



**Propuesta de mejora al desempeño del  
algoritmo de control de acceso al medio 2C**

**Para obtener el grado de  
Maestro en Ciencias  
(Ciencias y Tecnologías de la Información)**

**P R E S E N T A**


**Luis Ángel Alarcón Ramos**

Asesores

Dr. Miguel López Guerrero

Sinodales

Presidente: Dr. Víctor Rangel Licea 

Secretario: Dr. Miguel López Guerrero 

Vocal: Dr. Miguel Ángel Ruiz Sánchez 

21 de noviembre de 2007





# **Propuesta de mejora al desempeño del algoritmo de control de acceso al medio 2C**

Idónea Comunicación de Resultados para obtener el grado de

**MAESTRO EN CIENCIAS Y TECNOLOGÍAS DE LA INFORMACIÓN**

por

Alarcón Ramos Luis Ángel

Asesor:

Dr. Miguel López Guerrero

México D.F. a 21 de noviembre de 2007.



# Resumen

---

Los algoritmos de control de acceso, tienen la finalidad de distribuir el uso de un medio de transmisión entre un conjunto de estaciones o nodos que buscan transmitir información. En este trabajo se presenta una propuesta de mejora al algoritmo de control de acceso al medio 2C. En este algoritmo el tiempo se supone ranurado y existe una estación central que es la encargada de monitorear el medio y de informar su estado a los demás nodos. En condiciones óptimas de operación una sola estación accede al canal. Sin embargo, cuando dos o más nodos transmiten de forma simultánea, se dice que ha ocurrido una *colisión* y por lo tanto la pérdida de la información transmitida.

El algoritmo 2C resuelve las colisiones permitiendo el acceso al medio de transmisión de forma aleatoria. Cuando ocurre una colisión los nodos intentarán transmitir con probabilidad  $p$  y con probabilidad  $1 - p$  esperarán hasta que no hayan colisiones para poder intentar transmitir de nuevo. El algoritmo 2C original emplea  $p = \frac{1}{2}$ , lo que produce largos tiempos de espera cuando el número de estaciones es grande.

El presente trabajo propone la búsqueda de un valor de probabilidad  $p$  óptimo que disminuya los tiempos de espera. La solución planteada en este trabajo consiste en determinar el valor óptimo de probabilidad como función del número de estaciones en contienda. Para este fin se analizan las siguientes dos propuestas:

- Política de realimentación por ranura. En donde la estación central determinará y enviará a los nodos el número de estaciones en contienda en cada ranura de tiempo.
- Política de realimentación por fase. En este caso la estación central determinará y enviará a los nodos el número de estaciones en contienda cada determinado intervalo de tiempo.

De lo anterior surge una propuesta la cual se ha denominado *algoritmo 2C adaptable*, y que reduce en gran medida los tiempos de espera, además de que sólo requiere de ligeros cambios por parte del algoritmo original. El presente trabajo muestra que el tiempo de resolución de colisiones en el algoritmo 2C adaptable tiene un comportamiento lineal como función del número de estaciones en contienda, mientras que el algoritmo 2C original presenta un comportamiento logarítmico como función del factorial del número de estaciones en contienda.

# Agradecimientos

---

Quiero agradecer a la Universidad Autónoma Metropolitana por la beca que me otorgó durante la duración de la maestría. También, quiero externar un agradecimiento a los profesores de la Maestría en Ciencias y Tecnologías de la información en especial a mi asesor, el Dr. Miguel López Guerrero, cuya orientación a lo largo de la maestría resultó en la elaboración del presente trabajo.

A mis padres, ya que gracias a su apoyo he alcanzado esta meta.

Finalmente a todos los compañeros y amigos de la maestría, ya que pasamos buenos momentos durante esta experiencia.



# Índice

---

Resumen	iii
Agradecimientos	v
Índice	vii
Índice de figuras	xi
Índice de tablas	xiii
Lista de símbolos	xv
<b>1 Introducción</b>	<b>1</b>
1.1 Metodología de investigación . . . . .	3
1.2 Contribución . . . . .	4
1.3 Organización del documento . . . . .	5
<b>2 Algoritmos de control de acceso al medio</b>	<b>7</b>
2.1 Clasificación . . . . .	7
2.2 Descripción de algunos algoritmos . . . . .	10
2.2.1 Acceso múltiple por división de frecuencia (FDMA, <i>Frequency Division Multiple Access</i> ) . . . . .	10
2.2.2 Acceso múltiple por división en el tiempo (TDMA, <i>Time Division Multiple Access</i> ) . . . . .	10
2.2.3 ALOHA puro y ALOHA ranurado . . . . .	11

2.2.4	Acceso múltiple por detección de portadora (CSMA, <i>Carrier Sense Multiple Access</i> ) . . . . .	13
2.2.5	Algoritmo de árbol ( <i>Tree Algorithm</i> ) . . . . .	17
2.2.6	Protocolo de pila ( <i>Stack Protocol</i> ) . . . . .	19
2.2.7	Algoritmo de las dos celdas (2C, <i>Two-Cell Algorithm</i> ) . . . . .	21
2.3	Estado del conocimiento . . . . .	24
<b>3</b>	<b>El algoritmo 2C-adaptable y su análisis de desempeño</b>	<b>29</b>
3.1	Política de realimentación de la multiplicidad de la colisión por ranura . . . . .	31
3.1.1	Análisis . . . . .	31
3.1.2	Resultados de simulación para la política de realimentación por ranura . . . .	38
3.1.3	Margen de error permitido para la política de realimentación por ranura . . .	42
3.2	Política de realimentación de la multiplicidad de la colisión por fase . . . . .	45
3.2.1	Análisis . . . . .	45
3.2.2	Resultados de simulación para la política de realimentación por fase . . . . .	52
3.2.3	Margen de error permitido para la política de realimentación por fase . . . .	54
3.2.4	Comparación de los datos simulados <i>vs.</i> análisis . . . . .	55
3.3	Ejemplo de estimación de la multiplicidad de la colisión en el algoritmo 2C-adaptable	56
3.3.1	Fuente de tráfico . . . . .	58
3.3.2	Estimador de la multiplicidad de la colisión . . . . .	60
3.3.3	Resultados de la simulación con el estimador de la multiplicidad de la colisión	62
<b>4</b>	<b>Conclusiones y recomendaciones para trabajo futuro</b>	<b>67</b>
<b>A</b>	<b>Ecuaciones para el análisis de las políticas de difusión</b>	<b>71</b>
A.1	Intervalo de $p$ para el análisis de realimentación por ranura . . . . .	71
A.2	Probabilidad por rama para el árbol de transiciones . . . . .	73
A.3	Probabilidad de estados en un árbol de transiciones . . . . .	74
A.4	Demostración de la ecuación A.20 . . . . .	80
A.5	Longitud promedio de un CRI para el algoritmo 2C . . . . .	83
A.6	Longitud promedio de un CRI para el algoritmo 2C-adaptable . . . . .	90

