

Aportación al Estudio  
de Protocolos de Acceso Múltiple:  
El Protocolo ALOHA  
Estabilizado de Ventana

Jesús Angel Alcober Segura

Director de la Tesis Doctoral:  
Profesor Dr. Vicente Casares Giner  
Sección de Telemática  
Departamento de Matemática Aplicada y Telemática  
Universidad Politécnica de Cataluña (UPC)  
Barcelona  
España

1997

# Índice General

<b>1</b>	<b>Introducción</b>	<b>1</b>
<b>2</b>	<b>Los protocolos de acceso aleatorio</b>	<b>11</b>
2.1	Introducción . . . . .	11
2.2	Protocolos ALOHA . . . . .	12
2.2.1	ALOHA puro . . . . .	12
2.2.2	ALOHA ranurado (S-ALOHA) . . . . .	13
2.2.3	Inestabilidad de los protocolos ALOHA . . . . .	14
2.3	Protocolos de resolución de colisiones . . . . .	16
2.3.1	Protocolos de árbol binario . . . . .	17
2.3.2	Protocolos mejorados . . . . .	19
2.3.3	El mecanismo de ventana . . . . .	20
2.3.4	El protocolo de árbol binario truncado . . . . .	21
2.3.5	El protocolo de ventana de pila de dos células . . . . .	22
<b>3</b>	<b>El protocolo ALOHA estabilizado de ventana</b>	<b>25</b>
3.1	Introducción . . . . .	25
3.2	Modelo de sistema . . . . .	25
3.2.1	Modelo de canal . . . . .	28
3.3	Protocolo ALOHA estabilizado de ventana (WSA) . . . . .	31
3.3.1	Protocolo de acceso al canal . . . . .	31
3.3.2	Algoritmo de resolución de colisiones . . . . .	33
<b>4</b>	<b>Análisis del throughput, con errores en el feedback, caso feedback binario</b>	<b>37</b>
4.1	Introducción . . . . .	37
4.2	Análisis del throughput . . . . .	37
4.3	Sensibilidad a los errores en el feedback . . . . .	41
4.4	Apéndice . . . . .	45
4.4.1	Análisis de estabilidad . . . . .	45
4.4.2	Planteamiento del problema, $G_{k,i}$ . . . . .	46

4.4.3	Desarrollo de una cota superior de $E_k(i)$ , $E_k^0(i)$ . . . . .	47
4.4.4	Desarrollo de una cota superior ajustada de $E_k(i)$ , $E_k^v(i)$ . . . . .	59
<b>5</b>	<b>Análisis del retardo, caso feedback binario</b> . . . . .	<b>63</b>
5.1	Introducción . . . . .	63
5.2	El método . . . . .	64
5.3	El análisis del algoritmo . . . . .	68
5.3.1	Sistema de ecuaciones relacionadas a las cantidades por ciclo . . . . .	71
5.3.2	Solución del sistema y límites del retardo medio de paquete . . . . .	73
5.4	Apéndice . . . . .	77
5.4.1	Cálculo de la longitud media $E(i d)$ , $1 \leq d \leq \Delta$ . . . . .	77
5.4.2	Cálculo de la suma de los retardos media $E(\psi d)$ , $1 \leq d \leq \Delta$ . . . . .	82
5.4.3	Cálculo del segundo momento de la longitud $E(i^2 d)$ , $1 \leq d \leq \Delta$ . . . . .	86
<b>6</b>	<b>Análisis del throughput, con errores en el feedback, caso feedback ternario</b> . . . . .	<b>93</b>
6.1	Introducción . . . . .	93
6.2	El modelo del sistema y el algoritmo . . . . .	94
6.3	Análisis del throughput . . . . .	95
6.4	Sensibilidad a los errores en el feedback . . . . .	97
6.5	Apéndice . . . . .	99
6.5.1	Análisis de estabilidad . . . . .	99
6.5.2	Planteamiento del problema, $G_{k,i}$ . . . . .	100
6.5.3	Desarrollo de una cota superior de $E_k(i)$ , $E_k^0(i)$ . . . . .	101
6.5.4	Desarrollo de una cota superior ajustada de $E_k(i)$ , $E_k^v(i)$ . . . . .	104
<b>7</b>	<b>Análisis del retardo, caso feedback ternario</b> . . . . .	<b>107</b>
7.1	Introducción . . . . .	107
7.2	Límites del retardo medio de paquete . . . . .	108
7.3	Apéndice . . . . .	110
7.3.1	Cálculo de la longitud media $E(i   d)$ , $1 \leq d \leq \Delta$ . . . . .	110
7.3.2	Cálculo de la suma de los retardos media $E(\psi   d)$ , $1 \leq d \leq \Delta$ . . . . .	116
7.3.3	Cálculo del segundo momento de la longitud $E(i^2   d)$ , $1 \leq d \leq \Delta$ . . . . .	120

<b>8</b>	<b>Análisis del throughput del algoritmo ideal</b>	<b>127</b>
8.1	Introducción . . . . .	127
8.2	Análisis del throughput . . . . .	129
8.3	Apéndice . . . . .	132
8.3.1	Cálculo de la longitud media $E(l   d)$ , $1 \leq d \leq \Delta$ . . .	132
<b>9</b>	<b>Análisis del retardo del algoritmo ideal</b>	<b>135</b>
9.1	Introducción . . . . .	135
9.2	Límites del retardo medio de paquete . . . . .	136
9.3	Apéndice . . . . .	138
9.3.1	Cálculo de la suma de retardos media $E(\psi   d)$ , $1 \leq d \leq \Delta$ . . . . .	138
9.3.2	Cálculo del segundo momento de la longitud $E(l^2   d)$ , $1 \leq d \leq \Delta$ . . . . .	140
<b>10</b>	<b>Otros análisis del algoritmo</b>	<b>143</b>
10.1	Introducción . . . . .	143
10.2	Probabilidades de acceso iniciales . . . . .	143
10.3	Función de probabilidades de acceso . . . . .	146
10.4	Protocolos con sintonización limitada . . . . .	149
<b>11</b>	<b>Aplicación: Integración de voz y datos en redes de acceso inalámbricas</b>	<b>153</b>
11.1	Introducción . . . . .	153
11.2	Escenario . . . . .	156
11.3	Algoritmo de acceso aleatorio de reserva: WSA . . . . .	157
11.4	Reserva de slots para talkspurts . . . . .	159
11.5	Acceso de paquetes de datos . . . . .	160
11.6	Resultados de simulación . . . . .	161
	<b>Conclusiones</b>	<b>169</b>
	<b>Bibliografía</b>	<b>173</b>



# Glosario de Abreviaturas

- BAP:** Blocked-Access Protocol, Protocolo de Acceso Bloqueado.
- BSC:** Base Station Controller, Estación Base Controladora.
- CAP:** Channel Access Protocol, Protocolo de Acceso al Canal.
- CDMA:** Code Division Multiple Access, Acceso Múltiple por División en Código.
- CRA:** Collision Resolution Algorithm, Algoritmo de Resolución de Colisiones.
- CRAI:** Collision Resolution with Addition Information, Resolución de Colisiones con Información Añadida.
- CRAT:** Collision Resolution Algorithm (Tsybakov), Algoritmo de Resolución de Colisiones (Tsybakov).
- CRI:** Collision Resolution Interval, Intervalo de Resolución de Colisiones.
- CRP:** Collision Resolution Point, Punto de Resolución de Colisiones.
- CSMA:** Carrier Sense Multiple Access, Acceso Múltiple por Detección de Portadora.
- DBT:** Dynamic Binary Tree Protocol, Protocolo de Árbol Binario Dinámico.
- DFT:** Deferred First Transmission, Primera Transmisión Deferida.
- DLC:** Data Link Control, Control de Enlace de Datos.
- FAP:** Free-Access Protocol, Protocolo de Acceso Libre.
- FCFS:** First Come First Served, Primero en Llegar Primero en Servirse.

- FDD:** Frequency Division Duplex, Duplexación por División en Frecuencia.
- FDMA:** Frequency Division Multiple Access, Acceso Múltiple por División en Frecuencia.
- FIFO:** First In First Out, Primero en Entrar Primero en Salir.
- FTR:** First Transmission Rule, Regla de la Primera Transmisión.
- IFT:** Immediate First Transmission, Primera Transmisión Inmediata.
- ISO:** International Standards Organization, Organización de Estándares Internacionales.
- MAC:** Medium Access Control, Control de Acceso al Medio.
- OSI:** Open Systems Interconnection, Sistemas de Interconexión Abierta.
- PRMA:** Packet Reservation Multiple Access, Acceso Múltiple por Reserva de Paquete.
- RAS:** Random-Access System, Sistema de Acceso Aleatorio.
- RAA:** Random Access Algorithm, Algoritmo de Acceso Aleatorio.
- RRA:** Reservation Random Access Algorithm, Algoritmo de Acceso Aleatorio de Reserva.
- SNR:** Signal to Noise Rate, Relación Señal a Ruido.
- TCW:** Two Cell Stack Window Algorithm, Algoritmo de Pila de Dos Células de Ventana.
- TDMA:** Time Division Multiple Access, Acceso Múltiple por División en el Tiempo.
- WSA:** Window Stabilized ALOHA Algorithm, Algoritmo ALOHA Estabilizado de Ventana.

# Glosario de Anglicismos

El autor ha utilizado algunos términos ingleses tal y como fueron concebidos originariamente en lugar de substituirlos por una posible traducción al castellano. Esta decisión ha sido tomada por dos razones; en primer lugar, la escasa bibliografía especializada en castellano nos hace dudar de la bondad de la posible traducción y, en segundo lugar, el hecho de asociar un concepto y una palabra, aunque sea en otro idioma, dificulta la asociación de ese mismo concepto con otra bien distinta.

Por ello, en este Glosario de Anglicismos se ha intentado mostrar algunas posibilidades de traducción, reservándonos su adecuación.

**Feedback:** Realimentación, retroacción.

**Lag:** Retraso.

**Slot:** Ranura temporal.

**Throughput:** Tasa, caudal.





# Índice de Figuras

2.1	Gráfica de la ecuación del throughput ( $\tau$ ) en función del tráfico ( $\lambda_i$ ) del ALOHA puro y ranurado. . . . .	15
3.1	Ilustración de las relaciones entre las distintas variables relacionadas con la operación del algoritmo WSA (protocolo ALOHA estabilizado de ventana). . . . .	32
3.2	Ejemplo del funcionamiento del protocolo ALOHA estabilizado de ventana. . . . .	34
4.1	Representación gráfica de la obtención de $E_k^u(i)$ y $E_k^0(i)$ . . . . .	39
4.2	Representación gráfica de las regiones $R_i$ en que se ha dividido el espacio. . . . .	51
4.3	Representación gráfica de la función $0.9i^3 + 10.45i^2 - 30.7i + 21.15$ . . . . .	52
4.4	Representación gráfica de la función $g_1 = 1.312 - 2.625k - 0.125i$ . . . . .	53
4.5	Representación gráfica de la función $g_2 = -0.437k - 2.625k^2 + 1.625ik - i^2 + i$ . . . . .	55
4.6	Representación gráfica de las regiones en que se ha dividido el espacio para la demostración. . . . .	55
4.7	Representación gráfica de la función $0.0715i^2 - 5.436i + 3.057$ . . . . .	58
4.8	Representación gráfica de la función $0.024i^2 - 1.46i + 1.25$ . . . . .	60
5.1	Ejemplo del funcionamiento del protocolo ALOHA estabilizado de ventana. . . . .	70
11.1	Estrategia general de límites móviles. . . . .	155
11.2	Proceso de asignación de ranuras. (a) Distribución dinámica. (b) Asignación fija. . . . .	160
11.3	Probabilidad de pérdida de paquetes de voz y probabilidad de que un talkspurt sea enteramente enviado. . . . .	162

- 11.4 Esquema de distribución dinámica. Relación entre retardo medio de los paquetes de datos y tasa de datos  $\lambda_d$ , para  $N_v = 10, 15, 20, 25, 30, 35$  y  $40$ . . . . . 166
- 11.5 Esquema de asignación fija. Relación entre retardo medio de los paquetes de datos y tasa de datos, en términos del número de terminales de voz cuando  $N_d = 32$ . . . . . 168

# Índice de Tablas

4.1	Throughputs y tamaños de ventana óptimos. Los algoritmos representados son el WSA (ALOHA estabilizado de ventana), el algoritmo TCW ( <i>two cell window</i> ) de M. Paterakis y P. Papantoni-Kazakos y el algoritmo DBT ( <i>dynamic binary tree</i> ) de J. Capetanakis. . . . .	41
4.2	Throughputs en función de $\epsilon$ y $\delta$ . Los tamaños de ventana son optimizados para cada par $(\epsilon, \delta)$ . Los algoritmos representados son el WSA (ALOHA estabilizado de ventana), el algoritmo TCW ( <i>two cell window</i> ) de M. Paterakis y P. Papantoni-Kazakos y el algoritmo DBT ( <i>dynamic binary tree</i> ) de J. Capetanakis. . . . .	43
4.3	Throughputs en función de $\epsilon$ y $\delta$ . Los tamaños de ventana fijos para cada par $(\epsilon, \delta)$ . Los algoritmos representados son el WSA (ALOHA estabilizado de ventana), el algoritmo TCW ( <i>two cell window</i> ) de M. Paterakis y P. Papantoni-Kazakos y el algoritmo DBT ( <i>dynamic binary tree</i> ) de J. Capetanakis. . . . .	45
4.4	Coefficientes de $G_{k,i}^0$ , en función de $\epsilon$ y $\delta$ . . . . .	61
5.1	Límites superior e inferior de los retardos esperados. Los algoritmos representados son el WSA (ALOHA estabilizado de ventana), el algoritmo TCW ( <i>two cell window</i> ) de M. Paterakis y P. Papantoni-Kazakos y el algoritmo DBT ( <i>dynamic binary tree</i> ) de J. Capetanakis. . . . .	65
6.1	Throughputs y tamaños de ventana óptimos. Los algoritmos representados son el WSA (ALOHA estabilizado de ventana) en su versión ternaria y binaria. . . . .	97
6.2	Throughputs en función de $\epsilon$ y $\delta$ . Los tamaños de ventana son optimizados para cada par $(\epsilon, \delta)$ . Los algoritmos representados son el WSA (ALOHA estabilizado de ventana) en su versión ternaria y binaria. . . . .	98

6.3	Throughputs en función de $\epsilon$ y $\delta$ . Los tamaños de ventana fijos para cada par $(\epsilon, \delta)$ . Los algoritmos representados son el WSA (ALOHA estabilizado de ventana) en su versión ternaria y binaria. . . . .	99
7.1	Límites superior e inferior de los retardos de paquete esperados. Los algoritmos representados son el WSA (ALOHA estabilizado de ventana) en su versión ternaria y binaria. . . . .	110
8.1	Throughputs y tamaños de ventana óptimos. Los algoritmos representados son el WSA (ALOHA estabilizado de ventana) en su versión ideal, ternaria y binaria. . . . .	130
8.2	Throughputs óptimos. Los algoritmos representados son el protocolo ALOHA estabilizado de ventana ideal (WSA ideal), el protocolo de L. Georgiadis and P. Papantoni-Kazakos ( <i>collision resolution with addition information</i> , CRAI) y el protocolo de B. Tsybakov (CRAT). . . . .	132
9.1	Límites superior e inferior de los retardos esperados. Los algoritmos representados son el WSA (ALOHA estabilizado de ventana) en su versión ideal, ternaria y binaria. . . . .	136
10.1	Throughputs y tamaños de ventana para distintos $r_{t_0}$ y caso libre de error en el WSA (ALOHA estabilizado de ventana) en su versión binaria. . . . .	144
10.2	Throughputs en función de $\epsilon$ y $\delta$ para distintos valores de $r_{t_0}$ en el WSA (ALOHA estabilizado de ventana) en su versión binaria. Los tamaños de ventana son optimizados para cada par $(\epsilon, \delta)$ . . . . .	145
10.3	Ventanas en función de $\epsilon$ y $\delta$ para distintos valores de $r_{t_0}$ en el WSA (ALOHA estabilizado de ventana) en su versión binaria. . . . .	146
10.4	Throughputs en función de $\epsilon$ y $\delta$ para distintos valores de $r_{t_0}$ en el WSA (ALOHA estabilizado de ventana) en su versión binaria. Los tamaños de ventana fijos para cada par $(\epsilon, \delta)$ . . . . .	147
10.5	Throughputs y tamaños de ventana para dos funciones de probabilidad de acceso distintas en el WSA (ALOHA estabilizado de ventana) en su versión binaria. . . . .	149

# Capítulo 1

## Introducción

Uno de los campos de las telecomunicaciones que ha merecido mayor atención en los últimos 20 años ha sido la resolución de colisiones en un canal multiacceso, es decir, resolver el acceso de un numeroso grupo de usuarios a un canal común para comunicarse.

La solución clásica escogida ha sido implementar diferentes formas de multiplexado de la información; por ejemplo, multiplexado por división en el tiempo (TDM) o por división en frecuencia (FDM), con el fin de repartir equitativamente los recursos entre los usuarios. Sin embargo, esta solución se basa, como es sabido, en una división fija de los recursos, que es ineficiente cuando los requerimientos de los usuarios varían con el tiempo o, sencillamente, el número de usuarios es desconocido. Clásicamente se ha incorporado diversas formas de asignación por demanda de los recursos, de forma que la partición de estos se pueda adaptar a los requerimientos.

Una segunda alternativa más eficaz a problemas de acceso múltiple con las características citadas anteriormente consiste en emplear algún tipo de acceso aleatorio, es decir, permitir a todo usuario la posibilidad de ocupar los recursos de comunicaciones cuando tenga información a transmitir. Los problemas surgen, evidentemente, cuando dos o más usuarios transmiten a la vez, porque se producirá una interferencia mutua, llamada colisión, cuyo resultado será la no transmisión correcta de los mensajes implicados, por una parte, y la necesidad de articular los mecanismos necesarios para que no se produzcan ulteriores colisiones, o, al menos, minimizarlas, por otra.

La idea subyacente en este tipo de sistemas que lleva a una utilización más eficiente del medio radica en la esporádica utilización del medio por parte de los usuarios, y el completo desconocimiento que ellos tienen entre sí.

Todos los usuarios del sistema de acceso aleatorio saben de las colisiones que se producen en él, a través de un mensaje de realimentación llamado feedback.

Así pues, el objetivo de la resolución de colisiones se basa en el establecimiento de unas reglas de funcionamiento eficientes que permitan el acceso a un medio común de un número de usuarios independientes.

Desde los años setenta han ido apareciendo numerosos sistemas de acceso aleatorio más o menos eficientes. La eficacia de un algoritmo se basa en el cálculo de diversos parámetros, como el throughput (capacidad de información que se puede drenar) o el retardo medio de transmisión de un paquete. Una vez dados cuantitativamente tales parámetros, la bondad de éste será totalmente determinado y, por lo tanto, habrán unos criterios claros de comparación.

Sin duda alguna, la tarea más dificultosa no es la concepción de un algoritmo con altas prestaciones sino el cálculo analítico de sus características.

Someramente, el objetivo de esta tesis radica en la presentación de una nueva familia de algoritmos, y en el cálculo analítico de los parámetros que ésta presenta, de forma que puede ser comparado con los existentes.

La metodología utilizada proviene en parte del análisis realizado para otros algoritmos a finales de los años 80, pero que debido a la propia idiosincrasia de cada caso, las herramientas matemáticas se adecúan individualmente.

El vacío que esta tesis pretendía llenar era el análisis de un algoritmo que siguiera las pautas del celebrado protocolo ALOHA para su funcionamiento con acceso bloqueado. La forma más adecuada para ver cómo podría situarse un protocolo de tales características es a través de la clasificación de los protocolos. Y aquí es donde se aprecia ya la no existencia de un protocolo con la filosofía del ALOHA y con las prestaciones comparables a los algoritmos con acceso bloqueado.

## Planteamiento del problema

En general, podría decirse que las tres componentes que caracterizan las redes de comunicación son los nodos, los canales y los protocolos. Por una parte, los nodos son las entidades hardware que alberga las funciones de comunicación de datos; por otra, los protocolos son los conjuntos de reglas y acuerdos entre las partes comunicantes que dictan el comportamiento de los nodos, y, finalmente, el canal es el medio físico sobre el cual las señales, representando datos, viajan de un nodo a otro.

Las redes tradicionales hacen uso de los canales punto a punto, es decir, canales que están dedicados a un par de usuarios. Estos canales, más

que ser económicos, son ventajosos debido a que se caracterizan por la no interferencia; es decir, la transmisión entre un par de nodos no tiene efecto sobre la transmisión entre otro par de nodos aunque estos tengan un nodo en común. Los canales punto a punto, sin embargo, requieren una topología fija, determinada en gran parte en la fase de diseño de la red. Por tanto, los cambios topológicos subsiguientes serán bastante costosos de implementar.

En todas aquellas situaciones en las que los canales punto a punto no sean económicos, o cuando se necesite una topología dinámica, se puede utilizar entonces una canal broadcast. Informalmente hablando, un canal broadcast es aquel donde más de un receptor puede recibir potencialmente todos los mensajes transmitidos. Los canales broadcast aparecen de forma natural en radio, satélites, y en algunas redes de área local. La propiedad broadcast tiene sus ventajas y desventajas. Si, realmente, un mensaje está dirigido a un gran número de destinos entonces el canal broadcast es claramente superior. Sin embargo, en un caso típico éste se destina a un único objetivo, o como mucho a unos pocos; por lo que hay una parte del proceso malgastado, aquel que corresponde a los nodos para lo cuales el mensaje no estaba destinado. Además, las transmisiones sobre un canal broadcast interfieren, en el sentido que una transmisión coincidente en el tiempo con otra puede causar que ninguna de las dos sean recibidas. En otras palabras, el éxito de una transmisión no es ni mucho menos independiente de las demás transmisiones. Las ventajas más importantes de un sistema broadcast radican en la fácil implementación y en la posibilidad de habilitar usuarios móviles.

Con el fin de conseguir que la transmisión sea exitosa, la interferencia debe ser evitada, o, como mínimo, controlada. El canal se convierte entonces en el recurso compartido cuya distribución es crítica para una operación adecuada de la red. Estos esquemas de acceso a tales canales se les conoce en la literatura especializada como Protocolos de Acceso Múltiple. Estos protocolos no son más que esquemas de distribución del canal que mantienen unas prestaciones deseables. En términos de modelos de capas, tales como el modelo de referencia OSI de la ISO, estos protocolos residen en su mayor parte en una capa especial llamada Control de Acceso al Medio (*Medium Access Control*, MAC). La capa MAC está entre la capa de enlace (*Data Link Control*, DLC) y la capa física (*Physical*).

La necesidad de los protocolos de acceso múltiple radica no sólo en los sistemas de comunicación, sino también en muchos otros tales como sistemas de ordenadores, un sistema con facilidad de almacenamiento o un servidor de cualquier tipo, donde un recurso es compartido, y por tanto accedido, por un número de usuarios independientes. Brevemente, el entorno en el que estamos interesados se caracteriza por:



- El envío de un mensaje a múltiples usuarios a través de una única transmisión.
- Los usuarios oyen la existencia de otros a través del feedback.
- Nos centramos en el nivel MAC sin preocuparnos de otras funciones de la red tales como control de flujo.
- La comunicación entre dos pares de nodos pueden influir en la comunicación de otros pares; es decir, hay interferencia.

## Clasificación de los protocolos

Los protocolos de acceso múltiple sugeridos y analizados hasta la fecha son muy numerosos para ser todos mencionados aquí. Por lo tanto, en esta introducción clasificaremos estos protocolos para que de esta forma todos puedan ser situados conceptualmente. En la descripción de la clasificación consideramos el canal como el punto focal y nos referimos a los nodos transmitiendo a través del canal como sus usuarios.

Hay muchas maneras de clasificar los protocolos de acceso múltiple. Primero de todo, estamos interesados en protocolos de acceso múltiple no centralizados; esto es, protocolos en los cuales todos los nodos se comportan de acuerdo con el mismo conjunto de reglas. En particular, no hay ningún nodo coordinando las actividades de los demás (cuyo protocolo, por necesidad, difiere del resto).

En el nivel más alto de la clasificación distinguimos entre los protocolos libres de conflicto (*conflict-free*) y los protocolos de contienda (*contention*). Los protocolos libres de conflicto son aquellos que aseguran que la transmisión será exitosa una vez se haya hecho, es decir, no será interferida por otra transmisión. La transmisión libre de conflictos se puede conseguir distribuyendo el canal entre los usuarios tanto estática como dinámicamente. Los recursos del canal pueden ser vistos, para este propósito, desde el punto de vista del tiempo, de la frecuencia, o de ambos. Así, el canal se puede dividir proporcionando el rango entero de frecuencias (ancho de banda) a un usuario por una fracción de tiempo, como se hace en el Acceso Múltiple por Multiplexación en el Tiempo (*Time Division Multiple Access*, TDMA), o asignando una fracción del rango de frecuencias a cada usuario durante todo el tiempo, como se hace en el Acceso Múltiple por Multiplexación en Frecuencia (*Frequency Division Multiple Access*, FDMA), o seleccionando para cada usuario una porción del ancho de banda por una fracción de tiempo como se hace en los sistemas basados en el spread-spectrum, tales como Ac-

ceso Múltiple por Multiplexación de Código (*Code Division Multiple Access*, CDMA).

Para contar con la distribución estática, la dinámica distribuye el canal basado en demanda, de forma que el usuario que va a quedar vacío usa unos pocos recursos del canal, dejando la mayoría de su porción a los otros usuarios más activos. Tal distribución se puede llevar a cabo por varios esquemas de reserva en los cuales los usuarios anuncian primero su intención de transmitir y todos aquellos que lo anuncian son transmitidos antes que los nuevos usuarios tengan oportunidad de hacerlo. Otro esquema usual se refiere al traspaso de testigo (*token*), en los cuales un testigo se va pasando entre los usuarios permitiendo sólo transmitir al portador del testigo y por lo tanto garantizando la no interferencia.

Los esquemas de contienda se diferencian de los esquemas libres de conflictos en principio porque un usuario transmitiendo no se le asegura que lo haga exitosamente. El protocolo debe prescribir una manera de resolver los conflictos una vez ocurran para que todos los mensajes sean finalmente transmitidos exitosamente. El proceso de resolución consume recursos y es una de las mayores diferencias entre los diferentes protocolos de contienda. Si la probabilidad de interferencia es pequeña, tal podría ser el caso de usuarios a ráfagas, aún cabiendo la posibilidad de tener que resolver la interferencia, ésta compensa los recursos que se deberían utilizar para asegurar la no posibilidad de conflictos. Además, en la mayoría de los protocolos libres de conflictos, los usuarios vacíos consumen una porción de los recursos del canal; y ésta crece más y más cuando el número de usuarios potenciales del sistema es muy alto, haciendo imposible la implantación de un esquema libre de conflictos. En esquemas de contienda los usuarios vacíos no transmiten y entonces no consumen ninguna parte de los recursos del canal.

Cuando se utilizan los protocolos de acceso múltiple basados en contienda, la necesidad de resolver los conflictos, siempre que ellos ocurran, es la clave del problema. Como en el caso libre de conflicto existen resoluciones estáticas y dinámicas. La resolución estática significa que el comportamiento no se ve influenciado por la dinámica del algoritmo. Una resolución estática se puede basar, por ejemplo, en el identificador del usuario u otra prioridad prefijada, por lo que siempre que aparece un conflicto, el primer usuario a transmitir finalmente un mensaje será aquel que tenga el identificador más pequeño (este procedimiento se realiza en algunos protocolos de resolución por árbol). Una resolución estática puede ser probabilística, significando que la temporización de la transmisión para los usuarios que interfieren se escoje de una distribución que es independiente del número de usuarios interferentes, tal es el caso de los protocolos tipo ALOHA y de varias versiones de protocolos de Acceso Múltiple por Sintonización de Portadora (*Carrier Sense*

*Multiple Access, CSMA*).

La resolución dinámica es de igual forma posible en los protocolos basados en contienda, quienes siguen los cambios del sistema y toman ventaja de ellos. Por ejemplo, la resolución se puede basar en el tiempo de llegada, dando mayor (o menor) prioridad al mensaje más antiguo del sistema. Alternativamente, la resolución puede ser probabilística de forma que la estadística cambie dinámicamente de acuerdo con la extensión de la interferencia. La estimación de la multiplicidad de los paquetes interferentes, y el esquema exponencial back-off del estándar Ethernet se sitúan dentro de esta categoría.

## Objetivo de la tesis

El objetivo de esta tesis es presentar una nueva familia de protocolos basados en contienda, los algoritmos ALOHA estabilizados de ventana (*window stabilized ALOHA, WSA*). Según la clasificación dada anteriormente, estos podían enmarcarse en aquellos cuya resolución es dinámica y probabilística, es decir, los usuarios resuelven sus colisiones accediendo con una probabilidad que se adapta según la información recibida a través del feedback. El hecho que la resolución no se base en el tiempo de llegada, tiene la ventaja que estos algoritmos siguen siendo aplicables aún cuando el modelo de Poisson para el proceso de llegadas no sea válido.

Desde otro punto de vista, se podría considerar este algoritmo presentado como una variante del protocolo S-ALOHA estabilizado, propuesto por numerosos autores, que conseguían de esta forma que el algoritmo llegara de forma estable a conseguir un throughput de  $e^{-1} = 0.36\dots$ . Veremos posteriormente que esta cifra es superada.

Visto desde un punto de vista global, el propósito de la presente tesis ha sido aunar la idea simple de los protocolos ALOHA, con la eficiencia de los protocolos de resolución de colisiones con acceso bloqueado, resultando una nueva familia de protocolos que tienen unas prestaciones dignas de tener en cuenta.

El cálculo analítico de sus parámetros: throughput y retardo, nos permitirá cuantificar la bondad de estos algoritmos y compararlos con los ya existentes.

La motivación que ha conducido a plantear la existencia de esta nueva familia de algoritmos fue producto de una de las líneas de investigación llevadas a cabo por la sección de Telemática del Departamento de Matemática Aplicada y Telemática, enmarcados en sendos proyectos contiguos cronológicamente. En el primero, el proyecto PA86-D261, entre 1987 y 1990 y cuyo investigador principal fue el Profesor Dr. Emilio Sanvicente Gargallo, el objeto de investigación estaba centrado en sistemas distribuidos para conmutación dig-

ital integrada de voz y datos. En el segundo, los proyectos TIC90-0718 y TIC92-1180, entre 1990 y 1993 y cuyo investigador principal fue el Profesor Dr. Vicente Casares Giner, el objeto de investigación fue el entorno de emulación para sistemas de radiocomunicaciones digitales móviles y portátiles con acceso celular. Resultado de este esfuerzo fueron trabajos como el de V. Casares y J. Paradells [1], donde se estudió un modelo de colas con un servidor y dos corrientes de tráfico con prioridad y características distintas, con el fin de integrar voz y datos en Redes de Área Local con el mecanismo de paso de testigo; o también el de V. Casares y J. Paradells [2] en donde se estudiaron diferentes protocolos de acceso aleatorio en un esquema de multiplexación TDMA con dos estrategias de señalización: *in-slot* y *out-slot*, o más recientemente el de V. Casares y X. Viladegut [3], donde se estudia y simula un protocolo de acceso aleatorio para un servicio de voz, basado en el PRMA. Todo este trabajo realizado condujo a la concepción del hecho que era bastante lógico suponer que debían haber protocolos próximos a la filosofía de los protocolos ALOHA, pertenecientes a una resolución estática pero en un ámbito dinámico, en el camino de los intentos de estabilización del ALOHA.

El salto conceptual que hemos dado aquí, y nos a llevado más lejos que una más que estudiada estabilización del ALOHA es considerar un protocolo de acceso al canal (CAP) de acceso bloqueado. Por tanto, una gran parte de la metodología utilizada en el cálculo analítico proviene del estudio realizado para los distintos algoritmos pertenecientes a la resolución por tiempo de llegada, que debido a su buen comportamiento centró los trabajos de investigación de la última década.

En conclusión, esta nueva familia de algoritmos se puede considerar una evolución de los algoritmos ALOHA estabilizados hasta tal punto que pueden ser comparados con los protocolos cuya resolución es dinámica y basada en el tiempo de llegada, que son los considerados como más eficientes.

A lo largo de la tesis se van a exponer los distintos estudios analíticos y resultados correspondientes a esta nueva familia de algoritmos. Para ello, básicamente lo que se ha llevado a cabo ha sido el análisis completo de tres variantes del algoritmo, siguiendo una lógica preestablecida.

En el Capítulo 2 se da a conocer una visión histórica de los protocolos de resolución de colisiones, que nos lleva a situarnos en la problemática actual.

En el Capítulo 3 se presenta el protocolo ALOHA estabilizado de ventana propiamente dicho, determinando todas y cada una de las partes que se compone, para tener una concreta descripción del mismo. Ésta será la versión binaria del algoritmo, por otra parte considerado como el original; y ha sido así desde todos los puntos de vista.

En el Capítulo 4 se procede al análisis del throughput del protocolo ALO-

HA estabilizado de ventana, en su versión binaria y con errores en el feedback. En él también se incluye la explicación del método seguido.

En el Capítulo 5 se desarrolla el cálculo del retardo medio para la transmisión de un paquete del protocolo ALOHA estabilizado de ventana en su versión binaria. La metodología también está totalmente especificada.

En el Capítulo 6 se ha dispuesto el análisis del throughput del protocolo ALOHA estabilizado de ventana en su versión ternaria. A tal fin se especifica primero esta nueva variante del algoritmo. Para la metodología, el lector es remitido al Capítulo 4.

En el Capítulo 7 hay un análisis del retardo medio de paquete para el protocolo ALOHA estabilizado de ventana en su versión ternaria, especificado en el Capítulo 6. La metodología utilizada es la misma que en el Capítulo 5.

En el Capítulo 8 se presenta el protocolo ALOHA estabilizado de ventana ideal, caracterizándolo y calculando analíticamente su throughput.

En el Capítulo 9, y siguiendo con la misma lógica, se procede al cálculo del retardo del protocolo ALOHA estabilizado de ventana ideal.

En el Capítulo 10 se exponen diversos análisis paralelos del protocolo ALOHA estabilizado de ventana en su versión binaria, producto de profundas reflexiones intuitivas que nos ayudan a conocer de una forma más completa al algoritmo.

En el Capítulo 11 se aplica el protocolo ALOHA estabilizado de ventana en su versión binaria a un sistema de integración de voz y datos, reportando unos resultados numéricos obtenidos por simulación que corroboran lo que se ha ido obteniendo en el marco de un sistema de acceso aleatorio.

La tesis finaliza con las conclusiones y la bibliografía referenciada.

Llegado este punto deseo hacer constar mi más profundo agradecimiento al Profesor Dr. Vicente Casares Giner, que con su valiosa dirección, asesoramiento y ayuda ha hecho posible la redacción de la presente monografía. Deseo agradecer también a todo el Departamento de Matemática Aplicada y Telemática las facilidades que siempre me han ofrecido, y muy especialmente a la Sección de Telemática y su responsable, el Profesor Dr. Emilio Sanvicente Gargallo, por todo el apoyo recibido. Quiero también destacar la inestimable ayuda que me proporcionó el laboratorio WINLAB (*Wireless Information Network Laboratory*), Universidad de Rutgers, New Jersey, EUA, y en especial su Director el Profesor Dr. David Goodman y su Director Asociado el Profesor Dr. Jack Holtzman, durante mi estancia allí de 4 meses. De igual modo quiero agradecer la ayuda financiera recibida durante estos años al Ministerio de Educación y Ciencia, por el cual fui becado dentro del Programa Sectorial de Formación de Profesorado Universitario y Personal Investigador. Por último, deseo agradecer también a todas aquellas personas que de una u otra forma me han ayudado a superar el difícil camino que he

debido recorrer. De entre todas ellas quiero hacer especial mención, por su significado apoyo, al Dr. Eduard Bajet por sus inestimables consejos y su valioso tiempo; y, sobre todo, a mi familia y a Elisenda Gelpí por creer en mí.



# Capítulo 2

## Los protocolos de acceso aleatorio

### 2.1 Introducción

Según N. Abramson [4], a finales de los años 60 hubo numerosos esfuerzos dirigidos al uso de la red telefónica conmutada existente para proporcionar acceso remoto a sistemas de ordenadores por parte de terminales y, en algunos casos más ambiciosos, proporcionar conexiones entre sistemas de proceso de una gran información mediante una compartición de recursos. El término compartición de recursos en esa época significaba a menudo una compartición de hardware. Sin embargo, en seguida se pudo comprobar que la red telefónica existente no era adecuada a las necesidades de las redes de datos que emergieron rápidamente en los años 70. El objetivo original del sistema ALOHA fue investigar el uso de comunicaciones radio como una alternativa al sistema telefónico para comunicaciones entre ordenadores.

La publicación en 1970 de N. Abramson [5] de su ahora famoso artículo sobre el sistema ALOHA marcó el nacimiento de las comunicaciones de acceso aleatorio. Sin embargo, durante casi una década, la teoría de información de los sistemas de acceso aleatorio apenas progresó después del trabajo original de N. Abramson.

Esta encrucijada teórica se rompió con una tesis doctoral de 1977 por parte de J. Capetanakis, bajo la supervisión del profesor R. Gallager, que presentó el algoritmo de árbol binario, el primer algoritmo que se puede considerar como un protocolo de resolución de colisiones.

A partir de ahí surgieron numerosos algoritmos, con sus prestaciones particulares, que han llevado a configurar el panorama actual de los protocolos de acceso múltiple.



## 2.2 Protocolos ALOHA

La familia de protocolos ALOHA es probablemente la familia más rica de protocolos de acceso múltiple. Su popularidad es debida primero de todo a su madurez, ya que fue la primera técnica de acceso aleatorio introducida, concretamente fue en 1970 por parte de la red ALOHA de N. Abramson [5]. En segundo lugar, muchos de estos protocolos son tan simples que su implementación es sencilla. Muchas redes de área local hoy en día implementan algunas variantes sofisticadas de esta familia de protocolos.

La familia de protocolos ALOHA pertenecen a los protocolos de tipo contienda o retransmisión aleatoria en los cuales el éxito de la transmisión no es garantizada a priori. La razón es que siempre que dos o más usuarios estén transmitiendo en un canal compartido simultáneamente, se produce una colisión y los datos no pueden ser recibidos correctamente. La temporización de la transmisión es, por lo tanto, el tema principal de los protocolos de tipo contienda.

A causa de la gran popularidad de los protocolos ALOHA, se han realizado análisis para un gran número de variaciones. Las variaciones presentan diferentes protocolos para la temporización de la transmisión y retransmisión, así como adaptación a las diferentes circunstancias y características de canal. Estos estudios estaban enfocados a resolver cuestiones como la política de transmisión óptima en el report técnico de R. Metcalfe [6], el comportamiento del canal con un número finito de usuarios como en el artículo de M. Ferguson [7] o la estabilidad del canal como en el de A. Carleial y M. Hellman [8].

Un amplio tratamiento de estos temas se puede encontrar en el libro de L. Kleinrock [9].

A continuación nos centraremos en los dos tipos de ALOHA más extendidos: el ALOHA puro y el ranurado (S-ALOHA).

### 2.2.1 ALOHA puro

El protocolo ALOHA puro es el protocolo básico en la familia de los protocolos ALOHA. Se considera un sistema single-hop con una población infinita generando paquetes de igual longitud  $T$  de acuerdo con un proceso de Poisson de tasa  $\lambda$  paquetes por segundo. El canal es libre de error sin captura: siempre que una transmisión de un paquete no interfiera con alguna otra transmisión de paquete, el paquete transmitido es recibido correctamente mientras que si dos o más transmisiones de paquete se solapan en el tiempo, se produce una colisión y ninguno de los paquetes colisionados son recibidos correctamente y tienen que ser retransmitidos. Los usuarios cuyos paquetes colisionan con otro se llaman usuarios en colisión. Al final de cada trans-

misión cada usuario sabe si su transmisión ha sido exitosa o ha tenido lugar una colisión.

El protocolo ALOHA puro fue el primero y más simple, ver el trabajo de N. Abramson [5]. Establece que un nuevo paquete generado sea transmitido inmediatamente esperando que no haya interferencia por parte de los otros. Si la transmisión es no exitosa, cada usuario en contienda, independientemente de los otros, temporiza su retransmisión a un tiempo aleatorio en el futuro. Esta aleatoriedad se requiere para asegurar que este mismo conjunto de paquetes no colisionen indefinidamente. Un paquete empezando su transmisión en el tiempo  $t$  colisionará con cualquier otro paquete empezando en cualquier tiempo dentro del intervalo  $(t - 1, t + 1)$ . El límite superior del throughput ( $r_{max}$ ) es de  $1/2e = 0.18$ . este throughput se puede obtener mediante el mismo procedimiento que seguiremos para el S-ALOHA en el próximo Apartado.

Ya que la población es infinita cada paquete puede ser considerado como si perteneciera a un usuario diferente. Así pues, cada paquete recién llegado puede ser asignado a un usuario vacío, es decir, uno que no tiene paquetes a retransmitir. Esto nos permite intercambiar los papeles de los paquetes y los usuarios y considerar sólo los puntos en el tiempo cuando se realizan los intentos de retransmisión.

La mayor ventaja práctica del ALOHA puro es su capacidad para manejar paquetes de longitudes diversas. Para una más profunda visión consultar los trabajos de M. Fergusson [10] y S. Bellini y P. Borgonovo [11].

### 2.2.2 ALOHA ranurado (S-ALOHA)

La versión ranurada del ALOHA, llamada ALOHA ranurado o simplemente S-ALOHA, se obtiene dividiendo el tiempo en ranuras de igual duración a la de la transmisión de un único paquete (suponiendo que los paquetes son de longitud constante). Cada usuario tiene que sincronizarse al inicio de la transmisión de su paquete para coincidir con el límite de la ranura. Cuando dos paquetes colisionen, se solaparán completamente, no parcialmente como en el ALOHA puro, proporcionando un incremento en la eficiencia del canal.

Siempre que ocurra una colisión en una ranura, cada paquete envuelto en la colisión se dice que está retrasado (*backlogged*) hasta que es exitosamente transmitido. Cada uno de los paquetes retrasados se transmite en cada ranura subsiguiente con una probabilidad fija ( $p > 0$ ), independiente de las ranuras pasadas y de los otros paquetes. Hay que reseñar que, si  $p$  fuera 1, los paquetes retrasados continuarían colisionando y ningún paquete sería transmitido.

En el análisis del throughput del S-ALOHA, N. Abramson [5] postuló

que la política de transmisión era de tal forma que el número de paquetes retransmitidos en cualquier ranura podía ser una variable aleatoria de Poisson, con media  $\lambda_r$  paquetes/ranura, independiente ranura a ranura e independiente del proceso de generación de nuevos paquetes; según B. Hajek and T. Van Loon [12] esto es lo que se denomina aproximación local de Poisson. Como la suma de variables independientes de Poisson es de nuevo Poisson, esto implica que el número total de paquetes transmitidos en cualquier ranura es también una variable aleatoria de Poisson de media  $\lambda_t = \lambda + \lambda_r$ . Puesto que el throughput  $\tau$  de paquetes exitosos en el receptor es la fracción de ranuras en las cuales se transmite exáctamente un paquete, se concluye que  $\tau$  es la probabilidad que una variable aleatoria de Poisson de media  $\lambda_t$  tome el valor 1, es decir,

$$\tau = \lambda_t e^{-\lambda_t} \quad (2.1)$$

La ecuación anterior, que se conoce como la ecuación del throughput del S-ALOHA, se representa en la figura 2.1. Es fácil comprobar de la ecuación (2.1) que  $\tau$  se maximiza para  $\lambda_t = 1$  paquetes/ranura, lo que parece bastante natural, y que su máximo es

$$\tau_{max} = e^{-1} \approx 0.368 \text{ paquetes/ranura} \quad (2.2)$$

Estas relaciones son muy similares a las del ALOHA puro, excepto que el throughput se incrementa. La capacidad del canal es  $1/e \approx 0.36$  y se consigue para  $\lambda_t = 1$ . Los resultados anteriores fueron derivados primero por L. Roberts [13].

### 2.2.3 Inestabilidad de los protocolos ALOHA

Una suposición subyacente en el análisis del protocolo ALOHA es que el proceso de llegadas conjunto de paquetes nuevos y retransmitidos (debido a colisiones) es un proceso de Poisson. No hay justificación a este supuesto a no ser que simplifique el análisis del protocolo ALOHA. Bajo este supuesto, llamado aproximación local del ALOHA, se ha predicho que el throughput del protocolo S-ALOHA es

$$\tau = \lambda_t e^{-\lambda_t} \quad (2.3)$$

Otro supuesto que es implícitamente usado en el análisis del protocolo ALOHA es el supuesto de estabilidad; es decir, que el número de usuarios esperando a ser retransmitidos no está permanentemente creciendo. En otras palabras, se asume que los paquetes están entrando y saliendo del sistema

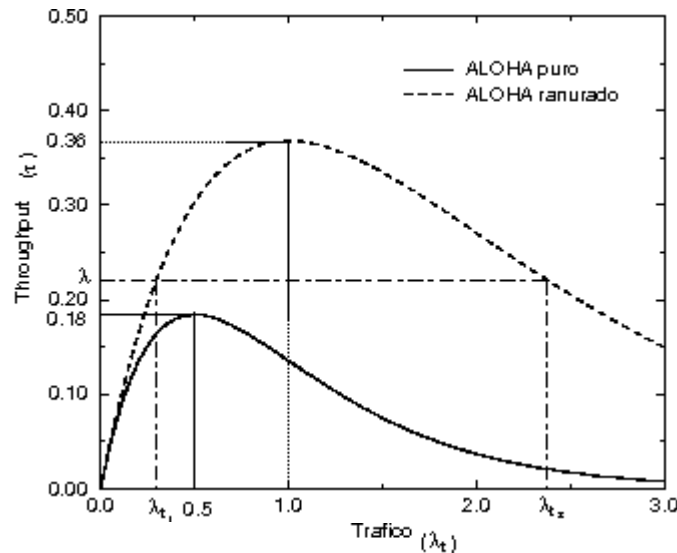


Figura 2.1: Gráfica de la ecuación del throughput ( $\tau$ ) en función del tráfico ( $\lambda_t$ ) del ALOHA puro y ranurado.

con la misma tasa. Razonaremos intuitivamente por qué este supuesto es falso.

Consideremos la figura 2.1, donde la función en la expresión (2.1) es representada gráficamente. Asumimos que la tasa de llegada de nuevos paquetes es  $\lambda$  paquetes por ranura y asumimos que  $\lambda < e^{-1}$ , ( $e^{-1}$  es el máximo throughput predicho para el protocolo S-ALOHA). Si prevalece el equilibrio entre tasas de llegada y de salida, entonces la tasa del tráfico total sobre el canal (paquetes nuevos y retransmitidos)  $\lambda_t$  será  $\lambda_{t1}$ , como muestra en la figura 2.1. Esta es naturalmente una tasa "media", y sobre cualquier intervalo fijo de tiempo, la tasa instantánea fluctuará alrededor de esta media. Si la tasa de tráfico instantáneo se mueve un poco por encima de  $\lambda_{t1}$ , el throughput instantáneo crece un poco por encima de  $\lambda$ . Así, los paquetes salen del sistema más rápido que llegaron, lo que produce que la tasa instantánea de tráfico decrezca hasta  $\lambda_{t1}$ . Si la tasa instantánea de tráfico se mueve un poco por debajo de  $\lambda_{t1}$ , el throughput instantáneo decrece un poco por debajo de  $\lambda$ . Así, los paquetes salen más lentos que llegaron, lo que produce que la tasa instantánea de tráfico decrezca hasta  $\lambda_{t1}$ . Consecuentemente, el punto  $(\tau, \lambda_t) = (\lambda, \lambda_{t1})$  es un punto condicionalmente estable, es decir, es estable bajo pequeñas variaciones de  $\lambda_{t1}$ . Sin embargo, si una larga variación (y esto

ocurrirá con probabilidad 1) provoca que el tráfico instantáneo exceda  $\lambda_{t_2}$ , entonces el throughput instantáneo decrece por debajo de  $\lambda$ . Así, los paquetes salen del sistema a una tasa más lenta que entraron, lo que provoca un mayor incremento de la tasa instantánea de tráfico, un mayor decremento del throughput instantáneo, etc. El sistema nunca vuelve al punto  $(\lambda, \lambda_{t_1})$ , sino que tiende irremediamente a un punto incondicionalmente estable, pero catastrófico,  $(\tau, \lambda_t) = (0, \infty)$ . Concluimos, pues, que el throughput máximo estable del protocolo ALOHA es cero.

Un análisis más completo y formal se puede encontrar en el trabajo de G. Fayolle, J. Labetoulle, D. Bastin y E. Gelenbe [14].

## 2.3 Protocolos de resolución de colisiones

Hemos visto que el protocolo ALOHA original era inherentemente inestable con la ausencia de control externo. Si miramos en la filosofía que subyace en el protocolo ALOHA, vemos que no hay un intento serio de resolver las colisiones entre los paquetes así como ocurren. En su lugar, los intentos de resolver colisiones son siempre trasladados al futuro, con la esperanza que las cosas funcionarán entonces mejor, pero nunca ocurre.

En esta sección se presentan y analizan protocolos de acceso múltiple con una filosofía distinta. En estos protocolos, llamados Protocolos de Resolución de Colisiones (CRP), los esfuerzos se concentran en resolver las colisiones tan pronto como ocurran. Además, en la mayoría de las versiones de estos protocolos, los paquetes nuevos que llegan al sistema se inhiben de ser transmitidos mientras la resolución de colisiones está en progreso. De esta manera se asegura que si la tasa de llegadas de paquetes nuevos al sistema es más pequeña que la tasa con la cual las colisiones pueden ser resueltas (la tasa máxima de paquetes salientes), entonces el sistema es estable.

La idea básica detrás de estos protocolos es explotar de una forma más sofisticada la información del feedback disponible por parte de los usuarios con el fin de controlar el proceso de retransmisión, para que las colisiones sean resueltas más eficientemente y sin eventos caóticos.

El modelo subyacente y los supuestos utilizados aquí son idénticos a aquellos asumidos por el protocolo S-ALOHA. El canal es ranurado y los usuarios pueden transmitir paquetes (cuya longitud es una ranura) sólo al inicio de éstas. Los paquetes nuevos llegan al sistema de acuerdo con un proceso de Poisson de tasa  $\lambda$  paquetes/ranura. Si dos o más paquetes son transmitidos en una ranura, se produce una colisión y los paquetes envueltos en la colisión tienen que ser retransmitidos. Al final de cada ranura los usuarios saben lo que ha ocurrido en ella, es decir, si ésta ha permanecido vacía

(ningún paquete fue transmitido), o ha contenido alguna transmisión exitosa (exáctamente un paquete fue transmitido) o se ha generado una colisión (al menos dos paquetes fueron transmitidos). Esto se conoce como modelo de feedback ternario. Para algunos protocolos es suficiente para los usuarios conocer si la ranura contuvo colisión o no. Este último se refiere al modelo de feedback binario. Es de esperar por tanto que un protocolo con feedback ternario tenga unas mayores prestaciones que el feedback binario.

### 2.3.1 Protocolos de árbol binario

El protocolo más básico de resolución de colisiones se llama CRP de árbol binario (o protocolo de árbol binario) y fue propuesto casi concurrentemente por J. Capetanakis [15], B. Tsybakov y V. Mikhailov [16], y J. Hayes [17]. De acuerdo con este protocolo cuando ocurre una colisión, digamos en la ranura  $k$ , todos los usuarios que no están envueltos en esta colisión esperan hasta que la colisión sea resuelta. Los usuarios envueltos en la colisión se dividen aleatoriamente en dos subconjuntos, lanzando, por ejemplo, una moneda. Los usuarios del primer subconjunto, los que sacaron cara, retransmiten en la ranura  $k + 1$ , mientras que aquellos que sacaron cruz esperan hasta que todos aquellos que sacaron cara transmitan exitosamente sus paquetes. Si la ranura  $k + 1$  está vacía o contiene una transmisión exitosa, los usuarios del segundo subconjunto, los que sacaron cruz, retransmiten en la ranura  $k + 2$ . Si la ranura  $k + 1$  contiene otra colisión, entonces el procedimiento se repite, es decir, los usuarios cuyos paquetes colisionaron en la ranura  $k + 1$  (los usuarios en contienda), lanzan una moneda de nuevo y operan de acuerdo con el resultado obtenido, y así sucesivamente. Nos referiremos a un usuario que tiene un paquete que colisiona (al menos uno) como usuario *backlogged*.

La explicación anterior muestra que el protocolo queda especificado por una recursión: un grupo de paquetes en contienda es dividido en dos subgrupos cada uno de los cuales es regido por el mismo procedimiento que el grupo inicial. Esta recursión se manifestará más tarde. Pero incluso a nivel de descripción la operación recursiva del protocolo se describe mejor por un árbol binario, en el cual cada vértice corresponde a una ranura temporal. La raíz del árbol corresponde a la ranura de la colisión original. Cada vértice del árbol también designa un subconjunto (quizás vacío) de usuarios *backlogged*. Los vértices cuyos subconjuntos contienen al menos dos usuarios (etiquetados  $\geq 2$ ) indican colisiones y tienen dos ramas salientes, correspondientes a la división del subconjunto en dos nuevos subconjuntos. Los vértices correspondientes a subconjuntos vacíos (etiquetados 0), o subconjuntos conteniendo un usuario (etiquetados 1) son hojas del árbol e indican una ranura vacía o exitosa, respectivamente.

Claro está que cada usuario puede construir este árbol binario siguiendo las señales del feedback correspondiente a cada ranura. Los usuarios que no están envueltos en la colisión, pueden también seguir el árbol binario y así saber exáctamente cuando se resuelve la colisión. De la misma forma, cada usuario backlogged puede guardar su propia posición en ese árbol (mientras se resuelve la colisión), y así poder determinar cuando transmitir su paquete. Para la correcta operación del protocolo de árbol binario, es suficiente el feedback binario, es decir, los usuarios no tienen que distinguir entre las ranuras vacías y las exitosas.

Decimos que una colisión se resuelve cuando los usuarios del sistema saben que todos los paquetes envueltos en una colisión han sido transmitidos exitosamente. El intervalo de tiempo que empieza con la colisión original (si hay) y acaba cuando la colisión se resuelve se llama intervalo de resolución de colisiones (CRI).

El protocolo de árbol binario dicta cómo resolver las colisiones una vez ocurren. Para completar la descripción del protocolo, necesitamos especificar cuando los paquetes nuevamente generados son transmitidos por primera vez o, en otras palabras, especificar la regla de transmisión de la primera vez. Una alternativa, que hemos asumido siempre (conocido por esquema de acceso bloqueado obvio), es que los nuevos paquetes son inhibidos de ser transmitidos mientras la resolución de una colisión sigue su curso. Esto es, los paquetes que llegan al sistema mientras la resolución de una colisión es en proceso, esperan hasta que la colisión sea resuelta, que será cuando serán transmitidos.

