest motiu, el programa de simulació es basa en alguns d'aquests algoritmes. En paral·lel, s'han capturat mesures del canal terrestre per realitzar un modelat.

Finalment, amb un programa molt més complet s'ha simulat el comportament d'una transmissió amb el modelat del canal físic, i s'ha assumit algunes consideracions, com les col·lisions. Per tant, s'ha obtingut un resultat més realista, amb el qual s'ha analitzat teòricament les possibilitats d'un enllaç entre el canal terrestre i el canal satèl·lit, per crear una xarxa híbrida.

---

The objective of this project is the packet loss prediction. For that we need the channel modelling. Thus, we will be able to determine when the transmission arrives successfully or not.

First, the rate adaptation algorithm has been studied. These algorithms improve the communication performance. For this motive, the simulation program is based in some of these algorithms. In parallel, measurements of the terrestrial channel have been captured to do the modelling.

Finally, with a program far more complete has been simulated the behaviour of a transmission with physic channel modelling, some considerations has been assumed, such as the collisions. Therefore, a result more realistic has been obtained with which the possibilities of the link between the terrestrial channel and the satellite channel has been analysed theoretically, to create a hybrid network.

# Bibliografía

- [1] IEEE Std 802.11b-1999, “Wireless LAN medium access control (MAC) and physical layer (PHY) specifications: higher-speed physical layer extension in the 2.4 GHz band,” Sep. 1999.
- [2] C. Siva Ram Murthy y B.S. Manoj, *Ad Hoc wireless networks: architectures and protocols*, Prentice Hall, 2004, ISBN: 0-13-147023-X.
- [3] Publicación electrónica *El NoticIEEEro*, IEEE sección Panamá, vol.33, no.4, Diciembre 2005, URL <http://www.ewh.ieee.org/r9/panama/noticIEEEro.htm>.
- [4] W. Stallings, *Data and computer communications*, Prentice Hall, 7<sup>a</sup> ed., 2004, Cap.17, ISBN: 0-13-183311-1.
- [5] A. Kerman y L. Monteban, “WaveLAN-II: a high-performance wireless LAN for the unlicensed band,” *Bell Labs Technical Journal*, vol.2, no.3, pag.118-133, Agosto 1997.
- [6] Q. Pang, V.C.M. Leung, y S.C. Liew, “A rate adaptation algorithm for IEEE 802.11 WLANs based on MAC-layer loss differentiation”, *2nd International Conference on Broadband Networks*, vol.1, pag.659- 667, 3-7 Octubre 2005, ISBN: 0-7803-9276-0.
- [7] M. Lacage, M. H. Manshaei, y T. Turletti, “IEEE 802.11 Rate Adaptation: A Practical Approach.”, *ACM MSWiM*, 2004.
- [8] J. Bicket, “Bit-rate Selection in Wireless Networks.”, *MIT Master’s Thesis*, 2005.
- [9] I. Haratcherev, K. Langendoen, R. Lagendijk y H. Sips, “Hybrid Rate Control for IEEE 802.11.”, *ACM MobiWac*, 2004.
- [10] S. H.Y. Wong , H. Yang , S. Lu y V. Bharghavan, “Robust Rate Adaptation for 802.11 Wireless Networks.”, *MobiCom’06*, 23–26 Septiembre 2006.
- [11] G. Holland, N. Vaidya, and P. Bahl, “A rate-adaptive MAC protocol for multi-hop wireless networks.”, *ACM/IEEE MobiCom*, Julio 2001.
- [12] B. Sadeghi, V. Kanodia, A. Sabharwal, y E. Knightly, “Opportunistic media access for multirate ad hoc networks.”, *MobiCom’02*, 2002.
- [13] E. N. Gilbert, “Capacity of a burst-noise channel,” *the Bell System Technical Journal*, pp. 1253–1265, Septiembre 1960.
- [14] J.E. O. Elliott, “Estimates of error rates for codes on burst-noise channels,” *the Bell System Technical Journal*, pp. 1977–1997, Septiembre 1963.

- [15] M. Mushkin y I. Bar-David, “Capacity and coding for the Gilbert-Elliott channels,” *IEEE Trans. Inform. Theory*, vol. 35, no. 6, pp. 1277–1290, Noviembre 1989.
- [16] P. Sun, “Similarity of Discrete Gilbert-Elliott and Polya Channel Models to Continuous Rayleigh Fading Channel Model”, For the Degree of Master of Science, National Chiao Tung University, Junio 2002.
- [17] Jiang Lei y M.A. Vázquez Castro, “Two-state Markov model of a DVB-S2/RCS mobile channel”, *Submitted to IEE Electronic Letters*.