<b>B</b>	<b>Simulador del algoritmo 2C-adaptable</b>	<b>95</b>
B.1	Descripción del Simulador . . . . .	96
B.2	Funcionamiento del Simulador . . . . .	97
B.3	Implantación del algoritmo 2C en el Simulador . . . . .	98
	<b>Referencias</b>	<b>101</b>



# Índice de figuras

---

2.1	Clasificación de los protocolos MAC . . . . .	9
2.2	FDMA y TDMA . . . . .	11
2.3	Algoritmo CSMA 1-persistente . . . . .	14
2.4	Algoritmo CSMA no persistente . . . . .	14
2.5	Algoritmo CSMA p-persistente . . . . .	15
2.6	Tiempo de propagación . . . . .	16
2.7	Árbol binario . . . . .	18
3.1	Diagrama de tiempo en un CRI . . . . .	30
3.2	Política de realimentación por ranura . . . . .	32
3.3	Curvas de $p_M(N)$ y $p_D(N)$ . . . . .	36
3.4	Curvas de $p_M(N)$ , $p_D(N)$ , $p_O(N)$ y $p_m(N)$ . . . . .	38
3.5	Resultados de la simulación para la política de realimentación por ranura, incluyendo el algoritmo original. . . . .	39
3.6	Resultados de la simulación para la política de realimentación por ranura . . . . .	40
3.7	Resultados de la simulación para la política de realimentación por ranura, para $N$ pequeño. . . . .	42
3.8	Simulaciones promedio para la política de realimentación por ranura. . . . .	43
3.9	Razón de error al estimar la multiplicidad de la colisión $N$ . . . . .	45
3.10	Árbol de transiciones . . . . .	48
3.11	Resultados de la simulación para la política de realimentación por fase . . . . .	53
3.12	Resultados de la simulación para la política de realimentación por fase . . . . .	54

3.13 Simulación <i>vs.</i> análisis para el algoritmo 2C original . . . . .	56
3.14 Simulación <i>vs.</i> análisis para la política de realimentación por fase en el algoritmo 2C-adaptable . . . . .	57
3.15 Talk Spurts . . . . .	58
3.16 Simulación con estimación para la política de realimentación por fase. Paquetes no transmitidos. . . . .	63
3.17 Simulación con estimación para la política de realimentación por fase. Retardo prome- dio. . . . .	65
A.1 Probabilidad para cada rama del árbol de transiciones. . . . .	74
A.2 Árbol de transiciones de nivel $n = 3$ . . . . .	75
A.3 Árbol de transiciones para longitud promedio de un CRI en el algoritmo 2C-Adaptable	92
B.1 Funcionamiento del simulador . . . . .	99
B.2 Comportamiento de un nodo en el Simulador contra el algoritmo 2C . . . . .	100

# Índice de tablas

---

3.1	Resultados numéricos de la simulación para política de realimentación por ranura . .	41
3.2	Parámetros de la simulación . . . . .	64



# Lista de símbolos

---

- $\mathbb{E}$  – Conjunto de números enteros.
- $\mathbb{N}$  – Conjunto de números enteros no negativos, incluyendo el cero.
- $N$  – Número de estaciones en colisión (multiplicidad de la colisión).
- $\hat{N}$  – Estimación del número de estaciones que colisionaron.
- $p$  – Probabilidad de permanecer en el estado de transmisión.
- $P_{N,k}(p)$  – Probabilidad de transición del estado  $N$  al estado  $k$ .
- $Q_{N,i}^{(n)}(p)$  – Probabilidad de transición del estado  $N$  al estado  $i$ , en  $n$  transiciones o saltos.
- $\lambda$  – Razón de paquetes por ranura.
- $\lambda_a$  – Razón de arribos por ranura.
- $\lambda_p$  – Razón de paquetes por arribo.
- $NE$  – Número de estaciones en el sistema.



# Capítulo 1

## Introducción

---

Generalmente al hablar de sistemas de comunicación, surge la idea de *transferir información* de un lugar a otro, en donde una entidad o nodo envía datos a otro; sin embargo, en la actualidad no sólo información es lo que se comparte. Un ejemplo claro es una *red de computadoras* (o simplemente *red*), en donde un conjunto de nodos o computadoras comparten información, recursos<sup>1</sup> o prestan servicios. El primer problema que surge al momento de establecer una red o en general un sistema de comunicación, es la forma de interconectar los nodos.

Básicamente existen dos formas de interconectar un conjunto de nodos<sup>2</sup>:

1. A través de una conexión *punto a punto* entre cada par de nodos.
2. Mediante el empleo de un canal o medio de transmisión *compartido*.