La operación del protocolo de árbol binario puede ser también descrito en términos de una pila. Es, de hecho, una descripción estándar de un árbol transversal por una representación de pila. En cada ranura la pila vacía su primera posición, y todos los usuarios que estaban encima de la pila transmiten sus mensajes. En caso de colisión, la pila se llena primero con los usuarios que sacaron cruz, y luego se llena de nuevo con los usuarios que sacaron cara. Los usuarios que sacaron cara permanecen por lo tanto en la cima de la pila para ser sacados y transmitir en la próxima ranura. En caso de una transmisión exitosa o de una ranura vacía no se hacen más operaciones en la pila. Claramente entonces, cuando la pila se vacía la colisión se ha resuelto, el CRI se acaba y los usuarios con paquetes recién llegados, si hay, son introducidos en la cima de la pila y el proceso opera como antes.

El análisis de las características del protocolo de árbol binario se puede hacer tanto con la representación de árbol como de pila alcanzando, naturalmente, los mismos resultados. Análisis vía representación de pila fueron desarrollados por G. Fayolle, P. Flajolet, M. Hofri y P. Jacquet [18], mientras que vía la representación de árbol se pueden encontrar en los trabajos

de J. Massey [19]. El protocolo de árbol binario descrito en este apartado consigue un throughput máximo estable de 0.4295 paquetes/ranura.

### 2.3.2 Protocolos mejorados

Las prestaciones del protocolo de árbol binario pueden ser mejorados de dos formas. La primera es acelerar el proceso de resolución de colisiones evitando ciertas colisiones. La segunda se basa en la observación que las colisiones entre un pequeño número de paquetes son resueltas más eficientemente que las colisiones entre un gran número de paquetes. Por lo tanto, si la mayoría de los CRI empiezan con un pequeño número de paquetes, será de esperar que las prestaciones del protocolo mejoren.

#### Protocolos de árbol binario modificado

El protocolo de árbol binario modificado es debido a J. Massey [19], éste evoluciona exáctamente como el protocolo de árbol binario, excepto en que elimina las colisiones evitables. Desde un punto de vista intuitivo, si se habilita un espacio temporal para transmitir y hay una colisión, este intervalo será subdividido en dos partes. Si en la primera parte resulta que no hay ningún usuario, podemos afirmar que en el intervalo restante habrá de nuevo una colisión si hacemos acceder a todos los usuarios. Por tanto la mejora consiste en que al llegar a un caso como el explicado, automáticamente se vuelva a subdividir el intervalo para evitar una colisión segura, que no aportaría ninguna información adicional.

La mejora que acabamos de describir tiene, como es de esperar, un precio a pagar, que se resume en los tres siguientes puntos,

1. El protocolo de árbol binario modificado requiere que la información del feedback distinga entre ranuras vacías y ranuras con un paquete, es decir, se necesita un feedback ternario.
2. El protocolo de árbol binario modificado es ligeramente más complejo de implementar por la necesidad de añadir un *flag* de control.
3. En el caso de errores en el canal, el protocolo de árbol binario modificado puede llegar a una situación irreversible, en el cual un CRI nunca acaba y no se puede transmitir ningún paquete más. Esta es la razón de más peso.



### 2.3.3 El mecanismo de ventana

En protocolos de árbol binario la resolución de colisiones entre un número pequeño de paquetes es más eficiente que entre un gran número de paquetes. Cuando se utiliza el acceso bloqueado obvio, es muy probable que un CRI empiece con una colisión entre un gran número de paquetes si el CRI anterior ha sido largo. Cuando los sistemas operan cerca de su throughput máximo la mayoría de los CRIs son largos, y de ahí que se tengan que resolver frecuentemente colisiones entre un gran número de paquetes, que hemos visto que no es muy eficiente.

Idealmente, si fuera posible empezar cada CRI con la transmisión de exactamente un paquete, el throughput del sistema sería la unidad. Ya que esto no es posible, se debería intentar diseñar el sistema para que en la mayoría de los casos un CRI empezara con la transmisión de aproximadamente un paquete. Hay varias formas de conseguir este objetivo determinando una regla de primera transmisión, es decir, cuando los paquetes son transmitidos por primera vez. Una forma, sugerida por Capetanakis, es tener una estimación del número de paquetes que llegaron en el CRI anterior y dividirlos en grupos más pequeños, cada uno con un número esperado de paquetes del orden de 1 y tratar cada uno de los grupos separadamente. Otra forma, conocida como el mecanismo de ventana, sugerido por R. Gallager [20] y por B. Tsybakov y V. Mikhailov [21], se describe a continuación.

Consideramos las llegadas de los paquetes al sistema y dividimos el eje temporal en ventanas consecutivas (llamadas ventanas de llegada), cada una de longitud  $\Delta$  ranuras temporales ( $\Delta$  no es necesariamente un entero). La  $i$ ésima ventana de llegada es el intervalo de tiempo  $(i\Delta, (i+1)\Delta)$ . Los paquetes que llegan durante la  $i$ ésima ventana de llegada son transmitidos por primera vez en la primera ranura tras la resolución de la colisión entre los paquetes que llegaron durante la ventana de llegada anterior, es decir, la  $(i-1)$ ésima. El parámetro  $\Delta$  se escoge de tal forma que optimice las prestaciones del sistema. En el canal observamos una secuencia de intervalos de resolución de colisiones cada uno correspondiente a las llegadas durante algún intervalo de tiempo en el eje de llegadas. Si numeramos secuencialmente estos intervalos de resolución de colisiones entonces en el  $i$ ésimo CRI todos los paquetes, si hay, que lleguen durante la  $i$ ésima ventana son transmitidos exitosamente. Todos los paquetes que llegaron en la ventana  $i$ ésima, es decir, aquellos que llegaron en el periodo  $(0, \Delta)$  son transmitidos en la primera ranura del CRI-0; colisionan y empieza un proceso de resolución. Mientras tanto, los paquetes recién llegados esperan y cuando el CRI-0 finaliza, todos los paquetes pertenecientes a la primera ventana, es decir, aquellos que llegaron en el periodo  $(\Delta, 2\Delta)$  son transmitidos, y así sucesivamente. En el

caso que llegara un punto en el que la ventana habilitada alcanzara la ranura en la que se está haciendo la transmisión, hay dos posibilidades: permitir el acceso a todos los usuarios, es decir, una ventana menor que  $\Delta$ , o esperar a que sea posible habilitar una ventana de longitud  $\Delta$ . En ambos casos el throughput que se consigue es el mismo, pero el retardo es ligeramente menor en el primer caso, según J. Huang y T. Berger [22], y éste será el que consideraremos en análisis posteriores.

### 2.3.4 El protocolo de árbol binario truncado

En el análisis previo llegamos a la conclusión que para mejorar las prestaciones un CRI debía empezar con una transmisión de aproximadamente un paquete. Sin embargo, esa idea no fue del todo explotada por las mejoras anteriores. Para ver el porqué, asumimos que empleamos el mecanismo de ventana y consideramos lo que ocurre cuando una colisión es seguida por otra colisión. La colisión en la primera ranura se produce entre los paquetes que llegaron durante un intervalo de longitud  $\Delta$ . Los usuarios cuyos paquetes colisionaron son divididos en dos grupos dividiendo el intervalo en dos partes (por razones de explicación asumimos que son iguales y de valor  $\Delta/2$ ). Los paquetes que llegaron en la parte izquierda son entonces transmitidos en la siguiente ranura y colisionan de nuevo. Tras una división repetida del intervalo los paquetes de la parte izquierda son finalmente transmitidos. En este punto los paquetes que llegaron en la parte derecha del intervalo original son transmitidos. Es en este punto que no se hace lo mejor que se puede hacer.

Si una colisión es seguida inmediatamente por otra colisión, entonces no tenemos información alguna acerca del número de paquetes en la segunda mitad del intervalo correspondiente a la anterior colisión. Así, la segunda mitad del intervalo puede ser fusionado con la porción no examinada del eje temporal de llegadas y no explorado como se haría en la continuación normal de la resolución de la colisión. Para incorporar esta estrategia en el protocolo adoptamos la regla que siempre que una colisión es seguida por dos transmisiones sucesivas exitosas, se permite una nueva ventana de longitud  $\Delta$  en el eje de llegadas. El protocolo que resulta de tal operación se llama protocolo de árbol binario truncado, ya que parte del árbol binario es truncado y no permitido. Se podría añadir que la idea de dividir puede ser usada tanto para el original protocolo de árbol binario como para el modificado, es decir, el que evita la colisión segura de los paquetes (este último se llama protocolo de árbol binario truncado y modificado).

Usando esta estrategia, sugerida por R. Gallager [20] y por B. Tsybakov y V. Mikhailov [21], R. Gallager [20] obtuvo un throughput de 0.4872 paquetes/ranura (comparado con los sólo 0.462 paquetes/ranura sin este "truco").

J. Mosely y P. Humblet [23], y B. Tsybakov y V. Mikhailov [21] refinaron el método optimizando a cada paso la longitud del intervalo de llegada que daba permiso para transmitir (lo que es equivalente a utilizar una moneda sesgada) hasta obtener un throughput máximo estable de 0.4878 paquetes/ranura. Otra mejora, aunque muy pequeña, de  $3.6 \cdot 10^{-7}$  fue obtenida por N. Vvedenskaya y M. Pinsker [24]. Esta mejora no es significativa desde el punto de vista práctico, sí en cambio desde el punto de vista teórico a la hora de determinar el óptimo en términos de throughput máximo estable. Todos los algoritmos anteriores necesitan un feedback ternario. Con un feedback binario, el protocolo de árbol binario truncado que consigue un mayor throughput es el de P. Studer y H. Pletscher [25], con 0.9443 paquetes/ranura. Este es el algoritmo binario que consigue un mayor throughput de los conocidos hasta el momento.

### 2.3.5 El protocolo de ventana de pila de dos células

Los protocolos de árbol binario truncado han resultado ser los más eficientes hasta ahora en términos de throughput máximo estable. Pero, como se pone de manifiesto en la literatura especializada, y así queremos reflejarlo a lo largo de esta monografía, las prestaciones de un algoritmo se deben medir tomando en consideración diversos parámetros, entre los que, indudablemente se encuentra el throughput, pero también el retardo y la robustez frente a no idealidades del canal.

Una de las mayores desventajas que presentan los protocolos de árbol binario truncado es precisamente su sensibilidad frente a los errores en el feedback, que pueden llevarle a una situación irreversible.

Otro de los grandes problemas que plantean radica en la propia concepción de dividir a los paquetes de entrada por el tiempo de llegada. Según el modelo de Poisson hay una probabilidad nula que dos paquetes lleguen simultáneamente. En un modelo práctico no puede despreciarse esta posibilidad, porque los relojes que sincronizan el sistema tienen un periodo finito y no nulo, de tal forma que si dos terminales se activan en un mismo periodo, estos no podrán ser discernidos por su tiempo de llegada; y si se aplica un protocolo basado en tiempo de llegada, los dos usuarios permanentemente seguirán las mismas reglas, siendo imposible la resolución de su colisión.

Por consiguiente, y desde este punto de vista, sería muy interesante la implementación de algoritmos robustos frente a errores en el feedback, y que pudieran soportar distribuciones estadísticas de tiempos de llegada no poissonianos, aunque ello implicara obtener throughputs máximos estables no tan elevados.

Después de consultada extensamente la literatura especializada, he llega-

do a la conclusión que uno de los algoritmos más importantes con las características anteriores es el algoritmo de ventana de pila de dos células (*two cell stack window*, TCW). Este algoritmo fue concebido por P. Papantoni-Kazakos and M. Paterakis [26] y [27].

En él, básicamente se utiliza un mecanismo de ventana para el protocolo de acceso al canal, que se explicará de forma extensiva en el capítulo siguiente y el protocolo de resolución de colisiones sigue las siguientes reglas para resolver las posibles colisiones.

En la primera ranura todos los usuarios acceden, de forma que si sólo hay 0 o 1 paquetes, el CRI durará una ranura. Durante un CRI que empiece con una ranura de colisión, cada usuario actúa independientemente vía la utilización de un contador cuyo valor en el tiempo  $t$  viene denotado por  $r_t$ . Los valores del contador pueden ser 1 o 2 y son actualizados dependiendo del contador  $r_t$  y de la información del feedback binario  $x_t$  (colisión si  $x_t = C$  y no colisión si  $x_t = NC$ ), de acuerdo con las reglas siguientes,

1. El usuario transmite en la ranura  $t$  si y sólo si  $r_t = 1$ . Un paquete es exitosamente transmitido en  $t$  si y sólo si  $r_t = 1$  y  $x_t = NC$ .
2. Las transiciones de los valores del contador en función del tiempo son,
  - (a) Si  $x_{t-1} = NC$  y  $r_{t-1} = 2$  entonces  $r_t = 1$ ,
  - (b) Si  $x_{t-1} = C$  y  $r_{t-1} = 2$  entonces  $r_t = 2$ ,
  - (c) Si  $x_{t-1} = C$  y  $r_{t-1} = 1$  entonces,

$$r_t = \begin{cases} 1 & \text{con probabilidad 0.5} \\ 2 & \text{con probabilidad 0.5} \end{cases}$$

De estas reglas puede verse que un CRI que empieza con una ranura de colisión acaba si y sólo si hay dos ranuras consecutivas de no colisión.

Este algoritmo es el que tendremos en nuestro punto de mira a la hora de estudiar y analizar el protocolo ALOHA estabilizado de ventana (WSA), por diversas razones,

1. Este algoritmo ha sido analizado con multitud de variantes y no idealidades en el canal. Entre ellas cabe destacar el análisis con canales ruidosos sin memoria, por parte de M. Paterakis y P. Papantoni-Kazakos [27], y con memoria, por parte de Y. Gong y M. Paterakis [28]; el análisis de la distribución de tráfico de salida de M. Paterakis y P. Papantoni-Kazakos [27]; el análisis con restricciones en el retardo de los paquetes, de M. Paterakis, L. Georgiadis y P. Papantoni-Kazakos [29]; el análisis en sistemas con tráfico de alta y baja prioridad

mezclados, de P. Papantoni-Kazakos [30]; y la aplicación como algoritmo de reserva a sistemas de integración de voz y datos, de trabajos como los de F. Babich [31], [32] y [33], de S. Jangi y L. Merakos [34], de M. Paterakis y A. Cleary [35], y de A. Cleary y M. Paterakis [36].

2. Ambos algoritmos (WSA y TCW) tienen características similares en lo que respecta a la robustez frente a los errores en el feedback y funcionamiento ante distribuciones de entrada no poissonianas.

A lo largo de los siguientes capítulos iremos obteniendo resultados que demuestran que el protocolo ALOHA estabilizado de ventana tiene mejores prestaciones que el protocolo de ventana de pila de dos células.

En ningún caso hay que caer en la tentación de considerar que uno es evolución del otro y viceversa, dada su similitud. Hay que dejar claro que son conceptualmente distintos. El algoritmo TCW resuelve las colisiones separando los usuarios en dos grupos y trasladándolos de un grupo a otro hasta que en el primero sólo hay un paquete que accede; por ello se puede decir que pertenece a los algoritmos *splitting*. Por su parte, el ALOHA estabilizado de ventana permite el acceso de *todos* los usuarios en contienda con una probabilidad que varía a lo largo de la resolución del CRI, como veremos en el capítulo siguiente.

# Capítulo 3

## El protocolo ALOHA estabilizado de ventana

### 3.1 Introducción

En el capítulo anterior se ha descrito los diferentes algoritmos que conforman el panorama actual en el campo de los algoritmos de acceso múltiple. Con el fin de desarrollar el estudio de una nueva familia de algoritmos de acceso aleatorio, objeto de esta tesis, vamos a estructurarlo adecuadamente.

En general, un sistema de acceso aleatorio está compuesto por un protocolo de acceso al canal y un algoritmo de resolución de colisiones.

En el presente capítulo vamos a describir un nuevo sistema de acceso aleatorio. Por consiguiente, para la descripción de dicho sistema nos bastará con especificar por una parte el protocolo de acceso al canal, y por otra el algoritmo de resolución de colisiones utilizado. El sistema global estará perfectamente caracterizado una vez elegido el modelo de canal multiacceso. Este será el que comúnmente se ha venido utilizando para problemas de acceso aleatorio.

### 3.2 Modelo de sistema

En cualquier sistema de acceso aleatorio (*random-access system*, RAS), el tema a tratar es la resolución de conflictos debidos a la utilización de un medio común.

Desde un punto de vista algorítmico, un sistema de acceso aleatorio está formado por un protocolo de acceso al canal (*channel access protocol*, CAP) y por un algoritmo de resolución de colisiones (*collision resolution algorithm*, CRA). Esta distinción es debida a P. Mathys y P. Flajolet [37]. J. Huang y

T. Berger [22] utilizan otra nomenclatura y denominan estrategia global a los CAPs y estrategia local a los CRAs. Durante los últimos años al protocolo de acceso al canal se le ha venido denominando también regla de la primera transmisión (*first transmission rule*, FTR), pero la idea subyacente sigue siendo la misma; como ejemplos de esto último se encuentran los trabajos de A. Bar-David y M. Sidi [38] y de Y. Gong y M. Paterakis [28].

**Protocolo de acceso al canal.** Básicamente éste es un algoritmo que determina, para cada transmisor, cuándo intentan transmitir por primera vez los usuarios recién llegados, según P. Mathys y P. Flajolet [37]; o, visto de otro modo, es el algoritmo que especifica el procedimiento para escoger el intervalo temporal habilitado para la transmisión en cada momento, según J. Huang y T. Berger [22].

Desde un punto de vista conceptual, se pueden distinguir dos tipos de protocolo de acceso al canal. Por orden cronológico son el protocolo de acceso libre y el protocolo de acceso bloqueado.

1. El protocolo de acceso libre, (*free-access protocol*, FAP). Es el más sencillo. En él, el transmisor envía el paquete generado en la ranura siguiente a su aparición.
2. El protocolo de acceso bloqueado, (*blocked-access protocol*, BAP). En él, ningún paquete recién llegado es transmitido hasta que se haya resuelto al menos la contienda que está en curso. Esta idea fue propuesta originariamente por R. Gallager [20] y J. Massey [19].

El FAP, por ser el más simple, fue el primero en aparecer. Ya con los protocolos ALOHA la idea utilizada inicialmente era aplicar el FAP. Los problemas que presentaba este mecanismo era el efecto avalancha por el hecho de permitir a todo usuario recién llegado al sistema entrar en contienda. Un primer paso hacia lo que posteriormente se consideró como BAP, fue el distinguir entre dos filosofías de funcionamiento: la primera transmisión inmediata (*immediate first transmission*, IFT) y la primera transmisión diferida (*deferred first transmission*, DFT). Esta nomenclatura es debida a F. Tobagi. Concretamente, la DFT fue el principio de lo que luego evolucionó hacia un BAP; en ella, cualquier paquete recién llegado inmediatamente pasaba a estar en modo bloqueado, es decir, como si hubiera sufrido una colisión. De esta manera se lograba reducir en parte el efecto avalancha.

Trabajos como el de G. Fayolle, E. Gelenbe y J. Labetoulle [39], el de L. Clare [40] y S. Thomopoulos [41], probaron estabilidad para IFT, mientras que R. Rivest [42] probó estabilidad para DFT y B. Hajek y T. Van Loon [41]

para ambos. Para una discusión adicional de uno y otro ver el trabajo de L. Merakos y D. Kazakos [43]. Según J. Massey [44] este protocolo tiene la ventaja práctica que los usuarios que transmiten necesitan monitorizar el feedback del canal sólo cuando tengan algún paquete a transmitir; además intuitivamente el acceso libre le sugiere que tendrá un mejor throughput que el acceso bloqueado.

Considero, en cambio, que la única ventaja que tienen los algoritmos de acceso libre es su fácil implementación. Como explican P. Mathys y P. Flajolet [37] al acceso bloqueado tiene que monitorizar durante todo el tiempo el canal porque ha de mantener una variable global que le indique cuando acceder por primera vez; todo lo demás son inconvenientes puesto que ha quedado probado en la literatura que los protocolos de acceso bloqueado han sido más eficientes. Además los protocolos de acceso bloqueado son más sencillos de analizar puesto que el hecho de desacoplar los tiempos de transmisión de los tiempos de llegada, como dicen I. Cidon, H. Kodesh y M. Sidi [45], hace que en su análisis se puedan aplicar potentes herramientas matemáticas de la teoría de colas, como es la teoría de la renovación.

Otra ventaja importante es que el acceso bloqueado asegura un comportamiento FIFO en una cierta medida, que, hasta cierto punto, es un comportamiento más justo.

**Algoritmo de resolución de colisiones.** Hay que dejar bien claro desde un principio que el algoritmo de resolución de colisiones es una parte de un sistema de acceso aleatorio. Podría decirse que es la más característica hasta el punto que incluso algunas veces se ha llegado a confundir uno con otro. A medida que el estudio de la resolución de colisiones ha evolucionado, han ido apareciendo diversas definiciones de CRA que, salvo ligeras variaciones, representan todas ellas un concepto común.

Una clásica definición de CRA es debida a J. Massey [19]: El CRA es un protocolo para la transmisión y retransmisión de paquetes por nodos individuales con la propiedad que después de una colisión inicial o una captura, todos los paquetes envueltos son finalmente retransmitidos exitosamente y todos los nodos, no sólo aquellos que retransmitieron en el nodo inicial, simultáneamente y permanentemente son conscientes que estos paquetes se han retransmitido exitosamente.

Decimos que una colisión se ha resuelto precisamente cuando todos los nodos son conscientes que todos los paquetes en contienda han sido exitosamente retransmitidos.

Según P. Mathys y P. Flajolet [37], un CRA se puede definir como un algoritmo (distribuido en el espacio y en el tiempo) que organiza la retrans-



misión de paquetes en contienda de tal manera que cada paquete se transmite exitosamente con un retardo finito y, además, todos los transmisores (o al menos todos aquellos para los que es relevante) son conscientes de este hecho.

El espacio temporal desde la ranura donde se ha producido una colisión inicial hasta la ranura donde todos los transmisores (o al menos todos aquellos para los que es relevante) reconocen que todos los paquetes envueltos en dicha colisión inicial han sido recibidos exitosamente se llama intervalo de resolución de colisiones (*collision resolution interval*, CRI). Notar que un CRI puede ser una ranura vacía, exitosa, o una colisión con su resolución subsiguiente.

Según R. Rom y M. Sidi [46], la idea básica detrás de estos protocolos es explotar de una forma más sofisticada la información del feedback que los usuarios disponen para controlar el proceso de retransmisión, con el fin que las colisiones sean resueltas más eficientemente y sin eventos caóticos. El objetivo de un CRA es distribuir el ancho de banda del canal entre los usuarios que hay al principio del CRI de una forma eficiente, según M. Paterakis y P. Papantoni-Kazakos [47].

De esta forma, un usuario que haya podido intentar transmitir porque así se lo ha permitido el protocolo de acceso al canal, seguirá las reglas dictadas por el CRA hasta su transmisión definitiva. En definitiva, y según G. Polyzos, M. Molle y A. Venetsanopoulos [48] un CRA puede ser descrito por las acciones que dicta paso a paso. La acción del algoritmo en cada paso consiste en conceder el permiso de acceso a un subconjunto de los participantes, determinando por el feedback en el presente paso junto al estado actual.

El siguiente paso será caracterizar el modelo multiacceso, es decir, el medio común a través del cual transmiten los usuarios.

### 3.2.1 Modelo de canal

El modelo idealizado a ser desarrollado nos permitirá enfocar el problema estandar que trata de la resolución de la contienda que ocurre cuando diversos terminales intentan usar el canal simultáneamente. Este modelo es el que tradicionalmente se ha tomado como base para el estudio de un sistema de acceso aleatorio. Ello no quita para que se estudien casos con ligeras modificaciones del modelo. Es muy usual, por ejemplo, abordar el estudio de la sensibilidad de las prestaciones del algoritmo a los errores en el feedback, por lo que se deja de cumplir la característica de colisión o recepción perfecta, como veremos más adelante. Canales cuyo modelo se corresponde con el expuesto aquí podrían ser un canal satélite en el cual éste actúa como transponder, una red radioterrestre donde un mínimo de terminales emplean frecuencias comunes, o un cable coaxial donde un número de terminales usan

un único canal para comunicarse mediante multiplexación en el tiempo. Las distintas características del modelo de canal son las siguientes.

**Sistema ranurado.** Un sistema es ranurado cuando el tiempo del canal es dividido en segmentos iguales, llamados ranuras temporales (*time slots*) o simplemente ranuras (*slots*). Todos los paquetes transmitidos tienen la misma longitud y coincide con la duración de una ranura, de tal forma que cada estación se sincroniza con el canal. Por lo tanto, aunque la fuente pueda generar en cualquier instante, transmite un solo paquete en una única ranura.

Según R. Gallager [49] tal sincronización no es difícil si se dan, en primer lugar, pequeños espacios de guarda entre paquetes, en segundo lugar, una pequeña cantidad de tiempo para el feedback del receptor y, en tercer lugar, relojes estables.

El tiempo se mide en unidades de ranura; es decir, la ranura  $t$  ocupa el intervalo temporal  $[t, t + 1)$ ,  $t \in \{0, 1, 2, \dots\}$ .

**Llegadas de Poisson.** Se supone que hay un infinito número de terminales idénticos e independientes que generan una tasa global  $\lambda$  paquetes por ranura. Además, el número de llegadas en una ranura es independiente de las llegadas en otras ranuras (no hay memoria). El modelo que más se aproxima a estas características es el proceso de Poisson. Cabe resaltar, entonces, que la probabilidad que dos paquetes lleguen en un mismo instante de tiempo es nula.

El modelo de Poisson también se puede deducir si se modela la generación de paquetes en cada uno de los usuarios mediante una distribución de Bernoulli, y se supone un número infinito.

De todo lo anterior, la probabilidad que se generen  $k$  paquetes en una ventana temporal  $\Delta$  dado que hay una tasa de  $\lambda$  paquetes por ranura será,

$$p(k|\Delta) = e^{-\lambda\Delta} \frac{(\lambda\Delta)^k}{k!} \quad (3.1)$$

Es muy importante resaltar que esta cantidad es independiente del estado del sistema y de su historia.

**Colisión o recepción perfecta.** Si más de un transmisor envía un paquete en una ranura, entonces interfieren entre ellos, se produce una colisión y el receptor no obtiene ninguna información acerca del contenido y origen de los paquetes transmitidos. Si sólo un transmisor ha enviado un paquete en una ranura, éste se recibe sin error y se dice que la transmisión ha sido exitosa.

El canal es libre de error; esto es, los errores sólo pueden ocurrir debido a colisión, donde las colisiones corresponden a transmisiones simultáneas de al menos dos paquetes. Debido a que el sistema es ranurado dos paquetes se solapan en su totalidad, o no se solapan en absoluto.

Según R. Gallager [49] esta característica es la que obvia tanto el ruido como los aspectos de comunicación del problema, que permite el estudio de la resolución de colisiones en el contexto más simple, pero limita severamente la clase de estrategias y compromisos que pueden ser considerados.

Según N. Mehravari y T. Berger [50], en esta característica se supone que la relación señal ruido (SNR) es suficientemente alta y/o se utilizan suficientes bits correctores de error que permiten que un único mensaje sin conflicto alcance su destino libre de error.

**Feedback inmediato y binario.** Al final de cada ranura, todos los usuarios reciben de forma instantánea información sobre lo que ha ocurrido en esa ranura; por ejemplo, si ha habido colisión o no. Ésta es la única información que cada transmisor obtiene acerca de la existencia de los paquetes. Según el feedback recibido los usuarios envueltos en una colisión actuarán de un modo u otro, siguiendo las reglas del CRA.

De aquí en adelante denotaremos el feedback que corresponde a la ranura  $t$  con  $x_t$ .

Según R. Gallager [49], la suposición de tomar un feedback inmediato es a menudo poco realista, pero los algoritmos de resolución de colisiones pueden ser modificados normalmente para tratar con feedbacks retardados. Ello complica el análisis sin un gran beneficio en los resultados.

Existen muchos tipos de feedback. Básicamente hay dos con un uso generalizado,

1. Feedback binario. A través de este feedback el transmisor puede distinguir una colisión ( $x_t = C$ ) de una no colisión ( $x_t = NC$ ).
2. Feedback ternario. El transmisor puede saber si la ranura estaba vacía (*hole*) ( $x_t = 0$ ), o contenía un único paquete (*success*) ( $x_t = 1$ ), o contenía más de un paquete (*collision*) ( $x_t = e$ ).

**Retransmisión de colisiones.** Una vez que un transmisor haya enviado un mensaje, éste no saldrá del sistema hasta que no sea exitosamente transmitido. Por tanto, en caso de una transmisión no exitosa debida a una colisión, se hará necesaria una retransmisión.

**Infinito número de terminales.** Se supone que el sistema está compuesto por un gran número de terminales, efectivamente infinito. De esta forma, cada paquete entrante llega a un transmisor que nunca ha recibido un paquete. Esto evita colas en los transmisores individuales y descarta el uso del TDM. Según R. Gallager [49] esta es una suposición poco razonable desde un punto de vista práctico pero cabe resaltar que dado un algoritmo, éste puede ser aplicado a un conjunto finito de terminales, relacionando cada llegada de paquete a un transmisor conceptualmente distinto. En este caso, físicamente podríamos estar en la situación que varios paquetes de un mismo transmisor estuvieran en contienda. Esto demuestra primero, que esta suposición proporciona el caso peor límite en un conjunto de usuarios finitos y, segundo, que la diferencia es solo significativa cuando son utilizados principalmente para tasas de entrada baja, donde hay una probabilidad relativamente baja que un transmisor tenga dos o más paquetes a transmitir. En este caso, las prestaciones en el caso de población infinita será una buena aproximación. El throughput máximo de un algoritmo bajo el supuesto de población infinita nos proporciona una medida cualitativa de la bondad del algoritmo.

### 3.3 Protocolo ALOHA estabilizado de ventana (WSA)

Una vez estructurado el estudio del algoritmo, vamos a pasar a la especificación concreta tanto del protocolo de acceso al canal utilizado, como del algoritmo de resolución de colisiones del protocolo ALOHA estabilizado de ventana (WSA).

#### 3.3.1 Protocolo de acceso al canal

La operación del protocolo para la primera transmisión que vamos a considerar en nuestro algoritmo es el mecanismo de ventana, sugerido por R. Gallager [20], por B. Tsybakov y V. Mikhailov [21] y por J. Massey [19], que pertenece al grupo de mecanismos de acceso bloqueado, según la clasificación dada por P. Mathys y P. Flajolet [37]. Los sistemas de acceso aleatorio que utilizan este protocolo de acceso al canal han demostrado ser los que tienen mejores prestaciones.

En este mecanismo, los nuevos paquetes son inhibidos de ser transmitidos durante la resolución de una contienda. Aún cuando ésta se ha completado, sólo un subconjunto de paquetes en espera pueden ser transmitidos. Este subconjunto está formado por todos aquellos paquetes que llegaron dentro de un intervalo temporal de longitud máxima  $\Delta$ , que llamaremos ventana

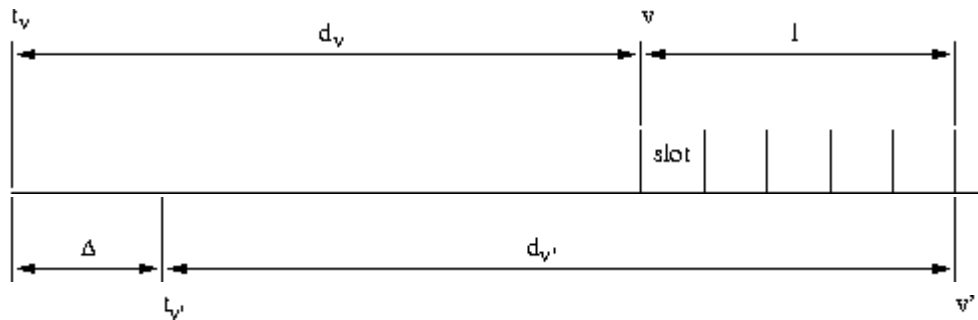


Figura 3.1: Ilustración de las relaciones entre las distintas variables relacionadas con la operación del algoritmo WSA (protocolo ALOHA estabilizado de ventana).

temporal. La ventana temporal  $\Delta$  será, pues, un parámetro del algoritmo que se escogerá de tal forma que maximice el throughput, y será tal que el número esperado de llegadas será aproximadamente 1.

El tiempo se mide en unidades de ranura; es decir, la ranura  $t$  ocupa el intervalo de tiempo  $[t, t + 1)$ . Sea  $v$  un instante de tiempo que corresponde al inicio de una ranura tal que, para cualquier  $t_v < v$ , todas las llegadas de paquetes en  $[0, t_v)$  han sido transmitidas exitosamente por el algoritmo y no se dispone de ninguna información referente al intervalo de llegada  $(t_v, v]$ , porque no han intentado transmitir nunca. El instante  $v$  se llama entonces punto de resolución de colisiones (*collision resolution point*, CRP), el intervalo de llegada  $[0, t_v)$  se llama el intervalo resuelto (*resolved interval*) y el intervalo  $(t_v, v]$  se llama el retraso (*lag*) en  $v$ . En la ranura  $v$  las llegadas de paquetes en  $(t_v, t_{v'} = t_v + \min(t_v + \Delta, v))$  intentan la transmisión, y el intervalo de llegada entonces se llama el intervalo examinado (*examined interval*). Si  $(t_v, t_{v'}$  contiene como mucho un paquete, entonces se resuelve en  $v$ , si  $(t_v, t_{v'}$  contiene al menos dos paquetes, entonces  $x_v = C$ , se produce una colisión en  $v$ , y su resolución empieza con el slot  $v + 1$ . Hasta que la colisión en  $v$  no sea resuelta, ningún paquete que llegue después de  $t_{v'}$  tiene permiso para transmitir. En la figura 3.1 se ilustran los tiempos  $t_v$ ,  $t_{v'}$  y  $v$ . En la figura 3.2 hay un ejemplo donde se especifican los tiempos  $t_v$ ,  $t_{v'}$  y  $v$  sobre un caso concreto.

Es de subrayar en este punto que mediante la aplicación de este protocolo de acceso al canal, en cada CRP tenemos un número aleatorio de paquetes en contienda, que dependerá única y exclusivamente del proceso de llegadas, en este caso de Poisson, y de la ventana temporal  $\Delta$ . Este mecanismo trata, pues, de desacoplar el número de paquetes en contienda al inicio de un CRI

y la longitud de los CRIs precedentes, que dependerá del CRA utilizado.

Este aparente contrasentido se resolverá más adelante cuando la ventana  $\Delta$  del CRP y la duración del CRI del CRA tendrán que cumplir unos requisitos mínimos para la estabilidad del sistema de acceso aleatorio.

### 3.3.2 Algoritmo de resolución de colisiones

La forma más usual de describir un CRA es mediante la explicación paso a paso de su evolución, dependiendo del feedback ( $x_t$ ) que reciben los distintos terminales, como razonan en su trabajo G. Polyzos, M. Molle y A. Venetsanopoulos [48].

El período de tiempo requerido para la resolución de la última colisión se llama intervalo de resolución de colisiones (CRI). Si el intervalo examinado contiene como mucho un paquete, entonces se resuelve en  $v$  y el CRI dura, por tanto, una ranura; si el intervalo examinado contiene al menos dos paquetes, entonces  $x_v = C$ , ocurre una colisión en  $v$ , y su resolución empieza con la ranura  $v + 1$ . Hasta que la colisión en  $t$  no sea resuelta, ningún paquete que llegue después de  $t_v$  tiene permiso para transmitir. Durante los CRI que empiezan con una ranura de colisión, cada usuario actúa independientemente vía la utilización de un contador cuyo valor en el tiempo  $t$  se denota por  $r_t$ . Este contador tiene el mismo valor para todos los usuarios, pero es cada usuario quien actualiza el suyo propio dependiendo del feedback. Cuando un usuario transmite por primera vez, fija  $r_t = r_{t_0} = 1$ . Los valores del contador son enteros, y son actualizados y utilizados de acuerdo con las reglas siguientes,

1. Los usuarios transmiten en la ranura  $t$  con probabilidad  $1/r_t$ .
2. Los valores del contador cambian con el tiempo de la manera siguiente:
  - (a) Si  $x_{t-1} = NC$ , entonces  $r_t = r_{t-1} - 1$
  - (b) Si  $x_{t-1} = C$ , entonces  $r_t = r_{t-1} + 1$

De estas reglas se puede ver que un CRI acaba con una ranura de no colisión y  $r_t = 1$ . En la figura 3.2 hay representado gráficamente un ejemplo del funcionamiento normal del algoritmo.

A pesar de las similitudes entre este algoritmo y el de M. Paterakis y P. Papantoni-Kazakos [27], se ha de resaltar que el algoritmo propuesto no pertenece a los algoritmos *splitting*. En el algoritmo propuesto todos los usuarios tienen la misma probabilidad de acceder y esta probabilidad se modifica dinámicamente hasta que sólo accede un paquete, mientras que

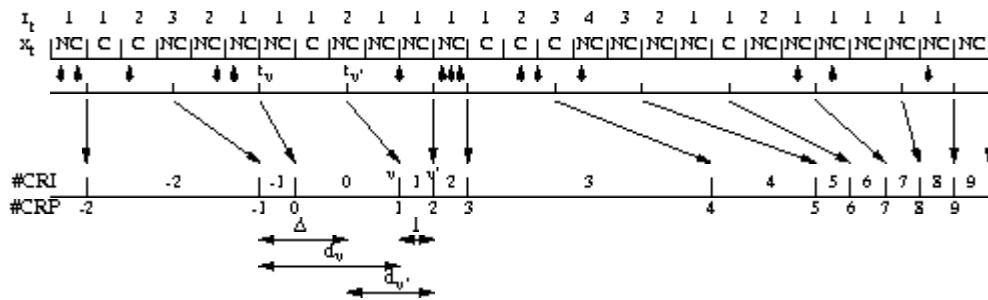


Figura 3.2: Ejemplo del funcionamiento del protocolo ALOHA estabilizado de ventana.

en los algoritmos *splitting* los usuarios se van distribuyendo en diferentes conjuntos, cada vez más pequeños, hasta que el conjunto sea compuesto sólo de un paquete.

El CRA que hemos pasado a presentar está inspirado por el trabajo de R. Rivest [42]. En él se estudia una estabilización del ALOHA decrementando las probabilidades de acceso en caso de colisión, e incrementándolo en el caso de un éxito y/o una colisión. De ahí, por tanto, el nombre de ALOHA estabilizado de ventana.

Pero, según él mismo, esta idea ya no era nueva. Por ejemplo M. Gerla y L. Kleinrock [51] tratan sobre distintas estrategias adaptativas para la red S-ALOHA, algunas de las cuales no distinguían entre paquetes nuevos y retrasados (*backlogged*) y que pueden ser sensibles a la congestión del canal observado. B. Ann y E. Gelenbe [52] trataron otras variantes, incluyendo una muy similar a la del mismo R. Rivest, éstas últimas políticas de control utilizan aproximaciones a los estimadores del retraso del canal encontrados por A. Segall [53]. D. Seret y C. Macchi [54] utilizan en cambio políticas basadas en el tráfico de canal. Por último mencionar el trabajo de S. Lam y L. Kleinrock [55], que introdujeron un algoritmo de control de la retransmisión que estaba basado en el tráfico estimado obtenido viendo una ventana de ranuras temporales.

El trabajo de R. Rivest [42] sugiere la siguiente estrategia. Al principio de la ranura  $t$ , cada estación tiene disponible la misma estimación  $\hat{n}_t$  de  $n_t$ , el número de terminales activos en el sistema. Cada estación con un paquete disponible intenta la transmisión con probabilidad  $1/\hat{n}_t$ . Condicionado a  $\hat{n}_t$ , las decisiones de las diferentes estaciones son estadísticamente independientes e independientes de otros eventos que han ocurrido en el pasado. Vamos a comprobar hasta qué punto son similares las técnicas de estabilización del ALOHA al CRA aquí presentado.

Según el trabajo de G. Cunningham [56], muchas de las técnicas de estabilización del ALOHA utilizan un feedback vacío ( $x_t = 0$ ), éxito ( $x_t = 1$ ) y colisión ( $x_t \geq 2$ ) para computar una estimación del número de terminales bloqueados  $\hat{n}$  de la ranura  $t$  a la ranura  $t + 1$  usando la ecuación,

$$\hat{n}_{t+1} = \max(\hat{n}_{\min}, \hat{n}_t + u_0 I[x_t = 0] + u_1 I[x_t = 1] + u_c I[x_t \geq 2]) \quad (3.2)$$

donde  $\hat{n}_{\min}$  es el mínimo valor permisible de  $\hat{n}$ .  $I$  es la función indicador y  $u_0, u_1, u_2$  son parámetros de control. Técnicas de este estilo son estudiadas por V. Mikhailov [57], L. Clare [40], R. Rivest [42] y S. Thomopoulos [41]. Por tanto, cada una de estas estabilizaciones tiene un conjunto  $u_0, u_1$  y  $u_2$  específicos.

Además, L. Clare [40] proporciona un método para probar que un sistema es estable para todo  $\lambda < 1/e$ , para un conjunto de parámetros ( $u_0, u_1$  y  $u_2$ ), si satisfacen la ecuación,

$$u_0 + u_1 + u_c(e - 2) = 0 \quad (3.3)$$

con  $u_c > 0$  y  $u_c \leq 0$ .

En el caso del CRA presentado, se observa que si escogemos una ventana  $\Delta = 1$ , nos encontramos ante un método de estabilización del ALOHA. Por consiguiente, el conjunto de parámetros será  $u_0 = -1, u_1 = -1$  y  $u_2 = 1$  y aplicando la fórmula de L. Clare:  $-1 - 1 + 1(e - 2) \neq 0$  resulta que no es estable. Este resultado es del todo coherente porque, como veremos posteriormente, para  $\Delta = 1$  nuestro algoritmo es inestable.

Después del siguiente razonamiento, es obvio el porqué del nombre del algoritmo presentado: ALOHA estabilizado de ventana (WSA).

La primera vez que el protocolo WSA salió a la luz pública fue en el trabajo V. Casares y el autor [58]. En él, el CRA del protocolo WSA formaba parte de un algoritmo aleatorio de reserva (RRA) de una aplicación de integración de datos que será descrita con más profundidad en el Capítulo 11.

Aproximadamente en la misma época aparece publicado un trabajo de S. Jangi y L. Merakos [34] donde utilizan en las simulaciones que presentan un algoritmo aleatorio de reserva llamado ALOHA controlado. Este es un algoritmo irrealizable porque los terminales en contienda acceden con una probabilidad el inverso del total de contendientes. En otro trabajo, esta vez de M. Paterakis y A. Cleary [35] se va un poco más allá y describen, además del ALOHA controlado ideal, basado en S. Jangi y L. Merakos [34] una versión implementable del ALOHA controlado, con unas reglas similares al CRA del WSA, pero partiendo de una probabilidad inicial distinta de 1.



La gran diferencia entre los trabajos de V. Casares y el autor [58], y los de S. Jangi y L. Merakos [34] y de M. Paterakis y A. Cleary [35] es que en el primero se hace uso en cierta medida del protocolo de acceso bloqueado, es decir, no se permite el acceso de nuevos usuarios hasta finalizar la contienda; mientras que en los segundos simplemente se reducen a técnicas de estabilización del ALOHA con protocolo de acceso libre, ya planteadas y estudiadas con profundidad anteriormente.

Todos ellos tienen en común que estudian mediante simulaciones algoritmos aleatorios de reserva variantes de estabilizaciones del ALOHA. Ello es totalmente distinto del trabajo realizado en la presente tesis, en que se estudia analíticamente un sistema de acceso aleatorio, obteniéndose parámetros como el throughput y el retardo.

# Capítulo 4

## Análisis del throughput, con errores en el feedback, caso feedback binario

### 4.1 Introducción

En este Capítulo vamos a llevar a cabo el análisis del throughput del algoritmo WSA (ALOHA estabilizado de ventana) en su versión binaria. Para ello, el primer paso será definirlo.

En la literatura especializada hay numerosas definiciones de throughput de un algoritmo. Para ver una serie de ellas un excelente trabajo es el de B. Tsybakov [59]. Muchas de estas definiciones son equivalentes, pero nosotros nos hemos decidido por utilizar la definición más comúnmente usada.

El throughput de un algoritmo se define, pues, como el supremo de todas las tasas de llegada de Poisson que el algoritmo es capaz de absorber con un retardo esperado finito para una longitud fijada de ventana temporal  $\Delta$ . Esta definición puede ser encontrada, entre otros, en el trabajo de J. Huang y T. Berger [22], el de B. Tsybakov [59] (definición de  $R_1$ ) y en el de L. Merakos y C. Bisdikian [60].

### 4.2 Análisis del throughput

El hecho que el protocolo de acceso al canal sea un protocolo de acceso bloqueado, concretamente el mecanismo de ventana, nos permite utilizar una metodología muy elegante. Esta metodología puede ser encontrada en diversos trabajos, como el de M. Paterakis y P. Papantoni-Kazakos [27].

Vamos a considerar el modelo de sistema y algoritmo del Capítulo anterior. El sistema empieza operando en el tiempo cero. Sea  $v_i, i \geq 1$ , la secuencia de sucesivos CRP, y sea  $d_v$  el retraso en  $v$ . La secuencia  $d_i, i \geq 1$ , es una cadena de Markov con el espacio de estados  $F$ . Si  $\Delta$  es racional, entonces  $F$  es como mucho un conjunto denumerable de  $[1, \infty)$ . La condición de ergodicidad obtenida por J. Huang y T. Berger [22], en su Teorema 1, indica que la cadena de Markov es ergódica y el sistema es estable si y sólo si,

$$E(i|\Delta) < \Delta \tag{4.1}$$

donde  $E(i|\Delta)$  denota la longitud esperada de un CRI dado que empieza con un intervalo examinado de longitud  $\Delta$ .

Cabe exponer en este punto una reflexión intuitiva de lo que significa la desigualdad (4.1). Podemos decir que ésta es la piedra angular que relaciona las dos partes en que dividimos a un sistema de acceso aleatorio. Por una parte, el protocolo de acceso al canal nos proporciona un parámetro, la ventana temporal de longitud  $\Delta$ . Este parámetro es en principio un grado de libertad que escogemos de tal forma que maximice el throughput. Por otra, y desde un punto de vista funcional, dada una  $\Delta$ , el comportamiento del protocolo de acceso al canal es independiente del algoritmo de resolución de colisiones, y del estado en que se encuentre el sistema. Ello evita, sin ir más lejos, el catastrófico efecto avalancha que sufre el protocolo ALOHA y que lo hace inherentemente inestable.

Siguiendo con la línea marcada por el citado Teorema 1, existe una tasa de llegadas crítica  $\lambda^*$  para la cual se cumple,

$$E(i|\Delta) = \Delta \tag{4.2}$$

Así, cualquier tasa  $\lambda < \lambda^*$  es alcanzable por el algoritmo, mientras que, para cualquier tasa  $\lambda \geq \lambda^*$  el algoritmo es inestable. Por consiguiente, dada una ventana  $\Delta$ ,  $\lambda^*$  es el supremo de todas las tasas alcanzables por el algoritmo.

Si denotamos como  $E_k(i)$  la longitud esperada de un CRI dado que empieza con una colisión de multiplicidad  $k$ , podemos entonces escribir,

$$E(i|\Delta) = \sum_{k=0}^{\infty} E(i|\Delta, k)p(k|\Delta) = \sum_{k=0}^{\infty} E_k(i)p(k|\Delta) \tag{4.3}$$

ya que,

$$E(i|\Delta, k) = E_k(i) \tag{4.4}$$

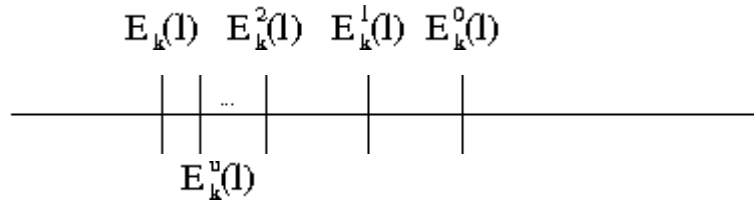


Figura 4.1: Representación gráfica de la obtención de  $E_k^u(l)$  y  $E_k^0(l)$ .

depende sólo de  $k$ , y donde,

$$p(k|\Delta) = e^{-\lambda\Delta} \frac{(\lambda\Delta)^k}{k!} \quad (4.5)$$

En el Apéndice de este Capítulo mostraremos cómo,

1.  $E_k(l)$  está acotada cuadráticamente,  $E_k(l) \leq E_k^0(l) = c_2 k^2 + c_1 k + c_0$ , ver Sección 4.4.3 del Apéndice.
2.  $E_k(l)$  puede ser recursiva y superiormente acotada de forma ajustada,  $E_k(l) \leq E_k^u(l)$ , ver Sección 4.4.4 del Apéndice.

Como veremos más adelante,  $E_k(l)$  es el límite de una secuencia monótona no creciente  $\{E_k^n(l); n \geq 0\}$ , donde  $E_k^0(l)$  es el primer elemento de la serie, y  $E_k^u(l)$  es un elemento con  $n$  suficientemente alto.

La expresión (4.1) junto con los resultados de los puntos 1 y 2 son usados en la computación del throughput del algoritmo. El procedimiento es como sigue.

Debido a la cota superior de  $E_k(l)$  y a la inecuación (4.1), concluimos que la siguiente condición es suficiente para la estabilidad,

$$E^u(l|\Delta) < \Delta \quad (4.6)$$

con,

$$E^u(l|\Delta) = \sum_{k=0}^{30} E_k^u(l)p(k|\Delta) + \sum_{k=31}^{\infty} E_k^0(l)p(k|\Delta) \quad (4.7)$$

Es decir,

$$E^u(l|\Delta) < \Delta \Rightarrow E(l|\Delta) < \Delta \quad (4.8)$$

ya que,

$$E^u(l|\Delta) \geq E(l|\Delta) \quad (4.9)$$

Después de algunas manipulaciones algebraicas, se obtiene que (4.7) es equivalente a,

$$\begin{aligned}
E^u(i|\Delta) &= \sum_{k=0}^{30} E_k^u(i) p(k|\Delta) + \alpha \left\{ (\lambda\Delta)^2 + \lambda\Delta - \sum_{k=0}^{30} k^2 p(k|\Delta) \right\} \\
&+ (\rho r_{t_0} + \mu) \left\{ \lambda\Delta - \sum_{k=0}^{30} k p(k|\Delta) \right\} \\
&+ (\beta r_{t_0}^2 + \eta r_{t_0} + \gamma) \left\{ 1 - \sum_{k=0}^{30} p(k|\Delta) \right\}
\end{aligned} \tag{4.10}$$

ya que,

$$E_k^0(i) = G_{k,r_{t_0}}^0 = \alpha k^2 + \beta r_{t_0}^2 + \rho k r_{t_0} + \mu k + \eta r_{t_0} + \gamma \tag{4.11}$$

y particularizando para  $r_{t_0} = 1$ ,

$$E_k^0(i) = G_{k,1}^0 = \alpha k^2 + (\rho + \mu)k + (\beta + \eta + \gamma) \tag{4.12}$$

Una vez calculado  $E^u(i|\Delta)$ , vamos a obtener una expresión para el cálculo de  $\lambda^*$ . Para ello definimos,

$$E^u(i|\Delta) \equiv f(\lambda\Delta) \tag{4.13}$$

$$x \equiv \lambda\Delta \tag{4.14}$$

Por consiguiente, sustituyendo las expresiones (4.13) y (4.14) en la ecuación (4.1), se concluye que, para la estabilidad del algoritmo, es suficiente que la tasa de entrada  $\lambda$  satisfaga la siguiente inecuación,

$$\lambda < \sup_{x \geq 0} \frac{x}{f(x)} \tag{4.15}$$

La siguiente condición especifica una región de valores de  $\lambda$  para los cuales el algoritmo es inestable,

$$\lambda > \sup_{x \geq 0} \frac{x}{g(x)} \tag{4.16}$$

donde,

$$g(x) = \sum_{k=0}^{30} E_k^u(i) e^{-x} \frac{x^k}{k!} \tag{4.17}$$

Algoritmo WSA	$\lambda^* = 0.4335$	$\Delta^* = 2.49$
Algoritmo TCW	$\lambda^* = 0.4295$	$\Delta^* = 2.33$
Algoritmo DBT	$\lambda^* = 0.4295$	$\Delta^* = 2.677$

Tabla 4.1: Throughputs y tamaños de ventana óptimos. Los algoritmos representados son el WSA (ALOHA estabilizado de ventana), el algoritmo TCW (*two cell window*) de M. Paterakis y P. Papantoni-Kazakos y el algoritmo DBT (*dynamic binary tree*) de J. Capetanakis.

La maximización de las expresiones (4.15) y (4.16) ha sido hecha numéricamente y proporciona el throughput ( $\lambda^*$ ) así como el tamaño de ventana óptimo  $\Delta^*$ . En todos los casos el orden de magnitud de la diferencia es menor que  $10^{-4}$ . El tamaño de ventana óptimo se encuentra como  $x^*(\lambda^*)^{-1}$ , donde  $x^*$  es el valor que alcanza el supremo en la expresión (4.15). El resultado obtenido se refleja en la tabla 4.1.

Además, como prueban en su trabajo M. Paterakis, L. Georgiadis, y P. Papantoni-Kazakos [61], para una gran clase de algoritmos de acceso aleatorio, entre los que se encuentra el aquí presentado, a medida que el tamaño de la población aumenta la estabilidad de un algoritmo es determinada por su throughput bajo el modelo de usuario de Poisson. Ello corrobora la decisión de considerar el proceso de Poisson como el proceso de llegadas.

Hay que resaltar aquí que el método es válido para cualquier sistema de acceso aleatorio que utilice nuestro protocolo de acceso al canal. Éste es, pues, un camino válido para la comparación de diversos algoritmos de resolución de colisiones. Cada uno de ellos tendrán su propia  $E(i|\Delta)$  y, por tanto, tendrán su propio throughput.

### 4.3 Sensibilidad a los errores en el feedback

En base al modelo de canal considerado en el Capítulo anterior no hay posibilidades de error en la transmisión a no ser que haya una colisión. Éste es un aspecto poco realista y por ello, a lo largo de la historia, se ha intentado introducir pequeñas variaciones en el modelo para reflejar el efecto del ruido en las prestaciones del algoritmo. Es obvio que un algoritmo con una baja sensibilidad al ruido será mejor que otros con una sensibilidad mayor. Por otra parte, está claro que el hecho de complicar el modelo, dificulta más si cabe su análisis, hasta el punto que en muchos algoritmos éste es un problema que aún hoy permanece abierto.

En lo que se refiere al protocolo WSA, vamos a estudiarlo analíticamente por completo.

Históricamente, los trabajos más importantes sobre el tratamiento de errores de ruido son los de J. Massey [19], de N. Vvedenskaya y B. Tsybakov [62], y el de D. Ryter [63]. En los trabajos de J. Massey [19] y N. Vvedenskaya y B. Tsybakov [62] el algoritmo propuesto está basado en los algoritmos de árbol binario de J. Capetanakis [15] y B. Tsybakov y V. Mikhailov [16], mientras que en el de D. Ryter [63] está basado en el algoritmo 0.487 de R. Gallager [20], [49] y de B. Tsybakov y V. Mikhailov [21] y de G. Ruget [64]. Cabe destacar que D. Ryter tuvo que modificar el algoritmo de R. Gallager para que no cayera en situación inestable.