La conexión punto a punto involucra un enlace único, la transmisión de datos entre los nodos no requiere de ningún intermediario<sup>3</sup>, y sólo es necesario emplear diversos mecanismos que determinan la manera en que se comunicarán. Estos mecanismos se conocen con el nombre de *protocolos*, y en la actualidad son un conjunto muy extenso y diverso. Sin embargo, la interconexión que cada nodo tendrá dentro de la red, es decir, lo que se conoce como *topología de la red*, deberá ser fija<sup>4</sup>. Una

---

<sup>1</sup>Por ejemplo software, tiempo de procesador, servicios de impresión, etc.

<sup>2</sup>En general emplearemos los términos *nodo*, *computadora* o *estación*, para referirnos a aquellas entidades autónomas capaces de transmitir información, o proporcionar un recurso o servicio.

<sup>3</sup>Salvo que se necesite intensificar la señal, por ejemplo un repetidor.

<sup>4</sup>Esto puede no ser cierto para una red de *conmutación de circuitos*, en donde un nodo puede conectarse con cualquier otro, requiriendo un enlace por conexión.

característica importante en este tipo de conexión es que, en ausencia de factores externos a la red<sup>5</sup>, los nodos no se preocupan porque su transmisión de datos se vea afectada por las transmisiones entre otro par de nodos. En los casos en los que se desea simplificar o dar cierta flexibilidad a la topología de la red no es conveniente usar conexiones punto a punto. Por ejemplo, al conectar computadoras éstas podrían cambiar su posición (incluso estar en movimiento) y alterar la topología de la red en cualquier instante, lo cual implicaría grandes cambios en una red punto a punto. En cambio cuando se interconectan *redes*<sup>6</sup> es adecuado emplear conexiones punto a punto, ya que en este caso cada *red* se ve como un nodo y en principio su posición geográfica prácticamente es fija. Existen otros criterios para no emplear conexiones punto a punto. Por ejemplo, es posible que bajo ciertos ambientes no sea posible colocar un enlace por cada par de nodos debido a que el número de nodos es muy grande. Esto puede resultar costoso en este tipo de conexión ya que la escalabilidad en cuanto a número de conexiones no es una característica en los enlaces punto a punto.

La siguiente forma de conexión es emplear un mismo enlace para todos los nodos, esto es lo que se conoce como *canal de comunicación compartido* o simplemente un *canal compartido*<sup>7</sup>. A diferencia de un enlace punto a punto, en un canal compartido los datos no sólo pueden sufrir errores en la transmisión debido a factores externos a la red, sino también debido a factores internos<sup>8</sup>. En una conexión punto a punto nada evita que cualquier nodo transmita información<sup>9</sup>; no así en un canal compartido, en donde el problema surge cuando dos o más nodos transmiten al mismo tiempo<sup>10</sup>. En este caso se dice que ha ocurrido una *colisión*, y al número estaciones involucradas en la colisión se le denomina *multiplicidad de la colisión*. Desde luego éste es un efecto no deseable, ya que es posible que la estación receptora no reciba la información de manera correcta<sup>11</sup>. Es necesario entonces usar un mecanismo que evite esto o que al menos lo controle; esto se conoce como *control de acceso al medio* (MAC, *Medium Access Control*) y es principalmente un conjunto de protocolos

---

<sup>5</sup>Un daño en el enlace o señales electromagnéticas externas que perturben la transmisión.

<sup>6</sup>Conocidas como redes WAN, Wide Area Network.

<sup>7</sup>En ocasiones se le llama también *medio de comunicación compartido* o *medio compartido*.

<sup>8</sup>Por ejemplo, un factor interno son las señales que generan los nodos de la red al emplear el canal.

<sup>9</sup>En principio ello depende del modo de transmisión, que puede ser en un sólo sentido, en ambos sentidos pero no simultáneamente y en ambos sentidos simultáneamente, lo que se conoce como transmisión *Simplex*, *Half-Duplex* y *Full-Duplex* respectivamente. La afirmación asume que la transmisión es Full-Duplex.

<sup>10</sup>En realidad no es necesario que la transmisión se realice al mismo tiempo, basta con que los tiempos de transmisión de cada nodo se intercepten.

<sup>11</sup>Aun cuando suceda una colisión, es posible que los nodos destinos reciban de manera correcta la información transmitida; a lo largo de este trabajo, se asume que una colisión implica la pérdida de la información transmitida.

y algoritmos que permiten que los nodos logren una transmisión exitosa en un canal compartido. Así, un canal compartido es mucho más fácil de implementar, pero debido a las colisiones es más difícil de controlar.

Generalmente, un protocolo MAC se compone de dos partes [13]:

1. Un algoritmo de acceso al canal (CAA, *Channel Access Algorithm*) que determina la forma en que las estaciones acceden al canal.
2. Un algoritmo de resolución de colisión<sup>12</sup> (CRA *Conflic Resolution Algoritm*) el cual una vez que sucede la colisión, se encarga de solucionar el problema.

El presente trabajo se enfoca en la parte de resolución de conflictos del algoritmo conocido como 2C (*Two-Cell Algorithm*) [15], el cual, como se verá más adelante, es un algoritmo de control de acceso al medio que permite el acceso al canal cuando las estaciones o nodos tienen información por transmitir. Cuando sucede una colisión soluciona el problema de manera aleatoria.

Dado lo anterior, la hipótesis del presente trabajo es la siguiente: la resolución aleatoria de las colisiones como función del número de estaciones en colisión puede mejorar el desempeño del algoritmo 2C. De esta forma, se reducen los tiempos de espera cuando el número de estaciones es grande.

Por lo tanto, el objetivo es proporcionar una propuesta de mejora al algoritmo 2C que permita una rápida respuesta del CRA; la búsqueda de esta propuesta se fundamenta en un análisis teórico y su evaluación mediante simulaciones, tanto del algoritmo original como de la propuesta. Se pondrá a modo de ejemplo, un modelo de tráfico para evaluar la mejora lograda al algoritmo 2C.

## 1.1 Metodología de investigación

El proceso de desarrollo se divide básicamente en las siguientes partes:

1. Investigación documental a fin de determinar el estado del conocimiento en la materia.

---

<sup>12</sup>A lo largo del trabajo se usaran las palabras *conflicto* y *colisión* indistintamente.

2. Un análisis teórico, que permitirá elaborar y sustentar la mejora al algoritmo. Dicho análisis se enfoca en modificar el comportamiento aleatorio del CRA en función del número de estaciones en colisión; es decir, el CRA considera la multiplicidad de la colisión, y conociendo este dato se podrá reducir el tiempo que cada nodo espera antes de poder transmitir su información con éxito.
3. Simulaciones, que permitirán comparar el desempeño de la propuesta con el algoritmo original.

## 1.2 Contribución

Desde la aparición del protocolo ALOHA<sup>13</sup> (principios de los 70s), se avanzó de manera muy lenta en la mejora del desempeño de los protocolos MAC, tal vez esto debido a la cantidad reducida de nodos e información en aquel entonces. En la actualidad, debido al crecimiento de los recursos de comunicación, es necesario proporcionar mecanismos que aseguren una buena calidad de servicio<sup>14</sup>. El acceso al medio no está exento de esto, ya que un buen control del tráfico permitiría ofrecer un buen servicio; por tanto, es necesario que los algoritmos sean capaces de aprovechar la información que puedan obtener del medio, y con ello tener un mejor desempeño que se traduzca en ofrecer una buena calidad de servicio.

Existen esfuerzos encaminados a ofrecer una mejor calidad de servicio, por ejemplo proporcionar tiempos de espera cortos para transmitir. En ese sentido, la aplicación del procesamiento de señales para recuperar las señales que han colisionado a partir de determinados criterios, permitiría disminuir el trabajo a los CRA en los protocolos MAC [7]; desde luego, este tipo de propuestas no están orientadas a modificar los protocolos MAC, sino más bien a realizar modificaciones tanto a la capa de enlace como a la capa física.

A diferencia de lo anterior, la presente propuesta está encaminada a realizar modificaciones en un protocolo MAC, aprovechando la información que el medio pueda ofrecer (por ejemplo el número de estaciones que colisionó). Nuestra meta es emplear la misma información que el propio algoritmo

---

<sup>13</sup>Descrito más adelante.

<sup>14</sup>La calidad de servicio puede ser medida por muchos parámetros, ello depende del contexto en que se aplique; por ejemplo para nuestra finalidad, la calidad de servicio implica retardos de transmisión cortos.

genera y que el medio puede ofrecer (o es capaz de conocer), y emplear mecanismos de predicción que permitan un mejor desempeño del algoritmo 2C, y que asegure una buena calidad de servicio.

### **1.3 Organización del documento**

Este trabajo está organizado de la siguiente manera:

- En el capítulo 2 se presentan algunos de los diversos protocolos de acceso al medio; asimismo se describe el estado del conocimiento acerca de los protocolos MAC.
- En el capítulo 3 se presenta la propuesta y el análisis teórico que la sustenta, asimismo se muestran los resultados de la simulación del algoritmo original y la propuesta. En esta sección se presenta un posible escenario que complementa la propuesta, y, cuya factibilidad se demuestra por simulaciones.
- Finalmente se presentan las conclusiones y recomendaciones para trabajo futuro.



# Capítulo 2

## Algoritmos de control de acceso al medio

---

En el capítulo anterior hemos introducido la noción de control de acceso al medio, en donde hemos descrito el problema que atacan los protocolos MAC. En este capítulo se ofrecen varias clasificaciones de los protocolos, y la descripción de algunos de ellos. Además, proporcionamos el estado del conocimiento en el que se encuentran los protocolos de acceso al medio relacionados al trabajo aquí presentado.

### 2.1 Clasificación

Debido a la gran cantidad de algoritmos para el control de acceso es de suma importancia ofrecer una clasificación y así poder posicionar al algoritmo de estudio; además, esto ayudará a tener una idea más clara del estado del conocimiento que guardan los protocolos de acceso al medio.

Existen varias clasificaciones para los algoritmos de acceso al medio; en este caso proponemos una clasificación sencilla, la cual estaría dada en términos de su funcionamiento. Así, es posible clasificar a los algoritmos de control de acceso al medio como:

- Estáticos: básicamente es una manera fija de dividir el canal compartido y permitir que los nodos accedan a él de forma organizada, prácticamente se tiene un algoritmo de acceso al canal (CAA). Debido a que la asignación del recurso es de manera fija, surgen problemas

cuando el número de nodos cambia en el tiempo, ya que en este caso es necesario hacer una reasignación del recurso, lo cual en ocasiones puede no resultar posible. Otro problema es el desperdicio del canal, ya que no todos los nodos emplean el recurso de forma continua; así, estos protocolos de control de acceso son adecuados cuando los nodos generan transmisiones de datos de forma constante.

- **Dinámicos:** en este caso los nodos acceden al canal y ante una colisión, los algoritmos de resolución de colisión (CRA) se encargan de asignar el canal de forma aleatoria a cada nodo que así lo solicite. A diferencia de los algoritmos estáticos, ningún nodo tiene asegurado el canal a menos que todos los demás nodos en competencia le permitan transmitir su información. En otras palabras, debe existir consenso por parte de los nodos involucrados, para que uno de ellos pueda transmitir su información. Este tipo de algoritmos, resulta muy adecuado cuando la transmisión de información se realiza a ráfagas.
- **Reservación:** esta clasificación bien podría caer dentro de los protocolos de acceso al medio dinámicos, pero en este caso el acceso al recurso se realiza de forma planeada; es decir, el recurso se reparte dinámicamente entre los usuarios pero bajo algún criterio que no es aleatorio. La asignación aleatoria en los protocolos dinámicos puede presentar una alta variabilidad temporal cuando el número de estaciones que solicitan el canal es muy grande, en cambio, organizar el acceso por medio de reservaciones requiere de un cierto tiempo específico.