Primero describiremos el modelo con errores en el feedback que ha sido el más comúnmente utilizado. Éste va a ser un modelo de error probabilístico, sin memoria y simétrico, es decir, todos los usuarios del sistema escuchan el mismo error debido a ruido con unas probabilidades concretas. Este modelo es utilizado por J. Massey [19], y por N. Vvedenskaya y B. Tsybakov [62].

Debido a condiciones de ruido, pueden ocurrir los siguientes tipos de errores en el feedback. Con probabilidad  $\epsilon$  una ranura vacía puede ser vista por los usuarios como una ranura de colisión. También, con probabilidad  $\delta$  una ranura ocupada por una transmisión sencilla puede ser vista por los usuarios como una ranura de colisión. Asumimos que una ranura de colisión es siempre reconocida correctamente por los usuarios. Consideramos el caso donde las probabilidades  $\epsilon$  y  $\delta$ , siendo características del sistema, son conocidas a priori. Entonces dados  $\epsilon$  y  $\delta$ , el tamaño de la ventana  $\Delta$  es optimizado para la maximización del throughput, y además se obtiene una  $E(I|\Delta)$  distinta para cada par  $(\epsilon, \delta)$ . Expondremos los resultados del análisis del throughput tanto para el algoritmo WSA (ALOHA estabilizado de ventana), como para el algoritmo TCW (*two cell window*) de M. Paterakis y P. Papantoni-Kazakos [27] y para el algoritmo DBT (*dynamic binary tree*) de J. Capetanakis [15]. Los resultados son expuestos en la tabla 4.2. De la tabla 4.2 la primera conclusión que se extrae es que el propuesto algoritmo es muy insensible a los errores en el feedback.

En una situación de acceso aleatoria real L. Merakos y D. Kazakos [43] anticipan la inecuación  $\epsilon < \delta \ll 1$ . Incluso para el caso considerado extremo por M. Paterakis y P. Papantoni-Kazakos [27],  $\epsilon = \delta = 0.1$ , el throughput es más del 90 por ciento de su valor en el caso libre de error. Sin embargo, la sensibilidad respecto a los errores en el feedback es mayor en el algoritmo WSA que en el algoritmo TCW. Notamos que si  $\epsilon \geq 0.5$ , el throughput para el algoritmo WSA y para el algoritmo DBT es cero, mientras que para el algoritmo TCW es mayor que cero. En cualquier caso estas situaciones están lejos del caso extremo  $\epsilon = \delta = 0.1$ , en el cual el algoritmo WSA consigue un mayor throughput que los algoritmos TCW y DBT.

Para justificar el modelo seguido, es útil la nomenclatura que utilizan

$\epsilon$	$\delta$	Alg. WSA	Alg. TCW	Alg. DBT
0.00	0.00	0.4335	0.4295	0.4295
0.00	0.01	0.4292	0.4248	0.4258
0.00	0.10	0.3904	0.3873	0.3920
0.00	0.20	0.3473	0.3463	0.3535
0.00	0.40	0.2610	0.2655	0.2731
0.00	0.50	0.2179	0.2251	0.2310
0.01	0.00	0.4321	0.4272	0.4272
0.10	0.00	0.4173	0.4117	0.4043
0.20	0.00	0.3940	0.3930	0.3706
0.40	0.00	0.2814	0.3503	0.2329
0.45	0.00	0.2006	0.3382	0.1524
0.10	0.10	0.3733	0.3706	0.3672
0.20	0.20	0.3050	0.3139	0.2972
0.30	0.30	0.2252	0.2589	0.2166
0.40	0.30	0.1591	0.2064	0.1205
0.30	0.50	0.1508	0.1885	0.1511
0.30	0.70	0.0854	0.1183	0.0886

Tabla 4.2: Throughputs en función de  $\epsilon$  y  $\delta$ . Los tamaños de ventana son optimizados para cada par  $(\epsilon, \delta)$ . Los algoritmos representados son el WSA (ALOHA estabilizado de ventana), el algoritmo TCW (*two cell window*) de M. Paterakis y P. Papantoni-Kazakos y el algoritmo DBT (*dynamic binary tree*) de J. Capetanakis.

L. Merakos y D. Kazakos [43], en la que  $P_{0\epsilon} = \epsilon$  y  $P_{1\epsilon} = \delta$ .

La decisión de considerar sólo no nulas las probabilidades que tanto una ranura vacía como exitosa pueden ser tomadas como colisiones ( $P_{0\epsilon}$  y  $P_{1\epsilon}$ ) no es arbitraria. Un argumento aducido podría ser que bajo una operación normal de la red,  $P_{0\epsilon}$  y  $P_{1\epsilon}$  son los errores más probables que pueden ocurrir. Como ya apunta J. Massey [19], es improbable que el ruido fuera interpretado como un único paquete ( $P_{0\epsilon}$ ) porque, en la práctica, cada paquete transmitido sería codificado con un código detector de error suficientemente potente. Por la misma razón es improbable que una colisión fuera interpretada como un único paquete ( $P_{1\epsilon}$ ) a no ser que el sistema esté operando en una red de comunicaciones seguras por clave pública, donde cada usuario, fuera de los que toman parte, no puede distinguir de forma fiable un único paquete de una colisión. Un argumento basado en la relación señal a ruido se puede usar también para justificar que es muy improbable que uno o más paquetes no fueran detectados (borrado) ( $P_{10}$  y  $P_{\epsilon 0}$ ). Esto sólo no es cierto en sistemas de



acceso aleatorio con propósitos especiales, como el empleado en las técnicas de modulación *spread spectrum*, donde los usuarios no pueden distinguir de forma fiable el ruido de canal del ruido de colisión porque intentan disimular el hecho que están comunicando manteniendo la potencia transmitida baja.

Por otra parte, una ranura vacía podría, a causa del ruido en el canal de transmisión, llegar a los usuarios como algo que están forzados a interpretar como colisión ( $P_{0e}$ ). Además una ranura con un único paquete podría acabar en una interpretación de colisión a causa de los errores de bit detectados en el paquete codificado en un canal ruidoso. Esta probabilidad ( $P_{1e}$ ) podría ser decrementada significativamente si se empleara un código corrector de errores, más que un código detector de errores.

En este punto cabe señalar lo siguiente. El modelo que acabamos de describir es el más sencillo que históricamente se ha venido utilizando. El objetivo de un análisis de este tipo es conocer mediante una primera aproximación la sensibilidad del algoritmo frente a los errores en el feedback. No obstante, hay algoritmos en los cuales la aplicación de este modelo ha forzado una variación de sus reglas, como el caso de D. Ryter [63].

Otros autores como A. Cleary y M. Paterakis [36] han estudiado algoritmos aplicándoles un modelo de error con memoria. Hasta ahora los errores eran modelados con un proceso de Bernoulli, es decir, los errores ocurrían independientemente de una ranura a otra. En los modelos de error con memoria se utiliza una cadena de Markov de dos estados (bueno y malo) que describe la condición del canal. De esta forma, tenemos un modelo de errores a ráfagas.

En la presente tesis hemos considerado el modelo de error sin memoria, porque el objetivo primero era el estudio de la sensibilidad frente a los errores en el feedback que nos llevan a conclusiones claras y sencillas, pudiendo ser comparados con las de otros algoritmos.

En algunos sistemas, las probabilidades  $\epsilon$  y  $\delta$  pueden no ser conocidas a priori. En este caso, un algoritmo puede ser diseñado sujeto a la suposición de feedback libre de error, y posteriormente se puede encontrar la correspondiente ventana temporal óptima ( $\Delta^*$ ), y a partir de ahí calcular los correspondientes throughputs ( $\lambda^*$ ). En el caso que nos concierne aquí, hemos encontrado los throughputs para el algoritmo WSA, el algoritmo TCW y el algoritmo DBT para varios valores de las probabilidades de error, que son las máximas tasas de Poisson para los cuales la condición (4.1) se satisface sujeto a la restricción que las ventanas son aquellas de la tabla 4.1. Nuestros resultados están expuestos en la tabla 4.3. De la tabla 4.3, observamos que el máximo valor para el que el algoritmo WSA sea estable es  $\epsilon = 0.32$ , mientras que para el algoritmo TCW es  $\epsilon = 0.378$  y para el algoritmo DBT es  $\epsilon = 0.33$ . Dentro del caso considerado real  $0 \leq \epsilon \leq 0.1$  y  $0 \leq \delta \leq 0.1$  el

$\epsilon$	$\delta$	Alg. WSA	Alg. TCW	Alg. DBT
0.00	0.00	0.433	0.429	0.429
0.00	0.01	0.429	0.424	0.410
0.00	0.10	0.389	0.386	0.391
0.00	0.40	0.250	0.255	0.262
0.01	0.00	0.432	0.427	0.427
0.10	0.00	0.414	0.408	0.401
0.20	0.00	0.375	0.377	0.355
0.30	0.00	0.259	0.320	0.248
0.10	0.10	0.365	0.363	0.361
0.10	0.20	0.317	0.318	0.318
0.10	0.30	0.268	0.273	0.272
0.20	0.10	0.319	0.328	0.311
0.20	0.20	0.265	0.279	0.266
0.20	0.30	0.212	0.232	0.221
0.30	0.10	0.173	0.261	0.192
0.30	0.20	0.085	0.205	0.138

Tabla 4.3: Throughputs en función de  $\epsilon$  y  $\delta$ . Los tamaños de ventana fijos para cada par  $(\epsilon, \delta)$ . Los algoritmos representados son el WSA (ALOHA estabilizado de ventana), el algoritmo TCW (*two cell window*) de M. Paterakis y P. Papantoni-Kazakos y el algoritmo DBT (*dynamic binary tree*) de J. Capetanakis.

algoritmo WSA consigue mayor throughput que el algoritmo TCW y DBT.

## 4.4 Apéndice

### 4.4.1 Análisis de estabilidad

En primer lugar vamos a definir qué se entiende por estabilidad de un algoritmo de acceso aleatorio. Para ello nos referimos a dos trabajos suficientemente significativos.

Según J. Capetanakis [15], un sistema se dice que es estable de orden  $k$  si los primeros  $k$  momentos del retardo de un paquete aleatoriamente escogido son finitos.

Para B. Tsybakov [59], un algoritmo de acceso aleatorio se dice que es inherentemente estable si su throughput es mayor que 0, e inherente inestable si su throughput es igual a 0.

Desde un punto de vista intuitivo no es difícil razonar la condición de

estabilidad para nuestro sistema. Si la longitud media de CRI, que es el tiempo de servicio correspondiente a una ventana  $\Delta$ , supera el tamaño de esta ventana, entonces el trabajo inacabado en el sistema, en media, se incrementará sin límite, haciendo el sistema inestable. Por el contrario, si la longitud media del CRI es menor que el tamaño de ventana  $\Delta$ , entonces escogiendo cualquier número suficientemente grande de tamaños de ventana, el sistema podrá procesar todos ellos.

Según los razonamientos anteriores, la condición de estabilidad viene dada por la inecuación (4.1). Matemáticamente ésta es la condición de estabilidad según B. Tsybakov [59]. En el cálculo del retardo del Capítulo siguiente veremos como esta condición aparece como necesaria en el cálculo del primer momento del retardo. Una vez calculado el retardo de paquete medio y comprobado que es finito, podemos asegurar que el sistema es estable de primer orden.

En la siguiente sección vamos a presentar el análisis del caso que puedan ocurrir errores en el feedback. Substituyendo  $\epsilon = \delta = 0$  en nuestros resultados, obtenemos las cantidades correspondientes en el caso libre de errores. La región de estabilidad del algoritmo proviene de la inecuación (4.1), donde el valor esperado en ella es dado por la expresión (4.3). Empezamos con el desarrollo de una cota superior ajustada en los valores esperados  $E_k(i)$ ,  $k \geq 0$ .

#### 4.4.2 Planteamiento del problema, $G_{k,i}$

En primer lugar definimos  $P_{k,i,s}$  como la probabilidad que se necesiten  $s$  ranuras por el algoritmo para la transmisión exitosa de  $k$  paquetes, dado que el valor del contador es igual a  $r_i = i$ .

De la operación del algoritmo se puede considerar una cadena de Markov que viene definida por el estado tridimensional  $(k, i, s)$  donde el proceso puede estar, siendo  $k = 0, 1, \dots$ ,  $i = 1, 2, \dots$ ,  $s = 0, 1, \dots$  aquellos referidos anteriormente.

Sean  $H_{k,i}$ ,  $S_{k,i}$  y  $C_{k,i}$  las probabilidades tales que, dados  $k$  paquetes en contienda, 0, 1, o más de 1 intentan transmitir, respectivamente, dado que el valor del contador es igual a  $r_i = i$ . Estas son dadas por,

$$H_{k,i} = \begin{cases} (1 - \frac{1}{i})^k & k > 0, i \geq 1 \\ 1 & k = 0, i \geq 1 \end{cases} \quad (4.18)$$

$$S_{k,i} = \begin{cases} \frac{k}{i} (1 - \frac{1}{i})^{k-1} & k \geq 0, i \geq 1 - \{(k = 0, i = 1), (k = 1, i = 1)\} \\ 1 & k = 1, i = 1 \\ 0 & k = 0, i = 1 \end{cases} \quad (4.19)$$

$$C_{k,i} = 1 - S_{k,i} - H_{k,i} \quad (4.20)$$

Observando el proceso de Markov, es lógico obtener que la distribución de probabilidad viene dada por la siguiente ecuación recursiva,

$$P_{k,i,s} = \begin{cases} (C_{k,i} + \epsilon H_{k,i} + \delta S_{k,i})P_{k,i+1,s-1} \\ + S_{k,i}(1 - \delta)P_{k-1,i-1,s-1} & k \geq 0, i \geq 1, s \geq 0 \\ + H_{k,i}(1 - \epsilon)P_{k,i-1,s-1} & \\ 0 & \text{en otro caso} \end{cases} \quad (4.21)$$

Tomando esperanzas en la expresión (4.21) respecto a  $s$ ,

$$G_{k,i} = \sum_{s=0}^{\infty} s P_{k,i,s} \quad (4.22)$$

y desarrollándolo, se obtiene,

$$G_{k,i} = \begin{cases} 1 + (C_{k,i} + \epsilon H_{k,i} + \delta S_{k,i})G_{k,i+1} \\ + S_{k,i}(1 - \delta)G_{k-1,i-1} & k \geq 0, i \geq 1 \\ + H_{k,i}(1 - \epsilon)G_{k,i-1} & \\ 0 & \text{en otro caso} \end{cases} \quad (4.23)$$

Siguiendo con esta línea, definimos  $G_{k,i}$  como el esperado número de ranuras necesarias por el algoritmo para la transmisión exitosa de  $k$  paquetes, dado que el valor del contador es igual a  $r_t = i$  en la primera ranura.

Cabe reseñar que  $E_k(t) = G_{k,r_{t_0}} = G_{k,1}$ .

#### 4.4.3 Desarrollo de una cota superior de $E_k(t)$ , $E_k^0(t)$

La ecuación (4.23) comprende un sistema numerable de ecuaciones lineales de la forma,

$$G_{k,i} = b_{k,i} + \sum_{q=0}^{\infty} \sum_{j=1}^{\infty} h_{q,j}^{k,i} G_{q,j} \quad (4.24)$$

para  $k, i \in \mathcal{I}$ , donde  $\mathcal{I}$  es un conjunto numerable de índices, y los coeficientes  $h_{q,j}^{k,i}$  y  $b_{k,i}$  son no negativos para cada  $k, i, q, j \in \mathcal{I}$ , donde,

$$b_{k,i} = \begin{cases} 1 & k \geq 0, i \geq 1 \\ 0 & \text{en otro caso} \end{cases} \quad (4.25)$$

y,

$$h_{qj}^{k,i} = \begin{cases} C_{k,i} + \epsilon H_{k,i} + \delta S_{k,i} & q = k, j = i + 1, k \geq 2, i \geq 1 \\ S_{k,i}(1 - \delta) & q = k - 1, j = i - 1, k \geq 2, i \geq 1 \\ H_{k,i}(1 - \epsilon) & q = k, j = i - 1, k \geq 2, i \geq 1 \\ 0 & \text{en otro caso} \end{cases} \quad (4.26)$$

Esta nomenclatura está inspirada en el trabajo de I. Stavrakakis y D. Kazakos [65].

Según el trabajo de L. Merakos y C. Bisdikian [60], los fundamentos de la teoría de sistemas de ecuaciones lineales de dimensión infinita, como el del sistema (4.24), fueron establecidos al final del siglo pasado, debido a sus aplicaciones en la solución de ecuaciones integrales y problemas de contorno de la física matemática.

En el libro de L. Kantorovich y V. Krylov [66] se pueden encontrar un buen número de resultados bastante generales sobre la existencia, unicidad y computación aproximada de la soluciones de tales sistemas. En el contexto de algoritmos de acceso aleatorio, los sistemas como el (4.24) fueron estudiados en primer lugar por B. Tsybakov y N. Vvedenskaya [67], y subsiguientemente en trabajos como los de N. Vvedenskaya y B. Tsybakov [62], de L. Georgiadis, L. Merakos, y P. Papantoni-Kazakos [68], de N. Vvedenskaya y B. Tsybakov [69], de W. Rosenkrantz [70], de P. Mathys y P. Flajolet [37] y de G. Fayolle, P. Flajolet, M. Hofri y P. Jacquet [18]. La aproximación a la solución del sistema de los últimos dos trabajos está basado en métodos de variable compleja, mientras que el resto está basado en técnicas de acotación, similares a aquellos desarrollados en el libro de L. Kantorovich y V. Krylov [66].

En nuestro caso también utilizaremos una técnica de acotación, que pasamos a describir brevemente, aplicándolo ya a nuestro caso particular. Este es paralelo al de L. Merakos y C. Bisdikian [60].

Procedemos ahora al desarrollo de una cota superior inicial de la solución del sistema (4.23),  $E_k^0(t) = G_{k,1}^0$ . Supongamos que existe una secuencia  $\{G_{k,i}^0; k, i \in \mathcal{I}\}$  tal que cumple los siguientes requisitos, para cualquier  $k, i \in \mathcal{I}$ ,

$$0 \leq G_{k,i}^0 < \infty \quad (4.27)$$

$$G_{k,i}^0 \geq b_{k,i} + \sum_{q=0}^{\infty} \sum_{j=1}^{\infty} h_{qj}^{k,i} G_{qj}^0 = G_{k,i}^1 \quad (4.28)$$

Definimos,

$$G_{k,i}^{n+1} = b_{k,i} + \sum_{q=0}^{\infty} \sum_{j=1}^{\infty} H_{q,j}^{k,i} G_{q,j}^n \quad (4.29)$$

para  $k, i \in \mathcal{I}$ ,  $n = 0, 1, 2, \dots$

Observamos que de las inecuaciones (4.27) y (4.28) y el hecho que  $H_{q,j}^{k,i}$  y  $b_{k,i}$  son no negativos para cada  $k, i, q, j \in \mathcal{I}$ , tenemos que,

$$0 \leq G_{k,i}^{n+1} \leq G_{k,i}^n < \infty \quad (4.30)$$

para cada  $k, i \in \mathcal{I}$  y cada  $n \geq 0$ .

*Lema 1.* Si una secuencia  $\{G_{k,i}^0; k, i \in \mathcal{I}\}$  existe que satisface las inecuaciones (4.27) y (4.28), entonces el sistema (4.24) tiene una solución finita no negativa  $\{G_{k,i}^*; k, i \in \mathcal{I}\}$  y  $G_{k,i}^*$  es el límite de la secuencia monótona no creciente  $\{G_{q,j}^n; n \geq 0\}$ , donde  $G_{k,i}^n$  se define en (4.29).

La principal dificultad en encontrar una forma apropiada de la secuencia  $G_{k,i}^0$  radica en que tiene que ser supuesta, según L. Merakos y C. Bisdikian [60]. Afortunadamente, para el sistema (4.23) se pueden usar secuencias cuadráticas de la forma,

$$G_{k,i}^0 = \begin{cases} \alpha k^2 + \beta i^2 + \rho ki + \mu k + \eta i + \gamma & k \geq 2, i \geq 1 \\ \frac{i}{1-2\epsilon} & k = 0, i \geq 1 \\ \frac{1}{1-2\epsilon} \left( i + \frac{2(\delta-\epsilon)}{1-\delta} \right) & k = 1, i \geq 1 \\ 0 & \text{en otro caso} \end{cases} \quad (4.31)$$

Estas secuencias tienen que satisfacer las inecuaciones (4.27) y (4.28), es decir, se tienen que determinar  $\alpha, \beta, \rho, \mu, \eta, \gamma$ .

Para  $n = 0$  en la expresión (4.29),

$$G_{k,i}^1 = \begin{cases} 1 + (C_{k,i} + \epsilon H_{k,i} + \delta S_{k,i}) G_{k,i+1}^0 \\ \quad + S_{k,i} (1 - \delta) G_{k-1,i-1}^0 \\ \quad + H_{k,i} (1 - \epsilon) G_{k,i-1}^0 & k \geq 0, i \geq 1 \\ 0 & \text{en otro caso} \end{cases} \quad (4.32)$$

Substituyendo  $G_{k,i}^0$  en la expresión (4.32), resulta que  $G_{k,i}^1$  es de la forma,

$$G_{k,i}^1 = \begin{cases} G_{k,i}^0 + M_{k,i} (1 - 2(1 - \epsilon) H_{k,i} - 2(1 - \delta) S_{k,i}) \\ \quad + N_{k,i} S_{k,i} + O_{k,i} & k \geq 2, i \geq 1 \\ G_{k,i}^0 & \text{en otro caso} \end{cases} \quad (4.33)$$

donde,

$$M_{k,i} = 2i\beta + \rho k + \eta \quad k \geq 2, i \geq 1 \quad (4.34)$$

$$N_{k,i} = \begin{cases} (1-\delta)(\alpha(1-2k) - \rho(i-1)) & k > 2, i \geq 1 \\ (1-\delta)(-\alpha k^2 - \beta(i-1)^2 - \rho k(i-1) - \mu k) & \\ -\gamma) + \frac{(5-3i)\delta + 2(i-2)\epsilon + i-1}{1-2\epsilon} & k = 2, i \geq 1 \end{cases} \quad (4.35)$$

$$O_{k,i} = 1 + \beta \quad k \geq 2, i \geq 1 \quad (4.36)$$

Se puede ver que  $G_{k,i}^* \leq G_{k,i}^1 \leq G_{k,i}^0$ , donde  $G_{k,i}^*$  es la solución del sistema (4.24) si,

$$f = M_{k,i}(1 - 2(1-\epsilon)H_{k,i} - 2(1-\delta)S_{k,i}) + N_{k,i}S_{k,i} + O_{k,i} \leq 0 \quad (4.37)$$

para  $k \geq 2, i \geq 1$ . Dados  $\delta$  y  $\epsilon$  por el sistema, en la expresión (4.37) tenemos una restricción lineal por cada par  $(k, i)$ . Procedemos como sigue,

1. Aplicamos el método dual simplex de programación lineal que se pueden encontrar en diversos libros, entre ellos el de S. Gass [71] para la Región  $R_1$ , dada por  $k = 0 \dots 20$  y  $i = 1 \dots 20$ , es decir, 420 restricciones lineales. La Región  $R_1$  está representada gráficamente en la figura 4.2. La función a minimizar escogida es la suma de todos los coeficientes. Los resultados están expuestos en la tabla 4.4.
2. Comprobamos que  $f$  es negativa en la región  $R_2$ , dada por  $k = 0 \dots 100$  y  $i = 1 \dots 100$ . La Región  $R_2$  está representada gráficamente en la figura 4.2. Hay que hacer notar que  $R_2$  incluye  $R_1$ , y que esta compuesto por un número finito de puntos.
3. Demostramos que los coeficientes obtenidos satisfacen las restricciones en la Región  $R_3$ , dada por todo par  $(k, i)$  tal que  $k > 100$  o  $i > 100$ . La Región  $R_3$  está representada gráficamente en la figura 4.2.

### Demostración

Dados  $\delta$  y  $\epsilon$  el objetivo de este apartado es demostrar analíticamente que los coeficientes obtenidos satisfacen las restricciones para un par general  $(k, i)$ . Mediante el paso 1, sabemos que en la zona  $R_2$  definida por  $0 \leq k \leq 100, 1 \leq$

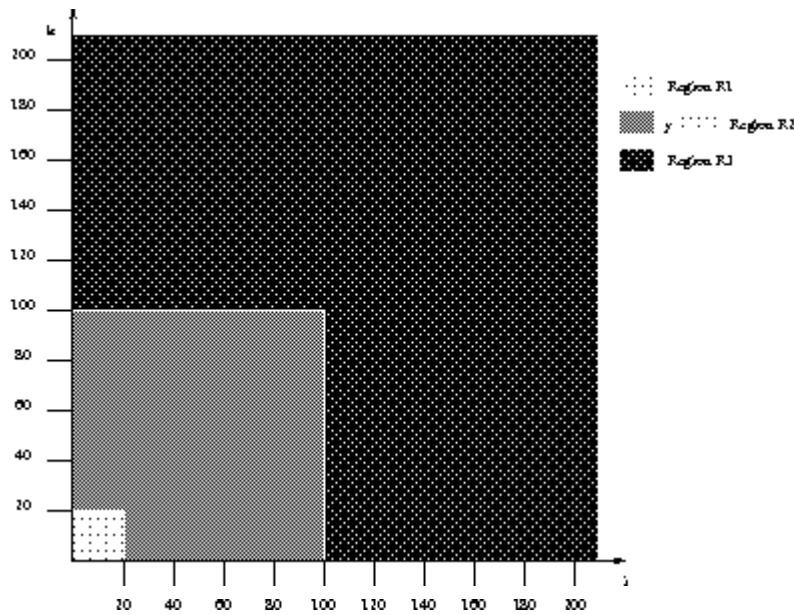


Figura 4.2: Representación gráfica de las regiones  $R_i$  en que se ha dividido el espacio.

$i \leq 100$ , la función  $f$  dada por la expresión (4.37) es no positiva. El objetivo, por tanto, es demostrar que en  $R_3$  también lo es, donde  $R_3$  corresponde a la región fuera de  $R_2$ , siempre en el cuadrante positivo dado por  $k \geq 0$ ,  $i \geq 1$ . Vamos a estudiar el caso  $\epsilon = 0$  y  $\delta = 0$ , el resto de posibilidades se desarrollarían siguiendo el mismo procedimiento.

**Caso  $\epsilon = 0$  y  $\delta = 0$ .** En este caso, el problema consiste en demostrar que la función  $f$  dada por la expresión (4.37) es menor o igual a 0, donde hemos substituido los valores correspondientes de la tabla 4.4. Tenemos, pues, dos funciones para dos zonas distintas,

1. Para  $k = 2$  y  $i \geq 1$ . Tenemos la función,

$$f = 1.25 + (.5i - 1.750) \left( 1 - 1.8 \left( 1 - \frac{1}{i} \right)^2 - 4 \frac{\left( 1 - \frac{1}{i} \right)^2}{i - 1} \right) + 2 \frac{(3.25i - 12.25 - 0.25(i - 1))^2 \left( 1 - \frac{1}{i} \right)^2}{i - 1} \quad (4.38)$$



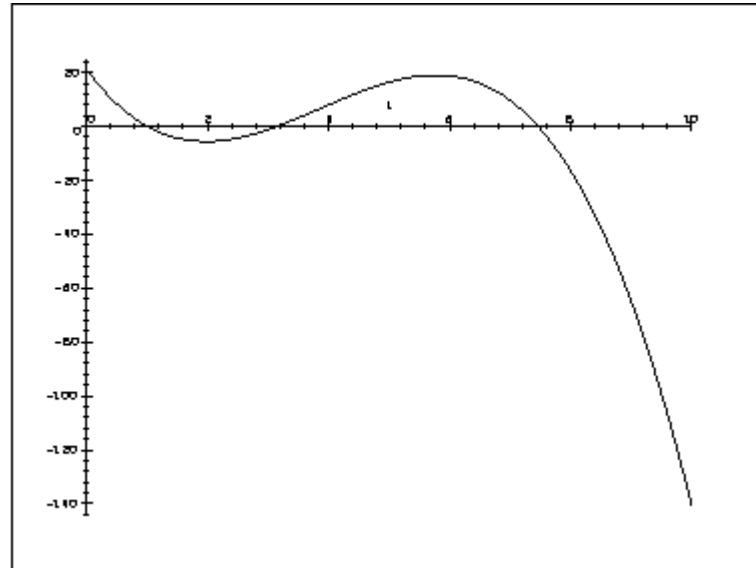


Figura 4.3: Representación gráfica de la función  $0.9i^3 + 10.45i^2 - 30.7i + 21.15$ .

Multiplicando por  $i^2$  resulta un polinomio de grado 3, representado en la figura 4.3.

$$0.9i^3 + 10.45i^2 - 30.7i + 21.15 \quad (4.39)$$

anulándose éste en

$$1, 7.461, 3.149$$

lo que significa que para  $i > 7.461$ ,  $f$ , (4.40), será negativa.

2. Para  $k > 2$  y  $i \geq 1$ . Tenemos la función,

$$f = 1.25 + \frac{(2.187k - 2.625k^2 + 0.875k(i-1) - ik)H_{k,i}}{i-1} + (0.50i - 0.875k)(1 - 2H_{k,i}) \quad (4.40)$$

Antes de continuar, vamos a escribir  $f$ , (4.40), de dos formas distintas,

$$f = 1.25 + \frac{g_1 k H_{k,i}}{i-1} + (0.50i - 0.875k)(1 - 2H_{k,i}) \quad (4.41)$$

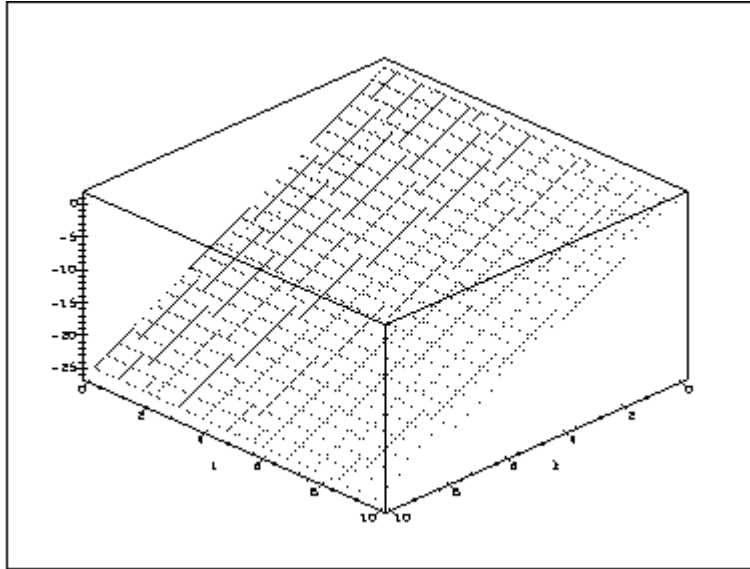


Figura 4.4: Representación gráfica de la función  $g_1 = 1.312 - 2.625k - 0.125i$ .

donde

$$g_1 = 2.187 - 2.625k + 0.875(i - 1) - i = 1.312 - 2.625k - 0.125i \quad (4.42)$$

y

$$f = 1.25 + \frac{g_2 H_{k,i}}{i - 1} + (0.50i - 0.875k) \quad (4.43)$$

donde

$$g_2 = -0.437k - 2.625k^2 + 1.625ik - i^2 + i \quad (4.44)$$

Vamos a demostrar que tanto  $g_1$ , (4.42), como  $g_2$ , (4.44), son negativas fuera de la región  $R_2$ , es decir, en  $R_3$ .

Para la función  $g_1$ , (4.42), es inmediato porque  $g_1$  es un plano que separa el espacio en dos mitades, quedando  $R_3$  situada en la parte negativa, por lo que  $g_1 \leq 0$  en  $R_3$ . La función  $g_1$ , (4.42), está representada gráficamente en la figura 4.4.

Por otra parte, la función  $g_2$ , (4.44), es cóncava porque,

$$\frac{\partial^2 g_2}{\partial k^2} = -5.25 < 0 \quad (4.45)$$

y

$$\frac{\partial^2 g_2}{\partial k^2} \frac{\partial^2 g_2}{\partial i^2} - \left( \frac{\partial^2 g_2}{\partial k \partial i} \right)^2 = 7.859 > 0 \tag{4.46}$$

Comprobando que en la frontera de una región que englobe  $R_2$  la función  $g_2$ , (4.44), es negativa, y que el máximo se encuentra dentro de esta región, demostramos que la función  $g_2$ , (4.44), es negativa en  $R_3$ .

Así,

$$\begin{aligned} g_2(k, 0) &= -0.437k - 2.625k^2 \\ g_2(-1, i) &= -2.188 - 0.625i - i^2 \\ g_2(k, 100) &= 162.063k - 2.625k^2 - 9900 \\ g_2(100, i) &= -26293.7 + 163.5i - i^2 \end{aligned}$$

Ambos casos son siempre negativos para  $k \geq -1$  e  $i \geq 0$ . Además el máximo de  $g_2$ , (4.44), se encuentra en la intersección de las siguientes rectas,

$$\begin{aligned} \frac{\partial g_2}{\partial k} &= -0.437 - 5.250k + 1.625i = 0 \\ \frac{\partial g_2}{\partial i} &= 1.625k - 2i + 1 = 0 \end{aligned}$$

El resultado es  $i = 0.578$ ,  $k = 0.096$ , que pertenece a la región. La función  $g_2$ , (4.44), está representada gráficamente en la figura 4.5.

Ya hemos demostrado que  $g_1$ , (4.42), y  $g_2$ , (4.44), son negativas en la región de interés  $R_3$ . A partir de ahora vamos a delimitar zonas del espacio donde se cumple que  $f$ , (4.40), es negativa.

Si  $g_2$ , (4.44), es negativa, entonces, de acuerdo con la expresión (4.39) de la función  $f$ , (4.43),

$$f = 1.25 + \frac{g_2 H_{k,i}}{i - 1} + (0.50i - 0.875k) \leq 1.25 + 0.50i - 0.875k \tag{4.47}$$

De aquí se obtiene que  $f$ , (4.40), será negativa si  $1.25 + 0.50i - 0.875k$  es negativa, es decir, para

$$k \geq 1.428 + 0.571i$$

que corresponde con la Región A en la figura 4.6.

Por otro lado, tenemos las siguientes inecuaciones,

$$1 - \frac{k}{i} \leq H_{k,i} = \left( 1 - \frac{1}{i} \right)^k \leq e^{-\frac{k}{i}} \tag{4.48}$$

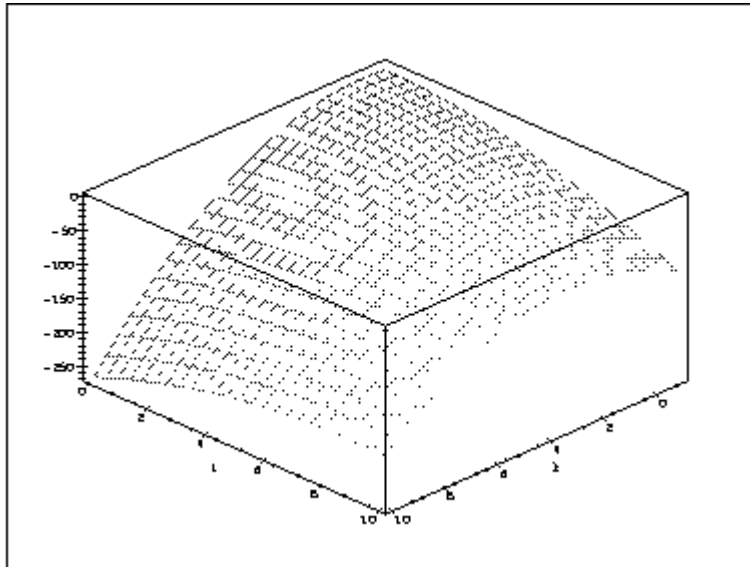


Figura 4.5: Representación gráfica de la función  $g_2 = -0.437k - 2.625k^2 + 1.625ik - i^2 + i$ .

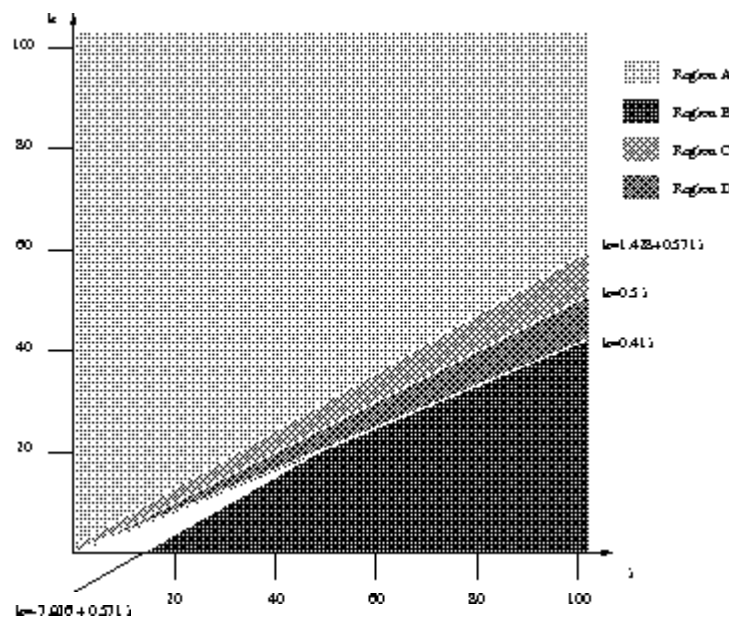


Figura 4.6: Representación gráfica de las regiones en que se ha dividido el espacio para la demostración.

Demostración de la primera inecuación de la expresión (4.48),

$$\begin{aligned} \left(1 - \frac{1}{i}\right)^k &= \sum_{n=0}^k \binom{k}{n} \left(-\frac{1}{i}\right)^n \\ &= 1 - \frac{k}{i} + \binom{k}{2} \frac{1}{i^2} + \dots \geq 1 - \frac{k}{i} \end{aligned} \quad (4.49)$$

Si construimos la función  $r(x) = (1-x)^k - (1-kx)$ , donde  $x = 1/i$  y demostramos que es siempre positiva o igual a 0, habremos demostrado la primera inecuación.

Tenemos, pues,

$$r(0) = 0 \quad (4.50)$$

y,

$$r'(x) = -k(1-x)^{k-1} + k = k(1 - (1-x)^{k-1}) \quad (4.51)$$

Vemos que  $r'(x)$  es siempre positiva si  $k > 1$ . Una función  $r(x)$  con derivada positiva que se anule en el origen, es siempre positiva para  $x \geq 0$  (q.e.d).

Demostración de la segunda inecuación de la expresión (4.48),

$$\begin{aligned} \left(1 - \frac{1}{i}\right)^i &\leq e^{-1} \Rightarrow \left(1 - \frac{1}{i}\right) \leq e^{-\frac{1}{i}} \\ &\Rightarrow \left(1 - \frac{1}{i}\right)^k \leq e^{-\frac{k}{i}} \end{aligned} \quad (4.52)$$

En nuestro caso, si  $0.50i - 0.875k \geq 0$ , ello implica que  $\frac{k}{i} \leq 0.57$ .

Por otro lado, si  $1 - 2H_{k,i} = 0$ , entonces,

$$k = -\frac{\ln 2}{\ln\left(1 - \frac{1}{i}\right)} \xrightarrow{i \rightarrow \infty} \left(i - \frac{1}{2}\right) \ln 2 \quad (4.53)$$

Entonces, si resolvemos el sistema,  $k = 0.5i$  y  $1 - 2H_{k,i} = 0$ , vemos que la solución está en  $k = 1$  e  $i = 2$ .

Y, por tanto, si  $k \leq 0.5i$  y  $k > 1$ , entonces,  $1 - 2H_{k,i} \leq 0$  y  $0.50i - 0.875k \geq 0$ .

Por tanto, si  $g_1$ , (4.42), es negativa, entonces, de acuerdo con la expresión (4.39) de la función  $f$ , (4.41),

$$\begin{aligned}
f &= 1.25 + (0.50i - 0.875k)(1 - 2H_{k,i}) + \frac{g_1 k H_{k,i}}{i - 1} \\
&\leq 1.25 + (0.50i - 0.875k)(1 - 2H_{k,i}) \\
&\leq 1.25 + (0.50i - 0.875k)(1 - 2\left(1 - \frac{k}{i}\right))
\end{aligned} \tag{4.54}$$

Si suponemos que  $k \leq 0.41i$ , entonces,

$$f \leq 1.25 - 0.09i + 0.1575k \tag{4.55}$$

Volvemos a tener especificada una zona. Por consiguiente  $f$ , (4.40), será negativa para

$$k \leq -7.9365 + 0.571i$$

que corresponde con la Región B en la figura 4.6. Esta Región B es la intersección de

$$k \leq -7.9365 + 0.571i$$

y  $k \leq 0.41i$ , que en  $R_3$  viene delimitada por  $k \leq 0.41i$ .

Vamos a dividir el espacio que nos falta,

$$0.41i \leq k \leq 1.428 + 0.571i$$

en dos partes.

La Región C representada en la figura 4.6 y determinada por

$$0.5i \geq k \geq 1.428 + 0.571i \tag{4.56}$$

La Región D representada en la figura 4.6 y determinada por

$$0.41i \geq k \geq 0.5i \tag{4.57}$$

**Región C** Dada una  $i$ , tenemos en esta Región C,

$$k_{\max} = 1.428 + 0.571i$$

$$k_{\min} = 0.5i$$

$$H_{k,i}^{\text{sup}} = e^{-\frac{k_{\min}}{i}} = e^{-0.5} = 0.6065$$

$$H_{k,i}^{\text{inf}} = 1 - \frac{k_{\max}}{i} = 1 - \frac{1.428 + 0.571i}{i}$$

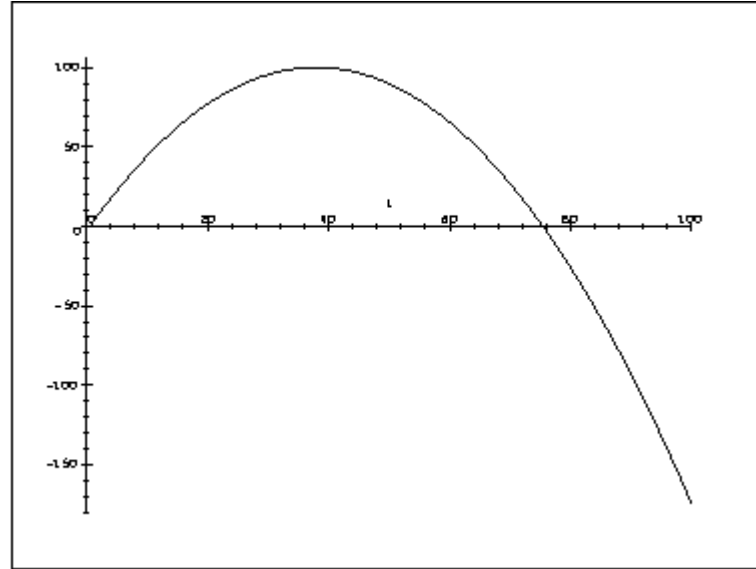


Figura 4.7: Representación gráfica de la función  $0.0715i^2 - 5.436i + 3.057$ .

Por consiguiente, obtenemos la siguiente desigualdad en esta Región de interés,

$$\begin{aligned}
 f &\leq 1.25 + 1.312 \frac{H_{k,i}k}{i-1} - 2.625 \frac{H_{k,i}k^2}{i-1} - 0.125 \frac{H_{k,i}ki}{i-1} \\
 &\quad + 0.5i - iH_{k,i} - 0.875k + 1.75kH_{k,i} \\
 &\leq 1.25 + 1.312 \frac{H_{k,i}^{\text{sup}}k_{\text{max}}}{i-1} - 2.625 \frac{H_{k,i}^{\text{inf}}k_{\text{min}}^2}{i-1} - 0.125 \frac{H_{k,i}^{\text{inf}}k_{\text{min}}i}{i-1} \\
 &\quad + 0.5i - iH_{k,i}^{\text{inf}} - 0.875k_{\text{min}} + 1.75k_{\text{max}}H_{k,i}^{\text{sup}}
 \end{aligned} \quad (4.58)$$

Substituyendo, se obtiene,

$$f \leq -\frac{0.0715i^2 - 5.436i + 3.057}{i-1} \quad (4.59)$$

Y multiplicando por  $(i-1)$ , que es siempre positivo, obtenemos una ecuación de segundo grado, representada gráficamente en la figura 4.7, y cuyos ceros son .56 y 75.41.

Por tanto, para  $i \geq 75.41$  la función  $f$ , (4.40), será negativa en esta zona.

**Región D** Dada una  $i$ , tenemos en esta Región D,

$$\begin{aligned} k_{\max} &= 0.5i \\ k_{\min} &= 0.41i \\ H_{k,i}^{\sup} &= e^{-\frac{k_{\min}}{i}} = e^{-0.41} = 0.6636 \\ H_{k,i}^{\inf} &= 1 - \frac{k_{\max}}{i} = 0.5 \end{aligned}$$

Por consiguiente, obtenemos la siguiente desigualdad en la Región de interés

$$\begin{aligned} f &\leq 1.25 + 1.312 \frac{H_{k,i} k}{i-1} - 2.625 \frac{H_{k,i} k^2}{i-1} - 0.125 \frac{H_{k,i} k i}{i-1} \\ &\quad + 0.5i - iH_{k,i} - 0.875k + 1.75kH_{k,i} \\ &\leq 1.25 + 1.312 \frac{H_{k,i}^{\sup} k_{\max}}{i-1} - 2.625 \frac{H_{k,i}^{\inf} k_{\min}^2}{i-1} - 0.125 \frac{H_{k,i}^{\inf} k_{\min} i}{i-1} \\ &\quad + 0.5i - iH_{k,i}^{\inf} - 0.875k_{\min} + 1.75k_{\max} H_{k,i}^{\sup} \end{aligned} \quad (4.60)$$

Substituyendo, se obtiene,

$$f \leq -\frac{0.024i^2 - 1.46i + 1.25}{i-1} \quad (4.61)$$

Y multiplicando por  $(i-1)$ , que es siempre positivo, obtenemos una ecuación de segundo grado, representada gráficamente en la figura 4.8, y cuyos ceros son 0.866 y 59.325.

Por tanto, para  $i \geq 59.325$  la función  $f$ , (4.40), será negativa en esta zona.

#### 4.4.4 Desarrollo de una cota superior ajustada de $E_k(l)$ , $E_k^u(l)$

Una vez tenemos  $G_{k,i}^0$  de la ecuación (4.31) y los coeficientes determinados, podemos conseguir una cota superior más ajustada de  $E_k(l)$ ,  $E_k^u(l)$  con el siguiente procedimiento.

Consideremos  $\mathcal{M}^0$  una matriz bidimensional  $(K_s \times I_s)$  cuyos componentes son  $G_{k,i}^0$  y sea  $\mathcal{M}^n$  la matriz bidimensional  $(K_s \times I_s)$  cuyos componentes  $G_{k,i}^n$  se obtienen de forma recursiva,

$$G_{k,i}^n = \begin{cases} 1 + (C_{k,i} + \varepsilon H_{k,i} + \delta S_{k,i}) G_{k,i+1}^{n-1*} \\ \quad + S_{k,i} (1 - \delta) G_{k-1,i-1}^{n-1*} + H_{k,i} (1 - \varepsilon) G_{k,i-1}^{n-1*} & (k, i) \in \mathcal{A}_s \\ G_{k,i}^0 & \text{en otro caso} \end{cases} \quad (4.62)$$



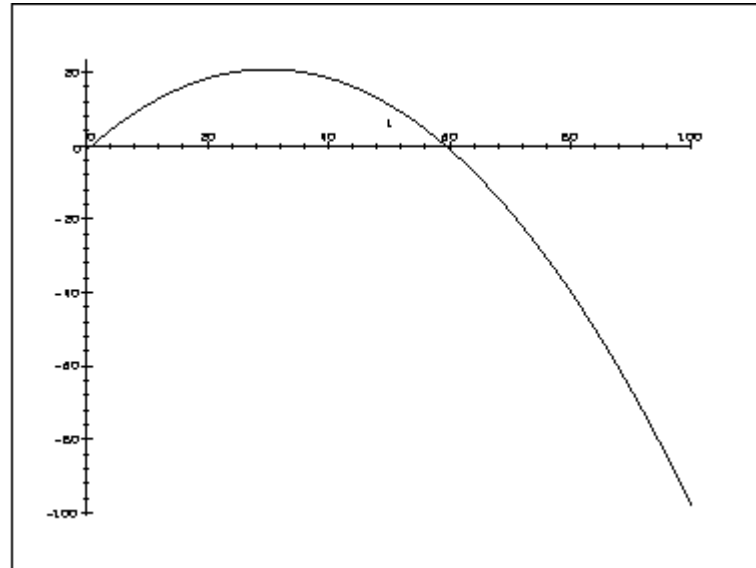


Figura 4.8: Representación gráfica de la función  $0.024i^2 - 1.46i + 1.25$ .

donde la región  $\mathcal{A}_s$  viene delimitada por  $0 \leq k \leq K_s$  y  $1 \leq i \leq I_s$ , y,

$$G_{k,i}^{n-1*} = \begin{cases} G_{k,i}^{n-1} & (k, i) \in \mathcal{A}_s \\ G_{k,i}^0 & \text{en otro caso} \end{cases} \quad (4.63)$$

Las componentes  $G_{k,i}^n = E_k^n(i)$  son la cota superior ajustada de  $E_k(i)$ . Cuanto mayor sea  $n$ , más ajustada será la cota. En nuestro caso, los valores de  $K_s$  e  $I_s$  considerados han sido  $K_s = I_s = 100$ .

$\epsilon$	$\delta$	$\alpha$	$\beta$	$\rho$	$\mu$	$\eta$	$\gamma$
0.00	0.00	2.187499	0.250000	-0.875001	0.00	0.00	0.00
0.00	0.01	2.189420	0.248314	-0.872471	0.00	0.00	0.00
0.00	0.10	2.206349	0.232143	-0.848214	0.00	0.00	0.00
0.00	0.20	2.224761	0.211538	-0.817307	0.00	0.00	0.00
0.00	0.40	2.251894	0.159091	-0.738636	0.00	0.00	0.00
0.00	0.50	2.174112	0.204758	-0.807136	0.00	0.00	0.00
0.01	0.00	2.203132	0.255323	-0.882984	0.00	0.00	0.00
0.10	0.00	2.391763	0.317745	-0.976617	0.00	0.00	0.00
0.20	0.00	2.773776	0.439705	-1.159558	0.00	0.00	0.00
0.30	0.00	3.608332	0.703704	-1.555556	0.00	0.00	0.00
0.40	0.00	7.276981	1.152253	-2.228379	0.00	0.00	0.00
0.45	0.00	13.631510	2.461186	-4.191780	0.00	0.00	0.00
0.10	0.10	2.404009	0.295461	-0.943192	0.00	0.00	0.00
0.10	0.20	2.412260	0.269231	-0.903846	0.00	0.00	0.00
0.10	0.30	2.412257	0.237944	-0.856916	0.00	0.00	0.00
0.20	0.10	2.781223	0.410853	-1.116279	0.00	0.00	0.00
0.20	0.20	2.779897	0.376082	-1.064123	0.00	0.00	0.00
0.20	0.30	2.761920	0.333318	-0.999978	0.00	0.00	0.00
0.30	0.10	3.618015	0.664365	-1.496548	0.00	0.00	0.00
0.30	0.30	3.579008	0.552609	-1.328913	0.00	0.00	0.00
0.40	0.30	6.958687	0.935381	-1.903072	0.00	0.00	0.00
0.30	0.50	3.457779	0.245678	-0.868518	0.00	0.00	0.00
0.30	0.70	3.969952	0.219482	-0.829222	0.00	0.00	0.00

Tabla 4.4: Coeficientes de  $G_{k,i}^0$ , en función de  $\epsilon$  y  $\delta$



# Capítulo 5

## Análisis del retardo, caso feedback binario

### 5.1 Introducción

El propósito de este Capítulo radica en el cálculo del otro gran parámetro que caracteriza un protocolo de acceso aleatorio: el retardo medio de paquete.

Antes de entrar en materia sería conveniente definir qué se entiende como retardo medio de paquete. A diferencia de throughput y estabilidad, hay prácticamente unanimidad en considerar que el retardo de un paquete es el tiempo transcurrido entre el instante que un paquete se genera o llega a la fuente y el instante que es exitosamente transmitido. Esta definición puede ser encontrada, entre otros, en los trabajos de J. Capetanakis [15], L. Merakos y C. Bisdikian [60], J. Huang y T. Berger [22] y P. Papantoni-Kazakos y M. Paterakis [26].

En la literatura especializada se pueden encontrar dos metodologías claramente diferenciadas para el cálculo del retardo. La primera es mediante ecuaciones funcionales, utilizada por ejemplo por P. Mathys y P. Flajolet [37] y por G. Fayolle, P. Flajolet, M. Hofri y P. Jacquet [18]; y la otra, que es la que usaremos nosotros, es mediante técnicas de acotación. A través de un resultado potente de la teoría de procesos regenerativos, se reduce el problema de determinar los momentos del proceso retardo al problema de resolución de sistemas de ecuaciones lineales de dimensión infinita, donde las técnicas de acotación obtienen cotas extremadamente ajustadas a sus soluciones. Otros análisis del retardo mediante esta metodología son los de N. Vvedenskaya y B. Tsybakov [69], de L. Merakos y C. Bisdikian [60], de P. Papantoni-Kazakos y M. Paterakis [26], de L. Georgiadis, L. Merakos y P. Papantoni-Kazakos [68]. A través de este método, llegaremos a un resultado que nos

permitirá calcular una cota superior y una inferior del retardo, en función de tres parámetros que deben ser calculados,

1. La longitud media  $E(l|d)$   $1 \leq d \leq \Delta$ ,
2. La suma de retardos media  $E(\psi|d)$   $1 \leq d \leq \Delta$ ,
3. El segundo momento de la longitud  $E(l^2|d)$   $1 \leq d \leq \Delta$ .

En el Apéndice de este Capítulo se encuentra desarrollado el cálculo de estos parámetros.

En la tabla 5.1 se exponen los resultados obtenidos tanto para el algoritmo WSA (ALOHA estabilizado de ventana), como para el algoritmo TCW (*two cell window*) de M. Paterakis y P. Papantoni-Kazakos [27] y para el algoritmo DBT (*dynamic binary tree*) de J. Capetanakis [15]. A grandes rasgos, se puede concluir que para bajas tasas los tres algoritmos tienen un comportamiento bastante similar en cuanto a retardo, y que para tasas altas, a partir de  $\lambda = 0.38$ , el retardo del algoritmo WSA es claramente inferior. Este resultado era de esperar puesto que, al fin y al cabo, si el algoritmo WSA llega hasta un throughput máximo estable superior, significa que a partir de una tasa dada el retardo será inferior.

El comportamiento que demuestran los tres algoritmos para bajas tasas es digno de resaltar, porque su retardo es considerablemente bajo. En la literatura han habido intentos de conseguir un retardo bajo para tasas altas a costa de elevar el retardo para bajas tasas. Esta última será la zona de funcionamiento normal, así que estos algoritmos tienen prestaciones peores, a mi entender, que los algoritmos que consiguen un retardo bajo para bajas tasas.

## 5.2 El método

En sistemas de acceso aleatorio, como en muchos sistemas de colas, es muy frecuente encontrarse con procesos estocásticos regenerativos. Vamos a exponer brevemente algunos resultados que utilizaremos posteriormente. Primeramente definiremos qué se entiende por proceso regenerativo.

*Definición.* El proceso  $\{X_n, n \geq 1\}$  se dice que es regenerativo con respecto a la secuencia de renovación  $\{\mathcal{R}_i, i \geq 1\}$  si para cada entero positivo  $M$  y cada secuencia  $t_1 \dots t_M$ , con  $0 \leq t_1 \leq \dots \leq t_M$ , la distribución conjunta de  $X_{t_1+\mathcal{R}_i} \dots X_{t_M+\mathcal{R}_i}$  es independiente de  $i$ .

Las variables aleatorias  $\mathcal{R}_i, i \geq 1$ , y  $C_i = \mathcal{R}_{i+1} - \mathcal{R}_i, i \geq 1$ , se llaman puntos de regeneración y ciclos de regeneración, respectivamente. Para

$\lambda$	Alg. WSA		Alg. TCW		Alg. DBT	
	$D^l$	$D^u$	$D^l$	$D^u$	$D^l$	$D^u$
0.02	1.560	1.560	1.562	1.563	1.563	1.564
0.06	1.710	1.716	1.708	1.716	1.713	1.719
0.10	1.887	1.905	1.888	1.917	1.903	1.921
0.16	2.257	2.310	2.257	2.363	2.308	2.362
0.20	2.616	2.709	2.607	2.812	2.712	2.809
0.24	3.138	3.298	3.103	3.467	3.308	3.476
0.30	4.588	4.946	4.412	5.197	4.976	5.365
0.32	5.443	5.923	5.162	6.170	5.973	6.501
0.36	8.660	9.619	7.941	9.665	9.798	10.883
0.38	12.174	13.669	11.008	13.398	14.121	15.855
0.40	20.003	22.715	18.262	22.024	24.427	27.736
0.42	51.351	58.994	57.354	67.665	78.530	90.212

Tabla 5.1: Límites superior e inferior de los retardos esperados. Los algoritmos representados son el WSA (ALOHA estabilizado de ventana), el algoritmo TCW (*two cell window*) de M. Paterakis y P. Papantoni-Kazakos y el algoritmo DBT (*dynamic binary tree*) de J. Capetanakis.

procesos regenerativos, se obtiene los siguientes resultados de L. Georgiadis, L. Merakos y P. Papantoni-Kazakos [68].

*Teorema 1.* Sea el proceso discreto en el tiempo  $\{X_n, n \geq 1\}$  regenerativo con respecto al proceso de renovación  $\{\mathcal{R}_i, i \geq 1\}$ . También denotemos  $C_i = \mathcal{R}_{i+1} - \mathcal{R}_i$ ,  $i = 1, 2, \dots$ , como la longitud del ciclo de regeneración  $i$ ésimo, y sea  $f$  una función no negativa con valores reales.

Siempre que se cumplan las tres inecuaciones siguientes,

$$X_n \geq 0 \quad (5.1)$$

$$C = E(C_i) < \infty \quad (5.2)$$

y,

$$S = E\left(\sum_{i=1}^{C_i} f(X_i)\right) < \infty \quad (5.3)$$

entonces,

$$\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n f(X_i) = \lim_{n \rightarrow \infty} \frac{1}{n} E\left(\sum_{i=1}^n f(X_i)\right) = \frac{S}{C} \quad (5.4)$$

con probabilidad 1. Además, si, adicionalmente a la existencia finita de  $C$  y  $S$ , la distribución de  $C$ ; no es periódica, entonces  $X_i$  converge en distribución a la variable aleatoria  $X_\infty$ , y,

$$\lim_{n \rightarrow \infty} E(f(X_n)) = E(f(X_\infty)) = \frac{S}{C} \quad (5.5)$$

Así, bajo las condiciones establecidas arriba, el promedio límite y la media de la distribución límite de  $\{f(X_n), n \geq 1\}$  existen, coinciden y son finitos. Más aún, su valor común se da en términos de las cantidades por ciclo  $S$  y  $C$ .

Dado un algoritmo de acceso aleatorio, sea  $\{X_n, n \geq 1\}$  el proceso de interés asociado con el sistema de acceso aleatorio; este proceso podría, por ejemplo, ser el proceso retardo inducido por el algoritmo. Entonces, una vez probado que  $X_n, n \geq 1$ , es regenerativo, el Teorema 1 muestra el camino para establecer la existencia de un estado permanente y para computar los momentos de este estado permanente y la distribución de  $X_n, n \geq 1$ , seleccionando adecuadamente la función  $f$ .

En muchos de los algoritmos de acceso aleatorio existentes es relativamente fácil identificar los puntos de regeneración (por ejemplo, cuando el sistema se vacía, o cuando la cadena de Markov alcanza un estado concreto) en los cuales el proceso de interés se reestablece. Dado un algoritmo de acceso aleatorio y una función  $f$ , el problema se convierte en la explotación de la dinámica del algoritmo, para encontrar aquellas propiedades por ciclo de la función muestra del proceso que podría ser usado posteriormente para evaluar las cantidades  $C$  y  $S$ . Cuando  $\{X_n, n \geq 1\}$  representa el proceso retardo, entonces la elección  $f(x) = x^i$  nos permitirá el computo del  $i$ ésimo momento del retardo en su estado permanente.

Resulta que en muchos algoritmos las cantidades  $S$  y  $C$  están íntimamente relacionadas a las soluciones de los sistemas de ecuaciones lineales de la forma,

$$x_i = b_i + \sum_{j \in \mathcal{A}} p_{ij} x_j \quad (5.6)$$

donde  $j \in \mathcal{A}$ ,  $p_{ij} \geq 0$ ,  $b_i \geq 0$  y  $\mathcal{A}$  es un conjunto numerable de índices. Hay que resaltar que este tipo de sistemas ya nos han aparecido y son del tipo del sistema (4.24), sólo que ahora tenemos un sumatorio en una única variable. Las cantidades  $S$  y  $C$  podrían ser un término de la secuencia solución  $X^* = \{x_i^*, i \in \mathcal{A}\}$  del sistema (5.6) (por ejemplo  $S = \{x_{i_0}^*, i_0 \in \mathcal{A}\}$  como en nuestro caso, o una función más general, como en el trabajo de L. Georgiadis y P. Papantoni-Kazakos [72]).

En casos donde los puntos de regeneración son los instantes en los cuales una cadena de Markov  $\{U_n\}$  vuelve a un estado específico  $i_0$ , una técnica

estándar para el desarrollo del sistema apropiado es el siguiente. Se introduce un "coste" apropiado  $\beta_{ij}$  por la transición simple de la cadena de Markov del estado  $i$  al estado  $j$  (este coste puede ser, por ejemplo, el tiempo retrasado o el retardo de paquete). Los  $x_i$  en el sistema (5.6) son entonces interpretados como el coste esperado total desde el estado  $i$  al estado  $i_0$ . En este caso, el  $b_i$  en el sistema (5.6) es el coste esperado de transición unitario en el estado  $i$ ,

$$b_i = E(\beta_{U_n+1, U_n} | U_n = i) \quad i \in \mathcal{A} \quad (5.7)$$

y lo que se busca es  $x_{i_0}^*$ , es decir, el coste esperado para la cadena para volver al estado  $i_0$ . Los  $b_i$ 's en la expresión (5.7) son a menudo fáciles de calcular.

Después de establecer los sistemas lineales asociados con las cantidades  $S$  y  $C$ , el próximo paso es estudiar sus soluciones. En el Apéndice A de L. Georgiadis, L. Merakos, y P. Papantoni-Kazakos [68] se dan un número de resultados generales que son útiles a la hora de establecer la existencia y unicidad de una solución y de desarrollar cotas superiores e inferiores de las soluciones de tales sistemas.

El sistema (5.6) es de la misma forma que el (4.24). Por tanto, se busca una secuencia  $X^0 = \{x_i^0, i \in \mathcal{A}\}$  tal que satisfaga la siguiente inecuación,

$$x_i^0 \geq b_i + \sum_{j \in \mathcal{A}} p_{ij} x_j^0 \geq 0 \quad \forall i \in \mathcal{A} \quad (5.8)$$

Para muchos de los algoritmo de acceso aleatorios existentes, el conjunto  $\mathcal{A}$  es un subconjunto de los números reales, y cuando se busca el retardo esperado, entonces la inecuación (5.8) se satisface por una secuencia lineal para el sistema asociado con  $C$ , mientras que se satisface por una secuencia cuadrática para el sistema asociado con  $S$ . Cuando se buscan los momentos superiores del retardo, entonces deben ser considerados secuencias polinómicas.

Si existe una secuencia  $X^0$  que satisfaga la inecuación (5.8), entonces se garantiza la existencia de una solución finita  $X^*$  del sistema (5.6), y  $X^0$  es una cota superior de esta solución, es decir,  $x_i^* \leq x_i^0$  para cada  $i \in \mathcal{A}$ . Además, si se puede encontrar otra secuencia  $Y^0 = \{y_i^0, i \in \mathcal{A}\}$ , tal que,

$$y_i^0 \leq x_i^0 \quad \forall i \in \mathcal{A} \quad (5.9)$$

$$y_i^0 \leq b_i + \sum_{j \in \mathcal{A}} p_{ij} y_j^0 \quad \forall i \in \mathcal{A} \quad (5.10)$$

entonces la secuencia  $Y^0$  es una cota inferior de la solución, es decir,  $y_i^0 \leq x_i^*$  para cada  $i \in \mathcal{A}$ . Este procedimiento se justifica por el Teorema A1 en el Apéndice A de L. Georgiadis, L. Merakos y P. Papantoni-Kazakos [68].



Inicialmente las secuencias  $X^0$  e  $Y^0$  están hechas para depender de ciertos parámetros, los cuales son elegidos para satisfacer las desigualdades (5.8)–(5.10). Este procedimiento también proporciona una cota inferior de la máxima tasa de entrada que el algoritmo mantiene con retardo esperado finito (throughput). A menudo esta cota está muy cerca del throughput o coincide con él.

### 5.3 El análisis del algoritmo

Empezaremos con una breve descripción de las características del algoritmo que son importantes para el análisis del retardo. Asumimos un canal ranurado al que acceden un número muy grande (efectivamente infinito) de usuarios idénticos e independientes que transmiten paquetes. La llegada de los paquetes se modela por un proceso de Poisson de tasa  $\lambda$  paquetes por ranura. Se asume que al final de cada ranura  $i$  (segmento de tiempo  $[i, i + 1)$ ) los usuarios reciben un feedback  $x_i$  igual a  $NC$  si en ranura había 0 o 1 paquetes transmitidos, o  $C$  si en ranura había más de 1 paquete. En la figura 5.1 hay un ejemplo gráfico de la evolución del protocolo WSA, donde se pueden observar los parámetros que seguidamente definiremos.

Supongamos que al inicio de la ranura  $v$  todos los paquetes que llegaron antes del tiempo  $t_v < v$  han sido exitosamente transmitidos y no hay información concerniente a los paquetes que pudieran haber llegado en el intervalo  $[t_v, v)$ ; es decir, la distribución de los tiempos entre llegadas de los paquetes en  $[t_v, v)$  es la misma que la supuesta originalmente. El inicio de tal ranura  $v$  se llama punto de resolución de conflictos (*conflict resolution point*, CRP). La diferencia de tiempo  $d_v = v - t_v$  será referida como el retraso en  $v$  (*lag at  $v$* ). En la ranura  $v$ , los usuarios que han generado paquetes en el intervalo  $[t_v, t_v + \tau_v)$  tienen permiso para transmitir, donde  $\tau_v = \min(d_v, \Delta)$ .  $\Delta$  es un parámetro adecuadamente escogido para maximizar el throughput. Después de un número de ranuras aleatorio, y siguiendo las reglas del algoritmo, se alcanza otro CRP,  $v'$ , con su correspondiente  $t_{v'} > t_v$ . El intervalo  $[v, v')$  entre dos CRP's consecutivos se llama intervalo de resolución de conflictos (*conflict resolution point*, CRI), el intervalo  $[t_v, t_v + \tau_v)$  se llama intervalo transmitido. En otros algoritmos donde no se resuelve todo el intervalo transmitido, existe el intervalo resuelto. En nuestro caso, tanto uno como otro coinciden.

Etiquetamos los paquetes con  $1, 2, 3, \dots$ , de acuerdo con el orden de llegada. El retardo  $D_n$  experimentado por el paquete  $n$ -ésimo se define como la diferencia de tiempo entre su llegada y el fin de su transmisión exitosa. Estaremos interesados en evaluar el valor esperado de  $D_n$ , cuando exista.

Sea  $\{v_i, i \geq 1\}$  la secuencia de sucesivos CRP's, y sea  $d_i$  el retraso en  $v_i$ . Se

puede demostrar que la secuencia  $\{d_i, i \geq 1\}$  es una cadena de Markov con los estados en el espacio  $\mathcal{F}$ , un conjunto denso y enumerable del intervalo  $[1, \infty)$  (en la parte inferior de la figura 5.1 hay representado un posible experimento de la cadena de Markov  $\{d_i, i \geq 1\}$  mediante pequeños círculos). Sea  $T_j = i$ ,  $d_{T_j} = 1$ , y definamos  $T_{j+1}$  como la primera ranura después de  $T_j$ , en la cual  $d_{T_{j+1}} = 1$ .

De la descripción del algoritmo se puede ver que el proceso probabilístico retardo inducido se reinicializa al inicio de cada ranura  $T_i$ ,  $i = 1, 2, \dots$ . El intervalo  $[T_i, T_{i+1})$  será referido como la sesión  $i$ ésima. Cabe resaltar que las sesiones tienen longitudes que son variables aleatorias independientes e idénticamente distribuidas.

Sea  $\mathcal{R}_i$ ,  $i = 1, 2, \dots$ , el número de paquetes exitosamente transmitidos en el intervalo  $[0, T_{i+1})$  (notar que  $\mathcal{R}_i$  también representa el número de paquetes que han llegado durante el intervalo  $[0, T_{i+1} - 1)$  ya que  $T_{i+1}$  es un CRP en el cual el retraso es igual a 1). Por tanto,  $C_i = \mathcal{R}_{i+1} - \mathcal{R}_i$ ,  $i = 1, 2, \dots$ , es el número de paquetes exitosamente transmitidos en el intervalo  $[T_i, T_{i+1})$  (estos son los paquetes que han llegado durante el intervalo  $[T_i - 1, T_{i+1} - 1)$ ). La secuencia  $\{\mathcal{R}_i, i \geq 1\}$  es un proceso de renovación ya que  $\{C_i, i \geq 1\}$  es una secuencia de variables aleatorias no negativas idénticas e independientes. Además, el proceso retardo  $\{D_n, n \geq 1\}$  es regenerativo con respecto al proceso de renovación  $\{\mathcal{R}_i, i \geq 1\}$ , con ciclo de regeneración  $C_i$ .

Del Teorema 1 de L. Georgiadis, L. Merakos y P. Papantoni-Kazakos [68], con  $f(D_i) = D_i$ , tenemos que si,

$$C = E(C_i) < \infty \quad (5.11)$$

y,

$$S = E\left(\sum_{i=1}^{C_i} D_i\right) < \infty \quad (5.12)$$

entonces existe un número real  $D$  tal que,

$$D = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n D_i = \lim_{n \rightarrow \infty} \frac{1}{n} E\left(\sum_{i=1}^n D_i\right) = \lim_{n \rightarrow \infty} E(D_n) = E(D_\infty) = \frac{S}{C} \quad (5.13)$$

con probabilidad 1.  $D$  será lo que hemos denominado como retardo medio de paquete.

A continuación desarrollaremos dos sistemas de ecuaciones cuya solución será usada para computar la longitud de ciclo media  $C$  y el retardo medio acumulado sobre un ciclo  $S$ .

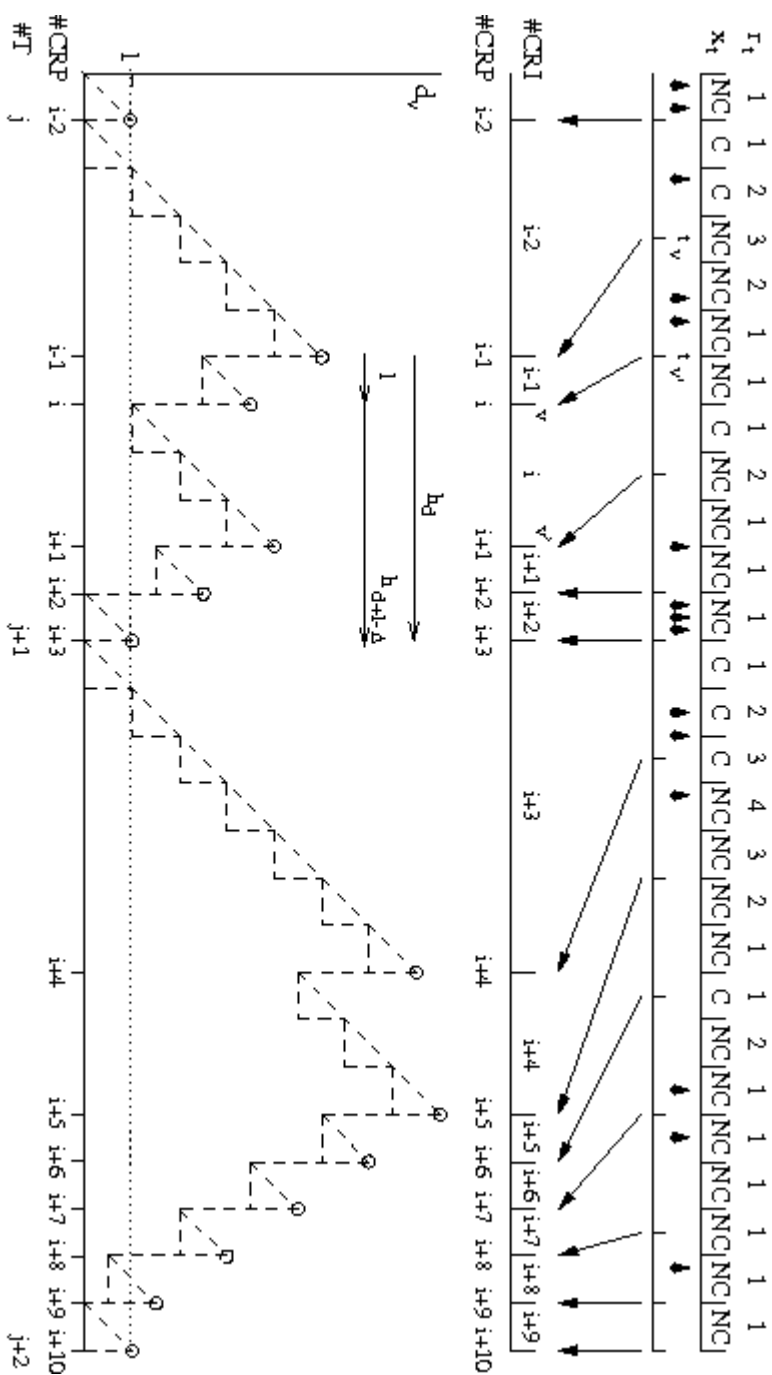


Figura 5.1: Ejemplo del funcionamiento del protocolo ALQHA estabilizado de ventana.

### 5.3.1 Sistema de ecuaciones relacionadas a las cantidades por ciclo

#### Longitud media por ciclo

Si la longitud de sesión media  $H \equiv E(T_2 - T_1)$  es finita, entonces se puede aplicar el teorema de Wald, página 59 del libro de S. Ross [73].

*Teorema.* Si  $X_1, X_2, \dots$  son variables aleatorias independientes e idénticamente distribuidas, y tienen esperanzas finitas, y si  $N$  es un tiempo de parada para  $X_1, X_2, \dots$  tal que  $E[N] < \infty$ , entonces,

$$E\left[\sum_1^N X_n\right] = E[N]E[X] \quad (5.14)$$

En nuestro caso tenemos que,

$$C = \sum_{T_i-1}^{T_{i+1}-1} \lambda_j \quad (5.15)$$

que es equivalente a

$$C = \sum_0^{T_{i+1}-T_i} \lambda_j \quad (5.16)$$

Puesto que las variables aleatorias longitud de sesión  $T_{i+1} - T_i$  son independientes e idénticamente distribuidas y las variables aleatorias número de usuarios que han llegado en el slot  $j$ ,  $\lambda_j$ , son independientes e idénticamente distribuidas slot a slot, entonces,

$$C = \sum_0^{T_{i+1}-T_i} \lambda_j = E[T_{i+1} - T_i]E[\lambda_j] = E[T_2 - T_1]E[\lambda_j] = H\lambda \quad (5.17)$$

Es decir,

$$C = H\lambda \quad (5.18)$$

Nuestro objetivo es desarrollar un sistema de ecuaciones a partir de los cuales  $H$  pueda ser determinado. Con este fin, definimos la variable aleatoria  $h_d$  como el número de ranuras necesarias para alcanzar un CRP en el cual el retraso sea 1, empezando desde un CRP en el cual el retraso es igual a  $d$ ,  $d \in \mathcal{F}$ .

Sea  $H_d = E(h_d)$ ,  $d \in \mathcal{F}$ , y notar que, por definición  $H = H_1$ .

Según el algoritmo, obtenemos las siguientes relaciones para los  $h_d$ 's,  $d \in \mathcal{F}$ , ver la figura 5.1 para una mayor claridad,

$$h_d = \begin{cases} l & l = 1, 1 \leq d \leq \Delta \\ l + h_l & l > 1, 1 \leq d \leq \Delta \\ l + h_{d-\Delta+l} & d > \Delta \end{cases} \quad (5.19)$$

El parámetro  $l$  podría ser considerado como un coste de transición de un paso unitario desde el estado  $d$  asociado con la cadena de Markov  $\{d_i, i \geq 1\}$  y  $h_d$  podría ser considerado como el coste para ir del estado  $d$  al estado 1.

Tomando esperanzas, tenemos,

$$H_d = \begin{cases} E(l|d) + \sum_{i \neq 1} p(l=i|d)H_i & 1 \leq d \leq \Delta \\ E(l|\Delta) + \sum_i p(l=i|\Delta)H_{d-\Delta+i} & d > \Delta \end{cases} \quad (5.20)$$

donde  $p(l=i|x)$  es la distribución de probabilidad de  $l$  en  $l=i$ , dado que el intervalo transmitido es  $x$ . Más adelante, para obtener límites superiores e inferiores, necesitaremos tan sólo  $p(l=1|x)$ , que resulta ser,

$$p(l=1|x) = (1 + \lambda x)e^{-\lambda x} \quad (5.21)$$

El sistema puede escribirse de una forma general,

$$H_d = b_d + \sum_{t \in \mathcal{F}} c_{dt} H_t \quad (5.22)$$

donde  $b_d$  y  $c_{dt}$ ,  $d, t \in \mathcal{F}$ , son coeficientes no negativos que podrían ser apropiadamente identificados.

### Retardo acumulativo medio sobre un ciclo

Desarrollaremos aquí un sistema de ecuaciones a partir de los cuales se puede determinar  $S$ . Consideramos un CRP  $v$  en el cual el retraso es igual a  $d$ ,  $d \in \mathcal{F}$ , y  $v_1$  entonces será el primer CRP después de  $v$  en el cual el retraso es igual a 1. Notar que  $v_1 - v = h_d$ . Definimos la variable aleatoria  $w_d$  como la suma de los retardos experimentados por todos los paquetes que fueron transmitidos exitosamente en el intervalo  $(v, v_1]$ .

Sea  $W_d = E(w_d)$ ,  $d \in \mathcal{F}$ , y notar que, por definición  $S = W_1$ . Consideramos ahora el CRI que empieza en  $v$  y finaliza en  $v'$ , se define las siguientes variables aleatorias,

$N$  es el número de paquetes en el intervalo transmitido  $[t_v, t_{v'})$ ,

$\theta$  es la suma de los retardos de los  $N$  paquetes hasta el instante  $t_{v'}$ , es decir en el intervalo  $[t_v, t_{v'})$ .

$\psi$  es la suma de los retardos de los  $N$  paquetes despues del CRP  $v$ , es decir en el intervalo  $[v, v')$ ,

Según el algoritmo se construye la siguiente relación,

$$w_d = \begin{cases} \theta + \psi & l = 1, 1 \leq d \leq \Delta \\ \theta + \psi + w_l & l > 1, 1 \leq d \leq \Delta \\ \theta + \psi + (d - \Delta)N + w_{d-\Delta+l} & d > \Delta \end{cases} \quad (5.23)$$

Tomando esperanzas, tenemos,

$$W_d = \begin{cases} E(\theta|d) + E(\psi|d) + \sum_{i \neq 1} p(l = i|d)W_i & 1 \leq d \leq \Delta \\ E(\theta|\Delta) + E(\psi|\Delta) + (d - \Delta)E(N|\Delta) \\ + \sum_i p(l = i|\Delta)W_{d-\Delta+i} & d > \Delta \end{cases} \quad (5.24)$$

donde  $p(l = i|d)$  es la misma distribución de probabilidad que anteriormente.

El sistema puede escribirse de una forma general,

$$W_d = b_d + \sum_{t \in \mathcal{F}} c_{dt} W_t \quad (5.25)$$

donde  $b_d$  y  $c_{dt}$ ,  $d$  y  $t \in \mathcal{F}$ , son coeficientes no negativos que podrían ser apropiadamente identificados.

### 5.3.2 Solución del sistema y límites del retardo medio de paquete

A continuación, el sistema en (5.22) sera referido como sistema  $H$  y al sistema en (5.25) sera referido como sistema  $W$ .

#### Sistema $H$

Buscamos secuencias  $X^0 = \{x_d^0, d \in \mathcal{F}\}$  y  $Y^0 = \{y_d^0, d \in \mathcal{F}\}$  que satisfagan las inecuaciones,

$$x_i^0 \geq b_i + \sum_{j \in \mathcal{F}} c_{ij} x_j^0 \geq 0 \quad (5.26)$$

$$0 \leq y_i^0 \leq b_i + \sum_{j \in \mathcal{F}} c_{ij} y_j^0 \quad (5.27)$$

Sea,

$$x_d^0 = \alpha_u d + \beta_u \quad (5.28)$$

donde  $\alpha_u, \beta_u$  son constantes reales y definimos,

$$x_d^1 = \begin{cases} E(i|d) + \sum_{i \neq 1} p(i=i|d) x_i^0 & 1 \leq d \leq \Delta \\ E(i|\Delta) + \sum_i p(i=i|\Delta) x_{d-\Delta+i}^0 & d > \Delta \end{cases} \quad (5.29)$$

Después de algunas operaciones, se obtiene,

$$x_d^1 = \begin{cases} x_d^0 + E(i|d) + \alpha_u (E(i|d) - d) \\ - (\alpha_u + \beta_u) (1 + \lambda d) e^{-\lambda d} & 1 \leq d \leq \Delta \\ x_d^0 + E(i|\Delta) - \alpha_u (\Delta - E(i|\Delta)) & d > \Delta \end{cases} \quad (5.30)$$

Y entonces, se puede ver que  $x_d^0 \geq x_d^1$ , para cada  $d \in \mathcal{F}$ , (la inecuación (5.8) se satisface) si,

$$\alpha_u = \frac{E(i|\Delta)}{\Delta - E(i|\Delta)} \quad (5.31)$$

y,

$$\beta_u = \max\{-\alpha_u, \sup_{1 \leq d \leq \Delta} \rho(d)\} \quad (5.32)$$

donde,

$$\rho(d) = \frac{E(i|d) + \alpha_u (E(i|d) - d - (1 + \lambda d) e^{-\lambda d})}{(1 + \lambda d) e^{-\lambda d}} \quad (5.33)$$

Similarmente, se puede demostrar que la secuencia  $Y_d^0$  con,

$$y_d^0 = \alpha_l d + \beta_l \quad (5.34)$$

satisface las desigualdades (5.9) y (5.10) si  $\alpha_l$  y  $\beta_l$  son escogidos como sigue,

$$\alpha_l = \alpha_u \quad (5.35)$$

$$\beta_l = \inf_{1 \leq d \leq \Delta} \rho(d) \quad (5.36)$$

donde  $\alpha_u$ ,  $\rho(d)$  son como los dados en las expresiones (5.31) y (5.33).

Así, tenemos que el sistema  $H$  tiene una solución no negativa  $S^H = \{s_d^H, d \in \mathcal{F}\}$  tal que,

$$\alpha_l d + \beta_l \leq y_d^1 \leq s_d^H \leq x_d^1 \leq \alpha_u d + \beta_u \quad (5.37)$$

para  $d \in \mathcal{F}$ , y donde,

$$y_d^1 = \begin{cases} x_d^1 - (\beta_u - \beta_l) (1 - (1 + \lambda d)e^{-\lambda d}) & 1 \leq d \leq \Delta \\ x_d^1 - (\beta_u - \beta_l) & d > \Delta \end{cases} \quad (5.38)$$

**Sistema  $W$**

Sea,

$$x_d^0 = \gamma_u d^2 + \delta_u d + \zeta_u \quad (5.39)$$

Siguiendo el mismo procedimiento que para el sistema  $H$ , definimos,

$$x_d^1 = \begin{cases} E(\theta|d) + E(\psi|d) + \sum_{i \neq 1} p(i=i|d)x_i^0 & 1 \leq d \leq \Delta \\ E(\theta|\Delta) + E(\psi|\Delta) + (d - \Delta)E(N|\Delta) \\ + \sum_i p(i=i|\Delta)x_{d-\Delta+i}^0 & d > \Delta \end{cases} \quad (5.40)$$

Después de algunas operaciones, se obtiene,

$$x_d^1 = \begin{cases} x_d^0 + E(\theta|d) + E(\psi|d) - \zeta_u(1 + \lambda d)e^{-\lambda d} \\ + \gamma_u (E(I^2|\Delta) - d^2 - (1 + \lambda d)e^{-\lambda d}) \\ + \delta_u (E(I|\Delta) - d - (1 + \lambda d)e^{-\lambda d}) & 1 \leq d \leq \Delta \\ x_d^0 + E(\theta|d) + E(\psi|d) + (d - \Delta)E(N|\Delta) \\ + \gamma_u (\Delta^2 - 2d\Delta + E(I^2|\Delta) + 2(d - \Delta)E(I|\Delta)) \\ + \delta_u (E(I|\Delta) - \Delta) & d > \Delta \end{cases} \quad (5.41)$$

De lo anterior, se puede ver que  $x_d^0 \geq x_d^1$ , y  $y_d^0 \leq y_d^1$  para cada  $d \in \mathcal{F}$ , (las inecuaciones (5.8)–(5.10) se satisfacen) si,

$$\gamma_u = \gamma_l = \frac{E(N|\Delta)}{2(\Delta - E(I|\Delta))} \quad (5.42)$$

$$\delta_u = \delta_l = \frac{E(\theta|\Delta) + E(\psi|\Delta) - \Delta E(N|\Delta) + \gamma_u (\Delta^2 + E(I^2|\Delta) - 2\Delta E(I|\Delta))}{\Delta - E(I|\Delta)} \quad (5.43)$$



$$\zeta_u = \sup_{1 \leq d \leq \Delta} \phi(d) \quad \zeta_l = \inf_{1 \leq d \leq \Delta} \phi(d) \quad (5.44)$$

donde

$$\phi(d) = \frac{E(\theta|d) + E(\psi|d) + \gamma_u (E(I^2|d) - d^2 - (1 + \lambda d)e^{-\lambda d}) - \delta_u (d - E(I|d) + (1 + \lambda d)e^{-\lambda d})}{(1 + \lambda d)e^{-\lambda d}} \quad (5.45)$$

Se puede ver de forma inmediata que,

$$E(N|\Delta) = \lambda \Delta \quad (5.46)$$

y,

$$E(\theta|t) = \frac{\lambda t^2}{2} \quad (5.47)$$

Partiendo del Teorema A1 en el Apéndice A de L. Georgiadis, L. Merakos y P. Papantoni-Kazakos [68], tenemos que el sistema  $W$  tiene una solución no negativa  $S^W = \{s_d^W, d \in \mathcal{F}\}$  tal que,

$$\gamma_l d^2 + \delta_l d + \zeta_l \leq y_d^1 \leq s_d^W \leq x_d^1 \leq \gamma_u d^2 + \delta_u d + \zeta_u \quad (5.48)$$

donde,

$$y_d^1 = \begin{cases} x_d^1 - (\zeta_u - \zeta_l) (1 - (1 + \lambda d)e^{-\lambda d}) & 1 \leq d \leq \Delta \\ x_d^1 - (\zeta_u - \zeta_l) & d > \Delta \end{cases} \quad (5.49)$$

Las soluciones  $S_H$  y  $S_W$  coinciden con las secuencias algorítmicas correspondientes de interés, es decir,  $s_d^H = H_d$  y  $s_d^W = W_d$ ,  $d \in \mathcal{F}$ . Por lo tanto tenemos que  $H_1^l \leq H_1 \leq H_1^u$ , y  $W_1^l \leq W_1 \leq W_1^u$  donde  $H_1^u = x_1^1$  según la expresión (5.30),  $H_1^l = y_1^1$  según la expresión (5.38),  $W_1^u = x_1^1$  según la expresión (5.41) y  $W_1^l = y_1^1$  según la expresión (5.49). Las expresiones correspondientes son,

$$H_1^u = E(I|1) + \alpha_u (E(I|1) - (1 + \lambda)e^{-\lambda}) + \beta_u (1 - (1 + \lambda)e^{-\lambda}) \quad (5.50)$$

$$H_1^l = H_1^u - (\beta_u - \beta_l) (1 - (1 + \lambda)e^{-\lambda}) \quad (5.51)$$

$$W_1^u = E(\psi|1) + E(\theta|1) + \gamma_u (E(I^2|1) - (1 + \lambda)e^{-\lambda}) + \delta_u (E(I|1) - (1 + \lambda)e^{-\lambda}) - \zeta_u (1 - (1 + \lambda)e^{-\lambda}) \quad (5.52)$$

$$W_1^l = W_1^u - (\zeta_u - \zeta_l) (1 - (1 + \lambda)e^{-\lambda}) \quad (5.53)$$

Finalmente, del Teorema 1 de L. Georgiadis, L. Merakos y P. Papantoni-Kazakos [68] y el hecho que  $S = W_1$ , tenemos los siguientes límites en el retardo de paquete medio,

$$\frac{W_1^l}{\lambda H_1^u} = D^l \leq D \leq D^u = \frac{W_1^u}{\lambda H_1^l} \quad (5.54)$$

## 5.4 Apéndice

### 5.4.1 Cálculo de la longitud media $E(l|d)$ , $1 \leq d \leq \Delta$

La longitud media del algoritmo es,

$$E(l|d) = \sum_{k=0}^{\infty} E_k(l) p(k|d) \quad (5.55)$$

donde,

$$p(k|d) = e^{-\lambda d} \frac{(\lambda d)^k}{k!} \quad (5.56)$$

Definimos, pues,  $E_k(l)$  como la longitud esperada de un CRI dado que empieza con una colisión de multiplicidad  $k$ .

Definimos también  $G_{k,i}$  como el número esperado de ranuras necesarias por el algoritmo para la transmisión exitosa de  $k$  paquetes, dado que el valor del contador es igual a  $r_i = i$  en la primera ranura.

Vamos a ver que en el algoritmo propuesto no se puede calcular de forma exacta  $E_k(l)$ . Nuestro objetivo será, pues, obtener una cota superior de  $E_k(l)$ , que llamaremos  $E_k^0(l)$ , utilizada para acotar el comportamiento en  $k \rightarrow \infty$ ; y a partir de ella obtendremos otra cota superior más ajustada, que llamaremos  $E_k^u(l)$ , que la utilizaremos para valores de  $k$  discretos. Así, obtendremos una cota superior bastante ajustada de  $E(l|d)$ , que llamaremos  $E^u(l|d)$ , y que será la que utilizaremos a todos los efectos como  $E(l|d)$ .

Según las definiciones dadas, se puede ver que  $E_k(l) = G_{k,r_{i_0}} = G_{k,1}$ , donde  $r_{i_0} = 1$  es el valor inicial del contador al principio del CRI. El primer paso será, pues, el cálculo de  $G_{k,r_{i_0}}$ . Como era lógico el resultado que se obtendrá para el cálculo de  $G_{k,r_{i_0}}$  será el mismo que el realizado en el Capítulo anterior, tomando  $\epsilon = \delta = 0$ , es decir, canal libre de error.

Cálculo de  $G_{k,r_i}$ 

Sea  $P_{k,i,s}$  la probabilidad que, dado que en el primer slot  $k$  terminales entran en contienda con probabilidad  $1/r_i = 1/i$ , exactamente se necesiten  $s$  slots para que accedan los  $k$  terminales. Recordar que han accedido todos los terminales sólo cuando una no colisión ocurre y  $r_i = 1$ . Entonces se aplica la siguiente ecuación recursiva,

$$P_{k,i,s} = \begin{cases} C_{k,i}P_{k,i+1,s-1} + S_{k,i}P_{k-1,i-1,s-1} & k \geq 0, i \geq 1 \\ + H_{k,i}P_{k,i-1,s-1} & \\ 0 & \text{en otro caso} \end{cases} \quad (5.57)$$

con,

$$H_{k,i} = \begin{cases} (1 - \frac{1}{i})^k & k > 0, i \geq 1 \\ 1 & k = 0, i \geq 1 \end{cases} \quad (5.58)$$

$$S_{k,i} = \begin{cases} \frac{k}{i} (1 - \frac{1}{i})^{k-1} & k \geq 0, i \geq 1 - \{(k=0, i=1), (k=1, i=1)\} \\ 1 & k=1, i=1 \\ 0 & k=0, i=1 \end{cases} \quad (5.59)$$

$$C_{k,i} = 1 - S_{k,i} - H_{k,i} \quad (5.60)$$

Tomando esperanza respecto de  $s$ , se llega a la siguiente expresión. Llamando,

$$G_{k,i} = \sum_{s=0}^{\infty} s P_{k,i,s} \quad (5.61)$$

se obtiene,

$$G_{k,i} = \begin{cases} 1 + C_{k,i}G_{k,i+1} + S_{k,i}G_{k-1,i-1} + H_{k,i}G_{k,i-1} & k \geq 2, i \geq 1 \\ i & 0 \leq k \leq 1, i \geq 1 \\ 0 & \text{en otro caso} \end{cases} \quad (5.62)$$

Tenemos, pues, un sistema de ecuaciones lineales de dimensión infinita de la forma,

$$G_{k,i} = l_{k,i} + \sum_{q=0}^{\infty} \sum_{j=1}^{\infty} a_{q,j}^{k,i} G_{q,j} \quad (5.63)$$

donde,

$$I_{k,i} = \begin{cases} 1 & k \geq 2, i \geq 1 \\ i & 0 \leq k \leq 1, i \geq 1 \\ 0 & \text{en otro caso} \end{cases} \quad (5.64)$$

y,

$$a_{qj}^{k,i} = \begin{cases} C_{k,i} & q = k, j = i + 1, k \geq 2, i \geq 1 \\ S_{k,i} & q = k - 1, j = i - 1, k \geq 2, i \geq 1 \\ H_{k,i} & q = k, j = i - 1, k \geq 2, i \geq 1 \\ 0 & \text{en otro caso} \end{cases} \quad (5.65)$$

Ya que no es posible resolver el sistema de ecuaciones, intentaremos encontrar límites de  $G_{k,i}$ .

Sea,

$$G_{k,i}^0 = \begin{cases} \alpha k^2 + \beta i^2 + \rho k i + \gamma & k \geq 2, i \geq 1 \\ i & 0 \leq k \leq 1, i \geq 1 \\ 0 & \text{en otro caso} \end{cases} \quad (5.66)$$

y definimos,

$$G_{k,i}^1 = \begin{cases} 1 + C_{k,i} G_{k,i+1}^0 + S_{k,i} G_{k-1,i-1}^0 + H_{k,i} G_{k,i-1}^0 & k \geq 2, i \geq 1 \\ i & 0 \leq k \leq 1, i \geq 1 \\ 0 & \text{en otro caso} \end{cases} \quad (5.67)$$

Después de operar, obtenemos los siguientes resultados,

$$G_{k,i}^1 = \begin{cases} G_{k,i}^0 + M_{k,i}(2C_{k,i} - 1) + N_{k,i}S_{k,i} + O_{k,i} & k \geq 2, i \geq 1 \\ G_{k,i}^0 & \text{en otro caso} \end{cases} \quad (5.68)$$

donde,

$$M_{k,i} = 2i\beta + \rho k \quad k \geq 2, i \geq 1 \quad (5.69)$$

$$N_{k,i} = \begin{cases} \alpha(1 - 2k) - \rho(i - 1) & k > 2, i \geq 1 \\ -(\alpha k^2 + \beta(i - 1)^2 + (\rho k - 1)(i - 1) + \gamma) & k = 2, i \geq 1 \end{cases} \quad (5.70)$$

$$O_{k,i} = 1 + \beta \quad k \geq 2, i \geq 1 \quad (5.71)$$

Se puede ver que  $G_{k,i}^1 \leq G_{k,i}^0$  si,

$$M_{k,i}(2C_{k,i} - 1) + N_{k,i}S_{k,i} + O_{k,i} \leq 0 \quad k \geq 2, i \geq 1 \quad (5.72)$$

Tenemos una restricción lineal por cada par  $(k, i)$ . Procedemos como sigue,

1. Aplicamos el método dual simplex de programación lineal que se pueden encontrar en diversos libros, entre ellos el de S. Gass [71] para la Región  $R_1$ , dada por  $k = 0 \dots 20$  y  $i = 1 \dots 20$ , es decir, 420 restricciones lineales. La Región  $R_1$  está representada gráficamente en la figura 4.2. La función a minimizar escogida es la suma de todos los coeficientes.
2. Comprobamos que  $f$  es negativa en la región  $R_2$ , dada por  $k = 0 \dots 100$  y  $i = 1 \dots 100$ . La Región  $R_2$  está representada gráficamente en la figura 4.2. Hay que hacer notar que  $R_2$  incluye  $R_1$ , y que esta compuesto por un número finito de puntos.
3. Demostramos que los coeficientes obtenidos satisfacen las restricciones en la Región  $R_3$ , dada por todo par  $(k, i)$  tal que  $k > 100$  o  $i > 100$ . La Región  $R_3$  está representada gráficamente en la figura 4.2.

El resultado obtenido es,

$$G_{k,i}^0 = \begin{cases} 2, 1875k^2 + 0, 25i^2 - 0, 875ki & k \geq 2, i \geq 1 \\ i & 0 \leq k \leq 1, i \geq 1 \\ 0 & \text{en otro caso} \end{cases} \quad (5.73)$$

De esta forma,  $G_{k,i}^0$  será un límite superior de  $G_{k,i}$ . Y, por tanto, tenemos una cota superior de  $E_k(l)$ ,  $E_k^0(l) = G_{k,r_{i_0}}^0 = G_{k,1}^0$ .

**Demostración.** El objetivo de este apartado es demostrar analíticamente que los coeficientes obtenidos satisfacen las restricciones para un par general  $(k, i)$ . El método a seguir sería totalmente paralelo al especificado en Apéndice del Capítulo anterior.

**Cálculo de  $E^u(l|d)$ ,  $1 \leq d \leq \Delta$**

Sea  $E_k(l)$  la longitud esperada de un CRI dado que empieza con una colisión de multiplicidad  $k$ .

Se puede ver que  $E_k(l) = G_{k,r_{i_0}}$ , donde  $r_{i_0}$  es el valor inicial del contador al principio del CRI. De los resultados anteriores conocemos un límite superior de  $G_{k,i}$  analíticamente, que nos servirá para calcular el throughput.

La longitud media del algoritmo es,

$$E(l|d) = \sum_{k=0}^{\infty} E_k(l)p(k|d) \quad (5.74)$$

donde

$$p(k|d) = e^{-\lambda d} \frac{(\lambda d)^k}{k!} \quad (5.75)$$

Ya hemos visto que no se puede calcular de forma exacta  $E_k(i)$ . Tenemos un límite superior analítico  $E_k^0(i) = G_{k,r_{i_0}}^0$ , que utilizaremos para acotar el comportamiento en  $k \rightarrow \infty$ . Para calcular la longitud media  $E(i|d)$ , que a todos los efectos substituiremos por una cota ligeramente superior  $E^u(i|d)$ , necesitamos una cota mas precisa de  $E_k(i)$  que llamaremos  $E_k^u(i) = G_{k,r_{i_0}}^u$ , y que utilizaremos para  $k$  discretos. Así tenemos,

$$E^u(i|d) = \sum_{k=0}^{30} E_k^u(i)p(k|d) + \sum_{k=31}^{\infty} E_k^0(i)p(k|d) \quad (5.76)$$

Después de algunas operaciones con el segundo sumando, se obtiene,

$$\begin{aligned} E^u(i|d) = & \sum_{k=0}^{30} E_k^u(i)p(k|d) + \alpha \left\{ (\lambda d)^2 + \lambda d - \sum_{k=0}^{30} k^2 p(k|d) \right\} \\ & + \rho r_{i_0} \left\{ \lambda d - \sum_{k=0}^{30} k p(k|d) \right\} + (\beta r_{i_0}^2 + \gamma) \left\{ 1 - \sum_{k=0}^{30} p(k|d) \right\} \end{aligned} \quad (5.77)$$

ya que,

$$E_k^0(i) = G_{k,r_{i_0}}^0 = \alpha k^2 + \beta r_{i_0}^2 + \rho k r_{i_0} + \gamma = 2,1875k^2 + 0,25r_{i_0}^2 - 0,875k r_{i_0} \quad (5.78)$$

Una vez tenemos  $G_{k,i}^0$ , expresión (5.73), podemos conseguir una cota superior más ajustada de  $E_k(i)$ ,  $E_k^u(i)$ , con el siguiente procedimiento. Sea  $\mathcal{M}^0$  una matriz bidimensional ( $K_s \times I_s$ ) cuyos componentes son  $G_{k,i}^0$  y sea  $\mathcal{M}^n$  la matriz bidimensional ( $K_s \times I_s$ ) cuyos componentes se obtienen de forma recursiva,

$$G_{k,i}^n = \begin{cases} 1 + C_{k,i} G_{k,i+1}^{n-1*} + S_{k,i} G_{k-1,i-1}^{n-1*} + H_{k,i} G_{k,i-1}^{n-1*} & (k,i) \in \mathcal{A}_s \\ G_{k,i}^0 & \text{en otro caso} \end{cases} \quad (5.79)$$

donde la región  $\mathcal{A}_s$  viene delimitada por  $0 \leq k \leq K_s$  y  $1 \leq i \leq I_s$ , y,

$$G_{k,i}^{n-1*} = \begin{cases} G_{k,i}^{n-1} & (k,i) \in \mathcal{A}_s \\ G_{k,i}^0 & \text{en otro caso} \end{cases} \quad (5.80)$$

Las componentes  $G_{k,r_{i_0}}^n = E_k^u(i)$  son la cota superior ajustada de  $E_k(i)$ . Cuanto mayor sea  $n$ , más ajustada será la cota. En este caso, tomaremos  $K_s = I_s = 100$ .

### 5.4.2 Cálculo de la suma de los retardos media $E(\psi|d)$ , $1 \leq d \leq \Delta$

La suma de los retardos media del algoritmo es,

$$E(\psi|d) = \sum_{k=0}^{\infty} E_k(\psi)p(k|d) \quad (5.81)$$

donde,

$$p(k|d) = e^{-\lambda d} \frac{(\lambda d)^k}{k!} \quad (5.82)$$

Definimos, pues,  $E_k(\psi)$  como la suma de los retardos esperada de un CRI dado que empieza con una colisión de multiplicidad  $k$ .

Definimos también  $\Psi_{k,i}$  como la suma de retardos esperado para la transmisión exitosa de  $k$  paquetes, dado que el valor del contador es igual a  $r_t = i$  en la primera ranura.

Vamos a ver que en el algoritmo propuesto no se puede calcular de forma exacta  $E_k(\psi)$ . Nuestro objetivo será, pues, obtener una cota superior de  $E_k(\psi)$ , que llamaremos  $E_k^o(\psi)$ , utilizada para acotar el comportamiento en  $k \rightarrow \infty$ ; y a partir de ella obtendremos otra cota superior más ajustada, que llamaremos  $E_k^u(\psi)$ , que la utilizaremos para valores de  $k$  discretos. Así, obtendremos una cota superior bastante ajustada de  $E(\psi|d)$ , que llamaremos  $E^u(\psi|d)$ , y que será la que utilizaremos a todos los efectos como  $E(\psi|d)$ .

Según las definiciones dadas, se puede ver que  $E_k(\psi) = \Psi_{k,r_{i_0}}$ , donde  $r_{i_0}$  es el valor inicial del contador al principio del CRI. El primer paso será, pues, el cálculo de  $\Psi_{k,r_{i_0}}$ .

#### Cálculo de $\Psi_{k,r_{i_0}}$

Sea  $\Psi_{k,i}$  la suma de los retardos de  $k$  terminales que entran en tienda con probabilidad  $1/r_t = 1/i$ , al principio del CRI. Recordar que han accedido todos los terminales solo cuando una no colisión ocurre y  $r_t = 1$ . Entonces se aplica la siguiente ecuación recursiva,

$$\Psi_{k,i} = \begin{cases} k + C_{k,i}\Psi_{k,i+1} + S_{k,i}\Psi_{k-1,i-1} + H_{k,i}\Psi_{k,i-1} & k \geq 2, i \geq 1 \\ \frac{i+1}{2} & k = 1, i \geq 1 \\ 0 & \text{en otro caso} \end{cases} \quad (5.83)$$

donde  $H_{k,i}$ ,  $S_{k,i}$ ,  $C_{k,i}$  vienen dados por las expresiones (5.58), (5.59) y (5.60) respectivamente.

Tenemos, pues, un sistema de ecuaciones lineales de dimension infinita de la forma,

$$\Psi_{k,i} = I_{k,i} + \sum_{q=0}^{\infty} \sum_{j=1}^{\infty} a_{q,j}^{k,i} \Psi_{q,j} \quad (5.84)$$

donde,

$$I_{k,i} = \begin{cases} k & k \geq 2, i \geq 1 \\ \frac{i+1}{2} & 0 \leq k \leq 1, i \geq 1 \\ 0 & \text{en otro caso} \end{cases} \quad (5.85)$$

y,

$$a_{q,j}^{k,i} = \begin{cases} C_{k,i} & q = k, j = i + 1, k \geq 2, i \geq 1 \\ S_{k,i} & q = k - 1, j = i - 1, k \geq 2, i \geq 1 \\ H_{k,i} & q = k, j = i - 1, k \geq 2, i \geq 1 \\ 0 & \text{en otro caso} \end{cases} \quad (5.86)$$

Sea,

$$\Psi_{k,i}^0 = \begin{cases} \alpha k^2 + \beta i^2 + \rho k i + \gamma & k \geq 2, i \geq 1 \\ \frac{i+1}{2} & k = 1, i \geq 1 \\ 0 & \text{en otro caso} \end{cases} \quad (5.87)$$

para  $i \geq 1$ , y definimos,

$$\Psi_{k,i}^1 = \begin{cases} k + C_{k,i} \Psi_{k,i+1}^0 + S_{k,i} \Psi_{k-1,i-1}^0 + H_{k,i} \Psi_{k,i-1}^0 & k \geq 2, i \geq 1 \\ \frac{i+1}{2} & k = 1, i \geq 1 \\ 0 & \text{en otro caso} \end{cases} \quad (5.88)$$

Despues de operar, obtenemos los siguientes resultados,

$$\Psi_{k,i}^1 = \begin{cases} \Psi_{k,i}^0 + M_{k,i} (2C_{k,i} - 1) + N_{k,i} S_{k,i} + O_{k,i} & k \geq 2, i \geq 1 \\ \Psi_{k,i}^0 & \text{en otro caso} \end{cases} \quad (5.89)$$

donde,

$$M_{k,i} = 2i\beta + \rho k \quad k \geq 2, i \geq 1 \quad (5.90)$$

$$N_{k,i} = \begin{cases} \alpha(1 - 2k) - \rho(i - 1) & k > 2, i \geq 1 \\ \alpha k^2 + \beta(i - 1)^2 - \frac{i}{2} + \rho k(i - 1) + \gamma & k = 2, i \geq 1 \end{cases} \quad (5.91)$$



$$O_{k,i} = k + \beta \quad k \geq 2, i \geq 1 \quad (5.92)$$

Se puede ver que  $\Psi_{k,i}^1 \leq \Psi_{k,i}^0$  si,

$$M_{k,i}(2C_{k,i} - 1) + N_{k,i}S_{k,i} + O_{k,i} \leq 0 \quad k \geq 2, i \geq 1 \quad (5.93)$$

Tenemos una restricción lineal por cada par  $(k, i)$ . Procedemos como sigue,

1. Aplicamos el método dual simplex de programación lineal que se pueden encontrar en diversos libros, entre ellos el de S. Gass [71] para la Región  $R_1$ , dada por  $k = 0 \dots 20$  y  $i = 1 \dots 20$ , es decir, 420 restricciones lineales. La Región  $R_1$  está representada gráficamente en la figura 4.2. La función a minimizar escogida es la suma de todos los coeficientes.
2. Comprobamos que  $f$  es negativa en la región  $R_2$ , dada por  $k = 0 \dots 100$  y  $i = 1 \dots 100$ . La Región  $R_2$  está representada gráficamente en la figura 4.2. Hay que hacer notar que  $R_2$  incluye  $R_1$ , y que esta compuesto por un número finito de puntos.
3. Demostramos que los coeficientes obtenidos satisfacen las restricciones en la Región  $R_3$ , dada por todo par  $(k, i)$  tal que  $k > 100$  o  $i > 100$ . La Región  $R_3$  está representada gráficamente en la figura 4.2.

El resultado obtenido es,

$$\Psi_{k,i}^0 = \begin{cases} 3,5077k^2 + 0,4410i^2 - 1,6615ki & k \geq 2, i \geq 1 \\ \frac{i+1}{2} & k = 1, i \geq 1 \\ 0 & \text{en otro caso} \end{cases} \quad (5.94)$$

De esta forma,  $\Psi_{k,i}^0$  sera un límite superior de  $\Psi_{k,i}$ . Y, por tanto, tenemos una cota superior de  $E_k(\psi)$ ,  $E_k^0(\psi) = \Psi_{k,r_{30}}^0 = \Psi_{k,1}^0$ .

**Demostración.** El objetivo de este apartado es demostrar analíticamente que los coeficientes obtenidos satisfacen las restricciones para un par general  $(k, i)$ . El método a seguir sería totalmente paralelo al especificado en Apéndice del Capítulo anterior.

Cálculo de  $E^u(\psi|d)$ ,  $1 \leq d \leq \Delta$

Sea  $E_k(\psi)$  la suma de los retardos esperada de un CRI dado que empieza con una colisión de multiplicidad  $k$ .

Se puede ver que  $E_k(\psi) = \Psi_{k,r_{t_0}}$ , donde  $r_{t_0}$  es el valor inicial del contador. De los resultados anteriores conocemos un límite superior de  $\Psi_{k,i}$  analíticamente, que nos servirá para cálculos posteriores.