En referencia a la clasificación anterior, se muestran ejemplos de mecanismos de control de acceso en la fig. 2.1; éstos son sólo algunos, y varios de ellos tienen algunas variantes que los hace ligeramente diferentes al original; más adelante algunos de ellos se comentarán.

La clasificación anterior posiciona al Algoritmo 2C, como un protocolo MAC dinámico, en el cual las estaciones o nodos compiten por el canal, y cuando suceden las colisiones de manera aleatoria se trata de asignar a un único nodo el derecho de uso del canal de comunicación.

Una clasificación más detallada [19], se centra en el éxito de la transmisión; así es posible clasificar los protocolos en dos grandes grupos

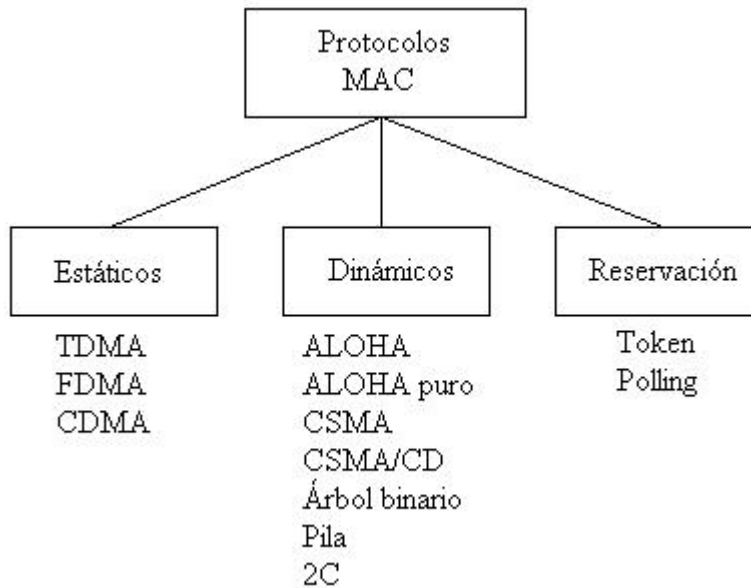


Figura 2.1: Clasificación de los protocolos MAC.

- *Libres de conflicto*, en donde se asegura que la transmisión será exitosa una vez realizada. Estos protocolos a su vez se catalogan en *estáticos* o *dinámicos*, pero ya no en el sentido antes descrito. El que un protocolo libre de conflicto sea estático o dinámico se refiere a la división que realiza del recurso; es decir, este tipo de protocolos son a los que anteriormente se denominó de tipo estático y de reservación. Ejemplos de protocolos libres de conflicto son TDMA, FDMA, CDMA, Token y Polling; en todos los casos el canal de comunicación se asigna de forma planeada. Para TDMA, FDMA y CDMA se realiza de forma estática; mientras que para los restantes se realiza de forma dinámica.
- De *contienda*, en donde por el contrario no se garantiza que la transmisión sea exitosa. Desde luego la transmisión tendrá lugar, pero se requiere de ciertas acciones para que ésta sea exitosa. Los protocolos de contienda de igual manera pueden ser *estáticos* y *dinámicos*. El que un protocolo de contienda sea estático significa que maneja cierto grado de preferencia; es decir, una vez que sucede la colisión, se selecciona el nodo que transmitirá en base a criterios establecidos; por ejemplo, un nivel de jerarquía o simplemente por su identificador. Cuando un protocolo de contienda es dinámico, significa que todos los nodos en colisión tienen la misma posibilidad de transmisión, y su resolución es de tipo aleatorio. Ejemplos de estos protocolos, que se describen más adelante, son: ALOHA, ALOHA puro, Algoritmo de Árbol

y el Algoritmo 2C; estos son protocolos de contienda dinámicos.

A la definición de algoritmos de contienda, hay que agregar que existen algoritmos que son *centralizados* y no *centralizados*; es decir, un algoritmo se dice centralizado cuando existe un nodo que está al pendiente de lo que sucede en el canal, y es capaz de informar a los demás nodos de ello. En cambio uno no centralizado requiere de la participación de todos los nodos para llevar a cabo el buen desarrollo del sistema.

## 2.2 Descripción de algunos algoritmos

### 2.2.1 Acceso múltiple por división de frecuencia (FDMA, *Frequency Division Multiple Access*)

El método FDMA consiste en dividir el ancho de banda<sup>1</sup> del medio de transmisión en subbandas de frecuencia, y asignar cada una de ellas a cada estación para que pueda transmitir. Como el ancho de banda es finito, el número de divisiones también lo será, de esta forma el número de estaciones que podrán transmitir será limitado, pero a la vez, cada estación podrá disponer de un canal para poder transmitir todo el tiempo, ver fig. 2.2a. Este método resulta efectivo cuando las estaciones transmiten continuamente ya que no requiere ningún tipo de coordinación o sincronización entre ellas; es decir, no es necesario un CRA. Cuando las estaciones sólo transmiten por cortos periodos de tiempo (ráfagas), se produce el desperdicio de una porción del canal. Además no posee flexibilidad, ya que si se desea agregar una estación, se requerirá modificar las asignaciones que se hayan realizado.

### 2.2.2 Acceso múltiple por división en el tiempo (TDMA, *Time Division Multiple Access*)

TDMA es semejante a FDMA, pero con la diferencia de que aquí se ranura el tiempo y no la banda de frecuencias. Cada estación puede emplear por completo el medio de transmisión por un determinado intervalo de tiempo; la asignación de estos intervalos se realiza de forma fija y periódica, en donde cada periodo es conocido como *ciclo*, ver fig. 2.2b.

---

<sup>1</sup>Por *ancho de banda* se debe entender en este caso, la banda de frecuencias que el medio permite transmitir sin afectar las características de la señal. En ocasiones el ancho de banda se relaciona con la velocidad de transmisión del medio; haremos esta distinción cuando sea necesario.

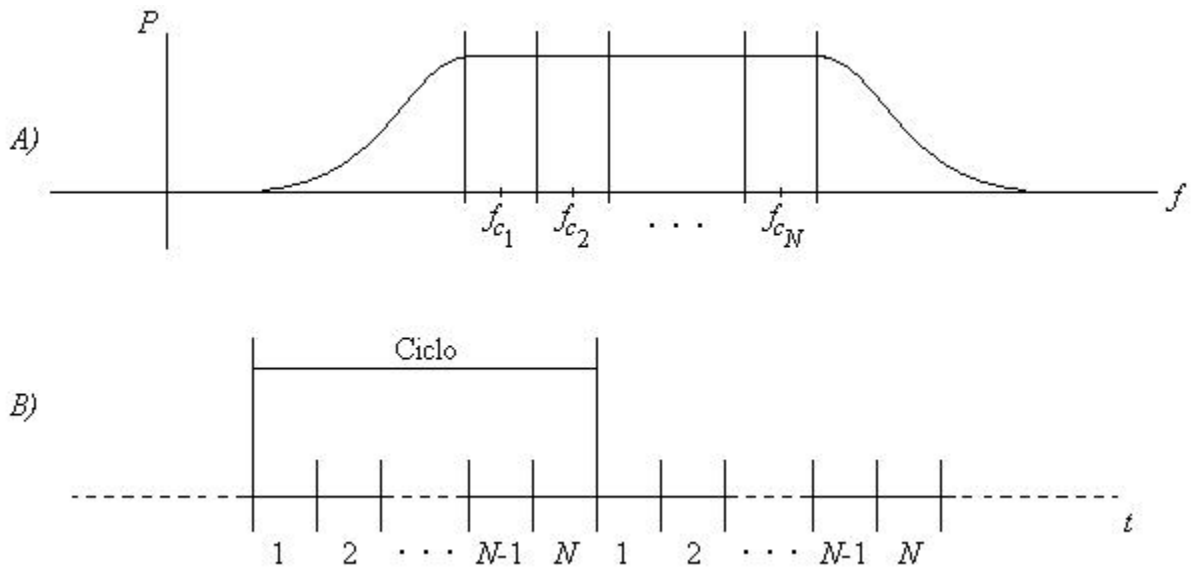


Figura 2.2: FDMA y TDMA, en A) se muestra la división del ancho de banda; la gráfica muestra potencia vs frecuencia, cada  $f_{ci}$  es la frecuencia a la que debe transmitir cada nodo  $i$  sin que interfiera con otro. En B) se ranura el tiempo, y cada estación puede transmitir durante un periodo fijo de tiempo; se muestran  $N$  estaciones en el eje  $t$ .

La forma más sencilla de emplear este método, es asignar una ranura de igual tamaño para cada estación; sin embargo, existe una generalización del TDMA, en el cual la asignación de la ranura se realiza en función de los requerimientos de la estación. Así una estación puede tener más de una ranura en un ciclo, y éstas podrán ser contiguas o espaciadas.

El TDMA requiere de cierta sincronización y coordinación, ya que cada estación debe estar "consciente" del momento a transmitir y durante cuánto tiempo. En general se hace un mejor uso del canal con la generalización descrita del TDMA [19], ya que las estaciones aprovechan las ranuras de tiempo ociosas.

### 2.2.3 ALOHA puro y ALOHA ranurado

El protocolo ALOHA tuvo su nacimiento en la Universidad de Hawaii, ante la necesidad de conectar varias terminales que estaban situadas en varias islas con la estación central de la universidad. Este protocolo es el más sencillo de todos los de contienda. Existen muchas variantes de este protocolo, pero los cambios resultan ser pequeños y tienen la finalidad de mejorar un poco el desempeño al

cambiar la política de retransmisión.

Se considera que la publicación de Abramson [1], marcó el nacimiento de las comunicaciones de acceso aleatorio; sin embargo, durante los siguientes años los algoritmos de acceso aleatorio apenas progresaron. Fue hasta 1977 que en su tesis doctoral, J. Capetanakis presentó el algoritmo de árbol binario [5], el cual es considerado como el primer protocolo de resolución de colisiones.

El funcionamiento del algoritmo ALOHA puro es el siguiente: cada estación transmite su información en forma de paquetes (en general la información se divide en segmentos o paquetes de longitud fija o variable); cuando una estación tiene un paquete por transmitir, se dice que ha *arribado un paquete* o simplemente hay un *arribo*, y sin importar si el canal está libre u ocupado, la estación transmitirá el paquete; existe una estación central que envía un mensaje de realimentación para informar lo que ha sucedido con la transmisión del paquete. Si la transmisión no es exitosa, la estación transmisora sabrá que hubo una colisión y retransmite el paquete después de cierto tiempo aleatorio. El protocolo funciona bien cuando el tráfico es pequeño, pero cuando éste aumenta, la razón de paquetes por unidad de tiempo que acepta el sistema (*caudal*), tiende a decrecer de forma exponencial negativa, esto en el caso de que la creación y retransmisión de paquetes siga una distribución tipo Poisson [19]. Otro problema es la elección del parámetro de retrasmisión, el cual define la longitud del tiempo aleatorio antes de intentar la retransmisión, un valor muy grande aumenta el retardo, mientras que un valor pequeño puede crear inestabilidad.