La suma media de los retardos del algoritmo es,

$$E(\psi|d) = \sum_{k=0}^{\infty} E_k(\psi)p(k|d) \quad (5.95)$$

donde,

$$p(k|d) = e^{-\lambda d} \frac{(\lambda d)^k}{k!} \quad (5.96)$$

Ya hemos visto que no se puede calcular de forma exacta  $E_k(\psi)$ . Tenemos un límite superior analítico  $E_k^0(\psi) = \Psi_{k,r_{t_0}}^0$ , que utilizaremos para acotar el comportamiento en  $k \rightarrow \infty$ . Para calcular la longitud media, que a todos los efectos substituiremos por una cota ligeramente superior  $E^u(\psi|d)$ , necesitamos una cota más precisa de  $E_k(\psi)$  que llamaremos  $E_k^u(\psi) = \Psi_{k,r_{t_0}}^u$ , y que utilizaremos para  $k$  discretos. Así tenemos,

$$E^u(\psi|d) = \sum_{k=0}^{30} E_k^u(\psi)p(k|d) + \sum_{k=31}^{\infty} E_k^0(\psi)p(k|d) \quad (5.97)$$

Después de algunas operaciones con el segundo sumando, se obtiene,

$$\begin{aligned} E^u(\psi|d) = & \sum_{k=0}^{30} E_k^u(\psi)p(k|d) + \alpha \left\{ (\lambda d)^2 + \lambda d - \sum_{k=0}^{30} k^2 p(k|d) \right\} \\ & + \rho r_{t_0} \left\{ \lambda d - \sum_{k=0}^{30} k p(k|d) \right\} + (\beta r_{t_0}^2 + \gamma) \left\{ 1 - \sum_{k=0}^{30} p(k|d) \right\} \end{aligned} \quad (5.98)$$

ya que,

$$\begin{aligned} E_k^0(\psi) = \Psi_{k,r_{t_0}}^0 = & \alpha k^2 + \beta r_{t_0}^2 + \rho k r_{t_0} + \gamma = \\ & 3,5077k^2 + 0,4410r_{t_0}^2 - 1,6615kr_{t_0} \end{aligned} \quad (5.99)$$

Una vez tenemos  $\Psi_{k,i}^0$ , expresión (5.94), podemos conseguir una cota superior más ajustada de  $E_k(\psi)$ ,  $E_k^u(\psi)$ , con el siguiente procedimiento. Sea  $\mathcal{M}^0$  una matriz bidimensional ( $K_s \times I_s$ ) cuyos componentes son  $\Psi_{k,i}^0$  y sea

$\mathcal{M}^n$  la matriz bidimensional ( $K_s \times I_s$ ) cuyos componentes se obtienen de forma recursiva,

$$\Psi_{k,i}^n = \begin{cases} 1 + C_{k,i} \Psi_{k,i+1}^{n-1*} + S_{k,i} \Psi_{k-1,i-1}^{n-1*} + H_{k,i} \Psi_{k,i-1}^{n-1*} & (k,i) \in \mathcal{A}_s \\ \Psi_{k,i}^0 & \text{en otro caso} \end{cases} \quad (5.100)$$

donde la región  $\mathcal{A}_s$  viene delimitada por  $0 \leq k \leq K_s$  y  $1 \leq i \leq I_s$ , y,

$$\Psi_{k,i}^{n-1*} = \begin{cases} \Psi_{k,i}^{n-1} & (k,i) \in \mathcal{A}_s \\ \Psi_{k,i}^0 & \text{en otro caso} \end{cases} \quad (5.101)$$

Las componentes  $\Psi_{k,r_{i_0}}^n = E_k^u(\psi)$  son la cota superior ajustada de  $E_k(\psi)$ . Cuanto mayor sea  $n$ , más ajustada será la cota. En nuestro caso hemos tomado  $K_s = I_s = 100$

### 5.4.3 Cálculo del segundo momento de la longitud $E(l^2|d)$ , $1 \leq d \leq \Delta$

El segundo momento de la longitud del algoritmo es,

$$E(l^2|d) = \sum_{k=0}^{\infty} E_k(l^2) p(k|d) \quad (5.102)$$

donde,

$$p(k|d) = e^{-\lambda d} \frac{(\lambda d)^k}{k!} \quad (5.103)$$

Definimos, pues,  $E_k(l^2)$  como el segundo momento de la longitud de un CRI dado que empieza con una colisión de multiplicidad  $k$ .

Definimos también  $Q_{k,i}$  como el segundo momento del número de ranuras necesarias por el algoritmo para la transmisión exitosa de  $k$  paquetes, dado que el valor del contador es igual a  $r_i = i$  en la primera ranura.

Vamos a ver que en el algoritmo propuesto no se puede calcular de forma exacta  $E_k(l^2)$ . Nuestro objetivo será, pues, obtener una cota superior de  $E_k(l^2)$ , que llamaremos  $E_k^0(l^2)$ , utilizada para acotar el comportamiento en  $k \rightarrow \infty$ ; y a partir de ella obtendremos otra cota superior más ajustada, que llamaremos  $E_k^u(l^2)$ , que la utilizaremos para valores de  $k$  discretos. Así, obtendremos una cota superior bastante ajustada de  $E(l^2|d)$ , que llamaremos  $E^u(l^2|d)$ , y que será la que utilizaremos a todos los efectos como  $E(l^2|d)$ .

Según las definiciones dadas, se puede ver que  $E_k(l^2) = Q_{k,r_{i_0}}$ , donde  $r_{i_0}$  es el valor inicial. El primer paso será, pues, el cálculo de  $Q_{k,r_{i_0}}$ .

Cálculo de  $Q_{k,r_i}$ 

Sea  $P_{k,i,s}$  la probabilidad que, dado que en el primer slot  $k$  terminales entran en contienda con probabilidad  $1/r_i = 1/i$ , exactamente se necesiten  $s$  slots para que accedan los  $k$  terminales. Recordar que han accedido todos los terminales solo cuando una no colision ocurre y  $r_i = 1$ . Entonces se aplica la siguiente ecuacion recursiva,

$$P_{k,i,s} = \begin{cases} C_{k,i}P_{k,i+1,s-1} + S_{k,i}P_{k-1,i-1,s-1} & k \geq 0, i \geq 1 \\ + H_{k,i}P_{k,i-1,s-1} & \\ 0 & \text{en otro caso} \end{cases} \quad (5.104)$$

donde  $H_{k,i}$ ,  $S_{k,i}$ ,  $C_{k,i}$  vienen dados por las expresiones (5.58), (5.59) y (5.60) respectivamente.

Llamando,

$$Q_{k,i} = \sum_{s=0}^{\infty} s^2 P_{k,i,s} \quad (5.105)$$

llegamos a la siguiente relacion recursiva,

$$Q_{k,i} = \begin{cases} 2G_{k,i} - 1 + C_{k,i}Q_{k,i+1} + S_{k,i}Q_{k-1,i-1} & k \geq 2, i \geq 1 \\ + H_{k,i}Q_{k,i-1} & \\ 0 & 0 \leq k \leq 1, i \geq 1 \\ & \text{en otro caso} \end{cases} \quad (5.106)$$

Tenemos, pues, un sistema de ecuaciones lineales de dimension infinita de la forma,

$$Q_{k,i} = l_{k,i} + \sum_{q=0}^{\infty} \sum_{j=1}^{\infty} a_{q,j}^{k,i} Q_{q,j} \quad (5.107)$$

donde,

$$l_{k,i} = \begin{cases} 2G_{k,i} - 1 & k \geq 2, i \geq 1 \\ i^2 & 0 \leq k \leq 1, i \geq 1 \\ 0 & \text{en otro caso} \end{cases} \quad (5.108)$$

y,

$$a_{q,j}^{k,i} = \begin{cases} C_{k,i} & q = k, j = i + 1, k \geq 2, i \geq 1 \\ S_{k,i} & q = k - 1, j = i - 1, k \geq 2, i \geq 1 \\ H_{k,i} & q = k, j = i - 1, k \geq 2, i \geq 1 \\ 0 & \text{en otro caso} \end{cases} \quad (5.109)$$

Ya que no es posible resolver el sistema de ecuaciones, intentaremos encontrar límites de  $Q_{k,i}$ .

Sea,

$$Q_{k,i}^0 = \begin{cases} \alpha k^4 + \beta i^4 + \rho k^2 i^2 + \mu k^3 i + \eta k i^3 & k \geq 2, i \geq 1 \\ i^2 & 0 \leq k \leq 1, i \geq 1 \\ 0 & \text{en otro caso} \end{cases} \quad (5.110)$$

para  $i \geq 1$ , y definimos,

$$Q_{k,i}^1 = \begin{cases} 2G_{k,i}^0 - 1 + C_{k,i}Q_{k,i+1}^0 + S_{k,i}Q_{k-1,i-1}^0 \\ \quad + H_{k,i}Q_{k,i-1}^0 & k \geq 2, i \geq 1 \\ i^2 & 0 \leq k \leq 1, i \geq 1 \\ 0 & \text{en otro caso} \end{cases} \quad (5.111)$$

Después de operar, obtenemos los siguientes resultados,

$$Q_{k,i}^1 = \begin{cases} Q_{k,i}^0 + M_{k,i}(2C_{k,i} - 1) + N_{k,i}S_{k,i} + O_{k,i} & k \geq 2, i \geq 1 \\ Q_{k,i}^0 & \text{en otro caso} \end{cases} \quad (5.112)$$

donde,

$$M_{k,i} = 2k^2 i \rho + k^3 \mu + k(1 + 3i^2)\eta + 4i(1 + i^2)\beta \quad k \geq 2, i \geq 1 \quad (5.113)$$

$$N_{k,i} = \begin{cases} (1-i)^3 \eta + (1-i)(1-3k+3k^2)\mu \\ \quad + (1-i)^2(1-2k)\rho & k > 2, i \geq 1 \\ + (1-4k^3 + 6k^2 - 4k)\alpha \\ (1-i)^2(1-k^2\rho) + k^3(1-i)\mu & k = 2, i \geq 1 \\ + k(1-i)^3 \eta - k^4 \alpha - (1-i)^4 \beta \end{cases} \quad (5.114)$$

$$O_{k,i} = 2G_{k,i}^0 - 1 + k^2 \rho + 3ki\eta + (6i^2 + 1)\beta \quad k \geq 2, i \geq 1 \quad (5.115)$$

Se puede ver que  $Q_{k,i}^1 \leq Q_{k,i}^0$  si,

$$M_{k,i}(2C_{k,i} - 1) + N_{k,i}S_{k,i} + O_{k,i} \leq 0 \quad k \geq 2, i \geq 1 \quad (5.116)$$

Tenemos una restricción lineal por cada par  $(k, i)$ . Procedemos como sigue,

1. Aplicamos el método dual simplex de programación lineal que se pueden encontrar en diversos libros, entre ellos el de S. Gass [71] para la Región  $R_1$ , dada por  $k = 0 \dots 20$  y  $i = 1 \dots 20$ , es decir, 420 restricciones lineales. La Región  $R_1$  está representada gráficamente en la figura 4.2. La función a minimizar escogida es la suma de todos los coeficientes.
2. Comprobamos que  $f$  es negativa en la región  $R_2$ , dada por  $k = 0 \dots 100$  y  $i = 1 \dots 100$ . La Región  $R_2$  está representada gráficamente en la figura 4.2. Hay que hacer notar que  $R_2$  incluye  $R_1$ , y que esta compuesto por un número finito de puntos.
3. Demostramos que los coeficientes obtenidos satisfacen las restricciones en la Región  $R_3$ , dada por todo par  $(k, i)$  tal que  $k > 100$  o  $i > 100$ . La Región  $R_3$  está representada gráficamente en la figura 4.2.

De esta forma,  $Q_{k,i}^0$  sera un límite superior de  $Q_{k,i}$ .  
El resultado obtenido es,

$$Q_{k,i}^0 = \begin{cases} 2,1218k^4 + 0,4933i^4 - 1,3141k^2i^2 & k \geq 2, i \geq 1 \\ i^2 & 0 \leq k \leq 1, i \geq 1 \\ 0 & \text{en otro caso} \end{cases} \quad (5.117)$$

De esta forma,  $Q_{k,i}^0$  sera un límite superior de  $Q_{k,i}$ . Y, por tanto, tenemos una cota superior de  $E_k(i^2)$ ,  $E_k(i^2) = Q_{k,r_{i_0}}^0 = Q_{k,1}^0$ .

**Demostración.** El objetivo de este apartado es demostrar analíticamente que los coeficientes obtenidos satisfacen las restricciones para un par general  $(k, i)$ . El método a seguir sería totalmente paralelo al especificado en Apéndice del Capítulo anterior.

Cálculo de  $E(i^2|d)$ ,  $1 \leq d \leq \Delta$

Sea  $E_k(i^2)$  el segundo momento de la longitud de un CRI dado que empieza con una colision de multiplicidad  $k$ .

Se puede ver que  $E_k(i^2) = Q_{k,r_{i_0}}$ , donde  $r_{i_0}$  es el valor inicial del contador. De los resultados anteriores conocemos un límite superior de  $Q_{k,i}$ , que nos servirá para calcular  $E_k(i^2)$ .

El segundo momento de la longitud de nuestro algoritmo se calcula como,

$$E(i^2|d) = \sum_{k=0}^{\infty} E_k(i^2)p(k|d) \quad (5.118)$$

donde,

$$p(k|d) = e^{-\lambda d} \frac{(\lambda d)^k}{k!} \quad (5.119)$$

Ya hemos visto que no se puede calcular de forma exacta  $E_k(I^2)$ . Tenemos un límite superior analítico  $E_k^0(I^2) = Q_{k,r_{10}}^0$ , que utilizaremos para acotar el comportamiento en  $k \rightarrow \infty$ . Para calcular la longitud media, que a todos los efectos substituiremos por una cota ligeramente superior  $E^u(I^2|d)$ , necesitamos una cota mas precisa de  $E_k(I^2)$  que llamaremos  $E_k^u(I^2) = Q_{k,r_{10}}^u$ , y que utilizaremos para  $k$  discretos. Así tenemos,

$$E^u(I^2|d) = \sum_{k=0}^{30} E_k^u(I^2)p(k|d) + \sum_{k=31}^{\infty} E_k^0(I^2)p(k|d) \quad (5.120)$$

Sabiendo que,

$$\sum_{k=0}^{\infty} e^{-x} \frac{(x)^k}{k!} = 1 \quad (5.121)$$

$$\sum_{k=0}^{\infty} k e^{-x} \frac{(x)^k}{k!} = x \quad (5.122)$$

$$\sum_{k=0}^{\infty} k^2 e^{-x} \frac{(x)^k}{k!} = x(1+x) \quad (5.123)$$

$$\sum_{k=0}^{\infty} k^3 e^{-x} \frac{(x)^k}{k!} = x(1+3x+x^2) \quad (5.124)$$

$$\sum_{k=0}^{\infty} k^4 e^{-x} \frac{(x)^k}{k!} = x(1+7x+6x^2+x^3) \quad (5.125)$$

Y despues de algunas manipulaciones algebraicas se obtiene que,

$$\begin{aligned}
E^u(I^2|d) &= \sum_{k=0}^{30} E_k^u(I^2)p(k|d) + \beta r_{t_0}^4 \left\{ 1 - \sum_{k=0}^{30} p(k|d) \right\} \\
&+ \eta r_{t_0}^3 \left\{ \lambda d - \sum_{k=0}^{30} k p(k|d) \right\} + \rho r_{t_0}^2 \left\{ (\lambda d)(1 + \lambda d) - \sum_{k=0}^{30} k^2 p(k|d) \right\} \\
&+ \mu r_{t_0} \left\{ (\lambda d)(1 + 3\lambda d + (\lambda d)^2) - \sum_{k=0}^{30} k^3 p(k|d) \right\} \\
&+ \alpha \left\{ (\lambda d)(1 + 7\lambda d + 6(\lambda d)^2 + (\lambda d)^3) - \sum_{k=0}^{30} k^3 p(k|d) \right\}
\end{aligned} \tag{5.126}$$

Una vez tenemos  $Q_{k,i}^0$ , expresión (5.117), podemos conseguir una cota superior más ajustada de  $E_k(I^2)$ ,  $E_k^u(I^2)$ , con el siguiente procedimiento. Sea  $\mathcal{M}^0$  una matriz bidimensional ( $K_s \times I_s$ ) cuyos componentes son  $Q_{k,i}^0$  y sea  $\mathcal{M}^n$  la matriz bidimensional ( $K_s \times I_s$ ) cuyos componentes se obtienen de forma recursiva,

$$Q_{k,i}^n = \begin{cases} 1 + C_{k,i} Q_{k,i+1}^{n-1*} + S_{k,i} Q_{k-1,i-1}^{n-1*} + H_{k,i} Q_{k,i-1}^{n-1*} & (k,i) \in \mathcal{A}_s \\ G_{k,i}^0 & \text{en otro caso} \end{cases} \tag{5.127}$$

donde la región  $\mathcal{A}_s$  viene delimitada por  $0 \leq k \leq K_s$  y  $1 \leq i \leq I_s$ , y,

$$Q_{k,i}^{n-1*} = \begin{cases} Q_{k,i}^{n-1} & (k,i) \in \mathcal{A}_s \\ Q_{k,i}^0 & \text{en otro caso} \end{cases} \tag{5.128}$$

Las componentes  $Q_{k,r_{t_0}}^n = E_k^u(I^2)$  son la cota superior ajustada de  $E_k(I^2)$ . Cuanto mayor sea  $n$ , más ajustada será la cota. En este caso, tomaremos  $K_s = I_s = 100$ .





## Capítulo 6

# Análisis del throughput, con errores en el feedback, caso feedback ternario

### 6.1 Introducción

En el Capítulo 4 se ha analizado el throughput que es capaz de absorber el algoritmo ALOHA estabilizado de ventana. En él, la información procedente del feedback era binaria, es decir, los usuarios eran capaces de distinguir entre una colisión y una no colisión. Con esta información la probabilidad de acceso de los usuarios se actualizaba de forma que intuitivamente se tendía a una probabilidad que correspondía al inverso del número de usuarios que había envueltos en la colisión.

Una pregunta que aparece de forma natural es: Por qué nos restringimos a un feedback binario? Cuanto mejoraría el algoritmo si pudieran disponer los usuarios de más información, sin romper con las premisas dadas en la descripción de un canal multiacceso?

En este Capítulo vamos a proponer otro algoritmo, que corresponderá con la versión ternaria del algoritmo WSA. Así pues, en este algoritmo los usuarios serán capaces de distinguir entre una ranura vacía y una ranura donde ha habido una transmisión exitosa. Por tanto, el algoritmo ha de aprovechar esta cantidad de información suplementaria que se le ofrece con el fin de mejorar en prestaciones.

Parece intuitivo que el hecho de tener más información en el feedback repercutirá en un incremento del throughput, pero no nos asegura si se va a mejorar el comportamiento frente al retardo y frente a la sensibilidad con respecto a los errores en el feedback.

Antes de entrar en el análisis del algoritmo WSA en su versión ternaria, podemos razonar parte de su comportamiento. A la hora de diseñar la operación del algoritmo, se ha de tener en cuenta que básicamente consistirá en hacer crecer y decrecer apropiadamente el contador propio de los usuarios tras obtener información del feedback. En la versión binaria del algoritmo el contador se incrementaba si había colisión y se decrementaba si había no colisión. A primera vista, parece que el siguiente paso hacia la versión ternaria sería que en el caso de ranura vacía, el contador no se moviera porque intuitivamente es la mejor opción. Meditando el planteamiento se llega a la conclusión que no es posible esta solución porque si llegamos al supósito que no hay más paquetes a transmitir y el contador es mayor que 1, nos quedaríamos indefinidamente en este estado, porque permanentemente tendríamos ranuras vacías.

La solución escogida, que no pretende ser la óptima, es disminuir el contador en una unidad en el caso de ranura vacía y disminuirla en dos en el caso de transmisión exitosa. Así pues, el algoritmo ternario y binario tendrán un comportamiento distinto cuando el contador  $r_t$  llegue a 3. La probabilidad de este evento es relativamente baja y, por tanto, lo primero que se deduce del comportamiento del algoritmo ternario es que indudablemente va a mejorar con respecto al binario, pero que esta mejora no será ni mucho menos espectacular.

## 6.2 El modelo del sistema y el algoritmo

Como ya se explicó en el Capítulo 3, un sistema de acceso aleatorio está formado por un protocolo de acceso al canal (CAP) y por un algoritmo de resolución de colisiones (CRA). El algoritmo en su versión ternaria va a coincidir con la versión binaria en el protocolo de acceso al canal.

Las diferencias entre ambos algoritmos radica en la parte del algoritmo de resolución de colisiones. En el caso de la versión ternaria,  $x_t$  denota el feedback que corresponde a la ranura  $t$ ; donde  $x_t = e$ ,  $x_t = 1$  y  $x_t = 0$  representan respectivamente colisión (más de un paquete intenta transmitir), transmisión exitosa (un único paquete intenta transmitir) y ranura vacía (ningún paquete intenta transmitir) en la ranura  $t$ , respectivamente.

Durante los CRI que empiezan con una ranura de colisión, cada usuario actúa independientemente vía la utilización de un contador cuyo valor en el tiempo  $t$  se denota por  $r_t$ . Este contador tiene el mismo valor para todos los usuarios, pero es cada usuario quien actualiza su propio contador dependiendo del feedback. Cuando un usuario transmite por primera vez, fija  $r_t = r_{t_0} = 1$ . Los valores del contador son enteros, y son actualizados y

utilizados de acuerdo con las reglas siguientes,

1. Los usuarios transmiten en la ranura  $t$  con probabilidad  $1/r_t$ .
2. Los valores del contador cambian con el tiempo de la manera siguiente:
  - (a) Si  $x_{t-1} = 0$ , entonces  $r_t = r_{t-1} - 1$
  - (b) Si  $x_{t-1} = 1$ , entonces  $r_t = \max\{r_{t-1} - 2, 1\}$
  - (c) Si  $x_{t-1} = e$ , entonces  $r_t = r_{t-1} + 1$

De estas reglas se puede ver que un CRI acaba con una ranura de no colisión y  $r_t = 1$ .

### 6.3 Análisis del throughput

En esta sección se describe el análisis del throughput del algoritmo WSA en su versión ternaria.

La metodología que vamos a utilizar en el cálculo del throughput del algoritmo WSA en su versión ternaria es totalmente paralela a la que ya se describió para la versión binaria. Ya en su momento se explicó que todo algoritmo que utilizara el mismo protocolo de acceso al canal podría analizarse por este camino; la única diferencia es que cada uno tendría una  $E(i|\Delta)$  distinta. Estamos indudablemente en uno de estos casos y, por tanto, las diferencias entre el algoritmo binario en lo que respecta al throughput vendrá directamente condicionado por sus valores de  $E(i|\Delta)$ .

Vamos a resumir, pues, la metodología del cálculo del throughput. Partíamos de la condición de estabilidad dada por J. Huang y T. Berger [74], que indica que la cadena de Markov es ergódica y el sistema es estable si y sólo si,

$$E(i|\Delta) < \Delta \quad (6.1)$$

donde  $E(i|\Delta)$  denota la longitud esperada de un CRI dado que empieza con un intervalo examinado de longitud  $\Delta$ .

Si denotamos como  $E_k(i)$  la longitud esperada de un CRI dado que empieza con una colisión de multiplicidad  $k$ , podemos entonces escribir,

$$E(i|\Delta) = \sum_{k=0}^{\infty} E(i|\Delta, k)p(k|\Delta) = \sum_{k=0}^{\infty} E_k(i)p(k|\Delta) \quad (6.2)$$

ya que,

$$E(i|\Delta, k) = E_k(i) \quad (6.3)$$

depende sólo de  $k$ , y donde,

$$p(k|\Delta) = e^{-\lambda\Delta} \frac{(\lambda\Delta)^k}{k!} \tag{6.4}$$

En el Apéndice de este Capítulo mostraremos cómo,

1.  $E_k(i)$ ,  $k \geq 0$  puede ser recursiva y superiormente acotada de forma ajustada,  $E_k(i) \leq E_k^u(i)$ , ver Sección 6.5.3 del Apéndice.
2.  $E_k(i)$  es acotada cuadráticamente,  $E_k(i) \leq E_k^0(i) = c_2k^2 + c_1k + c_0$ , ver Sección 6.5.4 del Apéndice.

La expresión (6.1) junto con los resultados de los puntos 1 y 2 son usados en la computación del throughput del algoritmo, siguiendo exáctamente el camino marcado en el Capítulo 4.

Debido a la cota superior de  $E(i|\Delta)$ ,  $E^u(i|\Delta)$  y a la inecuación (6.1), llegamos a que la siguiente condición es suficiente para la estabilidad,

$$E^u(i|\Delta) < \Delta \tag{6.5}$$

con,

$$E^u(i|\Delta) = \sum_{k=0}^{30} E_k^u(i)p(k|\Delta) + \sum_{k=31}^{\infty} E_k^0(i)p(k|\Delta) \tag{6.6}$$

Después de algunas manipulaciones algebraicas, concluimos que la expresión (6.6) es equivalente a,

$$\begin{aligned} E^u(i|\Delta) = & \sum_{k=0}^{30} E_k^u(i)p(k|\Delta) + \alpha \left\{ (\lambda\Delta)^2 + \lambda\Delta - \sum_{k=0}^{30} k^2 p(k|\Delta) \right\} \\ & + (\rho r_{t_0} + \mu) \left\{ \lambda\Delta - \sum_{k=0}^{30} k p(k|\Delta) \right\} \\ & + (\beta r_{t_0}^2 + \eta r_{t_0} + \gamma) \left\{ 1 - \sum_{k=0}^{30} p(k|\Delta) \right\} \end{aligned} \tag{6.7}$$

ya que,

$$E_k^0(i) = G_{k,r_{t_0}}^0 = \alpha k^2 + \beta r_{t_0}^2 + \rho k r_{t_0} + \mu k + \eta r_{t_0} + \gamma \tag{6.8}$$

Si definimos ahora,

$$E^u(i|\Delta) \equiv f(\lambda\Delta) \tag{6.9}$$

Algoritmo WSA ternario	$\lambda^* = 0.4365$	$\Delta^* = 2.38$
Algoritmo WSA binario	$\lambda^* = 0.4335$	$\Delta^* = 2.49$

Tabla 6.1: Throughputs y tamaños de ventana óptimos. Los algoritmos representados son el WSA (ALOHA estabilizado de ventana) en su versión ternaria y binaria.

$$x \equiv \lambda \Delta \quad (6.10)$$

entonces, de las expresiones (6.7) y (6.10) resulta que, para la estabilidad del algoritmo, es suficiente que la tasa de entrada  $\lambda$  satisfaga,

$$\lambda < \sup_{x \geq 0} \frac{x}{f(x)} \quad (6.11)$$

La siguiente condición especifica una región de valores de  $\lambda$  para los cuales el algoritmo es inestable,

$$\lambda > \sup_{x \geq 0} \frac{x}{g(x)} \quad (6.12)$$

donde,

$$g(x) = \sum_{k=0}^{30} E_k^u e^{-x} \frac{x^k}{k!} \quad (6.13)$$

La maximización de las expresiones (6.11) y (6.12) ha sido hecha numéricamente y proporciona el throughput ( $\lambda^*$ ) así como el tamaño de ventana óptimo ( $\Delta^*$ ). En todos los casos el orden de magnitud de la diferencia es menor que  $10^{-4}$ . El tamaño de ventana óptimo se encuentra como  $x^*(\lambda^*)^{-1}$ , donde  $x^*$  es el valor que alcanza el supremo en la expresión (6.11). Los resultados se reflejan en la tabla 6.1. En ella, se compara tanto la  $\lambda^*$  como la  $\Delta^*$  de los algoritmos WSA ternario y binario. Una conclusión inmediata que se puede extraer es que la mejora del throughput existe, como era previsible, pero se puede considerar que no compensa la complicación que supone el añadir un feedback ternario.

## 6.4 Sensibilidad a los errores en el feedback

De la misma forma que en el análisis del algoritmo binario, asumimos que debido a condiciones de ruido, pueden ocurrir los siguientes tipos de errores en el feedback. Con probabilidad  $\epsilon$  una ranura vacía puede ser vista por los

$\epsilon$	$\delta$	Algoritmo WSA	
		ternario	binario
0.00	0.00	0.4365	0.4335
0.00	0.01	0.4324	0.4292
0.00	0.10	0.3951	0.3904
0.00	0.20	0.3531	0.3473
0.00	0.40	0.2671	0.2610
0.00	0.50	0.2233	0.2179
0.01	0.00	0.4354	0.4321
0.10	0.00	0.4245	0.4173
0.20	0.00	0.4089	0.3940
0.40	0.00	0.3473	0.2814
0.45	0.00	0.3045	0.2006
0.10	0.10	0.3823	0.3733
0.20	0.20	0.3219	0.3050
0.30	0.30	0.2519	0.2252
0.30	0.50	0.1692	0.1508
0.30	0.70	0.0936	0.0854

Tabla 6.2: Throughputs en función de  $\epsilon$  y  $\delta$ . Los tamaños de ventana son optimizados para cada par  $(\epsilon, \delta)$ . Los algoritmos representados son el WSA (ALOHA estabilizado de ventana) en su versión ternaria y binaria.

usuarios como una ranura de colisión. De la misma forma, con probabilidad  $\delta$  una ranura ocupada por una transmisión sencilla puede ser vista por los usuarios como una ranura de colisión. Asumimos que una ranura de colisión es siempre reconocida correctamente por los usuarios. Consideramos el caso donde las probabilidades  $\epsilon$  y  $\delta$ , siendo características del sistema, son conocidas a priori. Entonces dados  $\epsilon$  y  $\delta$ , el tamaño de la ventana  $\Delta$  es optimizado para la maximización del throughput. Expondremos los resultados del análisis del throughput tanto para el algoritmo WSA ternario como binario. Los resultados son expuestos en la tabla 6.2.

En algunos sistemas, las probabilidades  $\epsilon$  y  $\delta$  pueden no ser conocidas a priori. En este caso, un algoritmo puede ser diseñado sujeto a la suposición de feedback libre de error. Encontramos los throughputs para el algoritmo WSA ternario y binario para varios valores de las probabilidades de error, que son las máximas tasas de Poisson para los cuales la condición (6.1) se satisface sujeto a la restricción que las ventanas son aquellas de la tabla 6.1. Nuestros resultados están expuestos en la tabla 6.3. Se ha podido observar que el máximo valor para el que el algoritmo tanto ternario como binario

$\epsilon$	$\delta$	Algoritmo WSA	
		ternario	binario
0.00	0.00	0.436	0.433
0.00	0.01	0.432	0.429
0.00	0.10	0.394	0.389
0.00	0.40	0.253	0.250
0.01	0.00	0.435	0.432
0.10	0.00	0.421	0.414
0.20	0.00	0.388	0.375
0.30	0.00	0.269	0.259
0.10	0.10	0.374	0.365
0.10	0.20	0.324	0.317
0.10	0.30	0.273	0.268
0.20	0.10	0.331	0.319
0.20	0.20	0.273	0.265
0.20	0.30	0.216	0.212

Tabla 6.3: Throughputs en función de  $\epsilon$  y  $\delta$ . Los tamaños de ventana fijos para cada par  $(\epsilon, \delta)$ . Los algoritmos representados son el WSA (ALOHA estabilizado de ventana) en su versión ternario y binario.

es estable es  $\epsilon = 0.32$ . De las tablas 6.2 y 6.3 se llega a la conclusión que dentro del caso considerado real  $0 \leq \epsilon \leq 0.1$  y  $0 \leq \delta \leq 0.1$  el algoritmo WSA ternario consigue mayor throughput que el algoritmo WSA binario, aunque ésta es una ligera mejoría.

## 6.5 Apéndice

### 6.5.1 Análisis de estabilidad

El análisis de estabilidad para el algoritmo WSA ternario (ALOHA estabilizado de ventana ternario) es totalmente paralelo al ya desarrollado para el WSA binario (ALOHA estabilizado de ventana binario). Por tanto, la región de estabilidad del algoritmo surge de la inecuación (6.1). Siguiendo con la misma línea de actuación, el objetivo inmediato será el cálculo de cotas superiores de  $E_k(i)$ ,  $k \geq 0$ .

Presentamos el análisis de estabilidad en el caso que puedan ocurrir errores en el feedback. Substituyendo  $\epsilon = \delta = 0$  en nuestros resultados, obtenemos las cantidades correspondientes en el caso libre de errores. La región de estabilidad del algoritmo proviene de la inecuación (6.1), donde el valor de



$E(l|\Delta)$  en ella es dado por la expresión (6.2). Empezamos con el desarrollo de una cota superior ajustada en los valores esperados  $E_k(l)$ ,  $k \geq 0$ .

### 6.5.2 Planteamiento del problema, $G_{k,i}$

Definimos  $P_{k,i,s}$  como la probabilidad que se necesiten  $s$  ranuras por el algoritmo para la transmisión exitosa de  $k$  paquetes, dado que el valor del contador es igual a  $r_i = i$ .

De la operación del algoritmo se puede considerar la cadena de Markov que envuelve el estado tridimensional  $(k, i, s)$  donde el proceso puede estar, donde  $k = 0, 1, \dots$ ,  $i = 1, 2, \dots$ ,  $s = 0, 1, \dots$  son definidos anteriormente.

Sean  $H_{k,i}$ ,  $S_{k,i}$  y  $C_{k,i}$  las probabilidades tales que, dados  $k$  paquetes en contienda, 0, 1, o más de 1 intentan transmitir, respectivamente. Éstas son dadas por,

$$H_{k,i} = \begin{cases} (1 - \frac{1}{i})^k & k > 0, i \geq 1 \\ 1 & k = 0, i \geq 1 \end{cases} \quad (6.14)$$

$$S_{k,i} = \begin{cases} \frac{k}{i} (1 - \frac{1}{i})^{k-1} & k \geq 0, i \geq 1 - \{(k = 0, i = 1), (k = 1, i = 1)\} \\ 1 & k = 1, i = 1 \\ 0 & k = 0, i = 1 \end{cases} \quad (6.15)$$

$$C_{k,i} = 1 - S_{k,i} - H_{k,i} \quad (6.16)$$

Según la operación del algoritmo, la distribución de probabilidad es dada por la siguiente ecuación recursiva,

$$P_{k,i,s} = \begin{cases} (C_{k,i} + \epsilon H_{k,i} + \delta S_{k,i})P_{k,i+1,s-1} + S_{k,i}(1 - \delta)P_{k-1,i-1,s-1} + H_{k,i}(1 - \epsilon)P_{k,i-1,s-1} & 1 \leq i \leq 2 \\ (C_{k,i} + \epsilon H_{k,i} + \delta S_{k,i})P_{k,i+1,s-1} + S_{k,i}(1 - \delta)P_{k-1,i-2,s-1} + H_{k,i}(1 - \epsilon)P_{k,i-1,s-1} & i > 2 \\ 0 & \text{en otro caso} \end{cases} \quad (6.17)$$

con  $k \geq 0$ .

Tomando esperanzas en la expresión (6.17) respecto a  $s$ ,

$$G_{k,i} = \sum_{s=0}^{\infty} sP_{k,i,s} \quad (6.18)$$

se obtiene,

$$G_{k,i} = \begin{cases} 1 + (C_{k,i} + \epsilon H_{k,i} + \delta S_{k,i})G_{k,i+1} \\ \quad + S_{k,i}(1 - \delta)G_{k-1,i-1} + H_{k,i}(1 - \epsilon)G_{k,i-1} & k \geq 2, 1 \leq i \leq 2 \\ 1 + (C_{k,i} + \epsilon H_{k,i} + \delta S_{k,i})G_{k,i+1} \\ \quad + S_{k,i}(1 - \delta)G_{k-1,i-2} + H_{k,i}(1 - \epsilon)G_{k,i-1} & k \geq 2, i > 2 \\ \frac{i}{1-2\epsilon} & k = 0, i \geq 1 \\ \frac{1}{1-2\epsilon} \left( i + \frac{2(\delta-\epsilon)}{1-\delta} \right) & k = 1, 1 \leq i \leq 2 \\ \frac{1}{1-2\epsilon} \left( i - 1 + \frac{2(\delta-\epsilon)}{1-\delta} \right) & k = 1, i > 2 \\ 0 & \text{en otro caso} \end{cases} \quad (6.19)$$

Así, definiremos  $G_{k,i}$  como el esperado número de ranuras necesarias por el algoritmo para la transmisión exitosa de  $k$  paquetes, dado que el valor del contador es igual a  $r_i = i$  en la primera ranura.

Notar que  $E_k(l) = G_{k,r_{i_0}} = G_{k,1}$ .

### 6.5.3 Desarrollo de una cota superior de $E_k(l)$ , $E_k^0(l)$

Procedemos ahora al desarrollo de una cota superior inicial de la solución del sistema (6.19),  $E_k^0(l)$ . La ecuación (6.19) comprende un sistema numerable de ecuaciones lineales de la forma,

$$G_{k,i} = b_{k,i} + \sum_{q=0}^{\infty} \sum_{j=1}^{\infty} h_{q,j}^{k,i} G_{q,j} \quad k, i \in \mathcal{I} \quad (6.20)$$

donde  $\mathcal{I}$  es un conjunto numerable de índices, y los coeficientes  $h_{q,j}^{k,i}$  y  $b_{k,i}$  son no negativos para cada  $k, i, q, j \in \mathcal{I}$ , donde,

$$b_{k,i} = \begin{cases} 1 & k \geq 2, i \geq 1 \\ \frac{i}{1-2\epsilon} & k = 0, i \geq 1 \\ \frac{1}{1-2\epsilon} \left( i + \frac{2(\delta-\epsilon)}{1-\delta} \right) & k = 1, 1 \leq i \leq 2 \\ \frac{1}{1-2\epsilon} \left( i - 1 + \frac{2(\delta-\epsilon)}{1-\delta} \right) & k = 1, i > 2 \\ 0 & \text{en otro caso} \end{cases} \quad (6.21)$$

y,

$$h_{q,j}^{k,i} = \begin{cases} C_{k,i} + \epsilon H_{k,i} + \delta S_{k,i} & q = k, j = i + 1, k \geq 2, i \geq 1 \\ S_{k,i}(1 - \delta) & q = k - 1, j = i - 1, k \geq 2, 1 \leq i \leq 2 \\ S_{k,i}(1 - \delta) & q = k - 1, j = i - 2, k \geq 2, i > 2 \\ H_{k,i}(1 - \epsilon) & q = k, j = i - 1, k \geq 2, i \geq 1 \\ 0 & \text{en otro caso} \end{cases} \quad (6.22)$$

Este sistema es de la misma forma que el sistema (4.24). Por consiguiente, la metodología de análisis es la misma que la seguida en aquella ocasión. A continuación se exponen los pasos seguidos de forma resumida.

Supongamos que una secuencia  $\{G_{q,j}; q, j \in \mathcal{I}\}$  existe tal que,

$$0 \leq G_{k,i}^0 < \infty \tag{6.23}$$

$$G_{k,i}^0 \leq b_{k,i} + \sum_{q=0}^{\infty} \sum_{j=1}^{\infty} h_{q,j}^{k,i} G_{q,j}^0 \tag{6.24}$$

Definimos,

$$G_{k,i}^{n+1} = \sum_{q=0}^{\infty} \sum_{j=1}^{\infty} h_{q,j}^{k,i} G_{q,j}^n + b_{k,i} \quad k, i \in \mathcal{I}, n = 0, 1, 2, \dots \tag{6.25}$$

Observamos que de (6.23) y (6.24) y el hecho que  $h_{q,j}^{k,i}$  y  $b_{k,i}$  son no negativos para cada  $k, i, q, j \in \mathcal{I}$ , tenemos que,

$$0 \leq G_{k,i}^{n+1} \leq G_{k,i}^n < \infty \tag{6.26}$$

para cada  $k, i \in \mathcal{I}$  y cada  $n \geq 0$ .

Aplicando el Lema 1 ya expuesto en el Apéndice del Capítulo 4, si una secuencia  $\{G_{q,j}^0; q, j \in \mathcal{I}\}$  existe tal que satisface las inecuaciones (6.23) y (6.24), el sistema (6.20) tiene una solución finita no negativa  $\{G_{q,j}^*; q, j \in \mathcal{I}\}$  y  $G_{q,j}^*$  es el límite de la secuencia monótona no creciente  $\{G_{q,j}^n; n \geq 0\}$ , donde  $G_{q,j}^n$  se define en la expresión (6.25). En nuestro caso, para el sistema (6.19) se pueden usar secuencias cuadráticas,

$$G_{k,i}^0 = \begin{cases} \alpha k^2 + \beta i^2 + \rho ki + \mu k + \eta i + \gamma & k \geq 2, i \geq 1 \\ \frac{i}{1-2\epsilon} & k = 0, i \geq 1 \\ \frac{1}{1-2\epsilon} \left( i + \frac{2(\delta-\epsilon)}{1-\delta} \right) & k = 1, 1 \leq i \leq 2 \\ \frac{1}{1-2\epsilon} \left( i - 1 + \frac{2(\delta-\epsilon)}{1-\delta} \right) & k = 1, i > 2 \\ 0 & \text{en otro caso} \end{cases} \tag{6.27}$$

Estas secuencias tienen que satisfacer las inecuaciones (6.23) y (6.24), es decir, se tienen que determinar  $\alpha, \beta, \rho, \mu, \eta, \gamma$ .

Para  $n = 0$  en la expresión (6.25),

$$G_{k,i}^1 = \begin{cases} 1 + (C_{k,i} + \epsilon H_{k,i} + \delta S_{k,i}) G_{k,i+1}^0 \\ \quad + S_{k,i}(1 - \delta) G_{k-1,i-1}^0 + H_{k,i}(1 - \epsilon) G_{k,i-1}^0 & k \geq 2, 1 \leq i \leq 2 \\ 1 + (C_{k,i} + \epsilon H_{k,i} + \delta S_{k,i}) G_{k,i+1}^0 \\ \quad + S_{k,i}(1 - \delta) G_{k-1,i-2}^0 + H_{k,i}(1 - \epsilon) G_{k,i-1}^0 & k \geq 2, i > 2 \\ \frac{i}{1-2\epsilon} & k = 0, i \geq 1 \\ \frac{1}{1-2\epsilon} \left( i + \frac{2(\delta-\epsilon)}{1-\delta} \right) & k = 1, 1 \leq i \leq 2 \\ \frac{1}{1-2\epsilon} \left( i - 1 + \frac{2(\delta-\epsilon)}{1-\delta} \right) & k = 1, i > 2 \\ 0 & \text{en otro caso} \end{cases} \quad (6.28)$$

Substituyendo  $G_{k,i}^0$  en la expresión (6.28), se puede ver que  $G_{k,i}^1$  es de la forma,

$$G_{k,i}^1 = \begin{cases} G_{k,i}^0 + M_{k,i}(1 - 2(1 - \epsilon)H_{k,i} - 2(1 - \delta)S_{k,i}) \\ \quad + N_{k,i}S_{k,i} + O_{k,i} & k \geq 2, i \geq 1 \\ G_{k,i}^0 & \text{en otro caso} \end{cases} \quad (6.29)$$

donde,

$$M_{k,i} = 2i\beta + \rho k + \eta \quad k \geq 2, i \geq 1 \quad (6.30)$$

$$N_{k,i} = \begin{cases} (1 - \delta)(\alpha(1 - 2k) - \rho(i - 1)) + J_{k,i} & k > 2, i \geq 1 \\ \frac{(5-3i)\delta + 2(i-2)\epsilon + F_{k,i}}{1-2\epsilon} & k = 2, i \geq 1 \\ -(1 - \delta)(\alpha k^2 + \beta(i - 1)^2 + \rho k(i - 1) + \mu k + \gamma) & k = 2, i > 1 \end{cases} \quad (6.31)$$

$$O_{k,i} = 1 + \beta \quad k \geq 2, i \geq 1 \quad (6.32)$$

con,

$$J_{k,i} = \begin{cases} 0 & k > 2, 1 \leq i \leq 2 \\ (1 - \delta)(-\rho(k - 1) - \mu + \beta(3 - 2i)) - 2\frac{\delta-\epsilon}{1-2\epsilon} & k > 2, i > 2 \end{cases} \quad (6.33)$$

y,

$$F_{k,i} = \begin{cases} i - 1 & k = 2, 1 \leq i \leq 2 \\ i - 1 - (1 - \delta) & k = 2, 3 \leq i \leq 4 \\ i - 1 - 2(1 - \delta) & k = 2, i > 4 \end{cases} \quad (6.34)$$

Se puede ver que  $G_{k,i}^1 \leq G_{k,i}^0$ , si,

$$M_{k,i}(2C_{k,i} - 1) + N_{k,i}S_{k,i} + O_{k,i} \leq 0 \quad k \geq 2, i \geq 1 \quad (6.35)$$

Dados  $\delta$  y  $\epsilon$  por el sistema, tenemos una restricción lineal por cada par  $(k, i)$ . Procedemos como sigue,

1. Aplicamos el método dual simplex de programación lineal que se pueden encontrar en diversos libros, entre ellos el de S. Gass [71] para la Región  $R_1$ , dada por  $k = 0 \dots 20$  y  $i = 1 \dots 20$ , es decir, 420 restricciones lineales. La Región  $R_1$  está representada gráficamente en la figura 4.2. La función a minimizar escogida es la suma de todos los coeficientes.
2. Comprobamos que  $f$  es negativa en la región  $R_2$ , dada por  $k = 0 \dots 100$  y  $i = 1 \dots 100$ . La Región  $R_2$  está representada gráficamente en la figura 4.2. Hay que hacer notar que  $R_2$  incluye  $R_1$ , y que esta compuesto por un número finito de puntos.
3. Demostramos que los coeficientes obtenidos satisfacen las restricciones en la Región  $R_3$ , dada por todo par  $(k, i)$  tal que  $k > 100$  o  $i > 100$ . La Región  $R_3$  está representada gráficamente en la figura 4.2.

**Demostración**

Dados  $\delta$  y  $\epsilon$  el objetivo de este apartado es demostrar analíticamente que los coeficientes obtenidos satisfacen las restricciones para un par general  $(k, i)$ . El método a seguir sería totalmente paralelo al especificado en Apéndice del Capítulo 4.

**6.5.4 Desarrollo de una cota superior ajustada de  $E_k(i)$ ,  $E_k^u(i)$**

Una vez tenemos  $G_{k,i}^0$  de la ecuación (6.27) y los coeficientes determinados, podemos conseguir una cota superior más ajustada de  $E_k(i)$ ,  $E_k^u(i)$  con el siguiente procedimiento. Sea  $M^0$  una matriz bidimensional  $(K_s \times I_s)$  cuyos componentes son  $G_{k,i}^0$  y sea  $M^n$  la matriz bidimensional  $(K_s \times I_s)$  cuyos componentes se obtienen de forma recursiva

$$G_{k,i}^n = \begin{cases} 1 + (C_{k,i} + \epsilon H_{k,i} + \delta S_{k,i})G_{k,i+1}^{n-1*} + S_{k,i}(1 - \delta)G_{k-1,i-1}^{n-1*} + H_{k,i}(1 - \epsilon)G_{k,i-1}^{n-1*} & (k, i) \in \mathcal{A}_1 \\ 1 + (C_{k,i} + \epsilon H_{k,i} + \delta S_{k,i})G_{k,i+1}^{n-1*} + S_{k,i}(1 - \delta)G_{k-1,i-2}^{n-1*} + H_{k,i}(1 - \epsilon)G_{k,i-1}^{n-1*} & (k, i) \in \mathcal{A}_2 \\ G_{k,i}^0 & \text{en otro caso} \end{cases} \quad (6.36)$$

donde la región  $\mathcal{A}_1$  viene delimitada por  $2 \leq k \leq K_s$  y  $1 \leq i \leq 2$ , y la región  $\mathcal{A}_2$  por  $2 \leq k \leq K_s$ ,  $2 < i \leq I_s$ , y,

$$G_{k,i}^{n-1*} = \begin{cases} G_{k,i}^{n-1} & 2 \leq k \leq K_s, 1 \leq i \leq I_s \\ G_{k,i}^0 & \text{en otro caso} \end{cases} \quad (6.37)$$

Las componentes  $G_{k,r_{10}}^n = E_k^u(l)$  son la cota superior ajustada de  $E_k(l)$ . Cuanto mayor sea  $n$ , más ajustada será la cota. En nuestro caso hemos tomado  $K_s = I_s = 100$ .



# Capítulo 7

## Análisis del retardo, caso feedback ternario

### 7.1 Introducción

En el capítulo anterior se ha calculado el throughput máximo estable del algoritmo WSA en su variante para feedback ternario. El paso siguiente será, pues, realizar el estudio del retardo del algoritmo en su versión ternaria. La metodología utilizada será la misma que en su momento se utilizó para el cálculo del retardo del algoritmo WSA binario, y desarrollada en el Capítulo 5. Ello es debido a que estamos utilizando el mismo protocolo de acceso al sistema y un protocolo de resolución de colisiones perteneciente a la misma familia.

En la próxima sección resumiremos brevemente la técnica utilizada para el cálculo del retardo que vamos a aplicar y que ya fue desarrollada ampliamente en el Capítulo 5. Por consiguiente, nos limitaremos a presentar los resultados finales. En estos se puede apreciar que el cálculo del retardo medio se reduce a aplicar las fórmulas obtenidas, que dependen a su vez del cálculo de la longitud media  $E(l | d)$ , de la suma de los retardos media  $E(\psi | d)$  y del segundo momento de la longitud  $E(l^2 | d)$ . En Secciones posteriores pueden encontrarse desarrollados estos cálculos, que son los que realmente diferencian las prestaciones de la versión binaria de la ternaria, y, en general, de cualquier otro algoritmo.

Una vez realizadas las operaciones que aquí se describen, se obtiene los resultados que vienen reflejados en la tabla 7.1. De ella se concluye lo que ya era presumible, que la versión ternaria del algoritmo WSA tiene un mejor retardo que la versión binaria, aunque la mejoría no es notoria.

Cabe resaltar un dato que en principio parece ser sorprendente: para muy



bajas tasas el retardo del algoritmo WSA binario es ligeramente mejor que el ternario. La única razón que explique este fenómeno cabe buscarla en el hecho que sendos algoritmos tienen su propia ventana temporal  $\Delta$ , optimizada para maximizar el throughput. A muy bajas tasas podemos considerar que los dos algoritmos tienen el mismo comportamiento, porque no tienen prácticamente la posibilidad de demostrar sus cualidades resolviendo colisiones porque éstas son muy esporádicas. Según la tabla 6.1, la ventana óptima del algoritmo WSA binario es mayor que la del ternario.

Como ya se ha explicado con anterioridad, la ventana óptima sería aquella que consiguiera  $\lambda\Delta \approx 1$ . En el caso que contemplamos de tasas muy bajas, se cumple que  $\lambda\Delta \ll 1$ , y, por tanto, el que tenga mayor tamaño de ventana se aproximará más al óptimo y tendrá, en principio, un comportamiento mejor.

## 7.2 Límites del retardo medio de paquete

Siguiendo este desarrollo se llega a las mismas expresiones que en el caso del algoritmo WSA en su versión binaria. Estas expresiones dependen básicamente de  $E(i | d)$ ,  $E(\psi | d)$  y  $E(i^2 | d)$  que son calculadas específicamente para el caso ternario, y están desarrolladas en las Secciones subsiguientes.

Vamos a recordar las expresiones con las que se puede calcular tanto una cota superior como inferior del retardo. Toda la metodología ha sido obtenida del trabajo de L. Georgiadis, L. Merakos y P. Papantoni-Kazakos [68].

Los límites en el retardo medio de paquete se obtienen a partir de la siguiente expresión,

$$\frac{W_1^i}{\lambda H_1^u} = D^l \leq D \leq D^u = \frac{W_1^u}{\lambda H_1^l} \quad (7.1)$$

En la expresión anterior,  $H_1^u$  y  $H_1^l$  son una cota superior e inferior de la longitud de sesión media, respectivamente.  $W_1^u$  y  $W_1^l$  son una cota superior e inferior de la suma de los retardos experimentados por todos los paquetes que fueron transmitidos exitosamente en el intervalo  $(v, v_1]$ , respectivamente. Las expresiones que permiten calcularlos, vienen dadas por,

$$H_1^u = E(i | 1) + \alpha_u (E(i | 1) - (1 + \lambda)e^{-\lambda}) + \beta_u (1 - (1 + \lambda)e^{-\lambda}) \quad (7.2)$$

$$H_1^l = H_1^u - (\beta_u - \beta_l) (1 - (1 + \lambda)e^{-\lambda}) \quad (7.3)$$

$$W_1^u = E(\psi | 1) + E(\theta | 1) + \gamma_u (E(I^2 | 1) - (1 + \lambda)e^{-\lambda}) + \delta_u (E(I | 1) - (1 + \lambda)e^{-\lambda}) - \zeta_u (1 - (1 + \lambda)e^{-\lambda}) \quad (7.4)$$

$$W_1^l = W_1^u - (\zeta_u - \zeta_l) (1 - (1 + \lambda)e^{-\lambda}) \quad (7.5)$$

donde

$$\alpha_u = \alpha_l = \frac{E(I | \Delta)}{\Delta - E(I | \Delta)} \quad (7.6)$$

$$\beta_u = \max\{-\alpha_u, \sup_{1 \leq d \leq \Delta} \rho(d)\} \quad \beta_l = \inf_{1 \leq d \leq \Delta} \rho(d) \quad (7.7)$$

$$\gamma_u = \gamma_l = \frac{E(N | \Delta)}{2(\Delta - E(I | \Delta))} \quad (7.8)$$

$$\delta_u = \delta_l = \frac{E(\psi | \Delta) + E(\theta | \Delta) - \Delta E(N | \Delta) + \gamma_u (\Delta^2 + E(I^2 | \Delta) - 2\Delta E(I | \Delta))}{\Delta - E(I | \Delta)} \quad (7.9)$$

$$\zeta_u = \sup_{1 \leq d \leq \Delta} \phi(d) \quad \zeta_l = \inf_{1 \leq d \leq \Delta} \phi(d) \quad (7.10)$$

con,

$$\rho(d) = \frac{E(I | d) + \alpha_u (E(I | d) - d - (1 + \lambda d)e^{-\lambda d})}{(1 + \lambda d)e^{-\lambda d}} \quad (7.11)$$

$$\phi(d) = \frac{E(\psi | d) + E(\theta | d) + \gamma_u (E(I^2 | d) - d^2 - (1 + \lambda d)e^{-\lambda d}) - \delta_u (d - E(I | d) + (1 + \lambda d)e^{-\lambda d})}{(1 + \lambda d)e^{-\lambda d}} \quad (7.12)$$

$$E(N | \Delta) = \lambda \Delta \quad (7.13)$$

$$E(\theta | t) = \frac{\lambda t^2}{2} \quad (7.14)$$

El siguiente paso será calcular  $E(I | d)$ ,  $E(\psi | d)$  y  $E(I^2 | d)$ , que se encuentran en las Secciones subsiguientes.

$\lambda$	Algoritmo WSA			
	ternario		binario	
	$D^l$	$D^u$	$D^l$	$D^u$
0.02	1.563	1.564	1.560	1.560
0.06	1.704	1.710	1.710	1.716
0.10	1.872	1.890	1.887	1.905
0.16	2.224	2.276	2.257	2.310
0.20	2.562	2.654	2.616	2.709
0.24	3.047	3.203	3.138	3.298
0.30	4.383	4.726	4.588	4.946
0.32	5.166	5.622	5.443	5.923
0.36	8.056	8.948	8.660	9.619
0.38	11.113	12.477	12.174	13.669
0.40	17.564	19.946	20.003	22.715
0.42	39.185	45.018	51.351	58.994

Tabla 7.1: Límites superior e inferior de los retardos de paquete esperados. Los algoritmos representados son el WSA (ALOHA estabilizado de ventana) en su versión ternaria y binaria.

## 7.3 Apéndice

### 7.3.1 Cálculo de la longitud media $E(l | d)$ , $1 \leq d \leq \Delta$

La longitud media del algoritmo es,

$$E(l|d) = \sum_{k=0}^{\infty} E_k(l)p(k|d) \quad (7.15)$$

donde,

$$p(k|d) = e^{-\lambda d} \frac{(\lambda d)^k}{k!} \quad (7.16)$$

Definimos, pues,  $E_k(l)$  como la longitud esperada de un CRI dado que empieza con una colisión de multiplicidad  $k$ .

Definimos también  $G_{k,i}$  como el número esperado de ranuras necesarias por el algoritmo para la transmisión exitosa de  $k$  paquetes, dado que el valor del contador es igual a  $r_i = i$  en la primera ranura.

Vamos a ver que en el algoritmo propuesto no se puede calcular de forma exacta  $E_k(l)$ . Nuestro objetivo será, pues, obtener una cota superior de  $E_k(l)$ , que llamaremos  $E_k^0(l)$ , utilizada para acotar el comportamiento en  $k \rightarrow \infty$ ; y

a partir de ella obtendremos otra cota superior más ajustada, que llamaremos  $E_k^v(i)$ , que la utilizaremos para valores de  $k$  discretos. Así, obtendremos una cota superior bastante ajustada de  $E(i|d)$ , que llamaremos  $E^u(i|d)$ , y que será la que utilizaremos a todos los efectos como  $E(i|d)$ .

Según las definiciones dadas, se puede ver que  $E_k(i) = G_{k,r_{i_0}}$ , donde  $r_{i_0}$  es el valor inicial del contador al principio del CRI. El primer paso será, pues, el cálculo de  $G_{k,r_{i_0}}$ . El resultado será, como era de esperar, el mismo que el del Apéndice del Capítulo anterior, sin más que substituir  $\epsilon = \delta = 0$ .

### Cálculo de $G_{k,r_{i_0}}$

Sea  $P_{k,i,s}$  la probabilidad que, dado que en el primer slot  $k$  terminales entran en contienda con probabilidad  $1/r_i = 1/i$ , exáctamente se necesiten  $s$  slots para que accedan los  $k$  terminales. Recordar que han accedido todos los terminales sólo cuando una no colisión ocurre y  $r_i = 1$ . Entonces se aplica la siguiente ecuación recursiva,

$$P_{k,i,s} = \begin{cases} C_{k,i}P_{k,i+1,s-1} + S_{k,i}P_{k-1,i-1,s-1} + H_{k,i}P_{k,i-1,s-1} & 1 \leq i \leq 2 \\ C_{k,i}P_{k,i+1,s-1} + S_{k,i}P_{k-1,i-2,s-1} + H_{k,i}P_{k,i-1,s-1} & i > 2 \\ 0 & \text{en otro caso} \end{cases} \quad (7.17)$$

con  $k \geq 0$ , y,

$$H_{k,i} = \begin{cases} (1 - \frac{1}{i})^k & k > 0, i \geq 1 \\ 1 & k = 0, i \geq 1 \end{cases} \quad (7.18)$$

$$S_{k,i} = \begin{cases} \frac{k}{i} (1 - \frac{1}{i})^{k-1} & k \geq 0, i \geq 1 - \{(k=0, i=1), (k=1, i=1)\} \\ 1 & k=1, i=1 \\ 0 & k=0, i=1 \end{cases} \quad (7.19)$$

$$C_{k,i} = 1 - S_{k,i} - H_{k,i} \quad (7.20)$$

Tomando esperanza respecto de  $s$ , se llega a la siguiente expresión. Llamando,

$$G_{k,i} = \sum_{s=0}^{\infty} s P_{k,i,s} \quad (7.21)$$

se obtiene,

$$G_{k,i} = \begin{cases} 1 + C_{k,i}G_{k,i+1} + S_{k,i}G_{k-1,i-1} + H_{k,i}G_{k,i-1} & k \geq 2, 1 \leq i \leq 2 \\ 1 + C_{k,i}G_{k,i+1} + S_{k,i}G_{k-1,i-2} + H_{k,i}G_{k,i-1} & k \geq 2, i > 2 \\ i & k = 0, i \geq 1 \\ i & k = 1, 1 \leq i \leq 2 \\ i - 1 & k = 1, i > 2 \\ 0 & \text{en otro caso} \end{cases} \quad (7.22)$$

Tenemos, pues, un sistema de ecuaciones lineales de dimensión infinita de la forma,

$$G_{k,i} = I_{k,i} + \sum_{q=0}^{\infty} \sum_{j=1}^{\infty} a_{q,j}^{k,i} G_{q,j} \quad (7.23)$$

donde,

$$I_{k,i} = \begin{cases} 1 & k \geq 2, i \geq 1 \\ i & k = 0, i \geq 1 \\ i & k = 1, 1 \leq i \leq 2 \\ i - 1 & k = 1, i > 2 \\ 0 & \text{en otro caso} \end{cases} \quad (7.24)$$

y,

$$a_{q,j}^{k,i} = \begin{cases} C_{k,i} & q = k, j = i + 1, k \geq 2, i \geq 1 \\ S_{k,i} & q = k - 1, j = i - 1, k \geq 2, 1 \leq i \leq 2 \\ S_{k,i} & q = k - 1, j = i - 2, k \geq 2, i > 2 \\ H_{k,i} & q = k, j = i - 1, k \geq 2, i \geq 1 \\ 0 & \text{en otro caso} \end{cases} \quad (7.25)$$

Ya que no es posible resolver el sistema de ecuaciones, intentaremos encontrar límites de  $G_{k,i}$ .

Sea,

$$G_{k,i}^0 = \begin{cases} \alpha k^2 + \beta i^2 + \rho ki & k \geq 2, i \geq 1 \\ i & k = 0, i \geq 1 \\ i & k = 1, 1 \leq i \leq 2 \\ i - 1 & k = 1, i > 2 \\ 0 & \text{en otro caso} \end{cases} \quad (7.26)$$

y definimos,

$$G_{k,i}^1 = \begin{cases} 1 + C_{k,i}G_{k,i+1}^0 + S_{k,i}G_{k-1,i-1}^0 + H_{k,i}G_{k,i-1}^0 & k \geq 2, 1 \leq i \leq 2 \\ 1 + C_{k,i}G_{k,i+1}^0 + S_{k,i}G_{k-1,i-2}^0 + H_{k,i}G_{k,i-1}^0 & k \geq 2, i > 2 \\ i & k = 0, i \geq 1 \\ i & k = 1, 1 \leq i \leq 2 \\ i - 1 & k = 1, i > 2 \\ 0 & \text{en otro caso} \end{cases} \quad (7.27)$$

Después de operar, obtenemos los siguientes resultados,

$$G_{k,i}^1 = \begin{cases} G_{k,i}^0 + M_{k,i}(2C_{k,i} - 1) + N_{k,i}S_{k,i} + O_{k,i} & k \geq 2, i \geq 1 \\ G_{k,i}^0 & \text{en otro caso} \end{cases} \quad (7.28)$$

donde,

$$M_{k,i} = 2i\beta + \rho k \quad k \geq 2, i \geq 1 \quad (7.29)$$

$$N_{k,i} = \begin{cases} \alpha(1 - 2k) - \rho(i - 1) + J_{k,i} & k > 2, i \geq 1 \\ F_{k,i} - (\alpha k^2 + \beta(i - 1)^2 + \rho k(i - 1)) & k = 2, i \geq 1 \end{cases} \quad (7.30)$$

$$O_{k,i} = 1 + \beta \quad k \geq 2, i \geq 1 \quad (7.31)$$

con,

$$J_{k,i} = \begin{cases} 0 & k > 2, 1 \leq i \leq 2 \\ \beta(3 - 2i) - \rho(k - 1) - \mu & k > 2, i > 2 \end{cases} \quad (7.32)$$

y,

$$F_{k,i} = \begin{cases} i - 1 & k = 2, 1 \leq i \leq 2 \\ i - 2 & k = 2, 3 \leq i \leq 4 \\ i - 3 & k = 2, i > 4 \end{cases} \quad (7.33)$$

Se puede ver que  $G_{k,i}^1 \leq G_{k,i}^0$  si,

$$M_{k,i}(2C_{k,i} - 1) + N_{k,i}S_{k,i} + O_{k,i} \leq 0 \quad k \geq 2, i \geq 1 \quad (7.34)$$

Tenemos pues una restricción lineal por cada par  $(k, i)$ . Procedemos como sigue,

1. Aplicamos el método dual simplex de programación lineal que se pueden encontrar en diversos libros, entre ellos el de S. Gass [71] para la Región  $R_1$ , dada por  $k = 0 \dots 20$  y  $i = 1 \dots 20$ , es decir, 420 restricciones lineales. La Región  $R_1$  está representada gráficamente en la figura 4.2. La función a minimizar escogida es la suma de todos los coeficientes. Los resultados están expuestos en la tabla 4.4.
2. Comprobamos que  $f$  es negativa en la región  $R_2$ , dada por  $k = 0 \dots 100$  y  $i = 1 \dots 100$ . La Región  $R_2$  está representada gráficamente en la figura 4.2. Hay que hacer notar que  $R_2$  incluye  $R_1$ , y que esta compuesto por un número finito de puntos.
3. Demostramos que los coeficientes obtenidos satisfacen las restricciones en la Región  $R_3$ , dada por todo par  $(k, i)$  tal que  $k > 100$  o  $i > 100$ . La Región  $R_3$  está representada gráficamente en la figura 4.2.

El resultado obtenido es,

$$G_{k,i}^0 = \begin{cases} 2,1875k^2 + 0,25i^2 - 0,875ki & k \geq 2, i \geq 1 \\ i & k = 0, i \geq 1 \\ i & k = 1, 1 \leq i \leq 2 \\ i - 1 & k = 1, i > 2 \\ 0 & \text{en otro caso} \end{cases} \quad (7.35)$$

De esta forma,  $G_{k,i}^0$  será un límite superior de  $G_{k,i}$ . Y, por tanto, tenemos una cota superior de  $E_k(l)$ ,  $E_k^0(l) = G_{k,r_{i0}}^0 = G_{k,1}^0$ .

**Demostración.** El objetivo de este apartado es demostrar analíticamente que los coeficientes obtenidos satisfacen las restricciones para un par general  $(k, i)$ . El método a seguir sería totalmente paralelo al especificado en el Apéndice del Capítulo 4.

**Cálculo de  $E^u(l|d)$ ,  $1 \leq d \leq \Delta$**

Sea  $E_k(l)$  la longitud esperada de un CRI dado que empieza con una colisión de multiplicidad  $k$ .

Se puede ver que  $E_k(l) = G_{k,r_{i0}}$ , donde  $r_{i0}$  es el valor inicial del contador al principio del CRI. De los resultados anteriores conocemos un límite superior de  $G_{k,i}$  analíticamente, que nos servirá para calcular el throughput.

La longitud media del algoritmo es,

$$E(l|d) = \sum_{k=0}^{\infty} E_k(l)p(k|d) \quad (7.36)$$

donde,

$$p(k|d) = e^{-\lambda d} \frac{(\lambda d)^k}{k!} \quad (7.37)$$

Ya hemos visto que no se puede calcular de forma exacta  $E_k(i)$ . Tenemos un límite superior analítico  $E_k^0(i) = G_{k,r_{i_0}}^0$ , que utilizaremos para acotar el comportamiento en  $k \rightarrow \infty$ . Para calcular la longitud media  $E(i|d)$ , que a todos los efectos substituiremos por una cota ligeramente superior  $E^u(i|d)$ , necesitamos una cota mas precisa de  $E_k(i)$  que llamaremos  $E_k^u(i) = G_{k,r_{i_0}}^u$ , y que utilizaremos para  $k$  discretos. Así tenemos,

$$E^u(i|d) = \sum_{k=0}^{30} E_k^u(i)p(k|d) + \sum_{k=31}^{\infty} E_k^0(i)p(k|d) \quad (7.38)$$

Después de algunas operaciones con el segundo sumando, se obtiene,

$$\begin{aligned} E^u(i|d) &= \sum_{k=0}^{30} E_k^u(i)p(k|d) + \alpha \left\{ (\lambda d)^2 + \lambda d - \sum_{k=0}^{30} k^2 p(k|d) \right\} \\ &+ \rho r_{i_0} \left\{ \lambda d - \sum_{k=0}^{30} k p(k|d) \right\} + \beta r_{i_0}^2 \left\{ 1 - \sum_{k=0}^{30} p(k|d) \right\} \end{aligned} \quad (7.39)$$

ya que,

$$E_k^0(i) = G_{k,r_{i_0}}^0 = \alpha k^2 + \beta r_{i_0}^2 + \rho k r_{i_0} = 2,1875k^2 + 0,25r_{i_0}^2 - 0,875kr_{i_0} \quad (7.40)$$

Una vez tenemos  $G_{k,i}^0$ , expresión (7.26), podemos conseguir una cota superior más ajustada de  $E_k(i)$ ,  $E_k^u(i)$ , con el siguiente procedimiento. Sea  $\mathcal{M}^0$  una matriz bidimensional ( $K_s \times I_s$ ) cuyos componentes son  $G_{k,i}^0$  y sea  $\mathcal{M}^n$  la matriz bidimensional ( $K_s \times I_s$ ) cuyos componentes se obtienen de forma recursiva,

$$G_{k,i}^n = \begin{cases} 1 + C_{k,i} G_{k,i+1}^{n-1*} + S_{k,i} G_{k-1,i-1}^{n-1*} + H_{k,i} G_{k,i-1}^{n-1*} & (k,i) \in \mathcal{A}_1 \\ 1 + C_{k,i} G_{k,i+1}^{n-1*} + S_{k,i} G_{k-1,i-2}^{n-1*} + H_{k,i} G_{k,i-1}^{n-1*} & (k,i) \in \mathcal{A}_2 \\ G_{k,i}^0 & \text{en otro caso} \end{cases} \quad (7.41)$$

donde la región  $\mathcal{A}_1$  viene delimitada por  $2 \leq k \leq K_s$  y  $1 \leq i \leq 2$ , y la región  $\mathcal{A}_2$  por  $2 \leq k \leq K_s$ ,  $2 < i \leq I_s$ , y,

$$G_{k,i}^{n-1*} = \begin{cases} G_{k,i}^{n-1} & 2 \leq k \leq K_s, 1 \leq i \leq I_s \\ G_{k,i}^0 & \text{en otro caso} \end{cases} \quad (7.42)$$

Las componentes  $G_{k,r_{i_0}}^n = E_k^u(i)$  son la cota superior ajustada de  $E_k(i)$ . Cuanto mayor sea  $n$ , más ajustada será la cota. En nuestro caso, tomaremos  $K_s = I_s = 100$ .



### 7.3.2 Cálculo de la suma de los retardos media $E(\psi | d)$ , $1 \leq d \leq \Delta$

La suma de los retardos media del algoritmo es,

$$E(\psi|d) = \sum_{k=0}^{\infty} E_k(\psi)p(k|d) \quad (7.43)$$

donde,

$$p(k|d) = e^{-\lambda d} \frac{(\lambda d)^k}{k!} \quad (7.44)$$

Definimos, pues,  $E_k(\psi)$  como la suma de los retardos esperada de un CRI dado que empieza con una colisión de multiplicidad  $k$ .

Definimos también  $\Psi_{k,i}$  como la suma de retardos esperado para la transmisión exitosa de  $k$  paquetes, dado que el valor del contador es igual a  $r_t = i$  en la primera ranura.

Vamos a ver que en el algoritmo propuesto no se puede calcular de forma exacta  $E_k(\psi)$ . Nuestro objetivo será, pues, obtener una cota superior de  $E_k(\psi)$ , que llamaremos  $E_k^0(\psi)$ , utilizada para acotar el comportamiento en  $k \rightarrow \infty$ ; y a partir de ella obtendremos otra cota superior más ajustada, que llamaremos  $E_k^u(\psi)$ , que la utilizaremos para valores de  $k$  discretos. Así, obtendremos una cota superior bastante ajustada de  $E(\psi|d)$ , que llamaremos  $E^u(\psi|d)$ , y que será la que utilizaremos a todos los efectos como  $E(\psi|d)$ .

Según las definiciones dadas, se puede ver que  $E_k(\psi) = \Psi_{k,r_{i_0}}$ , donde  $r_{i_0}$  es el valor inicial del contador al principio del CRI. El primer paso será, pues, el cálculo de  $\Psi_{k,r_{i_0}}$ .

#### Cálculo de $\Psi_{k,r_{i_0}}$

Sea  $\Psi_{k,i}$  la suma de los retardos de  $k$  terminales que entran en contienda con probabilidad  $1/r_t = 1/i$ , al principio del CRI. Recordar que han accedido todos los terminales solo cuando una no colision ocurre y  $r_t = 1$ . Entonces se aplica la siguiente ecuación recursiva,

$$\Psi_{k,i} = \begin{cases} k + C_{k,i}\Psi_{k,i+1} + S_{k,i}\Psi_{k-1,i-1} + H_{k,i}\Psi_{k,i-1} & k \geq 2, 1 \leq i \leq 2 \\ k + C_{k,i}\Psi_{k,i+1} + S_{k,i}\Psi_{k-1,i-2} + H_{k,i}\Psi_{k,i-1} & k \geq 2, i > 2 \\ \frac{i+1}{2} & k = 1, i \geq 1 \\ 0 & \text{en otro caso} \end{cases} \quad (7.45)$$

donde  $H_{k,i}$ ,  $S_{k,i}$ ,  $C_{k,i}$  vienen dados por las expresiones (7.18), (7.19) y (7.20) respectivamente.

Tenemos, pues, un sistema de ecuaciones lineales de dimension infinita de la forma,

$$\Psi_{k,i} = I_{k,i} + \sum_{q=0}^{\infty} \sum_{j=1}^{\infty} a_{q,j}^{k,i} \Psi_{q,j} \quad (7.46)$$

donde,

$$I_{k,i} = \begin{cases} k & k \geq 2, i \geq 1 \\ \frac{i+1}{2} & k = 1, i \geq 1 \\ 0 & \text{en otro caso} \end{cases} \quad (7.47)$$

y,

$$a_{q,j}^{k,i} = \begin{cases} C_{k,i} & q = k, j = i + 1, k \geq 2, i \geq 1 \\ S_{k,i} & q = k - 1, j = i - 1, k \geq 2, 1 \leq i \leq 2 \\ S_{k,i} & q = k - 1, j = i - 2, k \geq 2, i > 2 \\ H_{k,i} & q = k, j = i - 1, k \geq 2, i \geq 1 \\ 0 & \text{en otro caso} \end{cases} \quad (7.48)$$

Sea,

$$\Psi_{k,i}^0 = \begin{cases} \alpha k^2 + \beta i^2 + \rho k i + \gamma & k \geq 2, i \geq 1 \\ \frac{i+1}{2} & k = 1, i \geq 1 \\ 0 & \text{en otro caso} \end{cases} \quad (7.49)$$

para  $i \geq 1$ , y definimos,

$$\Psi_{k,i}^1 = \begin{cases} k + C_{k,i} \Psi_{k,i+1}^0 + S_{k,i} \Psi_{k-1,i-1}^0 + H_{k,i} \Psi_{k,i-1}^0 & k \geq 2, 1 \leq i \leq 2 \\ k + C_{k,i} \Psi_{k,i+1}^0 + S_{k,i} \Psi_{k-1,i-2}^0 + H_{k,i} \Psi_{k,i-1}^0 & k \geq 2, i > 2 \\ \frac{i+1}{2} & k = 1, i \geq 1 \\ 0 & \text{en otro caso} \end{cases} \quad (7.50)$$

Despues de operar, obtenemos los siguientes resultados,

$$\Psi_{k,i}^1 = \begin{cases} \Psi_{k,i}^0 + M_{k,i}(2C_{k,i} - 1) + N_{k,i}S_{k,i} + O_{k,i} & k \geq 2, i \geq 1 \\ \Psi_{k,i}^0 & \text{en otro caso} \end{cases} \quad (7.51)$$

donde,

$$M_{k,i} = 2i\beta + \rho k \quad k \geq 2, i \geq 1 \quad (7.52)$$

$$N_{k,i} = \begin{cases} \alpha(1 - 2k) - \rho(i - 1) + J_{k,i} & k > 2, i \geq 1 \\ \alpha k^2 + \beta(i - 1)^2 - \frac{i}{2} + \rho k(i - 1) + \gamma + F_{k,i} & k = 2, i \geq 1 \end{cases} \quad (7.53)$$

$$O_{k,i} = k + \beta \quad k \geq 2, i \geq 1 \quad (7.54)$$

con,

$$J_{k,i} = \begin{cases} 0 & k > 2, 1 \leq i \leq 2 \\ \beta(3 - 2i) - \rho(k - 1) - \mu & k > 2, i > 2 \end{cases} \quad (7.55)$$

y,

$$F_{k,i} = \begin{cases} \frac{i}{2} & k = 2, 1 \leq i \leq 2 \\ \frac{i-1}{2} & k = 2, i > 2 \end{cases} \quad (7.56)$$

Se puede ver que  $\Psi_{k,i}^1 \leq \Psi_{k,i}^0$  si,

$$M_{k,i}(2C_{k,i} - 1) + N_{k,i}S_{k,i} + O_{k,i} \leq 0 \quad k \geq 2, i \geq 1 \quad (7.57)$$

Tenemos pues una restricción lineal por cada par  $(k, i)$ . Procedemos como sigue,

1. Aplicamos el método dual simplex de programación lineal que se pueden encontrar en diversos libros, entre ellos el de S. Gass [71] para la Región  $R_1$ , dada por  $k = 0 \dots 20$  y  $i = 1 \dots 20$ , es decir, 420 restricciones lineales. La Región  $R_1$  está representada gráficamente en la figura 4.2. La función a minimizar escogida es la suma de todos los coeficientes. Los resultados están expuestos en la tabla 4.4.
2. Comprobamos que  $f$  es negativa en la región  $R_2$ , dada por  $k = 0 \dots 100$  y  $i = 1 \dots 100$ . La Región  $R_2$  está representada gráficamente en la figura 4.2. Hay que hacer notar que  $R_2$  incluye  $R_1$ , y que esta compuesto por un número finito de puntos.
3. Demostramos que los coeficientes obtenidos satisfacen las restricciones en la Región  $R_3$ , dada por todo par  $(k, i)$  tal que  $k > 100$  o  $i > 100$ . La Región  $R_3$  está representada gráficamente en la figura 4.2.

El resultado obtenido es,

$$\Psi_{k,i}^0 = \begin{cases} 3,5077k^2 + 0,4410i^2 - 1,6615ki & k \geq 2, i \geq 1 \\ \frac{i+1}{2} & k = 1, i \geq 1 \\ 0 & \text{en otro caso} \end{cases} \quad (7.58)$$

De esta forma,  $\Psi_{k,i}^0$  sera un límite superior de  $\Psi_{k,i}$ . Y, por tanto, tenemos una cota superior de  $E_k(\psi)$ ,  $E_k^0(\psi) = \Psi_{k,r_{i0}}^0 = \Psi_{k,1}^0$ .

**Demostración.** El objetivo de este apartado es demostrar analíticamente que los coeficientes obtenidos satisfacen las restricciones para un par general  $(k, i)$ . El método a seguir sería totalmente paralelo al especificado en Apéndice del Capítulo 4.

Cálculo de  $E^u(\psi|d)$ ,  $1 \leq d \leq \Delta$

Sea  $E_k(\psi)$  la suma de los retardos esperada de un CRI dado que empieza con una colisión de multiplicidad  $k$ .

Se puede ver que  $E_k(\psi) = \Psi_{k,r_{i_0}}$ , donde  $r_{i_0}$  es el valor inicial del contador. De los resultados anteriores conocemos un límite superior de  $\Psi_{k,i}$  analíticamente, que nos servirá para cálculos posteriores.

La suma media de los retardos del algoritmo es,

$$E(\psi|d) = \sum_{k=0}^{\infty} E_k(\psi)p(k|d) \quad (7.59)$$

donde,

$$p(k|d) = e^{-\lambda d} \frac{(\lambda d)^k}{k!} \quad (7.60)$$

Ya hemos visto que no se puede calcular de forma exacta  $E_k(\psi)$ . Tenemos un límite superior analítico  $E_k^0(\psi) = \Psi_{k,r_{i_0}}^0$ , que utilizaremos para acotar el comportamiento en  $k \rightarrow \infty$ . Para calcular la longitud media, que a todos los efectos substituiremos por una cota ligeramente superior  $E^u(\psi|d)$ , necesitamos una cota más precisa de  $E_k(\psi)$  que llamaremos  $E_k^u(\psi) = \Psi_{k,r_{i_0}}^u$ , y que utilizaremos para  $k$  discretos. Así tenemos,

$$E^u(\psi|d) = \sum_{k=0}^{30} E_k^u(\psi)p(k|d) + \sum_{k=31}^{\infty} E_k^0(\psi)p(k|d) \quad (7.61)$$

Después de algunas operaciones con el segundo sumando, se obtiene,

$$\begin{aligned} E^u(\psi|d) = & \sum_{k=0}^{30} E_k^u(\psi)p(k|d) + \alpha \left\{ (\lambda d)^2 + \lambda d - \sum_{k=0}^{30} k^2 p(k|d) \right\} \\ & + \rho r_{i_0} \left\{ \lambda d - \sum_{k=0}^{30} k p(k|d) \right\} + (\beta r_{i_0}^2 + \gamma) \left\{ 1 - \sum_{k=0}^{30} p(k|d) \right\} \end{aligned} \quad (7.62)$$

ya que,

$$\begin{aligned} E_k^0(\psi) = \Psi_{k,r_{i_0}}^0 = & \alpha k^2 + \beta r_{i_0}^2 + \rho k r_{i_0} + \gamma = \\ & 3,5077k^2 + 0,4410r_{i_0}^2 - 1.6615kr_{i_0} \end{aligned} \quad (7.63)$$

Una vez tenemos  $\Psi_{k,i}^0$ , expresión (7.58), podemos conseguir una cota superior más ajustada de  $E_k(\psi)$ ,  $E_k^y(\psi)$ , con el siguiente procedimiento. Sea  $\mathcal{M}^0$  una matriz bidimensional ( $K_s \times I_s$ ) cuyos componentes son  $\Psi_{k,i}^0$  y sea  $\mathcal{M}^n$  la matriz bidimensional ( $K_s \times I_s$ ) cuyos componentes se obtienen de forma recursiva,

$$\Psi_{k,i}^n = \begin{cases} 1 + C_{k,i}\Psi_{k,i+1}^{n-1*} + S_{k,i}\Psi_{k-1,i-1}^{n-1*} + H_{k,i}\Psi_{k,i-1}^{n-1*} & (k,i) \in \mathcal{A}_1 \\ 1 + C_{k,i}\Psi_{k,i+1}^{n-1*} + S_{k,i}\Psi_{k-1,i-2}^{n-1*} + H_{k,i}\Psi_{k,i-1}^{n-1*} & (k,i) \in \mathcal{A}_2 \\ \Psi_{k,i}^0 & \text{en otro caso} \end{cases} \quad (7.64)$$

donde la región  $\mathcal{A}_1$  viene delimitada por  $2 \leq k \leq K_s$  y  $1 \leq i \leq 2$ , y la región  $\mathcal{A}_2$  por  $2 \leq k \leq K_s$ ,  $2 < i \leq I_s$ , y,

$$\Psi_{k,i}^{n-1*} = \begin{cases} \Psi_{k,i}^{n-1} & 2 \leq k \leq K_s, 1 \leq i \leq I_s \\ \Psi_{k,i}^0 & \text{en otro caso} \end{cases} \quad (7.65)$$

Las componentes  $\Psi_{k,r_0}^n = E_k^y(\psi)$  son la cota superior ajustada de  $E_k(\psi)$ . Cuanto mayor sea  $n$ , más ajustada será la cota. En nuestro caso, tomaremos  $K_s = I_s = 100$ .

### 7.3.3 Cálculo del segundo momento de la longitud $E(i^2 | d)$ , $1 \leq d \leq \Delta$

El segundo momento de la longitud del algoritmo es,

$$E(i^2 | d) = \sum_{k=0}^{\infty} E_k(i^2) p(k | d) \quad (7.66)$$

donde,

$$p(k | d) = e^{-\lambda d} \frac{(\lambda d)^k}{k!} \quad (7.67)$$

Definimos, pues,  $E_k(i^2)$  como el segundo momento de la longitud de un CRI dado que empieza con una colisión de multiplicidad  $k$ .

Definimos también  $Q_{k,i}$  como el segundo momento del número de ranuras necesarias por el algoritmo para la transmisión exitosa de  $k$  paquetes, dado que el valor del contador es igual a  $r_i = i$  en la primera ranura.

Vamos a ver que en el algoritmo propuesto no se puede calcular de forma exacta  $E_k(i^2)$ . Nuestro objetivo será, pues, obtener una cota superior de  $E_k(i^2)$ , que llamaremos  $E_k^0(i^2)$ , utilizada para acotar el comportamiento en  $k \rightarrow \infty$ ; y a partir de ella obtendremos otra cota superior más ajustada,

que llamaremos  $E_k^u(l^2)$ , que la utilizaremos para valores de  $k$  discretos. Así, obtendremos una cota superior bastante ajustada de  $E(l^2|d)$ , que llamaremos  $E^u(l^2|d)$ , y que será la que utilizaremos a todos los efectos como  $E(l^2|d)$ .

Según las definiciones dadas, se puede ver que  $E_k(l^2) = Q_{k,r_{i_0}}$ , donde  $r_{i_0}$  es el valor inicial del contador. El primer paso será, pues, el cálculo de  $Q_{k,r_{i_0}}$ .

### Cálculo de $Q_{k,r_{i_0}}$

Sea  $P_{k,i,s}$  la probabilidad que, dado que en el primer slot  $k$  terminales entran en contienda con probabilidad  $1/r_i = 1/i$ , exáctamente se necesiten  $s$  slots para que accedan los  $k$  terminales. Recordar que han accedido todos los terminales solo cuando una no colisión ocurre y  $r_i = 1$ . Entonces se aplica la siguiente ecuación recursiva,

$$P_{k,i,s} = \begin{cases} C_{k,i}P_{k,i+1,s-1} + S_{k,i}P_{k-1,i-1,s-1} + H_{k,i}P_{k,i-1,s-1} & 1 \leq i \leq 2 \\ C_{k,i}P_{k,i+1,s-1} + S_{k,i}P_{k-1,i-2,s-1} + H_{k,i}P_{k,i-1,s-1} & i > 2 \\ 0 & \text{en otro caso} \end{cases} \quad (7.68)$$

con  $k \geq 0$ , y, donde  $H_{k,i}$ ,  $S_{k,i}$ ,  $C_{k,i}$  vienen dados por las expresiones (7.18), (7.19) y (7.20) respectivamente.

Llamando,

$$Q_{k,i} = \sum_{s=0}^{\infty} s^2 P_{k,i,s} \quad (7.69)$$

llegamos a la siguiente relación recursiva,

$$Q_{k,i} = \begin{cases} 2G_{k,i} - 1 + C_{k,i}Q_{k,i+1} + S_{k,i}Q_{k-1,i-1} + H_{k,i}Q_{k,i-1} & k \geq 2, 1 \leq i \leq 2 \\ 2G_{k,i} - 1 + C_{k,i}Q_{k,i+1} + S_{k,i}Q_{k-1,i-2} + H_{k,i}Q_{k,i-1} & k \geq 2, i > 2 \\ i^2 & k = 0, i \geq 1 \\ i^2 & k = 1, 1 \leq i \leq 2 \\ (i-1)^2 & k = 1, i > 2 \\ 0 & \text{en otro caso} \end{cases} \quad (7.70)$$

Tenemos, pues, un sistema de ecuaciones lineales de dimensión infinita de la forma,

$$Q_{k,i} = l_{k,i} + \sum_{q=0}^{\infty} \sum_{j=1}^{\infty} a_{qj}^{k,i} Q_{qj} \quad (7.71)$$

donde,

$$l_{k,i} = \begin{cases} 2G_{k,i} - 1 & k \geq 2, i \geq 1 \\ i^2 & k = 0, i \geq 1 \\ i^2 & k = 1, 1 \leq i \leq 2 \\ (i-1)^2 & k = 1, i > 2 \\ 0 & \text{en otro caso} \end{cases} \quad (7.72)$$

y,

$$a_{q,i}^{k,i} = \begin{cases} C_{k,i} & q = k, j = i + 1, k \geq 2, i \geq 1 \\ S_{k,i} & q = k - 1, j = i - 1, k \geq 2, 1 \leq i \leq 2 \\ S_{k,i} & q = k - 1, j = i - 2, k \geq 2, i > 2 \\ H_{k,i} & q = k, j = i - 1, k \geq 2, i \geq 1 \\ 0 & \text{en otro caso} \end{cases} \quad (7.73)$$

Ya que no es posible resolver el sistema de ecuaciones, intentaremos encontrar límites de  $Q_{k,i}$ .

Sea,

$$Q_{k,i}^0 = \begin{cases} \alpha k^4 + \beta i^4 + \rho k^2 i^2 + \mu k^3 i + \eta k i^3 & k \geq 2, i \geq 1 \\ i^2 & k = 0, i \geq 1 \\ i^2 & k = 1, 1 \leq i \leq 2 \\ (i-1)^2 & k = 1, i > 2 \\ 0 & \text{en otro caso} \end{cases} \quad (7.74)$$

para  $i \geq 1$ , y definimos,

$$Q_{k,i}^1 = \begin{cases} 2G_{k,i} - 1 + C_{k,i}Q_{k,i+1}^0 + S_{k,i}Q_{k-1,i-1}^0 + H_{k,i}Q_{k,i-1}^0 & k \geq 2, 1 \leq i \leq 2 \\ 2G_{k,i} - 1 + C_{k,i}Q_{k,i+1}^0 + S_{k,i}Q_{k-1,i-2}^0 + H_{k,i}Q_{k,i-1}^0 & k \geq 2, i > 2 \\ i^2 & k = 1, 1 \leq i \leq 2 \\ i^2 & k = 0, i \geq 1 \\ (i-1)^2 & k = 1, i > 2 \\ 0 & \text{en otro caso} \end{cases} \quad (7.75)$$

Después de operar, obtenemos los siguientes resultados,

$$Q_{k,i}^1 = \begin{cases} Q_{k,i}^0 + M_{k,i}(2C_{k,i} - 1) + N_{k,i}S_{k,i} + O_{k,i} & k \geq 2, i \geq 1 \\ Q_{k,i}^0 & \text{en otro caso} \end{cases} \quad (7.76)$$

donde,

$$M_{k,i} = 2k^2 i \rho + k^3 \mu + k(1 + 3i^2)\eta + 4i(1 + i^2)\beta \quad k \geq 2, i \geq 1 \quad (7.77)$$

$$N_{k,i} = \begin{cases} (1-i)^2(1-2k)\rho + (1-i)(1-3k+3k^2)\mu \\ \quad + (1-i)^3\eta + (1-4k^3+6k^2-4k)\alpha + J_{k,i} & k > 2, i \geq 1 \\ (i-1)^2 - k^2(1-i)^2\rho + k^3(1-i)\mu \\ \quad + k(1-i)^3\eta - k^4\alpha - (1-i)^4\beta + F_{k,i} & k = 2, i \geq 1 \end{cases} \quad (7.78)$$

$$O_{k,i} = 2G_{k,i}^0 - 1 + k^2\rho + 3ki\eta + (6i^2 + 1)\beta \quad k \geq 2, i \geq 1 \quad (7.79)$$

con,

$$J_{k,i} = \begin{cases} 0 & k > 2, 1 \leq i \leq 2 \\ \beta(3-2i)(2i^2-6i+5) + \rho(3-2i)(k-1)^2 \\ \quad - \mu(k-1)^3 - \eta(k-1)(3i^2-9i+7) & k > 2, i > 2 \end{cases} \quad (7.80)$$

y,

$$F_{k,i} = \begin{cases} 2i-1 & k=2, 1 \leq i \leq 2 \\ 3-2i & k=2, 3 \leq i \leq 4 \\ 8-4i & k=2, i > 4 \end{cases} \quad (7.81)$$

Se puede ver que  $Q_{k,i}^1 \leq Q_{k,i}^0$  si,

$$M_{k,i}(2C_{k,i} - 1) + N_{k,i}S_{k,i} + O_{k,i} \leq 0 \quad k \geq 2, i \geq 1 \quad (7.82)$$

Tenemos pues una restricción lineal por cada par  $(k, i)$ . Procedemos como sigue,

1. Aplicamos el método dual simplex de programación lineal que se pueden encontrar en diversos libros, entre ellos el de S. Gass [71] para la Región  $R_1$ , dada por  $k = 0 \dots 20$  y  $i = 1 \dots 20$ , es decir, 420 restricciones lineales. La Región  $R_1$  está representada gráficamente en la figura 4.2. La función a minimizar escogida es la suma de todos los coeficientes. Los resultados están expuestos en la tabla 4.4.
2. Comprobamos que  $f$  es negativa en la región  $R_2$ , dada por  $k = 0 \dots 100$  y  $i = 1 \dots 100$ . La Región  $R_2$  está representada gráficamente en la figura 4.2. Hay que hacer notar que  $R_2$  incluye  $R_1$ , y que esta compuesto por un número finito de puntos.
3. Demostramos que los coeficientes obtenidos satisfacen las restricciones en la Región  $R_3$ , dada por todo par  $(k, i)$  tal que  $k > 100$  o  $i > 100$ . La Región  $R_3$  está representada gráficamente en la figura 4.2.



El resultado obtenido es,

$$Q_{k,i}^0 = \begin{cases} 2, 1218k^4 + 0, 4933i^4 - 1, 3141k^2i^2 & k \geq 2, i \geq 1 \\ i^2 & k = 0, i \geq 1 \\ i^2 & k = 1, 1 \leq i \leq 2 \\ (i-1)^2 & k = 1, i > 2 \\ 0 & \text{en otro caso} \end{cases} \quad (7.83)$$

De esta forma,  $Q_{k,i}^0$  será un límite superior de  $Q_{k,i}$ . Y, por tanto, tenemos una cota superior de  $E_k(I^2)$ ,  $E_k^0(I^2) = Q_{k,r_{i_0}}^0 = Q_{k,1}^0$ .

**Demostración.** El objetivo de este apartado es demostrar analíticamente que los coeficientes obtenidos satisfacen las restricciones para un par general  $(k, i)$ . El método a seguir sería totalmente paralelo al especificado en Apéndice del Capítulo 4.

**Cálculo de  $E(I^2|d)$ ,  $1 \leq d \leq \Delta$**

Sea  $E_k(I^2)$  el segundo momento de la longitud de un CRI dado que empieza con una colisión de multiplicidad  $k$ .

Se puede ver que  $E_k(I^2) = Q_{k,r_{i_0}}$ , donde  $r_{i_0}$  es el valor inicial del contador. De los resultados anteriores conocemos un límite superior de  $Q_{k,i}$ , que nos servirá para calcular  $E_k(I^2)$ .

El segundo momento de la longitud de nuestro algoritmo se calcula como,

$$E(I^2|d) = \sum_{k=0}^{\infty} E_k(I^2)p(k|d) \quad (7.84)$$

donde,

$$p(k|d) = e^{-\lambda d} \frac{(\lambda d)^k}{k!} \quad (7.85)$$

Ya hemos visto que no se puede calcular de forma exacta  $E_k(I^2)$ . Tenemos un límite superior analítico  $E_k^0(I^2) = Q_{k,r_{i_0}}^0$ , que utilizaremos para acotar el comportamiento en  $k \rightarrow \infty$ . Para calcular la longitud media, que a todos los efectos substituiremos por una cota ligeramente superior  $E^u(I^2|d)$ , necesitamos una cota mas precisa de  $E_k(I^2)$  que llamaremos  $E_k^u(I^2) = Q_{k,r_{i_0}}^u$ , y que utilizaremos para  $k$  discretos. Así tenemos,

$$E^u(I^2|d) = \sum_{k=0}^{30} E_k^u(I^2)p(k|d) + \sum_{k=31}^{\infty} E_k^0(I^2)p(k|d) \quad (7.86)$$

Sabiendo que,

$$\sum_{k=0}^{\infty} e^{-x} \frac{(x)^k}{k!} = 1 \quad (7.87)$$

$$\sum_{k=0}^{\infty} k e^{-x} \frac{(x)^k}{k!} = x \quad (7.88)$$

$$\sum_{k=0}^{\infty} k^2 e^{-x} \frac{(x)^k}{k!} = x(1+x) \quad (7.89)$$

$$\sum_{k=0}^{\infty} k^3 e^{-x} \frac{(x)^k}{k!} = x(1+3x+x^2) \quad (7.90)$$

$$\sum_{k=0}^{\infty} k^4 e^{-x} \frac{(x)^k}{k!} = x(1+7x+6x^2+x^3) \quad (7.91)$$

Y despues de algunas manipulaciones algebraicas se obtiene que,

$$\begin{aligned} E^u(I^2|d) &= \sum_{k=0}^{30} E_k^u(I^2) p(k|d) + \beta r_{t_0}^4 \left\{ 1 - \sum_{k=0}^{30} p(k|d) \right\} \\ &+ \gamma r_{t_0}^3 \left\{ \lambda d - \sum_{k=0}^{30} k p(k|d) \right\} + \rho r_{t_0}^2 \left\{ (\lambda d)(1 + \lambda d) - \sum_{k=0}^{30} k^2 p(k|d) \right\} \\ &+ \mu r_{t_0} \left\{ (\lambda d)(1 + 3\lambda d + (\lambda d)^2) - \sum_{k=0}^{30} k^3 p(k|d) \right\} \\ &+ \alpha \left\{ (\lambda d)(1 + 7\lambda d + 6(\lambda d)^2 + (\lambda d)^3) - \sum_{k=0}^{30} k^4 p(k|d) \right\} \end{aligned} \quad (7.92)$$

Una vez tenemos  $Q_{k,i}^0$ , expresión (7.83), podemos conseguir una cota superior más ajustada de  $E_k(I^2)$ ,  $E_k^u(I^2)$ , con el siguiente procedimiento. Sea  $\mathcal{M}^0$  una matriz bidimensional ( $K_s \times I_s$ ) cuyos componentes son  $Q_{k,i}^0$ , y sea  $\mathcal{M}^n$  la matriz bidimensional ( $K_s \times I_s$ ) cuyos componentes se obtienen de forma recursiva,

$$Q_{k,i}^n = \begin{cases} 1 + C_{k,i} Q_{k,i+1}^{n-1*} + S_{k,i} Q_{k-1,i-1}^{n-1*} + H_{k,i} Q_{k,i-1}^{n-1*} & (k,i) \in \mathcal{A}_1 \\ 1 + C_{k,i} Q_{k,i+1}^{n-1*} + S_{k,i} Q_{k-1,i-2}^{n-1*} + H_{k,i} Q_{k,i-1}^{n-1*} & (k,i) \in \mathcal{A}_2 \\ Q_{k,i}^0 & \text{en otro caso} \end{cases} \quad (7.93)$$

donde la región  $\mathcal{A}_1$  viene delimitada por  $2 \leq k \leq K_s$  y  $1 \leq i \leq 2$ , y la región  $\mathcal{A}_2$  por  $2 \leq k \leq K_s$ ,  $2 < i \leq I_s$ , y,

$$Q_{k,i}^{n-1*} = \begin{cases} Q_{k,i}^{n-1} & 2 \leq k \leq K_s, 1 \leq i \leq I_s \\ Q_{k,i}^0 & \text{en otro caso} \end{cases} \quad (7.94)$$

Las componentes  $Q_{k,i}^n = E_k^u(i^2)$  son la cota superior ajustada de  $E_k(i^2)$ . Cuanto mayor sea  $n$ , más ajustada será la cota. En nuestro caso, tomaremos  $K_s = I_s = 100$ .

# Capítulo 8

## Análisis del throughput del algoritmo ideal

### 8.1 Introducción

En las secciones precedentes se ha presentado una nueva familia de protocolos de resolución de colisiones. En ella todos los usuarios que tienen un paquete a transmitir, lo intentan con una probabilidad que aumenta o disminuye dependiendo de la información del feedback, de tal forma que el algoritmo pretende seguir el número de usuarios que están en contienda, es decir, la multiplicidad de la colisión. En los Capítulos 4 y 5 la información del feedback era binaria y en los Capítulos 6 y 7 la información del feedback era ternaria.

Es bastante intuitivo considerar que las prestaciones del algoritmo mejorarán a medida que el feedback nos proporcione más información. Siguiendo por esta línea, la máxima información que podríamos obtener del feedback sería el número de paquetes que hay en cada colisión. Es inmediato pensar entonces que éste sería un caso óptimo en este sentido. Un análisis de esta situación nos llevaría a conocer un límite máximo de prestaciones, con todo lo que ello comporta. Nos serviría, en primera instancia, para conocer cuanto de lejos se encuentran del caso ideal los algoritmos estudiados en los Capítulos anteriores.

Esta idea del conocimiento del número de paquetes que colisionan es interesante porque la probabilidad óptima para el acceso de estos usuarios sería, aproximadamente, el inverso de la multiplicidad de la colisión. La demostración de esta última afirmación puede encontrarse en el libro de R. Rom y M. Sidi [46], pág. 70.

A partir de este punto es obvio que dentro de la misma familia de algoritmos, la máxima información que podemos obtener a partir del feedback es

la multiplicidad de la colisión. Visto de otra forma, si el sistema fuera tan inteligente que, mediante la información obtenida a través del feedback, fuera capaz de saber en cada momento la multiplicidad de la colisión, la probabilidad de acceso de los usuarios sería óptima: su inverso. Vemos pues, que el throughput máximo que se puede obtener con esta familia de protocolos viene dada por aquel algoritmo en el que la probabilidad de acceso de los usuarios es siempre el inverso de la multiplicidad de la colisión.

En la literatura se han estudiado los casos en que se conoce la multiplicidad para otros algoritmos. Los primeros en tratar con el problema fueron Y. Yemini y L. Kleinrock [75]. Posteriormente fueron B. Tsybakov [76] y L. Georgiadis y P. Papantoni-Kazakos [77]. B. Tsybakov [76] considera el caso donde el número de paquetes en contienda se detecta siempre sin importar lo grande que sea este número, el resultado obtenido es un throughput máximo de 0.533. L. Georgiadis y P. Papantoni-Kazakos [77] estudian un caso que se puede considerar más real y llegan prácticamente al mismo resultado, un throughput máximo de 0.532. Ellos asumen que después de cada colisión, aunque la identidad de los paquetes en contienda se pierde completamente, se conoce el número (hasta un posible límite) de paquetes envueltos en la colisión y se retransmite a todos los usuarios. Esta información se puede proporcionar a través de un número de detectores de energía. Si existieran un número infinitamente grande de detectores de energía y se permitieran retardos infinitamente largos, N. Pippenger [78] demostró que se puede alcanzar un throughput igual a la unidad. Satisfacer este límite implica complejidad infinita.

Según L. Georgiadis y P. Papantoni-Kazakos [77] en el caso que no existan detectores de energía, diversos autores han estudiado los posibles límites superiores de los throughputs inducidos por la clase de protocolos estables. N. Pippenger [78] encontró mediante argumentos de la teoría de la información un límite igual a 0.744. Este límite fue recortado a 0.704 por P. Humblet, y más tarde a 0.6731 por M. Molle [79], donde utilizó un argumento de "genio mágico", a 0.6125 por R. Cruz y B. Hajek [80], a 0.587 por B. Tsybakov y V. Mikhailov [81] y a 0.5254 por T. Berger, N. Mehravari y G. Munson [82] (en este último a través de una conjetura).

En nuestro caso vamos a estudiar el caso en que la multiplicidad es conocida. Siguiendo con la idea de B. Tsybakov [76], consideramos que el sistema conoce la multiplicidad de la colisión. Por otro lado, y siguiendo con la idea de L. Georgiadis y P. Papantoni-Kazakos [77], consideramos que hay detectores de energía que nos informan de la multiplicidad de la colisión. Así pues, al inicio de un CRI no conocemos cuantos usuarios hay en contienda, pero lo sabremos desde la primera colisión, hasta el final del CRI.

Siguiendo las premisas desarrolladas hasta ahora, la operación del al-

goritmo es de la siguiente manera. Al principio de un CRI, y puesto que no sabemos cuantos usuarios han llegado en el periodo permitido, todos los usuarios intentarán acceder con una probabilidad 1. En caso que haya una colisión, los detectores de energía nos informarán que hay una colisión de multiplicidad  $k$ . Por tanto, la probabilidad de acceso para todos los usuarios en la siguiente ranura será  $1/k$ . A partir de entonces, y suponiendo que no hay errores en los detectores de energía, sabremos en cada ranura cuantos usuarios entrarán en contienda y, por tanto, podremos optimizar la probabilidad de acceso.

El hecho de no considerar errores en los detectores de energía se basa en que estamos analizando un caso ideal, y no tiene mucho sentido complicar un modelo añadiendo la posibilidad de errores, es decir, intentar acercarlo a un modelo real, cuando el modelo es de por sí ideal. Tampoco no se ha considerado que haya un límite en la multiplicidad detectable por los detectores de energía, porque complica en exceso el problema. De todas formas, L. Georgiadis y P. Papantoni-Kazakos [77] lo analizan y llegan a la conclusión que con un detector de multiplicidad 8, se obtiene prácticamente el mismo resultado que con un detector de multiplicidad ilimitada.

## 8.2 Análisis del throughput

En esta sección vamos a calcular el throughput máximo estable para el algoritmo que hemos presentado, resultante de la idealización del algoritmo WSA. La metodología que vamos a seguir va a ser exactamente la misma que para el caso no ideal. A grandes rasgos, los pasos son los siguientes.

La condición de ergodicidad de la cadena de Markov formada por la secuencia  $d_i$ ,  $i \geq 1$ , correspondiente al retraso en  $i$ , es,

$$E(i|\Delta) < \Delta \quad (8.1)$$

donde  $E(i|\Delta)$  denota la longitud esperada de un CRI dado que empieza con un intervalo examinado de longitud  $\Delta$ .

Para el cálculo de  $E(i|\Delta)$ , condicionamos a que haya una colisión de multiplicidad  $k$ , y el resultado es,

$$E(i|\Delta) = \sum_{k=0}^{\infty} E(i|\Delta, k)p(k|\Delta) = \sum_{k=0}^{\infty} E_k(i)p(k|\Delta) \quad (8.2)$$

ya que  $E(i|\Delta, k) = E_k(i)$  es la longitud esperada de un CRI dado que empieza con una colisión de multiplicidad  $k$  y depende sólo de  $k$ , y donde, debido al

Algoritmo WSA ideal	$\lambda^* = 0.5052$	$\Delta^* = 2.77$
Algoritmo WSA ternario	$\lambda^* = 0.4365$	$\Delta^* = 2.38$
Algoritmo WSA binario	$\lambda^* = 0.4335$	$\Delta^* = 2.49$

Tabla 8.1: Throughputs y tamaños de ventana óptimos. Los algoritmos representados son el WSA (ALOHA estabilizado de ventana) en su versión ideal, ternaria y binaria.

proceso de llegadas poissoniano,

$$p(k|\Delta) = \exp(-\lambda\Delta) \frac{(\lambda\Delta)^k}{k!} \quad (8.3)$$

En el Apéndice de este Capítulo mostraremos cómo,

1.  $E_k(i)$  está acotada linealmente,  $E_k(i) \leq E_k^0(i) = c_1 k + c_0$ .
2.  $E_k(i)$  puede ser calculada exáctamente para un  $k$  concreto,  $E_k(i)$ .

La expresión (8.1) junto con los resultados de los puntos 1 y 2 son usados en la computación del throughput del algoritmo. El procedimiento es como sigue.

Debido a la cota superior de  $E_k(i)$  y a la inecuación (8.1), concluimos que la siguiente condición es suficiente para la estabilidad,

$$E^u(i|\Delta) < \Delta \quad (8.4)$$

con,

$$E^u(i|\Delta) = \sum_{k=0}^{30} E_k(i)p(k|\Delta) + \sum_{k=31}^{\infty} E_k^0(i)p(k|\Delta) \quad (8.5)$$

Es decir,

$$E^u(i|\Delta) < \Delta \Rightarrow E(i|\Delta) < \Delta \quad (8.6)$$

ya que,

$$E^u(i|\Delta) \geq E(i|\Delta) \quad (8.7)$$

Así, para calcular  $E^u(i|\Delta)$ , descomponemos en dos sumandos,

$$E^u(i|\Delta) = \sum_{k=0}^{30} E_k(i)p(k|\Delta) + \sum_{k=31}^{\infty} E_k^0(i)p(k|\Delta) \quad (8.8)$$

Después de algunas operaciones con el segundo sumando, se obtiene,

$$E^u(i|d) = \sum_{k=0}^{30} E_k(i)p(k|d) + \varepsilon \left\{ \lambda d - \sum_{k=0}^{30} kp(k|d) \right\} \\ + (2 - \varepsilon) \left\{ 1 - \sum_{k=0}^{30} p(k|d) \right\} \quad (8.9)$$

ya que,

$$E_k^0(i) = 2 + (k - 1)\varepsilon \quad (8.10)$$

y donde,

$$p(k|d) = \exp(-\lambda d) \frac{(\lambda d)^k}{k!} \quad (8.11)$$

Una vez calculado  $E^u(i|\Delta)$ , vamos a obtener una expresión para el cálculo de  $\lambda^*$ . Para ello definimos,

$$E^u(i|\Delta) \equiv f(\lambda\Delta) \quad (8.12)$$

$$x \equiv \lambda\Delta \quad (8.13)$$

Por consiguiente, sustituyendo las expresiones (8.12) y (8.13) en la ecuación (8.1), se concluye que, para la estabilidad del algoritmo, es suficiente que la tasa de entrada  $\lambda$  satisfaga la siguiente inecuación,

$$\lambda < \sup_{x \geq 0} \frac{x}{f(x)} \quad (8.14)$$

La siguiente condición especifica una región de valores de  $\lambda$  para los cuales el algoritmo es inestable,

$$\lambda > \sup_{x \geq 0} \frac{x}{g(x)} \quad (8.15)$$

donde,

$$g(x) = \sum_{k=0}^{30} E_k^u(i) e^{-x} \frac{x^k}{k!} \quad (8.16)$$

La maximización de las expresiones (8.14) y (8.15) ha sido realizada numéricamente y proporciona el throughput ( $\lambda^*$ ) así como el tamaño de ventana óptimo  $\Delta^*$ . En todos los casos el orden de magnitud de la diferencia es menor que  $10^{-4}$ . El tamaño de ventana óptimo se encuentra como  $x^*(\lambda^*)^{-1}$ , donde  $x^*$  es el valor que alcanza el supremo en la expresión (8.14). Los resultados comparativos con las otras versiones del algoritmo ALOHA estabilizado de ventana son expuestos en la tabla 8.1, mientras que la comparación con los otros algoritmos donde se conoce la multiplicidad se reflejan en la tabla 8.2.



Algoritmo WSA ideal	$\lambda^* = 0.5052$
Algoritmo CRAI	$\lambda^* = 0.5324$
Algoritmo CRAT	$\lambda^* = 0.5330$

Tabla 8.2: Throughputs óptimos. Los algoritmos representados son el protocolo ALOHA estabilizado de ventana ideal (WSA ideal), el protocolo de L. Georgiadis and P. Papantoni-Kazakos (*collision resolution with addition information*, CRAI) y el protocolo de B. Tsybakov (CRAT).