Existe una variante llamada ALOHA ranurado cuya principal diferencia con el ALOHA puro es que éste ranura el tiempo<sup>2</sup>, por lo que es necesario cierta sincronización por parte de los nodos involucrados; así, cada nodo sólo puede transmitir al inicio de una ranura y el tiempo de espera antes de volver a transmitir es proporcional a la longitud del tiempo de transmisión de una ranura. Por el contrario, ALOHA puro no ranura el tiempo por lo que las estaciones pueden transmitir en cualquier instante.

---

<sup>2</sup>Generalmente la longitud de una ranura de tiempo es igual al tiempo que toma transmitir un paquete.

#### 2.2.4 Acceso múltiple por detección de portadora (CSMA, *Carrier Sense Multiple Access*)

Gran parte de la razón del bajo rendimiento de los protocolos ALOHA, se debe a que intentan transmitir cuando el canal se encuentra ocupado. El protocolo CSMA sugiere una ligera modificación, la cual consiste en revisar el canal antes de transmitir. De esta forma, cada estación es responsable de determinar el estado del medio de transmisión antes de intentar transmitir.

Existen 3 esquemas bajo los que trabaja el protocolo CSMA, éstos son:

- 1-Persistente. Cuando una estación desea transmitir, sondea el canal de manera continua hasta que determina que éste está libre. La estación transmite en cuanto detecta que el canal está libre. Si hay más estaciones en el mismo proceso, con toda seguridad ocurrirá una colisión. Cuando la colisión sucede, las estaciones esperan un tiempo aleatorio antes de volver a sensar el canal e intentar transmitir, ver la fig. 2.3.
- No-Persistente. A diferencia del esquema anterior, si detectan que el canal está ocupado seleccionan un tiempo de espera aleatorio antes de revisar de nuevo el canal, ver fig. 2.4; generalmente este tiempo de espera corresponde al tiempo de propagación. Las colisiones son resueltas de la misma manera que con CSMA 1-persistente; es decir, las estaciones esperan un tiempo aleatorio antes de volver a sensar el canal.
- p-Persistente. Este esquema es algo parecido a la unión de los anteriores. La revisión del canal se realiza de forma persistente (como en el 1-Persistente), pero si el canal está libre, intentará transmitir con probabilidad  $p$ , de otra forma difiere el intento de transmisión con probabilidad  $1 - p$ , ver fig. 2.5. Este esquema sólo aplica cuando el tiempo se ha ranurado. De igual manera cuando hay una colisión, esperan un tiempo aleatorio antes de volver a sensar.

En el algoritmo CSMA la propagación de la señal juega un papel importante para su funcionamiento, ya que cuando las estaciones están transmitiendo, sondan el canal sólo por un determinado intervalo de tiempo; este intervalo corresponde al doble de tiempo de propagación de la señal ( $\tau$ ). Por ejemplo, suponga que hay dos estaciones  $A$  y  $B$  en los extremos del sistema de comunicación, y que la estación  $A$  tiene un paquete por transmitir, por lo que el primer paso es sondear el canal

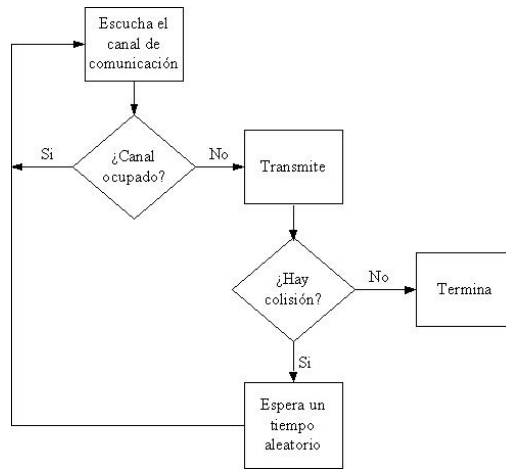


Figura 2.3: Algoritmo CSMA 1-persistente.

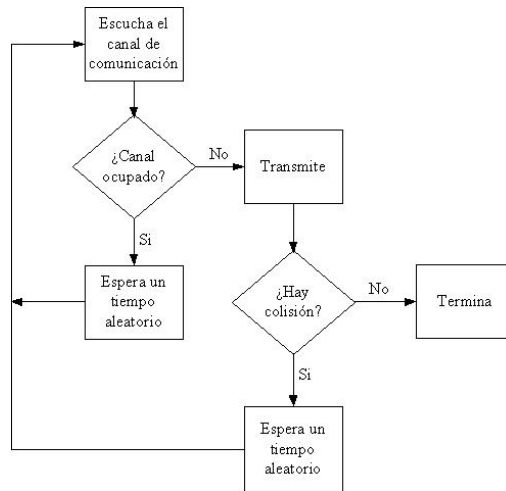


Figura 2.4: Algoritmo CSMA no persistente.

y si está libre, iniciar la transmisión. Suponga que el canal está libre y la estación  $A$  inicia su transmisión, ver fig. 2.6a. La estación  $B$  no detectara la transmisión de  $A$  hasta que ésta se haya propagado un tiempo  $\tau$ , por lo que  $B$  puede iniciar su transmisión al sondear el canal y encontrarlo libre, ver fig. 2.6b. La colisión sucede y estación  $B$  es quien se entera casi de inmediato de ello, en cambio a  $A$  le tomara un tiempo casi igual a  $2\tau$  enterarse de ello. Debido a esto, las estaciones que estan transmitiendo su información, deben sondear el canal durante un tiempo igual a  $2\tau$ , para asegurarse de que su transmisión se haya realizado con éxito.