## 8.3 Apéndice

### 8.3.1 Cálculo de la longitud media $E(l | d)$ , $1 \leq d \leq \Delta$

La longitud media del algoritmo es,

$$E(l|d) = \sum_{k=0}^{\infty} E_k(l)p(k|d) \quad (8.17)$$

donde,

$$p(k|d) = \exp(-\lambda d) \frac{(\lambda d)^k}{k!} \quad (8.18)$$

Definimos, pues,  $E_k(l)$  como la longitud esperada de un CRI dado que empieza con una colisión de multiplicidad  $k$ .

Definimos también  $G_{k,i}$  como el número esperado de ranuras necesarias por el algoritmo para la transmisión exitosa de  $k$  paquetes, dado que el valor del contador es igual a  $r_i = i$  en la primera ranura. Debido a que se conoce la multiplicidad de la colisión, es decir, se conoce el número de paquetes en contienda, el valor óptimo de  $i$  será  $k$ . Notar que se conoce la multiplicidad a partir de la primera colisión. Será, por lo tanto,  $G_k = G_{k,i}|_{i=k}$  el número esperado de ranuras necesarias por el algoritmo para la transmisión exitosa de  $k$  paquetes.

Nuestro objetivo será obtener una cota superior de  $E_k(l)$ , que llamaremos  $E_k^0(l)$ , para acotar el comportamiento en  $k \rightarrow \infty$ . Así, obtendremos una cota superior bastante ajustada de  $E(l|d)$ , que llamaremos  $E^u(l|d)$ , y que será la que utilizaremos a todos los efectos como  $E(l|d)$ .

Según las definiciones dadas, se puede ver que,

$$E_k(l) = \begin{cases} 1 + G_k & k \geq 2 \\ 1 & k < 2 \end{cases} \quad (8.19)$$

De la expresión (4.23), con  $\epsilon = \delta = 0$ , encontramos que,

$$G_k = 1 + (1 - S_{k,k})G_k + S_{k,k}G_{k-1} \quad k \geq 2 \quad (8.20)$$

con el valor inicial  $G_1 = 1$ .

De la expresión (8.20), despejamos  $G_k$ ,

$$G_k = \frac{1}{s_{k,k}} + G_{k-1} \quad k \geq 2 \quad (8.21)$$

donde,

$$s_{k,k} = \left(1 - \frac{1}{k}\right)^{k-1} \quad (8.22)$$

Entonces,

$$G_k = 1 + \left(\frac{2}{1}\right)^1 + \left(\frac{3}{2}\right)^2 + \dots + \left(\frac{k}{k-1}\right)^{k-1} \quad k \geq 2 \quad (8.23)$$

Tenemos una expresión para el cálculo de  $E_k(l)$ ,  $k \geq 2$ . El siguiente paso será el desarrollo de una cota superior de  $E_k(l)$ .

Se puede ver que ,

$$\left(\frac{k}{k-1}\right)^{k-1} < e \quad k \geq 2 \quad (8.24)$$

Entonces, de la expresión (8.23), se obtiene,

$$G_k \leq 1 + (k-1)e \quad k \geq 2 \quad (8.25)$$

Y finalmente encontramos,

$$E_k(l) = 1 + G_k \leq E_k^0(l) = 2 + (k-1)e = ek + (2-e) \quad k \geq 2 \quad (8.26)$$

Cabe resaltar en este punto que  $E_k(l)$  está acotada por una función lineal en  $k$ .



# Capítulo 9

## Análisis del retardo del algoritmo ideal

### 9.1 Introducción

En la sección anterior se ha calculado el throughput máximo estable del algoritmo propuesto en su vertiente ideal. La idealidad considerada se basaba en que el feedback era capaz de informar a los usuarios de la multiplicidad de la colisión. El estudio del throughput en el caso ideal era interesante puesto que nos daba una idea de lo que se puede llegar a conseguir con esta familia de algoritmos propuesta. Parece obvio que un estudio del retardo en el caso ideal sería también muy útil.

En la literatura no hay muchos estudios sobre el retardo ideal. El autor sólo conoce de la existencia de los trabajos de B. Tsybakov y N. Likhanov [83] y [84], sobre este tema, y un trabajo posterior, de B. Hajek, N. Likhanov y B. Tsybakov [85], y en ningún caso la idealidad viene dada por la cognición de la multiplicidad de la colisión.

Así pues, en esta sección vamos a realizar un estudio del retardo del algoritmo ALOHA estabilizado de ventana (WSA) propuesto en el Capítulo 4 en su caso ideal. La metodología utilizada será la misma que en su momento se utilizó para el cálculo del retardo del algoritmo propuesto, y desarrollada en el Capítulo 5. Ello es debido a que estamos utilizando el mismo protocolo de acceso al canal y un protocolo de resolución de colisiones perteneciente a la misma familia.

Aplicando la metodología resumida en la próxima sección, y los cálculos explicados en las secciones subsiguientes, se obtienen los resultados reflejados en la tabla 9.1.

En ellos se comprueba que el hecho que estemos analizando el algoritmo

$\lambda$	Algoritmo WSA					
	ideal		ternaria		binaria	
	$D^l$	$D^u$	$D^l$	$D^u$	$D^l$	$D^u$
0.02	1.549	1.550	1.563	1.564	1.560	1.560
0.06	1.661	1.666	1.704	1.710	1.710	1.716
0.10	1.794	1.810	1.872	1.890	1.887	1.905
0.16	2.034	2.082	2.224	2.276	2.257	2.310
0.20	2.246	2.327	2.562	2.654	2.616	2.709
0.24	2.523	2.651	3.047	3.203	3.138	3.298
0.30	3.153	3.420	4.383	4.726	4.588	4.946
0.32	3.458	3.764	5.166	5.622	5.443	5.923
0.36	4.334	4.814	8.056	8.948	8.660	9.619
0.38	4.993	5.606	11.113	12.477	12.174	13.669
0.40	5.913	6.714	17.564	19.946	20.003	22.715
0.42	7.280	8.363	39.185	45.018	51.351	58.994
0.44	9.508	11.056	–	–	–	–
0.46	13.750	16.189	–	–	–	–
0.48	24.794	29.568	–	–	–	–
0.50	121.108	146.342	–	–	–	–

Tabla 9.1: Límites superior e inferior de los retardos esperados. Los algoritmos representados son el WSA (ALOHA estabilizado de ventana) en su versión ideal, ternaria y binaria.

ideal de la familia ALOHA estabilizado de ventana, proporciona un retardo mínimo y que es menor, por coherencia, al retardo introducido por el algoritmo ALOHA estabilizado de ventana tanto en su versión binaria como ternaria.

## 9.2 Límites del retardo medio de paquete

Siguiendo el desarrollo expuesto en el Capítulo 5, se llega a las mismas expresiones que en el caso del algoritmo WSA. Estas expresiones dependen básicamente de la longitud media  $E(l | d)$ , de la suma de los retardos media  $E(\psi | d)$  y del segundo momento de la longitud  $E(l^2 | d)$  que son calculadas específicamente para el caso ideal, y están desarrolladas en secciones posteriores.

Vamos a recordar las expresiones con las que se puede calcular tanto una cota superior como inferior del retardo. Toda la metodología ha sido obtenida del trabajo de L. Georgiadis, L. Merakos y P. Papantoni-Kazakos [68].

Los límites en el retardo medio de paquete se obtienen a partir de la siguiente expresión,

$$\frac{W_1^l}{\lambda H_1^u} = D^l \leq D \leq D^u = \frac{W_1^u}{\lambda H_1^l} \quad (9.1)$$

En la expresión anterior,  $H_1^u$  y  $H_1^l$  son una cota superior e inferior de la longitud de sesión media, respectivamente.  $W_1^u$  y  $W_1^l$  son una cota superior e inferior de la suma de los retardos experimentados por todos los paquetes que fueron transmitidos exitosamente en el intervalo  $(v, v_1]$ , respectivamente. Las expresiones que permiten calcularlos, vienen dadas por,

$$H_1^u = E(i | 1) + \alpha_u (E(i | 1) - (1 + \lambda)e^{-\lambda}) + \beta_u (1 - (1 + \lambda)e^{-\lambda}) \quad (9.2)$$

$$H_1^l = H_1^u - (\beta_u - \beta_l) (1 - (1 + \lambda)e^{-\lambda}) \quad (9.3)$$

$$W_1^u = E(\psi | 1) + E(\theta | 1) + \gamma_u (E(i^2 | 1) - (1 + \lambda)e^{-\lambda}) + \delta_u (E(i | 1) - (1 + \lambda)e^{-\lambda}) - \zeta_u (1 - (1 + \lambda)e^{-\lambda}) \quad (9.4)$$

$$W_1^l = W_1^u - (\zeta_u - \zeta_l) (1 - (1 + \lambda)e^{-\lambda}) \quad (9.5)$$

donde,

$$\alpha_u = \alpha_l = \frac{E(i | \Delta)}{\Delta - E(i | \Delta)} \quad (9.6)$$

$$\beta_u = \max\{-\alpha_u, \sup_{1 \leq d \leq \Delta} \rho(d)\} \quad \beta_l = \inf_{1 \leq d \leq \Delta} \rho(d) \quad (9.7)$$

$$\gamma_u = \gamma_l = \frac{E(N | \Delta)}{2(\Delta - E(i | \Delta))} \quad (9.8)$$

$$\delta_u = \delta_l = \frac{E(\psi | \Delta) + E(\theta | \Delta) - \Delta E(N | \Delta) + \gamma_u (\Delta^2 + E(i^2 | \Delta) - 2\Delta E(i | \Delta))}{\Delta - E(i | \Delta)} \quad (9.9)$$

$$\zeta_u = \sup_{1 \leq d \leq \Delta} \phi(d) \quad \zeta_l = \inf_{1 \leq d \leq \Delta} \phi(d) \quad (9.10)$$

con,

$$\rho(d) = \frac{E(l | d) + \alpha_u (E(l | d) - d - (1 + \lambda d)e^{-\lambda d})}{(1 + \lambda d)e^{-\lambda d}} \quad (9.11)$$

$$\phi(d) = \frac{E(\psi | d) + E(\theta | d) + \gamma_u (E(l^2 | d) - d^2 - (1 + \lambda d)e^{-\lambda d}) - \delta_u (d - E(l | d) + (1 + \lambda d)e^{-\lambda d})}{(1 + \lambda d)e^{-\lambda d}} \quad (9.12)$$

$$E(N | \Delta) = \lambda \Delta \quad (9.13)$$

$$E(\theta | t) = \frac{\lambda t^2}{2} \quad (9.14)$$

En definitiva, sólo nos queda por calcular  $E(\psi | d)$  y  $E(l^2 | d)$ , que se encuentran en las secciones siguientes. El cálculo de  $E(l | d)$  se puede encontrar en el Capítulo 8.

## 9.3 Apéndice

### 9.3.1 Cálculo de la suma de retardos media $E(\psi | d)$ , $1 \leq d \leq \Delta$

La suma de retardos media del algoritmo es,

$$E(\psi | d) = \sum_{k=0}^{\infty} E_k(\psi) p(k | d) \quad (9.15)$$

donde,

$$p(k | d) = e^{-\lambda d} \frac{(\lambda d)^k}{k!} \quad (9.16)$$

Definimos, pues,  $E_k(\psi)$  como la suma de retardos esperada en un CRI dado que empieza con una colisión de multiplicidad  $k$ .

Definimos también  $\Psi_{k,i}$  como suma de retardos esperada en ranuras para la transmisión exitosa de  $k$  paquetes, dado que el valor del contador es igual a  $r_i = i$  en la primera ranura. Debido a que se conoce la multiplicidad de la colisión, es decir, se conoce el número de paquetes en contienda, el valor óptimo de  $i$  será  $k$ . Notar que se conoce la multiplicidad a partir de la primera

colisión. Será, por lo tanto,  $\Psi_k = \Psi_{k,i}|_{i=k}$  el número esperado de ranuras necesarias por el algoritmo para la transmisión exitosa de  $k$  paquetes.

Nuestro objetivo será obtener una cota superior de  $E_k(\psi)$ , que llamaremos  $E_k^0(\psi)$ , para acotar el comportamiento en  $k \rightarrow \infty$ . Así, obtendremos una cota superior bastante ajustada de  $E(\psi|d)$ , que llamaremos  $E^u(\psi|d)$ , y que será la que utilizaremos a todos los efectos como  $E(\psi|d)$ .

Según las definiciones dadas, se puede ver que,

$$E_k(\psi) = \begin{cases} k + \Psi_k & k \geq 2 \\ k & k < 2 \end{cases} \quad (9.17)$$

De la expresión (5.83), con  $\varepsilon = \delta = 0$ , encontramos que,

$$\Psi_k = k + (1 - S_{k,k})\Psi_k + S_{k,k}\Psi_{k-1} \quad k \geq 2 \quad (9.18)$$

con el valor inicial  $\Psi_1 = 1$ .

De la expresión (9.18), despejamos  $\Psi_k$ ,

$$\Psi_k = \frac{k}{S_{k,k}} + \Psi_{k-1} \quad k \geq 2 \quad (9.19)$$

donde  $S_{k,k}$  viene dado por la expresión (8.22).

Entonces,

$$\Psi_k = 1 + 2 \binom{2}{1}^1 + 3 \binom{3}{2}^2 + \dots + k \binom{k}{k-1}^{k-1} \quad k \geq 2 \quad (9.20)$$

Tenemos una expresión para el cálculo de  $E_k(\psi)$ ,  $k \geq 2$ . El siguiente paso será el desarrollo de una cota superior de  $E_k(\psi)$ .

Se puede ver que,

$$\left(\frac{k}{k-1}\right)^{k-1} < e \quad k \geq 2 \quad (9.21)$$

Entonces, de la expresión (9.20), se obtiene,

$$\begin{aligned} \Psi_k &\leq 1 + e(2 + 3 + \dots + k) = 1 + e \left( \frac{k(k+1)}{2} - 1 \right) \\ &= \frac{\varepsilon}{2}k^2 + \frac{\varepsilon}{2}k + (1 - e) \end{aligned} \quad k \geq 2 \quad (9.22)$$

Y finalmente encontramos,

$$E_k(\psi) = k + \Psi_k \leq E_k^0(\psi) = \frac{\varepsilon}{2}k^2 + \left(\frac{\varepsilon}{2} + 1\right)k + (1 - e) \quad k \geq 2 \quad (9.23)$$



Cabe destacar en este punto que  $E_k(\psi)$  está acotada superiormente por una función cuadrática en  $k$ . Así, para calcular  $E^u(\psi|d)$ , descomponemos en dos sumandos,

$$E^u(\psi|d) = \sum_{k=0}^{30} E_k(\psi)p(k|d) + \sum_{k=31}^{\infty} E_k^0(\psi)p(k|d) \quad (9.24)$$

Después de algunas operaciones con el segundo sumando, se obtiene,

$$\begin{aligned} E^u(\psi|d) &= \sum_{k=0}^{30} E_k(\psi)p(k|d) + \frac{\epsilon}{2} \left\{ (\lambda d)^2 + \lambda d - \sum_{k=0}^{30} k^2 p(k|d) \right\} \\ &+ \left( \frac{\epsilon}{2} + 1 \right) \left\{ \lambda d - \sum_{k=0}^{30} k p(k|d) \right\} + (1 - \epsilon) \left\{ 1 - \sum_{k=0}^{30} p(k|d) \right\} \end{aligned} \quad (9.25)$$

ya que,

$$E_k^0(\psi) = \frac{\epsilon}{2} k^2 + \left( \frac{\epsilon}{2} + 1 \right) k + (1 - \epsilon) \quad (9.26)$$

y donde,

$$p(k|d) = e^{-\lambda d} \frac{(\lambda d)^k}{k!} \quad (9.27)$$

### 9.3.2 Cálculo del segundo momento de la longitud $E(l^2 | d)$ , $1 \leq d \leq \Delta$

El segundo momento de la longitud del algoritmo es,

$$E(l^2|d) = \sum_{k=0}^{\infty} E_k(l^2)p(k|d) \quad (9.28)$$

donde,

$$p(k|d) = e^{-\lambda d} \frac{(\lambda d)^k}{k!} \quad (9.29)$$

Definimos, pues,  $E_k(l^2)$  como el segundo momento de la longitud de un CRI dado que empieza con una colisión de multiplicidad  $k$ .

Definimos también  $Q_{k,i}$  como el segundo momento del número esperado de ranuras necesarias por el algoritmo para la transmisión exitosa de  $k$  paquetes, dado que el valor del contador es igual a  $r_i = i$  en la primera ranura. Debido a que se conoce la multiplicidad de la colisión, es decir, se conoce el número

de paquetes en contienda, el valor óptimo de  $i$  será  $k$ . Notar que se conoce la multiplicidad a partir de la primera colisión. Será, por lo tanto,  $Q_k = Q_{k,i}|_{i=k}$  el número esperado de ranuras necesarias por el algoritmo para la transmisión exitosa de  $k$  paquetes.

Nuestro objetivo será obtener una cota superior de  $E_k(I^2)$ , que llamaremos  $E_k^0(I^2)$ , para acotar el comportamiento en  $k \rightarrow \infty$ . Así, obtendremos una cota superior bastante ajustada de  $E(I^2|d)$ , que llamaremos  $E^u(I^2|d)$ , y que será la que utilizaremos a todos los efectos como  $E(I^2|d)$ .

Según las definiciones dadas, se puede ver que,

$$E_k(I^2) = \begin{cases} 2G_k + 1 + Q_k & k \geq 2 \\ 1 & k < 2 \end{cases} \quad (9.30)$$

De la expresión (5.106), con  $\varepsilon = \delta = 0$ , encontramos que,

$$Q_k = 2G_k - 1 + (1 - S_{k,k})Q_k + S_{k,k}Q_{k-1} \quad k \geq 2 \quad (9.31)$$

con el valor inicial  $Q_1 = 1$ .

De la expresión (9.31), despejamos  $Q_k$ ,

$$Q_k = \frac{2G_k - 1}{S_{k,k}} + Q_{k-1} \quad k \geq 2 \quad (9.32)$$

donde  $S_{k,k}$  viene dado por la expresión (8.22).

Entonces,

$$Q_k = 1 + (2G_2 - 1) \left(\frac{2}{1}\right)^1 + (2G_3 - 1) \left(\frac{3}{2}\right)^2 + \dots + (2G_k - 1) \left(\frac{k}{k-1}\right)^{k-1} \quad k \geq 2 \quad (9.33)$$

donde  $G_k$  viene dado por la expresión (8.21).

Tenemos una expresión para el cálculo de  $E_k(I^2)$ ,  $k \geq 2$ . El siguiente paso será el desarrollo de una cota superior de  $E_k(I^2)$ .

Se puede ver que,

$$\left(\frac{k}{k-1}\right)^{k-1} < e \quad k \geq 2 \quad (9.34)$$

Entonces, de la expresión (9.33), se obtiene,

$$\begin{aligned} Q_k &\leq 1 + e((2G_2 - 1) + (2G_3 - 1) + \dots + (2G_k - 1)) \\ &= 1 + e(2(G_2 + G_3 + \dots + G_k) - (k - 1)) \end{aligned} \quad k \geq 2 \quad (9.35)$$

Viendo que,

$$\begin{aligned} G_2 &= 1 + \left(\frac{2}{1}\right)^1 \\ G_3 &= 1 + \left(\frac{2}{1}\right)^1 + \left(\frac{3}{2}\right)^2 \\ G_4 &= 1 + \left(\frac{2}{1}\right)^1 + \left(\frac{3}{2}\right)^2 + \left(\frac{4}{3}\right)^3 \\ &\vdots \\ G_k &= 1 + \left(\frac{2}{1}\right)^1 + \left(\frac{3}{2}\right)^2 + \left(\frac{4}{3}\right)^3 + \dots + \left(\frac{k}{k-1}\right)^k \end{aligned} \quad (9.36)$$

entonces,

$$G_2 + G_3 + \dots + G_k \leq (k-1) + (k-1) \left(\frac{e}{1}\right)^1 + (k-2) \left(\frac{e}{2}\right)^2 + \dots + (k - (k-1)) \left(\frac{e}{k-1}\right)^{k-1} \quad k \geq 2 \quad (9.37)$$

Y, aplicando la inecuación,

$$\left(\frac{e}{k-1}\right)^{k-1} < e \quad k \geq 2 \quad (9.38)$$

obtenemos,

$$\begin{aligned} G_2 + G_3 + \dots + G_k &\leq (k-1) + e(k(k-1) - (1+2+3+\dots \\ &+ (k-1))) = (k-1) + e \left( k(k-1) - \frac{k(k-1)}{2} \right) \quad k \geq 2 \\ &= (k-1) + \frac{e}{2}k(k-1) \end{aligned} \quad (9.39)$$

Así pues,

$$\begin{aligned} Q_k &\leq 1 + e(2(G_2 + G_3 + \dots + G_k) - (k-1)) \\ &\leq 1 + e \left( 2 \left( (k-1) + \frac{e}{2}k(k-1) \right) - (k-1) \right) \quad k \geq 2 \\ &= 1 + e(k-1)(1+ke) = e^2k^2 - (e^2 - e)k + (1 - e) \end{aligned} \quad (9.40)$$

Y finalmente encontramos,

$$\begin{aligned} E_k(I^2) = 2G_k + 1 + Q_k &\leq E_k^0(I^2) = 2(1 + e(k-1)) + 1 \\ &+ (e^2k^2 - (e^2 - e)k + (1 - e)) = e^2k^2 - (e^2 - 3e)k + (4 - 3e) \quad k \geq 2 \end{aligned} \quad (9.41)$$

Cabe señalar en este punto que  $E_k(I^2)$  está acotada superiormente por una función cuadrática en  $k$ .

Así, para calcular  $E^u(I^2|d)$ , descomponemos en dos sumandos,

$$E^u(I^2|d) = \sum_{k=0}^{30} E_k(I^2)p(k|d) + \sum_{k=31}^{\infty} E_k^0(I^2)p(k|d) \quad (9.42)$$

Después de algunas operaciones con el segundo sumando, se obtiene,

$$\begin{aligned} E^u(I^2|d) &= \sum_{k=0}^{30} E_k(I^2)p(k|d) + e^2 \left\{ (\lambda d)^2 + \lambda d - \sum_{k=0}^{30} k^2 p(k|d) \right\} \\ &- (e^2 - 3e) \left\{ \lambda d - \sum_{k=0}^{30} k p(k|d) \right\} + (4 - 3e) \left\{ 1 - \sum_{k=0}^{30} p(k|d) \right\} \end{aligned} \quad (9.43)$$

ya que,

$$E_k^0(I^2) = e^2k^2 - (e^2 - 3e)k + (4 - 3e) \quad (9.44)$$

y donde

$$p(k|d) = e^{-\lambda d} \frac{(\lambda d)^k}{k!} \quad (9.45)$$

# Capítulo 10

## Otros análisis del algoritmo

### 10.1 Introducción

A lo largo de los Capítulos anteriores hemos realizado un análisis del throughput y el retardo de una nueva familia de protocolos de acceso aleatorio, el ALOHA estabilizado de ventana, en su variante binaria, ternaria e ideal.

Esta familia de protocolos de acceso aleatorio han demostrado tener unas muy buenas características de throughput, de retardo y de robustez frente a errores, dignos de ser considerados ante protocolos pertenecientes a otras familias.

En este Capítulo volveremos de nuevo sobre el algoritmo original, la versión binaria, y reflexionaremos sobre él mismo de tal forma que podamos extraer ideas que nos ayuden a conocerlo mejor.

Para ello, los razonamientos que siguen a continuación tienen como objetivo analizar desde un punto de vista intuitivo y heurístico las razones que llevaron a su concepción.

La primera idea que aparece clara es que el algoritmo aquí presentado es susceptible de ser mejorado, por la sencilla razón que en su concepción no se pretendía optimizar ningún parámetro en concreto.

Surge, por consiguiente, una pregunta inmediata: Por qué las reglas del algoritmo fueron configuradas de esta forma?

### 10.2 Probabilidades de acceso iniciales

Partimos de la base que nuestro objetivo es regular el acceso de un número aleatorio de paquetes al sistema, sabiendo que hasta que estos no sean totalmente transmitidos, los demás paquetes estarán en espera.

Algoritmo WSA binario		
$r_{t_0} = 1$	$\lambda^* = 0.4335$	$\Delta^* = 2.494$
$r_{t_0} = 2$	$\lambda^* = 0.4202$	$\Delta^* = 4.734$
$r_{t_0} = 3$	$\lambda^* = 0.4123$	$\Delta^* = 6.922$

Tabla 10.1: Throughputs y tamaños de ventana para distintos  $r_{t_0}$  y caso libre de error en el WSA (ALOHA estabilizado de ventana) en su versión binaria.

Sabemos también que el número esperado de paquetes intentando acceder al inicio de un CRI será aproximadamente 1 y que el proceso de llegadas es de Poisson. Según estas premisas, es obvio que la probabilidad de acceso de los citados paquetes debe ser 1.

En este punto aparece un punto digno de discusión, el hecho que el número esperado de paquetes sea aproximadamente 1 es debido a que  $\lambda\Delta$  es aproximadamente 1, donde  $\lambda$  es la tasa de llegadas y  $\Delta$  es la ventana temporal del algoritmo. Por consiguiente, si elegimos  $\Delta$  tal que  $\lambda\Delta$  sea del orden de 2, seguirá siendo cierta la afirmación que la probabilidad óptima inicial es 1? O visto desde otro punto de vista, qué prestaciones tendrá el algoritmo si forzamos a una probabilidad inicial  $1/2$ ?

Intuitivamente, lo que podemos asegurar en este caso es que la ventana óptima inicial será tal que  $\lambda\Delta$  será aproximadamente 2. Aparece, por tanto, la necesidad de comprobar cómo son las prestaciones del algoritmo para las probabilidades iniciales del algoritmo distintas de 1. A primera vista podemos prever que no serán tan buenas como en el caso inicial porque el CRI durará siempre como mínimo 2 ranuras, mientras que con anterioridad la duración mínima era de una ranura.

Vamos, por tanto, a definir y analizar el algoritmo WSA con distintos  $r_{t_0}$ . En primer lugar describiremos brevemente el algoritmo que analizaremos posteriormente.

Durante los CRI que empiezan con una ranura de colisión, cada usuario actúa independientemente vía la utilización de un contador cuyo valor en el tiempo  $t$  se denota por  $r_t$ . Este contador tiene el mismo valor para todos los usuarios, pero es cada usuario quien actualiza su propio contador dependiendo del feedback. Cuando un usuario transmite por primera vez, fija  $r_t = r_{t_0}$ . Los valores del contador son enteros, y son actualizados y utilizados de acuerdo con las reglas siguientes.

1. Los usuarios transmiten en la ranura  $t$  con probabilidad  $1/r_t$ .
2. Los valores del contador cambian con el tiempo de la manera siguiente:
  - (a) Si  $x_{t-1} = NC$ , entonces  $r_t = r_{t-1} - 1$

$\epsilon$	$\delta$	Alg. WSA binario		
		$r_{t_0} = 1$	$r_{t_0} = 2$	$r_{t_0} = 3$
0.00	0.00	0.4335	0.4202	0.4123
0.00	0.01	0.4292	0.4161	0.4082
0.00	0.10	0.3904	0.3786	0.3714
0.00	0.20	0.3473	0.3367	0.3302
0.00	0.40	0.2610	0.2524	0.2471
0.00	0.50	0.2179	0.2102	0.2055
0.01	0.00	0.4321	0.4191	0.4113
0.10	0.00	0.4173	0.4066	0.4005
0.20	0.00	0.3940	0.3860	0.3820
0.40	0.00	0.2814	0.2802	0.2816
0.45	0.00	0.2006	0.2026	0.2060
0.10	0.10	0.3733	0.3642	0.3589
0.20	0.20	0.3050	0.3001	0.2975
0.30	0.30	0.2252	0.2241	0.2238
0.40	0.30	0.1591	0.1609	0.1624
0.30	0.50	0.1508	0.1498	0.1494
0.30	0.70	0.0854	0.0842	0.0836

Tabla 10.2: Throughputs en función de  $\epsilon$  y  $\delta$  para distintos valores de  $r_{t_0}$  en el WSA (ALOHA estabilizado de ventana) en su versión binaria. Los tamaños de ventana son optimizados para cada par  $(\epsilon, \delta)$

(b) Si  $x_{t-1} = C$ , entonces  $r_t = r_{t-1} + 1$

De estas reglas se puede ver que un CRI acaba con una ranura de no colisión y  $r_t = 1$ .

Los resultados son expuestos en la tabla 10.1, donde muestra los throughputs y ventanas óptimos para distintos valores de  $r_{t_0}$  y el caso libre de error. Por su parte, la tabla 10.2 representa los throughputs obtenidos por el algoritmo para  $r_{t_0} = 1, 2$  y  $3$ , en el caso que las probabilidades de error  $\epsilon$  y  $\delta$  no sean nulas. En este caso los tamaños de ventana son optimizados para cada par  $(\epsilon, \delta)$ . Ello produce que para cada uno de estos valores tengamos un tamaño de ventana distinto. Ello viene reflejado en la tabla 10.3.

Por último, si no conocemos de antemano las probabilidades de error, fijamos el tamaño de ventana al óptimo para el caso libre de error, que se encuentran en la tabla 10.1. Bajo este supuesto se obtienen otros valores para el throughput que están reflejados en la tabla 10.4.

De sendas tablas se extrae un par de conclusiones. Como era de esperar, el throughput máximo estable se consigue para  $r_{t_0} = 1$ , pero es sorprendente

		Alg. WSA binario		
$\epsilon$	$\delta$	$r_{t_0} = 1$	$r_{t_0} = 2$	$r_{t_0} = 3$
0.00	0.00	2.494	4.734	6.922
0.00	0.01	2.517	4.779	6.992
0.00	0.10	2.735	5.229	7.682
0.00	0.20	3.022	5.825	8.602
0.00	0.40	3.807	7.449	11.121
0.00	0.50	4.367	8.585	12.880
0.01	0.00	2.530	4.800	7.017
0.10	0.00	2.917	5.511	8.038
0.20	0.00	3.562	6.697	9.744
0.30	0.00	4.721	8.855	12.863
0.40	0.00	7.817	14.900	21.828
0.45	0.00	13.855	56.344	51.720
0.10	0.10	3.230	6.154	9.026
0.10	0.20	3.607	6.937	10.236
0.10	0.30	4.065	7.901	11.739
0.20	0.10	4.005	7.602	11.129
0.20	0.20	4.540	8.716	12.854
0.20	0.30	5.197	10.106	15.031
0.30	0.10	5.451	10.333	15.120
0.30	0.30	7.447	14.549	21.713
0.40	0.30	14.831	30.352	46.014
0.30	0.50	10.623	21.572	33.002
0.30	0.70	16.458	34.441	54.118

Tabla 10.3: Ventanas en función de  $\epsilon$  y  $\delta$  para distintos valores de  $r_{t_0}$  en el WSA (ALOHA estabilizado de ventana) en su versión binaria.

notar cómo para los restantes  $r_{t_0}$ , el comportamiento del algoritmo sigue siendo destacable. De ahí se concluye que el algoritmo WSA no es muy sensible en su comportamiento al tamaño de ventana, si ésta provoca que aproximadamente haya un número de usuarios en la ventana inverso de la probabilidad inicial.

### 10.3 Función de probabilidades de acceso

Volviendo al caso original que analizábamos con  $\lambda\Delta \approx 1$ , donde hemos razonado el porqué era óptimo la probabilidad inicial unidad. Si suponemos que el feedback recibido es de colisión  $x_t = C$ , qué se puede deducir de forma

$\epsilon$	$\delta$	Alg. WSA binario		
		$r_{t_0} = 1$	$r_{t_0} = 2$	$r_{t_0} = 3$
0.00	0.00	0.433	0.420	0.412
0.00	0.01	0.429	0.416	0.408
0.00	0.10	0.389	0.377	0.370
0.00	0.40	0.250	0.239	0.234
0.01	0.00	0.432	0.419	0.411
0.10	0.00	0.414	0.403	0.397
0.20	0.00	0.375	0.366	0.362
0.30	0.00	0.259	0.239	0.228
0.10	0.10	0.365	0.356	0.350
0.10	0.20	0.317	0.307	0.302
0.10	0.30	0.268	0.258	0.253
0.20	0.10	0.319	0.311	0.307
0.20	0.20	0.265	0.255	0.250
0.20	0.30	0.212	0.201	0.195
0.30	0.10	0.173	0.000	0.000
0.30	0.20	0.085	0.000	0.000

Tabla 10.4: Throughputs en función de  $\epsilon$  y  $\delta$  para distintos valores de  $r_{t_0}$  en el WSA (ALOHA estabilizado de ventana) en su versión binaria. Los tamaños de ventana fijos para cada par  $(\epsilon, \delta)$

lógica? En el caso libre de error, que consideramos ahora por simplicidad, lo que es obvio es que tenemos como mínimo dos paquetes intentando transmitir. Como hemos visto con anterioridad, debido a que el proceso de llegada es de Poisson y  $\lambda\Delta \approx 1$ , hay una probabilidad relativamente alta que haya 0 o 1 paquetes intentando acceder, pero éste no es el caso porque sabemos que hay 2 o más.

Es totalmente intuitivo que podamos suponer que hay aproximadamente 2 usuarios intentando acceder, porque es más probable que 3, 4 o más. Pero hemos de considerar que la contienda va más allá de  $i = 2$ , tarde o temprano volveremos a pasar por este estado con un número de usuarios menor o igual que a la ida, supuestamente 2. De ahí que estando en este estado deberíamos suponer que hay ligeramente menos de 2 usuarios en contienda. Por consiguiente la probabilidad óptima deberá ser ligeramente mayor que  $1/2$ .

En el caso que  $i = 2$  y volviéramos a recibir un feedback de colisión  $x_i = C$ , el CRA original propone decrementar la probabilidad hasta  $1/3$ , porque entonces "prevé" que hay 3 usuarios en contienda.



En este punto es donde se debe hacer hincapié en una cuestión, es óptimo en este caso "suponer" que hay 3 usuarios y "necesitan" ver su probabilidad de acceso decrementada? No es posible que haya 2 que debido a su mala suerte hayan vuelto a colisionar? El hecho que la probabilidad que haya 2 usuarios al inicio del CRI es mayor que 3 nos lleva a la conclusión que esta última posibilidad no es ni mucho menos despreciable. Pero sin lugar a dudas lo que no se puede hacer es dejar de decrementar la probabilidad por si hubiera el caso, con probabilidad no nula, que hubiera 3 o más usuarios al inicio del CRI.

Hemos razonado, pues, desde un punto de vista intuitivo, que la probabilidad de acceso  $p(i) = 1/i$  no es óptima para  $i \geq 2$ . En primera instancia podemos suponer que  $1/2 \geq p(3) \geq 1/3$ . Surge, por tanto, la necesidad de analizar las prestaciones del algoritmo maximizando el throughput respecto a la probabilidad de acceso.

Partiendo de las reflexiones planteadas anteriormente, hemos llegado a configurar el comportamiento de la función de probabilidad de acceso  $p(i)$ , tal que las prestaciones, en este caso el throughput, se mejoren.

En ella se ha obtenido cualitativamente que las probabilidades óptimas y el número de usuarios en contienda esperado óptimo son, aproximadamente,

$$\begin{aligned} p_{opt} = 1 &\Rightarrow k_{opt} = 1 \\ p_{opt} > \frac{1}{2} &\Rightarrow k_{est} < 2 \\ p_{opt} > \frac{1}{3} &\Rightarrow k_{est} < 3 \\ p_{opt} > \frac{1}{4} &\Rightarrow k_{est} < 4 \end{aligned} \quad (10.1)$$

Una de las funciones que cumplen los requisitos anteriores es de la forma,

$$p(i) = \frac{1}{i^x} \quad 0 \leq x \leq 1 \quad (10.2)$$

Hemos maximizado numéricamente el throughput del algoritmo con respecto a  $x$ , y el resultado obtenido es  $x = 0.91$ , es decir, substituyendo en la expresión (10.2),

$$\begin{aligned} p(1) &= 1.000 \\ p(2) &= 0.532 \\ p(3) &= 0.368 \\ p(4) &= 0.283 \end{aligned} \quad (10.3)$$

En la tabla 10.5 se exponen los resultados obtenidos para el algoritmo binario con una función probabilidad de acceso mejor. La mejora no es realmente significativa, tan sólo es de  $1.2 \cdot 10^{-3}$ .

Vamos a describir a continuación el algoritmo que hemos considerado.

$p(i) = 1/i$	$\lambda^* = 0.4335$	$\Delta^* = 2.494$
$p(i) = 1/i^{0.91}$	$\lambda^* = 0.4347$	$\Delta^* = 2.489$

Tabla 10.5: Throughputs y tamaños de ventana para dos funciones de probabilidad de acceso distintas en el WSA (ALOHA estabilizado de ventana) en su versión binaria.

Durante los CRI que empiezan con una ranura de colisión, cada usuario actúa independientemente vía la utilización de un contador cuyo valor en el tiempo  $t$  se denota por  $r_t$ . Este contador tiene el mismo valor para todos los usuarios, pero es cada usuario quien actualiza su propio contador dependiendo del feedback. Cuando un usuario transmite por primera vez, fija  $r_t = r_{t_0} = 1$ . Los valores del contador son enteros, y son actualizados y utilizados de acuerdo con las reglas siguientes.

1. Los usuarios transmiten en la ranura  $t$  con probabilidad  $1/r_t^{0.91}$ .
2. Los valores del contador cambian con el tiempo de la manera siguiente:
  - (a) Si  $x_{t-1} = NC$ , entonces  $r_t = r_{t-1} - 1$
  - (b) Si  $x_{t-1} = C$ , entonces  $r_t = r_{t-1} + 1$

De estas reglas se puede ver que un CRI acaba con una ranura de no colisión y  $r_t = 1$ .

## 10.4 Protocolos con sintonización limitada

Los protocolos que hemos descrito a lo largo del presente trabajo requieren que cada usuario monitoree el feedback del canal en todo momento, incluso si ese usuario no tiene ningún paquete para transmitir. Esto es necesario porque los paquetes recién generados se pueden transmitir por primera vez sólo al final del presente CRI, requiriendo a su vez que cada usuario determine el final del CRI. Este tipo de monitorización del feedback se conoce como sintonización continua (*full-sensing*). Este modo de operación es considerado poco práctico porque un usuario que deje de sintonizar el feedback no puede reengancharse y, por lo tanto, un usuario que debido a algún fallo no recibiera adecuadamente todas las señales puede de hecho perturbar a los otros y decrementar la eficiencia del protocolo.

Así, sería deseable que los usuarios monitorizaran la señales del feedback solamente durante periodos limitados, preferiblemente entre que genera un paquete y se transmite exitosamente. Este tipo de monitorización se

conoce como sintonización limitada (*limited-sensing*), y los protocolos con tal monitorización son referidos como protocolos que operan en un entorno de sintonización limitada.

Varios CRA han sido creados para un entorno de sintonización limitada. El más sencillo, aunque no el más eficiente, es el protocolo de libre acceso (*free-access*) analizado por P. Mathys y P. Flajolet [37] y por G. Fayolle, P. Flajolet, M. Hofri y P. Jacquet [18]. En este protocolo, los paquetes nuevos se transmiten tan pronto como sea posible, es decir, en la ranura siguiente a su tiempo de llegada; por lo tanto, un usuario que transmitió un nuevo paquete monitoriza las señales del feedback y continúa operando como si fuera un usuario "antiguo". El máximo throughput alcanzado por este protocolo de sintonización limitada lo consigue en conjunción con el protocolo de árbol binario modificado, y es de 0.360 paquetes/ranura.

Hasta la fecha, el protocolo más eficiente para un entorno de sintonización limitada es el que introdujeron P. Humblet [86], y L. Georgiadis y P. Papantoni-Kazakos [87], el cual es básicamente una adaptación del protocolo de árbol binario modificado truncado (sintonización continua). Como ya se mencionó en su momento, una característica clave para la correcta operación de este algoritmo es la posibilidad de todos los usuarios de determinar el final de un CRI. Esto es necesario para que los usuarios con paquetes nuevos sepan exactamente cuando pueden transmitir por primera vez, de tal forma que no interfiera la resolución de la colisión en curso. El mayor cambio realizado en la adaptación del protocolo de árbol binario truncado a la sintonización limitada es por lo tanto el controlar la duración del CRI para que su fin sea fácil y únicamente detectable. Como antes, los paquetes recién generados no son considerados para la transmisión hasta que se haga la resolución de la colisión en curso (y todos los usuarios reciban suficiente feedback para detectar el final del CRI). Un feedback de colisión indica obviamente que un CRI está en curso; la dificultad está en saber si un CRI ha acabado cuando se recibe una serie de feedbacks de no colisión.

Volviendo al protocolo de árbol binario truncado descrito en el Capítulo 2, cabe remarcar que el fin de un CRI que empezó con una colisión se caracteriza por dos ranuras exitosas consecutivas. Además, hay CRIs que consisten en una simple ranura, sea vacía o transmisión exitosa. Así, si tenemos una ranura exitosa seguida de otra ranura de no colisión, nos aseguramos que el CRI ha acabado entonces. Tomamos, pues, este evento como un marcador de fin de CRI. Sin embargo, este marcador no es suficiente. Si observamos el canal con una ranura vacía y sin ningún paquete para ser transmitido, se sucederán una serie de ranuras vacías que un usuario recién llegado no detectará como fin de CRI, y el sistema quedará permanentemente bloqueado. Para evitar este bloqueo potencial, también se considerará como fin de CRI

cuando el usuario que monitorice el canal detecte un número prefijado de ranuras vacías ( $R + 1$ ), que será un parámetro del protocolo. De esta manera, un usuario que observe  $R + 1$  ranuras vacías consecutivamente considera que el CRI ha acabado. Esta indicación de fin de CRI es correcta sólo si  $R + 1$  ranuras vacías consecutivas no se pueden producir en el transcurso de un CRI. Sin embargo, en el protocolo de árbol binario truncado modificado, tal evento es posible ya que el protocolo evita algunas colisiones seguras. En consecuencia, para poder aplicar lo visto hasta ahora a este protocolo, si los usuarios ven  $R$  ranuras consecutivas vacías, en la siguiente ranura accederán todos.

A partir de lo anteriormente expuesto, la pregunta obligada es existe una versión del algoritmo ALOHA estabilizado de ventana con sintonización limitada? Si es así, qué prestaciones tiene?

Después de una profunda reflexión, podemos asegurar que la respuesta no es inmediata. Y la razón es clara. La característica necesaria para que un algoritmo se pueda convertir en otro de sintonización limitada es, como se ha explicado anteriormente, que los usuarios puedan detectar el final de un CRI sin tener que monitorizarlo desde el principio de éste. Así, los paquetes que llegan en un CRI esperan a la finalización de éste para intentar la primera transmisión, si se les permite.

Nuestro algoritmo tal y como está concebido ahora no tiene esta peculiaridad, o casi no la tiene. Imaginemos que implementamos una versión de nuestro algoritmo con sintonización limitada tal que si un usuario despierta en un CRI que está en marcha, y escucha dos feedbacks consecutivos de no colisión, considera que el CRI ha finalizado.

Lo que está claro es que todo CRI acaba con dos feedbacks de no colisión consecutivos, pero ello no significa que dos feedbacks de no colisión consecutivos impliquen la finalización del CRI. Lo que es claro es que dos feedbacks de no colisión consecutivos no supondrá una finalización real del CRI con una probabilidad relativamente baja. Y en el caso de ser así, esto hace suponer que  $k \leq i$ , por lo que, en este punto, acceder un usuario más no sobrecargará el sistema.

El estudio analítico de lo anteriormente explicado es un problema abierto, por el momento.



# Capítulo 11

## Aplicación: Integración de voz y datos en redes de acceso inalámbricas

### 11.1 Introducción

En este Capítulo<sup>1</sup> presentamos una simulación sobre la capacidad de redes inalámbricas para un esquema combinado de integración de voz y datos. Como servicio portador tenemos una estructura de multiplexación por división en el tiempo y duplexación en frecuencia, es decir, TDMA/FDD. Simultáneamente, se ha tenido en cuenta un número fijo de fuentes de voz combinado con un número finito o infinito de fuentes de datos. Cada fuente de voz es modelada por un proceso de Markov de dos estados, mientras que cada fuente de datos individual genera una corriente de Poisson, con un paquete de datos por mensaje.

Los terminales de voz operan con detectores de conversación (*speech detectors*), es decir, a nivel de talkspurt. Durante el tiempo de vida de un talkspurt, los paquetes de voz son generados periódicamente, uno por trama TDMA. El protocolo de acceso múltiple implementado para las entidades talkspurt está basado en un esquema de acceso aleatorio más reserva. Cada vez que un terminal de voz genera un talkspurt, se necesita una ranura para llevar el paquete de voz. Para reservar esta ranura, utilizamos el algoritmo de acceso aleatorio WSA, explicado en el Capítulo 3. Debido a la sensibilidad de las fuentes de voz al retardo, opera un mecanismo de pérdida de paquetes cuando el acceso se retarda más allá de un cierto umbral temporal.

---

<sup>1</sup>Este Capítulo es deudor del trabajo realizado y posteriormente publicado por V. Casares y el autor [58].

Para paquetes de datos, no es necesario un esquema de reserva, ya que se asume la transmisión de un paquete de datos por mensaje. Debido a esto, se usa el protocolo ALOHA con un procedimiento estabilizado. Además, no se implementa ningún mecanismo de pérdidas para paquetes de datos, ya que no hay restricción de retardo para ellos.

En el esquema de integración propuesto hay prioridad de voz sobre el tráfico de datos. Esto se lleva a cabo por una asignación dinámica de las ranuras libres de una trama dada a terminales de voz y datos. Así, en el mecanismo de prioridad, el tráfico de datos no influye en el desarrollo del tráfico de voz.

En la literatura han sido propuestos muchos esquemas de integración de voz y datos. Su tratamiento tradicional consiste en considerar que la voz opera en conmutación de circuitos y los datos en conmutación de paquetes. En ese caso, la probabilidad de bloqueo de las llamadas de voz y el tiempo medio de retardo de los paquetes de datos son los parámetros de interés. Adicionalmente para la clase de fuentes de información, voz y datos en nuestro caso, los esquemas propuestos difieren en la forma en la que se distribuyen los canales de comunicación. En el trabajo de U. Bhat y M. Fischer [88] y del mismo M. Fischer [89], se analiza un modelo continuo temporal para integración de tráfico de conmutación de circuitos y paquetes, con una total compartición de canales, para ambos, con una política de prioridad con y sin expulsión. En el trabajo de K. Kummerle [90] y P. Zafiropoulo [91], se introduce un sistema parcialmente distribuido en un TDM digital, llamado estrategia de límites móviles (*movable-boundary strategy*). Consiste en dividir el número total de ranuras  $N_s$  por trama en  $N_1$  ranuras para voz, y  $N_2$  ranuras para datos. Sin embargo, los terminales de datos tienen permiso para usar las ranuras de voz cuando todos las ranuras de datos están ocupadas y algunas ranuras para voz están libres. Con una estructura TDM, todos aquellos esquemas pueden ser vistos como un caso particular del caso más general representado en la figura 11.1. De aquí, por ejemplo, obtenemos el esquema de U. Bhat y M. Fischer [88] y del mismo M. Fischer [89] cuando  $N_1 = N_2 = N_s$ , y el esquema de K. Kummerle [90] y P. Zafiropoulo [91] cuando  $N_1$  es fijo y  $N_2$  varía dinámicamente entre  $N_s$  y  $N_s - N_1$ .

Se ha llevado a cabo un análisis continuo temporal del esquema de límites móviles en el libro de M. Schwartz [92] para  $N_1 = N_2 = 1$ . Para el caso general, se han desarrollado en la literatura algunos métodos aproximados, principalmente basados en una aproximación al flujo de fluidos, como los artículos de D. Gaver y P. Lehoczky [93] y A. L. García, R. Kwong y G. Williams [94].

Sin embargo, cuando intentamos trasladar estos esquemas a redes inalámbricas, aparece el problema de los protocolos de acceso múltiple. Esto

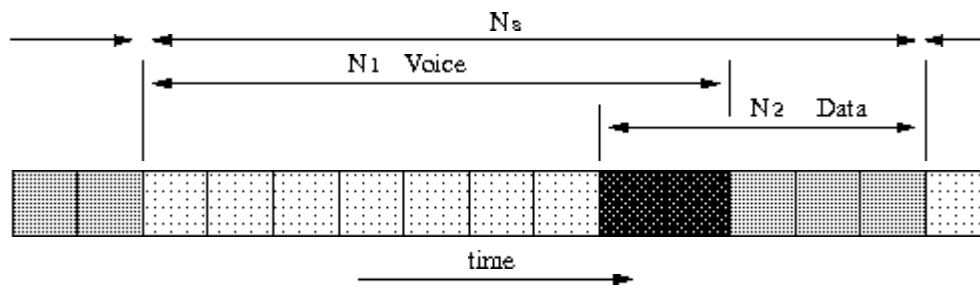


Figura 11.1: Estrategia general de límites móviles.

es, debido a la carencia de coordinación entre terminales se debe implementar un protocolo de acceso aleatorio con el fin de ganar el medio y obtener, si es necesario, una reserva de un canal físico que sea mantenido durante el tiempo de vida de la entidad de voz o datos. Para cubrir estos puntos en entornos inalámbricos, se han propuesto estos últimos años esquemas de integración de voz y datos. De aquí, por ejemplo, en los trabajos de S. Nanda [95] y de H. Chung y D. Goodman [96], voz y datos están integrados, de acuerdo con un esquema de compartición total; de acuerdo con la figura 11.1 cuando  $N_1$  y  $N_2$  están fijados a  $N_s$ . Con este propósito, el protocolo PRMA presentado por D. Goodman, R. Valenzuela, K. Gayliard y B. Ramamurthi [97] es usado para terminales de voz y datos, donde los terminales de voz operan con detectores de conversación activo/silencio, y los terminales de datos generan un paquete por mensaje.

Recientemente, un esquema de integración voz más datos ha sido propuesto por F. Babich [33]. Esta solución es también un caso particular del caso general representado en la figura 11.1. F. Babich [33] sugiere una estrategia de límites móviles, es decir,  $N_1$  y  $N_2$  varía dinámicamente de acuerdo con las condiciones de tráfico. Los terminales de voz usan tantas ranuras temporales como les haga falta, y el resto de ranuras son ofrecidas como disponibles a los terminales de datos. En el trabajo de F. Babich [33], los métodos de acceso propuestos para las entidades de voz (talkspurts) y datos (paquetes) están basados en un algoritmo de resolución de colisiones (*collision resolution algorithm*, CRA), introducido por M. Paterakis y P. Papantoni-Kazakos [27], basado en una pila de dos células (algoritmo TCW).

En este trabajo proponemos una integración voz más datos que está en línea con lo propuesto en el de F. Babich [33]. Sin embargo, usamos diferente CRAs. Para talkspurt, utilizamos un algoritmo de acceso aleatorio (RAA), el protocolo ALOHA estabilizado de ventana (WSA). El CRA usado para paquetes de datos es el protocolo slotted ALOHA con una ligera modificación



del procedimiento estabilizado de R. Rivest [42]. Comparamos nuestro esquema de integración con una técnica llamada de asignación fija, en los cuales las ranuras están dedicadas o distribuidas entre terminales de voz o datos. Esto será detallado en la próxima sección.

## 11.2 Escenario

**Modelo de fuente.** El entorno que consideramos consiste en un número combinado de terminales de voz y datos. Cada terminal de voz opera con un detector de conversación activo/silencioso, así que cada uno puede ser modelado por un modelo de talkspurts y huecos. Para modelar este efecto, se ha escogido un proceso de Markov de dos estados on/off, ya utilizado por P. Reilly y A. Ghani [98], y D. Goodman y S. Wei [99], con una duración exponencial para los talkspurts y huecos de media  $1/\mu = 1$  y  $1/\lambda = 1.35$  segundos, respectivamente. No se ha considerado un modelo de tres estados debido al hecho que no se obtiene una ganancia de multiplexado significativa, para una discusión adicional el lector puede consultar el trabajo de D. Goodman, R. Valenzuela, K. Gayliard y B. Ramamurthi en [97]. Las fuentes de datos generan una corriente de tráfico de Poisson, con un paquete de datos por mensaje de longitud constante. Los paquetes generados por cada terminal son servidos en una cola con disciplina FIFO, así que sólo paquetes de datos pertenecientes a diferentes terminales acceden mediante contienda para ganar el medio de transmisión. La tasa poissoniana global de paquetes de datos se mantiene constante, mientras se varía el número de terminales, de forma que cuando el número de terminales crece, la tasa por terminal decrece, y viceversa.

**Estructura del servicio portador.** La estructura del servicio portador considerada es TDMA/FDD (Time Division Multiple Access y Frequency Division Duplex) como servicio portador. Los parámetros usados en nuestro trabajo han sido tomados de H. Chung y D. Goodman [96]. La duración de la trama es de 16 ms y el número de ranuras por trama es  $N_s = 20$ .

**Esquema de asignación de recursos.** Las ranuras como recursos pueden ser asignadas de una forma estática y dinámica. Para comparar, se han considerado dos asignaciones. La primera, que llamamos ubicación dinámica (*dynamic allocation*), donde las ranuras están asignadas inicialmente a terminales de voz, y el resto de las ranuras libres están asignadas a terminales de datos, ver el trabajo de F. Babich [33]. Este esquema se compara con una asignación fija (*fixed assignment*), donde el conjunto de  $N_s$  ranuras por

trama es partido en dos grupos, y separadamente dedicadas para terminales de voz  $N_1$ , y para terminales de datos  $N_2$ , ( $N_s = N_1 + N_2$ ).

Para entidades talkspurt, se usa un protocolo de acceso múltiple consistente en acceso aleatorio más reserva. Para la ubicación dinámica, se utiliza la versión de algoritmo de reserva del ALOHA estabilizado de ventana (WSA) explicado en el Capítulo 3, que se presenta en la próxima sección. Cuando se considera una asignación fija, se implementa el PRMA, según D. Goodman, R. Valenzuela, K. Gayliard y B. Ramamurthi [97]. Para ambos casos, y debido a la sensibilidad en el retardo de las fuentes de voz, opera un mecanismo de pérdida de paquetes de voz cuando el retardo en el acceso va más allá que un cierto umbral temporal.

Para entidades paquete de datos no se necesita esquema de reserva ya que asumimos un paquete de datos por mensaje. Debido a este hecho, se usa para el esquema de ubicación dinámica el protocolo slotted ALOHA con un procedimiento estabilizado, y cuando se considera la asignación fija, se elige el slotted ALOHA sin estabilización. Esta consideración es debida a que la estabilización del ALOHA implica una coordinación entre terminales que la asignación fija no contempla y que lo hace más robusto frente a errores del feedback. Además, no se implementa ningún mecanismo de pérdida para paquetes de datos, ya que no existe restricción en el retardo para ellos.

Implícitamente, el esquema de ubicación dinámica proporciona prioridad de voz sobre datos, partiendo las ranuras libres de una trama dada en ranuras para terminales de voz, y ranuras para terminales de datos. La asignación de ranura es dictada por el algoritmo de acceso aleatorio (RAA) usado para la reserva del talkspurt. En este proceso de reserva, normalmente, los terminales de voz no necesitan todas las ranuras libres de una trama dada. En cada trama, las ranuras libres que no han sido utilizadas por los terminales de voz, pueden ser fácilmente identificadas y, por ello, pueden ser utilizadas por los terminales de datos. Como veremos, este proceso de identificación está implícitamente conseguido de una forma muy sencilla por el RAA. Así, en el mecanismo de prioridad, la carga del tráfico de datos no influye en voz.

### 11.3 Algoritmo de acceso aleatorio de reserva: WSA

En esta sección introducimos un algoritmo de acceso aleatorio simple (RAA) que es usado por las entidades talkspurt para conseguir y reservar el medio, el protocolo ALOHA estabilizado de ventana (WSA). Con ese fin, primeramente consideramos usuarios que envían paquetes de reserva sobre un canal

ranurado, con un canal con feedback binario de colisión o no colisión después de cada ranura, retardo de propagación nulo y sin error en el feedback. Un controlador de la estación base (BSC) monitoriza el canal ranurado. No se consideran ni efectos de captura ni desvanecimientos como en trabajo de C. Namislo [100], es decir, hay una transmisión exitosa cuando un paquete es enviado a través de una ranura, y una colisión ocurre cuando dos o más paquetes coinciden en la misma ranura, en este caso el paquete es totalmente destruido y se requiere una retransmisión.

Para la presentación usaremos la misma notación que en el Capítulo 3. Las unidades de tiempo son ranuras, así que la ranura  $t$  ocupa el intervalo de tiempo  $[t, t + 1)$ . Denotaremos por  $x_t$  el feedback que corresponde a la ranura  $t$ ;  $x_t = C$  y  $x_t = NC$  representa colisión y no colisión en la ranura  $t$  respectivamente. El algoritmo opera partiendo el eje temporal en ventanas separadas, todas de longitud  $\Delta$ . En nuestra aplicación, escogemos  $\Delta$  un número entero de ranuras. Todos los usuarios que lleguen durante la ventana temporal  $\Delta_i$  son transmitidos durante un conjunto de ranuras disponibles que conforman el intervalo de resolución de colisiones (*collision resolution interval*, CRI). Como veremos, el fin de un CRI es fácilmente identificado, de tal forma que, una vez un CRI finaliza, el próximo CRI puede empezar cuando la correspondiente ventana temporal  $\Delta$  haya pasado.

El algoritmo procede como sigue. Cada usuario tiene una variable entera  $r_t$  que es actualizada al final de cada ranura de acuerdo con el feedback  $x_t$  transmitido por el BSC. Todos los usuarios que compiten en el mismo CRI, tendrán el mismo valor para  $r_t$ . Después de una actualización, todos los usuarios transmitirán con probabilidad  $1/r_t$ . Cuando el BSC señala el inicio de un CRI todos los usuarios inicializan con  $r_t = 1$ . De acuerdo con el feedback recibido  $x_t$ , el proceso de actualización sigue los siguientes pasos:

Si  $x_{t-1} = NC$ , entonces  $r_t = r_{t-1} - 1$ .

Si  $x_{t-1} = C$ , entonces  $r_t = r_{t-1} + 1$ .

De estas reglas, se puede ver que un CRI dura una ranura cuando la ventana temporal examinada contiene a lo sumo un paquete. De otra forma, el CRI empieza con una colisión. El fin de un CRI es detectado por el BSC cuando una no colisión (NC) ocurre y  $r_t = 1$ . Por ello el BSC siempre guarda una copia de  $r_t$ . Comparando con el algoritmo de M. Paterakis y P. Papantoni-Kazakos [27], hay que resaltar que dos o más de dos NC consecutivos no implica un fin de CRI.

## 11.4 Reserva de slots para talkspurts

En esta sección trataremos el procedimiento de reserva de ranura para talkspurts. En la primera parte explicaremos como el RAA introducido en la sección anterior se aplica al esquema de ubicación dinámica. Después, introduciremos la asignación fija, para ser comparada con la anterior.

Cuando se genera un talkspurt, éste intentará reservar una ranura libre, probablemente en competición con otros talkspurts, mediante el RAA de la Sección anterior. Una vez el terminal obtiene una reserva, la ranura reservada se mantiene hasta el fin del talkspurt. Después de ese evento, la ranura está de nuevo disponible para otros usuarios.

En el esquema de ubicación dinámica, usamos el RAA previamente introducido. Escogemos una ventana temporal  $\Delta$  de longitud igual a la duración de trama. Así, se dedica un CRI para cada trama. Todo talkspurt que se generó en la misma trama no intenta el acceso hasta que el correspondiente CRI sea permitido por el BSC. El inicio de un CRI coincide siempre con el inicio de una trama. Una vez que el CRI haya empezado, todos los talkspurts de la ventana temporal permitida compiten entre ellos para obtener una ranura libre. Esta competición sigue las reglas del mencionado RAA. Como consecuencia, la duración de un CRI puede ocasionalmente consumir más de 16 ms. En aquellos casos, opera el mecanismo de pérdida de paquetes, y  $k$  paquetes de voz de un talkspurt son descartados si el acceso del terminal se produce en la trama  $(k + 1)$ ésima del CRI. Esto es, el retardo máximo de acceso permitido por los talkspurts antes de empezar a descartar paquetes es dos veces la duración de trama. La figura 11.2(a) representa un ejemplo consistente donde se observa un intervalo temporal de varias tramas, con 10 ranuras por trama. Observamos que los terminales 1 y 3 generan cada uno un talkspurt en la trama  $i$  y obtienen un acceso exitoso en la trama  $i + 2$ , así que se descarta un paquete de voz por cada uno. También observamos que los terminales 6 y 10 ganan un par de ranuras libres en la trama  $i + 2$ , pero el correspondiente  $CRI_{i+1}$  dura un slot adicional (H en la trama  $i + 3$ ) ya que cuando el terminal 3 accede exitosamente el parámetro de sistema  $r_i = 2$ . Como detalle adicional, se ha escogido la opción que el último paquete del talkspurt está marcado como tal; la otra opción hubiera sido que un slot se libera cuando no se transmite nada por él, criterio utilizado por D. Goodman y S. Wei en [99] y por D. Goodman, R. Valenzuela, K. Gayliard y B. Ramamurthi [97].

Cuando se usa el esquema de asignación fija, operamos con el PRMA, para una descripción de éste, ver los trabajos de S. Nanda [95], de H. Chung y D. Goodman [96], de D. Goodman y S. Wei [99] y de D. Goodman, R. Valenzuela, K. Gayliard y B. Ramamurthi [97]. La probabilidad de permiso es

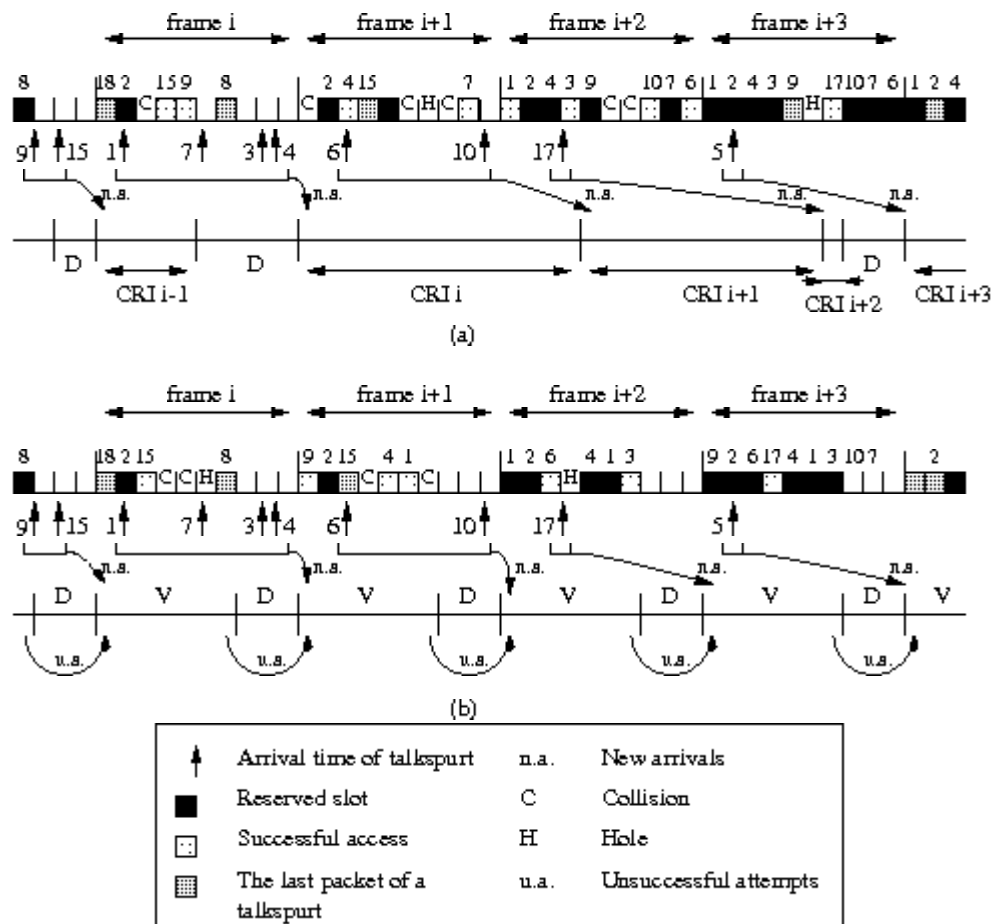


Figura 11.2: Proceso de asignación de ranuras. (a) Distribución dinámica. (b) Asignación fija.

$p = 0.3$ , y los paquetes de voz son descartados de acuerdo con el esquema previo. En ese caso, figura 11.2(b), observamos por ejemplo que el terminal 9 descarta un paquete de voz, lo mismo que el terminal 3. Sin embargo, el terminal 7 descarta por lo menos 3 paquetes de voz. Notamos que a nivel de trama, y relacionado con el procedimiento de reserva de ranuras, no se garantiza una disciplina FIFO para los talkspurts.

### 11.5 Acceso de paquetes de datos

Para los dos esquemas de integración de voz más datos propuestos, ubicación dinámica y asignación fija, destacamos del ejemplo de la figura 11.2, que

algunas ranuras están disponibles para paquetes de datos. Asumimos que  $N_d$  fuentes de datos generan de acuerdo con una ley de Poisson una tasa global de  $\lambda_d$ .

En la ubicación dinámica se usa una ligera modificación del algoritmo de R. Rivest [42], que sólo necesita feedback binario, ya que sólo necesita distinguir colisiones de no colisiones. Siguiendo con la misma notación que en la sección previa, se opera de acuerdo con las siguientes reglas:

1. Cada terminal mantiene una copia de  $r_t$  y, durante cada ranura:
  - (a) Transmite con probabilidad  $1/r_t$  si tiene un paquete.
  - (b) Si  $x_t = NC$  entonces  $r_{t+1} = r_t - 1$ .
  - (c) Si  $x_t = C$  entonces  $r_{t+1} = r_t + (\epsilon - 2) - 1 = r_t + 1.39221\dots$
2. Si  $\Delta_t$  es el tiempo transcurrido entre el inicio del último slot de datos y el inicio del próximo slot de datos, entonces actualiza  $r_{t+1}$  a  $\min[Nd, \max(r_t + 1 + \lambda_d, 1)]$

Si se adopta el esquema de asignación fija, se escoge el protocolo s-ALOHA no estabilizado. En ese caso, el throughput debería ser limitado a alrededor del 7 por ciento, según D. Chakraborty [101].

## 11.6 Resultados de simulación

En esta sección presentamos los esquemas de simulación previamente discutidos. La trama tiene una duración total de 16 ms, y está configurada por un total de  $N_s = 20$  ranuras. Para talkspurts, la probabilidad de pérdida se mantiene por debajo del 1%; y en ese caso la capacidad máxima es aproximadamente 37 terminales de voz. En la figura 11.3 se representa también la probabilidad que un talkspurt sea enteramente enviado, es decir, sin ningún paquete de voz descartado.

En el esquema de ubicación dinámica el retardo para paquetes de datos está representado en la figura 11.4, para los tres casos  $N_v = 10, 25$  y  $35$ . Para cada caso el número de terminales de datos está parametrizado:  $N_d = 1, 2, 8$  y  $32$ . Observamos que cuando  $N_d = 1$ , tenemos una cola  $M/D/1$  sin colisiones. Cuando  $N_d = 2, 8$  y  $32$ , podemos registrar colisiones de multiplicidad  $N_d$  a lo sumo. En aquellos casos, y comparando con  $N_d = 1$ , notamos que el throughput se reduce considerablemente. También observamos que no hay diferencias significativas entre los casos de colisión y  $N_d$  tendiendo a infinito, y estas diferencias son aún menores cuando aumenta el número de terminales de voz ( $N_v = 25$  y  $35$ ).

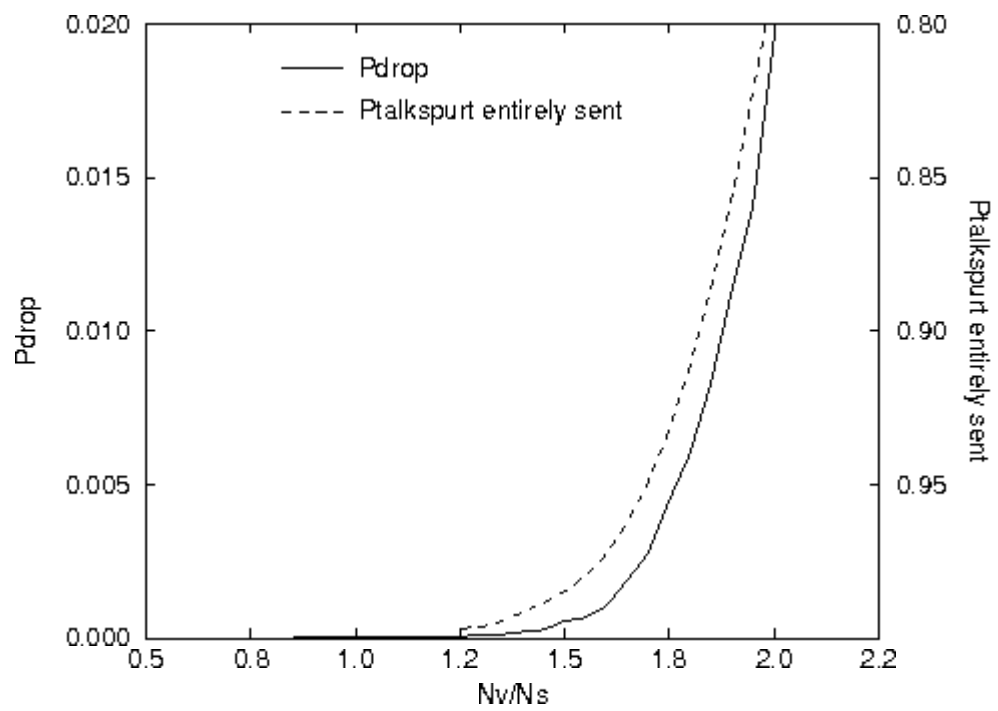
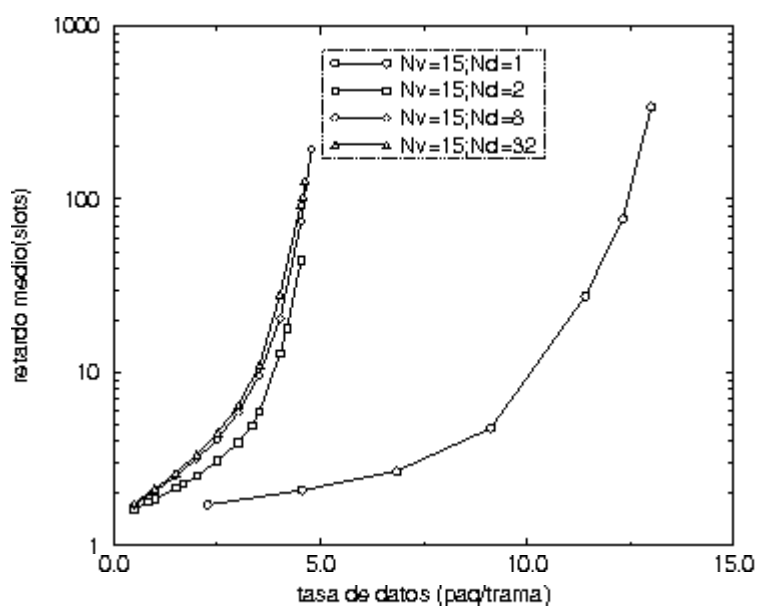
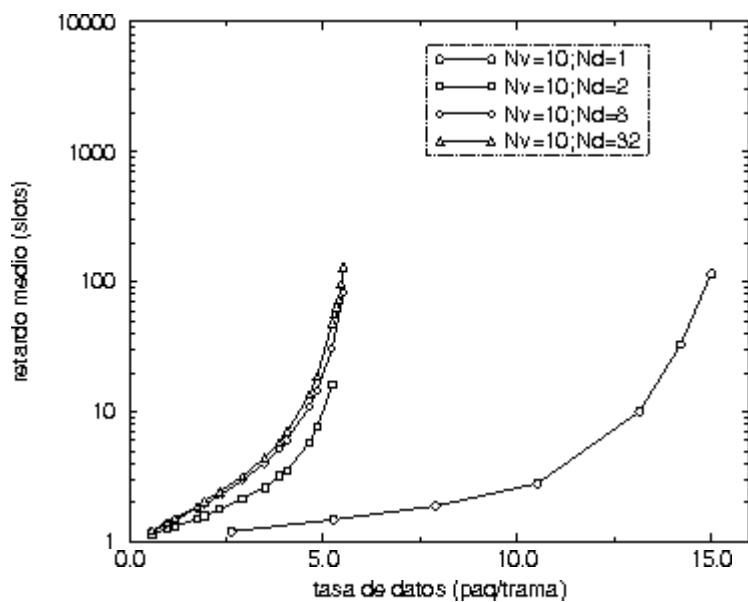
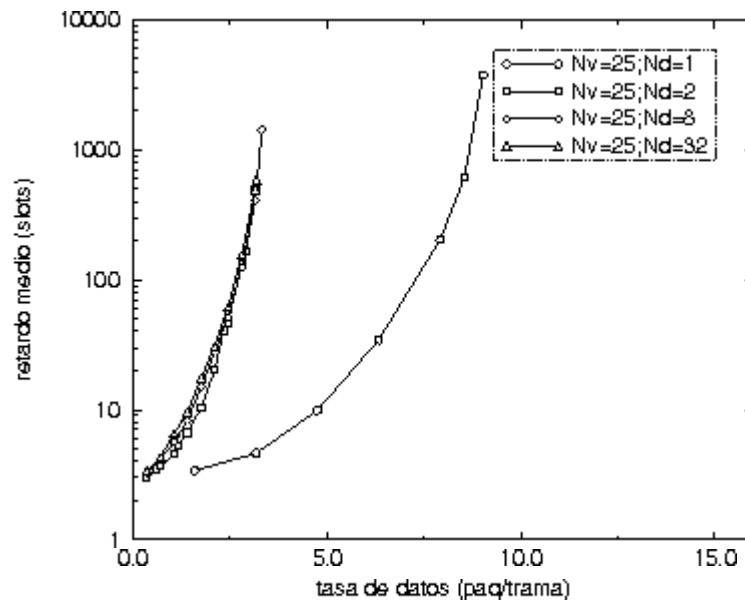
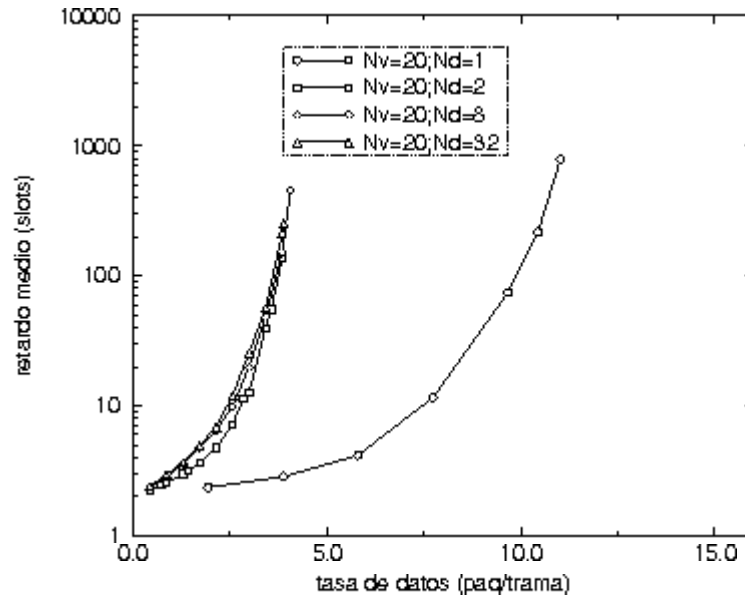
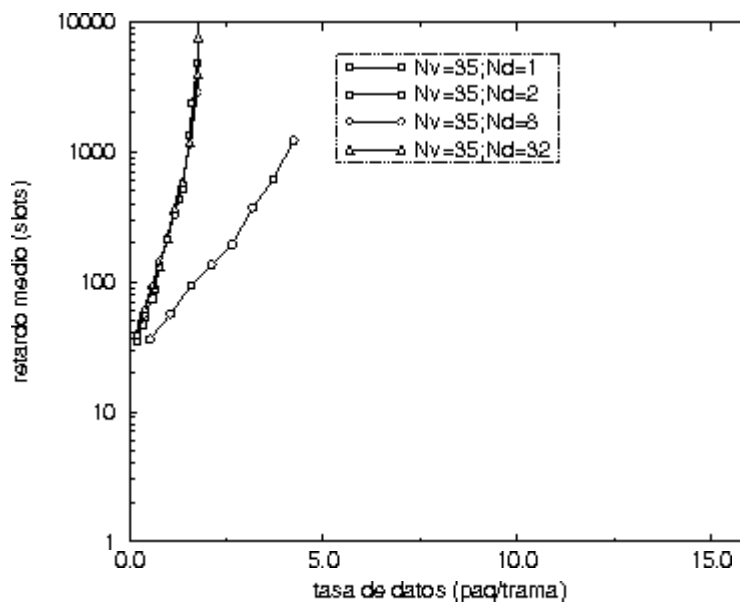
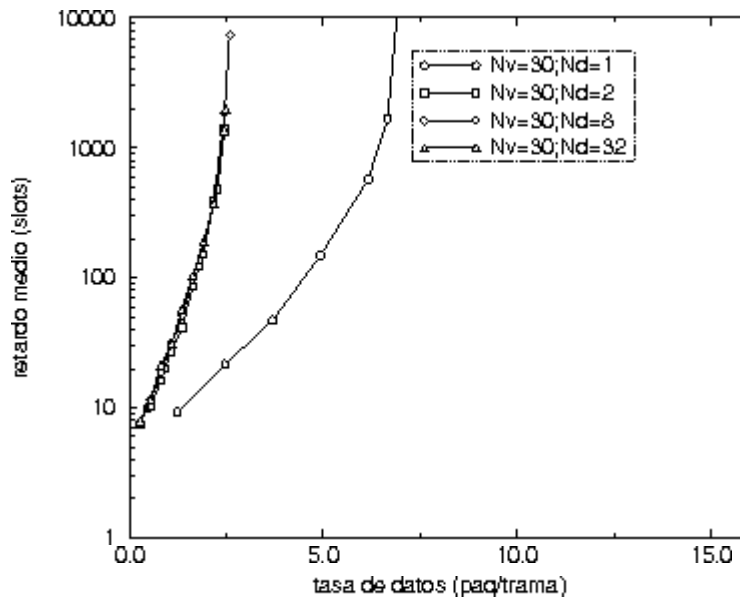


Figura 11.3: Probabilidad de pérdida de paquetes de voz y probabilidad de que un talkspurt sea enteramente enviado.









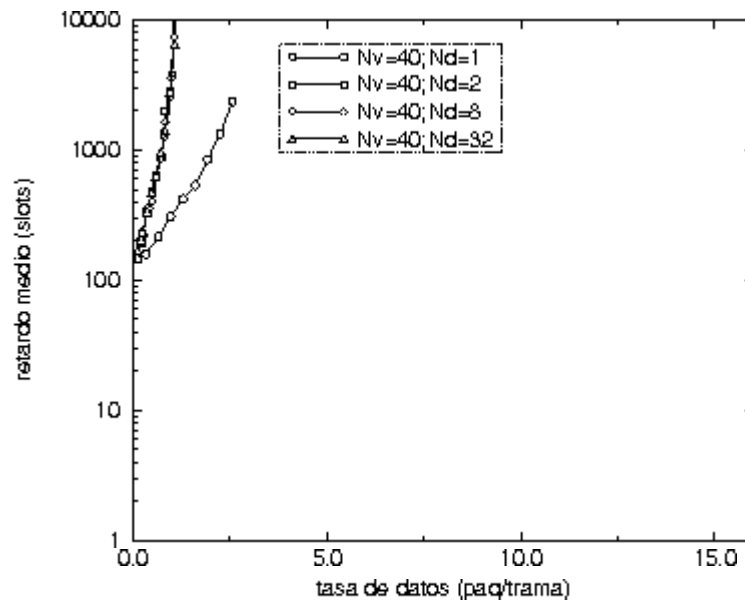


Figura 11.4: Esquema de distribución dinámica. Relación entre retardo medio de los paquetes de datos y tasa de datos  $\lambda_d$ , para  $N_v = 10, 15, 20, 25, 30, 35$  y  $40$ .

En la asignación fija se reportan dos situaciones en paralelo a la ubicación dinámica, figura 11.5. La primera es para  $N_v = 10$  y  $N_d = 32$ , y la segunda para  $N_v = 25$  y  $N_d = 32$ . Teniendo en cuenta la probabilidad máxima de pérdida para paquetes de voz del 1%, por simulaciones extensivas obtenemos que el número mínimo de slots dedicados es  $N_1 = 7$  ( $N_2 = 13$ ) y  $N_1 = 15$  ( $N_2 = 5$ ) respectivamente. Las probabilidades de permiso usadas son iguales a  $p_v = p_d = 0.3$ . Es claro que la capacidad en el esquema de asignación fija es notoriamente reducida comparada con la dinámica. Notar que en el esquema de asignación fija, cuando  $N_v = 35$ , ninguna ranura puede ser dedicada a terminales de datos.

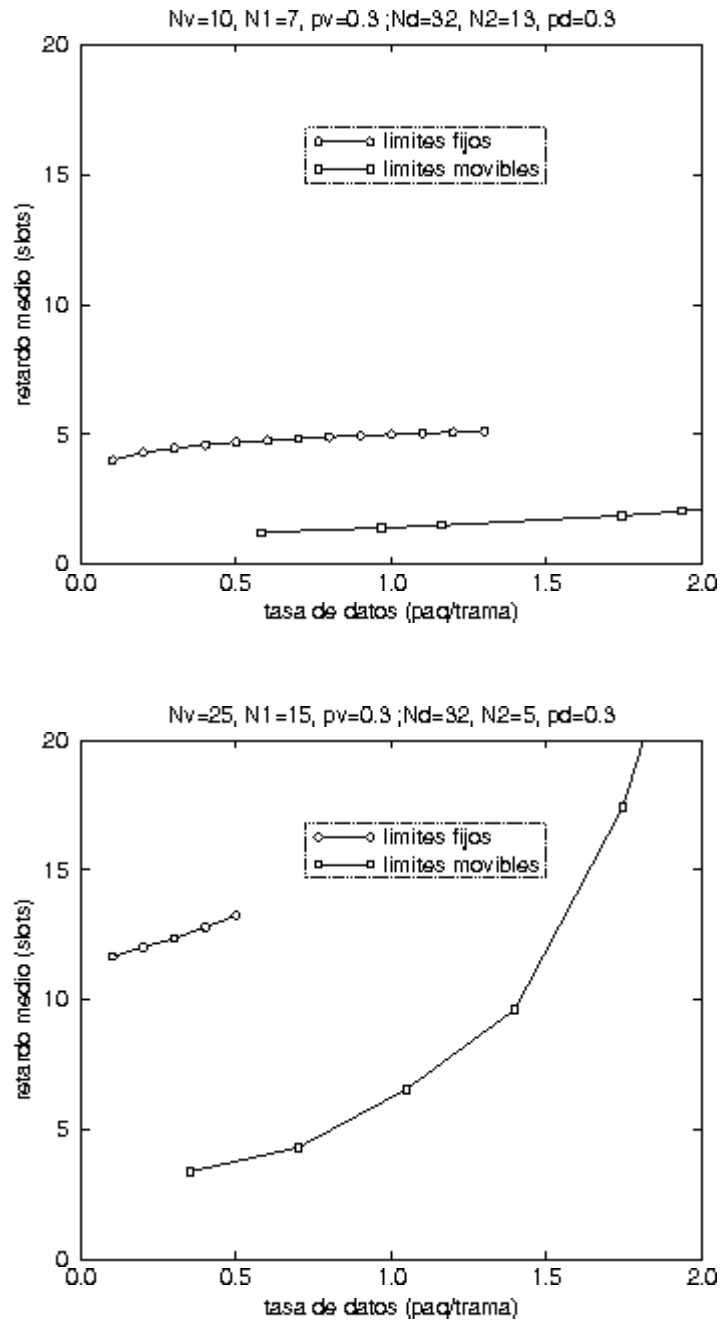


Figura 11.5: Esquema de asignación fija. Relación entre retardo medio de los paquetes de datos y tasa de datos, en términos del número de terminales de voz cuando  $N_d = 32$ .

# Conclusiones

A continuación, vamos a enumerar de forma sucinta las conclusiones a las que hemos ido llegando a lo largo de la presente tesis, y las líneas futuras que esta plantea:

1. Se ha presentado y analizado una nueva familia de algoritmos de acceso aleatorio de tipo ventana. De forma global, se ha analizado uno por uno hasta tres miembros o versiones de esta familia. La progresión que se ha escogido está basada en la cantidad de información que los usuarios reciben a través del feedback.

A través de esta metodología, el resultado final es el conocimiento de las prestaciones que esta familia posee y las que se podrían esperar de las posibles variantes que a partir de aquí podrían surgir. Esta es la conclusión global de la tesis. A partir de ahora las conclusiones estarán más enfocadas a cada resultado obtenido.

2. Desde el punto de vista del throughput, se ha estudiado analíticamente las tres versiones anteriormente citadas: binaria, ternaria e ideal. Las conclusiones a las que se han llegado son las siguientes:

- (a) El algoritmo WSA binario presenta un throughput que se puede considerar elevado (0.4335 paquetes/ranura). Éste es en principio menor que el algoritmo de P. Studer y H. Pletscher [25] (0.4493 paquetes/ranura). El problema que presenta el citado algoritmo es que no funciona para procesos de llegada distintos de Poisson, es decir, que haya una probabilidad no nula que dos paquetes lleguen simultáneamente. Bajo estas circunstancias el algoritmo presentado puede seguir funcionando. La razón es bien sencilla, los algoritmos FCFS se basan en que un paquete realizará una acción u otra para evitar la colisión dependiendo del tiempo de llegada de

éste. Si dos paquetes llegan al unísono, estos seguirán las mismas acciones permanentemente, colisionando continuamente y llevando el algoritmo irremediablemente a la inestabilidad. En nuestro caso, los paquetes varían su probabilidad de acceso dependiendo del feedback, y dos paquetes que lleguen simultáneamente seguirán caminos distintos.

En primera instancia podría parecer que la simultaneidad de las llegadas de los paquetes es una característica teórica, casi filosófica, pero desde un punto de vista pragmático, entendemos como simultaneidad de llegadas el hecho que estos no puedan ser distinguidos por el orden de llegada, debido a que lo hacen dentro del mismo periodo de reloj que sincroniza el sistema.

El algoritmo WSA binario consigue el mayor throughput de los conocidos hasta ahora con esta interesante característica. El algoritmo que hasta la fecha desempeñaba esta función era el propuesto por M. Paterakis y P. Papantoni-Kazakos [27], con un throughput de 0.4295 paquetes/ranura.

- (b) El algoritmo WSA binario posee una alta robustez frente a los errores en el feedback. Esta robustez es también estudiada a lo largo de la tesis y hemos llegado a la conclusión que dentro de los valores que se pueden considerar prácticos, los valores del throughput se mantienen dentro de un margen del 90%. Se ha encontrado también que la sensibilidad frente a los errores en el feedback en los márgenes anteriormente citados son similares a los del algoritmo de M. Paterakis y P. Papantoni-Kazakos [27].
- (c) El algoritmo WSA ternario tiene un throughput ligeramente superior al binario (0.4345 paquetes/ranura). Se mantienen las mismas características especificadas para el algoritmo WSA binario que le aventajan respecto a los algoritmos FCFS: el funcionamiento para procesos de llegada distintos de Poisson y una alta robustez frente a los errores en el feedback.

Una lectura posible es que no es suficiente este ligero incremento del throughput comparado con el hecho que el feedback recibido es sustancialmente mayor, es decir, puede discriminar el hecho que haya 0 o 1 usuarios accediendo. El desarrollo de este estudio a lo largo de la presente tesis ha podido posibilitar esta lectura.

- (d) El algoritmo WSA ideal tiene un throughput de 0.5052 paquetes/ranura. Este algoritmo no es implementable en la práctica, pero a lo largo de la historia todos los estudios sobre los algoritmos de acceso aleatorio han estado centrados en algún momento

en el estudio de la versión ideal correspondiente. La razón es la siguiente: conociendo los límites de las prestaciones de un algoritmo podemos conocer con exactitud a qué distancia se encuentran los algoritmos implementables del ideal, porque si realmente están próximos no es necesario invertir esfuerzos para mejorarlos; o visto desde otro punto de vista, cualquier mejora añadida será lo suficientemente pequeña para que no compense la más que presumible complicación introducida.

En esta tesis se ha estudiado el algoritmo WSA ideal en su totalidad, lo que proporciona un espejo donde mirar si se pretende mejorar los algoritmos existentes.

3. Desde el punto de vista del retardo, se ha analizado también la versión binaria, ternaria e ideal del protocolo ALOHA estabilizado de ventana (WSA) y se ha llegado a las siguientes conclusiones:

- (a) El algoritmo WSA binario tiene un retardo relativamente reducido, comparado con el algoritmo de M. Paterakis y P. Papantoni-Kazakos [27]. Una característica adicional a destacar es que el retardo es muy reducido para tasas bajas, tendiendo a 1.56 ranuras. Esta cualidad se ha de tener en cuenta porque hay algoritmos que consiguen reducir retardo para altas tasas a costa de mantener un retardo alto para tasas bajas, que es claramente un inconveniente, ya que ésta será la zona de trabajo habitual.
- (b) El algoritmo WSA ternario presenta un retardo ligeramente menor que el algoritmo WSA binario. Este resultado era el esperado, puesto que el feedback posee más información en el caso ternario, y el retardo medio esperado por un paquete para ser transmitido será menor.
- (c) El algoritmo WSA ideal es el que consigue un retardo medio menor. Era de esperar que debido a la idealidad del algoritmo, es decir, la elección óptima de las probabilidades de acceso, el retardo esperado fuera menor; pero no es el de retardo mínimo, porque el parámetro optimizado es el throughput.

Es de destacar también que el estudio del retardo para los algoritmos en su versión ideal son muy escasos en la literatura especializada.

4. Centrándonos en el algoritmo en su versión binaria y analizando posibles ligeras variantes en la probabilidad de acceso, hemos llegado a la



conclusión que la probabilidad de acceso inicial óptima es 1, y que una posible función de probabilidad de la forma  $p(i) = 1/i^{0.91}$  mejora el máximo throughput, llegando hasta 0.4347 paquetes/ranura.

5. Dado que el algoritmo presentado es mejor en cuanto a prestaciones que el algoritmo de M. Paterakis y P. Papantoni-Kazakos [27], es de suponer que esta mayor bondad se mantendrá en todas las aplicaciones donde el citado algoritmo ha venido siendo aplicado desde su concepción, en 1987.

A título de ejemplo, el Capítulo 11 se presenta una aplicación a la integración de voz y datos mostrando por resultados de simulación lo anteriormente expuesto.

6. Como líneas futuras, es inmediato que el estudio de las aplicaciones hechas para el algoritmo de M. Paterakis y P. Papantoni-Kazakos [27] es un claro camino a seguir.

Además, un tema abierto es la modificación del algoritmo propuesto para funcionar en sintonización limitada. Desde un punto de vista analítico, su estudio es un problema de difícil solución; pero desde un punto de vista práctico se ha esbozado en el Capítulo 8 una vía de aplicación.

# Bibliografia

- [1] V. Casares and J. Paradells, "An integrated voice/data traffic model with variable voice service rate," in *ICC*, pp. 971–976, June 1989.
- [2] V. Casares and J. Paradells, "Variable voice service rate in wireless access packet networks," in *Wireless*, pp. 609–623, July 1993.
- [3] V. Casares and X. Viladegut, "An approximated analysis of stabilized PRMA in a FDMA/TDMA/FDD bearer service," in *Wireless*, pp. 360–370, July 1993.
- [4] N. Abramson, "Development of the ALOHANET," *IEEE Transactions on Information Theory*, vol. 31, pp. 119–123, March 1985.
- [5] N. Abramson, "The ALOHA system—another alternative for computer communications," in *Fall Joint Computer Conference*, pp. 281–285, 1970.
- [6] R. Metcalfe, "Packet communication," tech. rep., Project MAC, Massachusetts Inst. Technol., July 1973.
- [7] M. Ferguson, "On the control, stability, and waiting time in a slotted ALOHA random-access system," *IEEE Transactions on Communications*, vol. 23, pp. 1306–1311, November 1975.
- [8] A. Carleial and M. Hellman, "Bistable behavior of aloha-type systems," *IEEE Transactions on Communications*, vol. 23, pp. 401–410, April 1975.
- [9] L. Kleinrock, *Queueing Systems*. John Wiley and Sons, 1975.
- [10] M. Ferguson, "A approximate analysis of delay for fixed and variable length packets in an unslotted ALOHA channel," *IEEE Transactions on Communications*, vol. 25, pp. 644–654, July 1977.

- [11] S. Bellini and P. Borgonovo, "On the throughput of an ALOHA channel with variable length packets," *IEEE Transactions on Communications*, vol. 28, pp. 1932–1935, November 1980.
- [12] B. Hajek and T. Van Loon, "Decentralized dynamic control of a multi-access broadcast channel," *IEEE Transactions on Automatic Control*, vol. 27, pp. 559–569, June 1982.
- [13] L. Roberts, "ALOHA packet system with and without slots and capture," *Computer Communications Review*, vol. 5, pp. 28–42, April 1975.
- [14] G. Fayolle, J. Labetoulle, D. Bastin, and E. Gelenbe, "The stability problem of broadcast packet switching computer networks," *Acta Informatica*, vol. 4, no. 1, pp. 49–53, 1974.
- [15] J. Capetanakis, "Tree algorithms for packet broadcast channels," *IEEE Transactions on Information Theory*, vol. 25, pp. 505–515, September 1979.
- [16] B. Tsybakov and V. Mikhailov, "Free synchronous packet access in a broadcast channel with feedback," *Problemy Peredachi Informatsii*, vol. 14, pp. 32–59, October–December 1978.
- [17] J. Hayes, "An adaptative technique for local distribution," *IEEE Transactions on Communications*, vol. 26, pp. 1178–1186, August 1978.
- [18] G. Fayolle, P. Flajolet, M. Hofri, and P. Jacquet, "Analysis of a stack algorithm for random multiple-access communications," *IEEE Transactions on Information Theory*, vol. 31, pp. 244–254, March 1985.
- [19] J. Massey, "Collision resolution algorithms and random-access communications," in *Multi-user Communication Systems*, CISM Courses and Lectures Series, pp. 73–137, G. Longo, Springer-Verlag, 1981.
- [20] R. Gallager, "Conflict resolution in random access broadcast networks," in *AFOSS Workshop Communication Theory and Applications*, pp. 74–76, IEEE, September 1978.
- [21] B. Tsybakov and V. Mikhailov, "Random multiple access of packets: part-and-try algorithm," *Problemy Peredachi Informatsii*, vol. 16, pp. 65–79, October–December 1980.

- [22] J. Huang and T. Berger, "Delay analysis of interval-searching contention resolution algorithms," *IEEE Transactions on Information Theory*, vol. 31, pp. 264–273, March 1985.
- [23] J. Mosely and P. Humblet, "A class of efficient contention resolution algorithms for multiple access channels," *IEEE Transactions on Communications*, vol. 33, pp. 145–151, February 1985.
- [24] N. Vvedenskaya and M. Pinsker, "Non-optimality of the part-and-try algorithm," in *Abstracts of the International Workshop of Convolutional Codes, Multiuser Communication*, pp. 141–148, 1983.
- [25] P. Studer and H. Pletscher, "Q-ary part-and-try algorithm for packet conflict resolution," tech. rep., Inst. Telecommun. Swiss Fed. Inst. Technol., Zurich, Switzerland, October 1984.
- [26] P. Papantoni-Kazakos and M. Paterakis, "Full analysis of a powerful window random access algorithm," tech. rep., School of Engineering and Applied Science, University of Virginia, Charlottesville, March 1988.
- [27] M. Paterakis and P. Papantoni-Kazakos, "A simple window random access algorithm with advantageous properties," *IEEE Transactions on Information Theory*, vol. 35, pp. 1124–1130, September 1989.
- [28] Y. Gong and M. Paterakis, "A robust random multiple-access algorithm for packet transmissions over noisy channels with error memory," *IEEE Transactions on Communications*, vol. 42, pp. 2664–2669, September 1994.
- [29] M. Paterakis, L. Georgiadis, and P. Papantoni-Kazakos, "A full sensing window random-access algorithm for messages with strict delay constraints," in *Algoritmica*, vol. 46, pp. 313–328, Springer-Verlag, 1989.
- [30] P. Papantoni-Kazakos, "Multiple-access algorithms for a system with mixed traffic: high and low priority," *IEEE Transactions on Communications*, vol. 40, pp. 541–555, March 1992.
- [31] F. Babich, "A packet access method for handling mixed traffic on wireless networks," in *Wireless*, pp. 579–587, 1993.
- [32] F. Babich, "Analysis of a contention resolution multiple-access algorithm for handling mixed traffic on wireless networks," *International Journal of Wireless Information Networks*, vol. 1, no. 4, pp. 271–287, 1994.

- [33] F. Babich, "Contention resolution multiple access techniques for handling speech and data traffic on wireless networks," in *The International Zurich Seminar on Digital Communications*, pp. 476–487, March 1994.
- [34] S. Jangi and L. Merakos, "Performance analysis of reservation random access protocols for wireless access networks," *IEEE Transactions on Communications*, vol. 42, pp. 1223–1234, February 1994.
- [35] M. Paterakis and A. Cleary, "On the voice–data integration in third generation wireless access communications systems," *European Transactions on Telecommunications and Related Technologies*, vol. 5, pp. 11–18, January–February 1994.
- [36] A. Cleary and M. Paterakis, "An investigation of reservation random access algorithms for voice–data integration in microcellular wireless environments," *International Journal of Wireless Information Networks*, vol. 2, no. 1, pp. 1–16, 1995.
- [37] P. Mathys and P. Flajolet, "Q–ary collision resolution algorithms in random–access systems with free or blocked channel access," *IEEE Transactions on Information Theory*, vol. 31, pp. 614–622, March 1985.
- [38] A. Bar-David and M. Sidi, "Collision resolution algorithms in multi-station packet–radio networks," in *Performance'87: Proceedings of the 12th IFIP WG 7.3 International Symposium on Computer Performance Modelling, Measurement, and Evaluation* (P. Courtois and G. Latouche, eds.), pp. 385–399, Elsevier Science Publishers B. V., 1988.
- [39] G. Fayolle, E. Gelenbe, and J. Labetoulle, "Stability and optimal control of the packet switching broadcast channel," *J. Assoc. Comput. Machinery*, vol. 24, pp. 375–386, July 1977.
- [40] L. Clare, "Control procedures for slotted ALOHA systems that achieve stability," in *ACM SIGCOMM Symp., Commun., Arch. and Protocols*, August 1986.
- [41] S. Thomopoulos, "A simple and versatile decentralized control for slotted ALOHA, reservation ALOHA, and local area networks," *IEEE Transactions on Communications*, vol. 36, pp. 662–674, June 1988.
- [42] R. Rivest, "Network control by bayesian broadcast," *IEEE Transactions on Information Theory*, vol. 33, pp. 323–328, May 1987.

- [43] L. Merakos and D. Kazakos, "On retransmission control policies in multiple-access communication networks," *IEEE Transactions on Automatic Control*, vol. 30, pp. 109–117, February 1985.
- [44] J. Massey, "Some new approaches to random-access communications," in *Performance'87: Proceedings of the 12th IFIP WG 7.3 International Symposium on Computer Performance Modelling, Measurement, and Evaluation* (P. Courtois and G. Latouche, eds.), pp. 551–569, Elsevier Science Publishes B. V., 1988.
- [45] I. Cidon, H. Kodesh, and M. Sidi, "Erasure, capture, and random power level selection in multiple-access systems," *IEEE Transactions on Communications*, vol. 36, pp. 263–271, March 1988.
- [46] R. Rom and M. Sidi, *Multiple Access Protocols: Performance and Analysis*. Telecommunication Networks and Computer Systems, Springer-Verlag, 1990.
- [47] M. Paterakis and P. Papantoni-Kazakos, "A limited sensing random-access algorithm with binary success-failure feedback," *IEEE Transactions on Communications*, vol. 37, pp. 526–530, May 1989.
- [48] G. Polyzos, M. Molle, and A. Venetsanopoulos, "Performance analysis of finite nonhomogeneous population tree conflict resolution algorithms using constant size window access," *IEEE Transactions on Communications*, vol. 35, pp. 1124–1137, November 1987.
- [49] R. Gallager, "A perspective on multiaccess channels," *IEEE Transactions on Information Theory*, vol. 31, pp. 124–142, March 1985.
- [50] N. Mehravari and T. Berger, "Poisson multiple-access contention with binary feedback," *IEEE Transactions on Information Theory*, vol. 30, pp. 745–751, September 1984.
- [51] M. Gerla and L. Kleinrock, "Closed loop stability controls for S-ALOHA satellite communications," in *Fifth Data Communications Symp.*, pp. 2.10–2.19, September 1977.
- [52] B. Ann and E. Gelenbe, "Etude et simulation de reseaux de diffusion sous controle," *RAIRO Informatique/Comput. Sci.*, vol. 11, pp. 301–321, November 1977.
- [53] A. Segall, "Recursive estimation from discrete-time point processes," *IEEE Transactions on Information Theory*, vol. 22, pp. 422–431, July 1976.

- [54] D. Seret and C. Macchi, "Control adaptatif d'un reseau á diffusion de paquets," in *Flow Control in Computer Networks*, J.L. Grange and M. Gien Eds., 1979.
- [55] S. Lam and L. Kleinrock, "Packet switching in a multiaccess broadcast channel: Dynamic control procedures," *IEEE Transactions on Communications*, vol. 23, pp. 891-904, September 1975.
- [56] G. Cunningham, "Delay versus throughput comparisons for stabilized slotted ALOHA," *IEEE Transactions on Communications*, vol. 38, pp. 1932-1934, November 1990.
- [57] V. Mikhailov, "Metody sluchainogo mnojestvennogo dostupa (methods of random multiple access)," Candidate Eng. thesis, Moscow Inst. Physics and Technol., 1979.
- [58] V. Casares and J. Alcober, "Voice and data integration in wireless networks," in *Wireless*, pp. 1-10, July 1994.
- [59] B. Tsybakov, "Survey of USSR contributions to random multiple-access communications," *IEEE Transactions on Information Theory*, vol. 31, pp. 143-165, March 1985.
- [60] L. Merakos and C. Bisdikian, "Delay analysis of the n-ary stack random-access algorithm," *IEEE Transactions on Information Theory*, vol. 34, pp. 931-942, September 1988.
- [61] M. Paterakis, L. Georgiadis, and P. Papantoni-Kazakos, "On the relation between the finite and the infinite population models for a class of RAA's," *IEEE Transactions on Communications*, vol. 35, pp. 1239-1240, November 1987.
- [62] N. Vvedenskaya and B. Tsybakov, "Random multiple-access of packets to a channel with errors," *Problemy Peredachi Informatsii*, vol. 19, pp. 52-68, April-June 1983.
- [63] D. Ryter, "A conflict resolution algorithm for noisy multiaccess channels," tech. rep., Mass. Inst. Technol., Cambridge, June 1987.
- [64] G. Ruget, "Some tools for study of channel-sharing algorithms," in *Multi-user Communication Systems*, CISM Courses and Lectures Series, pp. 201-231, G. Longo, Springer-Verlag, 1981.

- [65] I. Stavrakakis and D. Kazakos, "A multi user random access communication system for users with different priorities," tech. rep., Department of Electrical Engineering and Computer Science, University of Vermont, Burlington, 1987.
- [66] L. Kantorovich and V. Krylov, *Approximate Methods of Higher Analysis*. Interscience Publ., 1964.
- [67] B. Tsybakov and N. Vvedenskaya, "Random multiple-access stack algorithm," *Problemy Peredachi Informatsii*, vol. 16, pp. 80–94, July–September 1980.
- [68] L. Georgiadis, L. Merakos, and P. Papantoni-Kazakos, "A method for the delay analysis of random multiple-access algorithms whose delay process is regenerative," *IEEE Journal on Selected Areas in Communication*, vol. 5, pp. 1051–1062, July 1987.
- [69] N. Vvedenskaya and B. Tsybakov, "Packet delay in multi-access stack algorithm," *Problemy Peredachi Informatsii*, vol. 20, pp. 77–97, April–June 1984.
- [70] W. Rosenkrantz, "Some remarks on the asymptotic behavior of the lengths of a collision resolution interval," *IEEE Transactions on Communications*, vol. 34, pp. 962–965, September 1986.
- [71] S. Gass, *Linear Programming: Methods and Applications*. McGraw-Hill, fifth ed., 1985.
- [72] L. Georgiadis and P. Papantoni-Kazakos, "Limited feedback sensing algorithms for the packet broadcast channel," *IEEE Transactions on Information Theory*, vol. 31, pp. 280–294, March 1985.
- [73] S. Ross, *Stochastic Processes*. Wiley Series in Probability and Mathematical Statistics. Probability and Mathematical Statistics, J. Wiley, 1983.
- [74] J. Huang and T. Berger, "Delay analysis of 0.487 contention resolution algorithms," *IEEE Transactions on Communications*, vol. 34, pp. 916–926, September 1986.
- [75] Y. Yemini and L. Kleinrock, "An optimal adaptative scheme for multiple access broadcast communications," in *ICC*, pp. 7.2.1–7.2.5, 1978.
- [76] B. Tsybakov, "Resolution of a conflict of known multiplicity," *Probl. Information Transmission*, vol. 16, pp. 134–144, April–June 1980.



- [77] L. Georgiadis and P. Papantoni-Kazakos, "A collision resolution protocol for random access channels with energy detectors," *IEEE Transactions on Communications*, vol. 30, pp. 2413–2420, November 1982.
- [78] N. Pippenger, "Bounds on the performance of protocols for a multiple access broadcast channel," *IEEE Transactions on Information Theory*, vol. 27, pp. 145–152, March 1981.
- [79] M. Molle, "On the capacity of infinite population multiple access protocols," *IEEE Transactions on Information Theory*, vol. 28, pp. 396–401, May 1982.
- [80] R. Cruz and B. Hajek, "A new upper bound to the throughput of a multi-access broadcast channel," *IEEE Transactions on Information Theory*, vol. 28, pp. 402–405, May 1982.
- [81] B. Tsybakov and V. Mikhailov, "An upper bound for maximum throughput of random access system," in *Int. Symp. Inform. Theory*, 1981.
- [82] N. M. T. Berger and G. Munson, "On genie-aided upper bounds to multiple access contention resolution efficiency," in *Annu. Conf. Inform. Sci. Syst.*, 1981.
- [83] B. Tsybakov and N. Likhanov, "An upper bound on packet delay in a multiple access channels," *Problemy Peredachi Informatsii*, vol. 18, pp. 76–84, October 1980.
- [84] B. Tsybakov and N. Likhanov, "Lower bound for the delay in multiple access systems," *Problemy Peredachi Informatsii*, vol. 27, pp. 73–88, July–September 1991.
- [85] B. Hajek, N. Likhanov, and B. Tsybakov, "On the delay in a multiple-access system with large propagation delay," *IEEE Transactions on Information Theory*, vol. 40, pp. 1158–1166, July 1994.
- [86] P. Humblet, "On the throughput of channels access algorithms with limited sensing," *IEEE Transactions on Communications*, vol. 34, pp. 345–347, April 1986.
- [87] L. Georgiadis and P. Papantoni-Kazakos, "A 0.487 throughput limited sensing algorithm," *IEEE Transactions on Information Theory*, vol. 33, pp. 233–237, March 1987.

- [88] U. Bhat and M. Fischer, "Multichannel queueing systems with heterogeneous classes of arrival," *Naval Res. Logist Quarterly*, vol. 23, pp. 271–283, February 1976.
- [89] M. Fischer, "A queueing analysis of an integrated telecommunications systems with priorities," *INFOR*, vol. 15, pp. 277–288, October 1977.
- [90] K. Kummerle, "Multiple performance for integrated line and packet switched traffic," in *ICCC*, pp. 508–515, 1974.
- [91] P. Zafiropoulo, "Flexible multiplexing for networks supporting line-switched and packet-switched data traffic," in *ICCC*, pp. 517–523, 1974.
- [92] M. Schwartz, *Telecommunications Networks. Protocols, Modeling and Analysis*. Addison-Wesley Series in Electrical and Computer Engineering, Addison Wesley, 1987.
- [93] D. Gaver and P. Lehoczky, "Channel that cooperative service a data stream and voice messages," *IEEE Transactions on Communications*, vol. 30, pp. 1153–1162, May 1982.
- [94] A. L. García, R. Kwong, and G. Williams, "Performance evaluation methods for an integrated voice–data link," *IEEE Transactions on Communications*, vol. 30, pp. 1848–1858, August 1982.
- [95] S. Nanda, "Analysis of packet reservation multiple access: Voice data integration for wireless network," in *GLOBECOM*, pp. 1984–1988, December 1990.
- [96] H. Chung and D. Goodman, "Transmission of speech and data using packet reservation multiple access," in *ICC*, pp. 99–104, 1991.
- [97] D. Goodman, R. Valenzuela, K. Gayliard, and B. Ramamurthi, "Packet reservation multiple access for local wireless communications," *IEEE Transactions on Communications*, vol. 37, pp. 885–890, August 1989.
- [98] P. Reilly and A. Ghani, "Data performance in burst switching when the voice silence periods have a hyper exponential distribution," *IEEE Transactions on Communications*, vol. 35, pp. 1109–1112, October 1987.
- [99] D. Goodman and S. Wei, "Efficiency of packet reservation multiple access," *IEEE Transactions on Vehicular Technology*, vol. 40, pp. 280–294, February 1991.

- [100] C. Námislo, "Analysis of mobile radio slotted ALOHA networks," *IEEE Journal on Selected Areas in Communications*, vol. 2, pp. 563–574, July 1984.
- [101] D. Chakraborty, "VSAT communications networks—an overview," *IEEE Communications Magazine*, vol. 26, pp. 10–24, May 1988.