Si las estaciones que transmiten no sondean el canal durante un tiempo  $2\tau$  (no lo hacen o lo

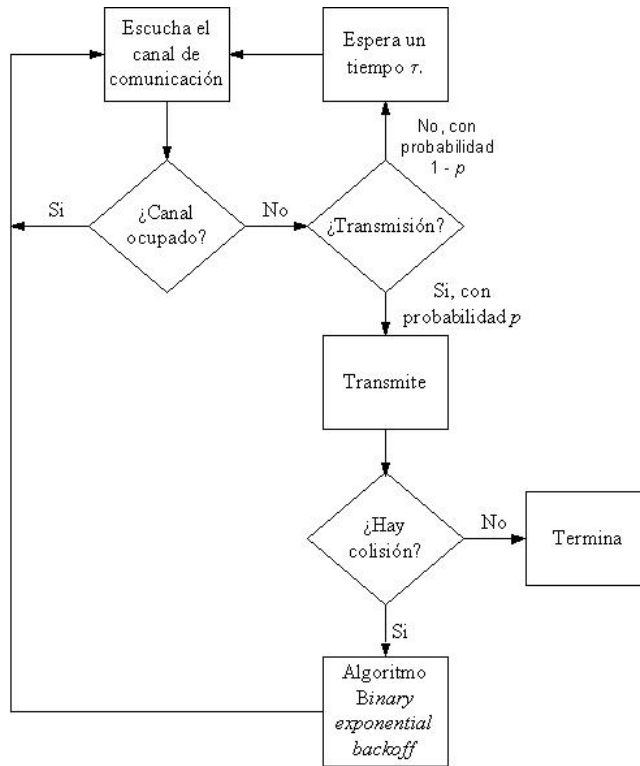


Figura 2.5: Algoritmo CSMA p-persistente.

hacen un tiempo menor a  $2\tau$ ), entonces no podran saber si hay una colisión en el canal, por lo que pueden asumir erroneamente que se transmisión fue exitosa. Este tiempo de sondeo fija una longitud mínima de tamaño para cada paquete que se va transmitir, por ejemplo, el estandar 802.3 define una longitud máxima de 2500m y cuatro repetidoras, en donde el tiempo de propagación es de  $\tau = 25.6\mu s$ , a una velocidad de 10Mbps; para este caso, la longitud mínima del paquete a transmitir es de 64 bytes debido a que en  $\tau$  se mantienen en transito 256 bits (este valor es lo que se conoce como *producto retardo-ancho de banda*) [21].

Existe una variante de CSMA con detección de portadora (CSMA/CD, *Carrier Sense Multiple Access with Collision Detection*), en donde las estaciones interrumpen la transmisión cuando se detecta una colisión. Por ejemplo en CSMA, dos o más estaciones pueden empezar a trasmitir antes de que una o varias se hayan dado cuenta de que existe una transmisión en progreso (desde luego, si detectan que otra estación está transmitiendo esperan), para el caso de CSMA/CD cuando las estaciones se dan cuenta de que ocurrió una colisión, detienen de inmediato la transmisión, y envían

a las demás estaciones información de lo ocurrido.

Cuando los paquetes colisionan, las estaciones inician un algoritmo llamado *binary exponential backoff* [21], el cual coloca a las estaciones en estado de espera antes de volver a sondear el canal y transmitir de nuevo. El funcionamiento es el siguiente:

En este algoritmo el tiempo se divide en ranuras de longitud igual al doble del tiempo de propagación. Después de la colisión las estaciones involucradas en la colisión seleccionan de forma aleatoria un número entre 0 y  $2^i - 1$ , en donde  $i$  representa el número de intentos. Este número aleatorio representa el número de ranuras que la estación debe esperar antes de sondear el canal y transmitir; así, en la primera colisión  $i = 1$ , y de ocurrir otra colisión incrementan una unidad a  $i$ , de esta forma continúan hasta que alguna estación haya podido completar con éxito su transmisión. El proceso no puede continuar de forma infinita, ya que sólo se permiten hasta 16 intentos, después de lo cual se informa que no es posible transmitir; además cuando  $i > 10$ , la selección del número aleatorio se hace entre 0 y 1023.

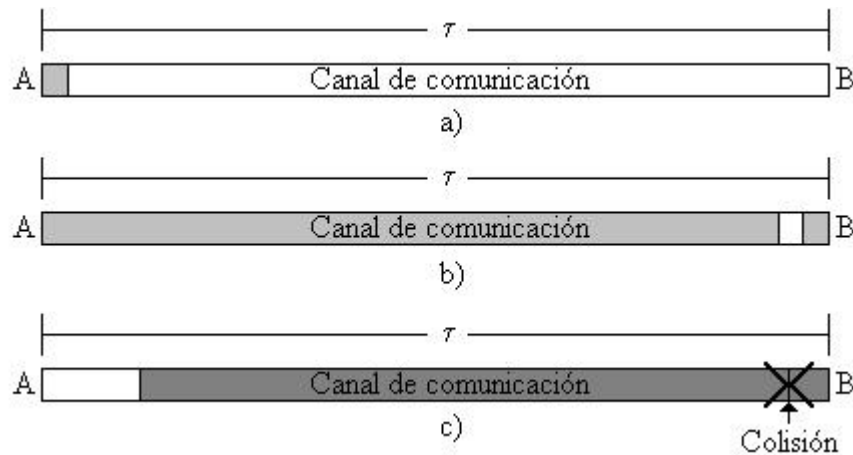


Figura 2.6: En a) la estación A inicia su transmisión y alcanzara a B hasta después de un tiempo de propagación  $\tau$ . En b) la estación B inicia su transmisión sin saber que A lleva transmitiendo casi un tiempo  $\tau$ . Cuando la colisión sucede, ver c), la estación B se entera de forma casi inmediata, en cambio A no se entera hasta después de un tiempo casi igual a  $2\tau$ .

### 2.2.5 Algoritmo de árbol (*Tree Algorithm*)

Este algoritmo fue propuesto por Capetankis [6], y al mismo tiempo por Tsybakov y Mikhailov [23] y por Hayes [10] (también se le conoce como *algoritmo de árbol binario*). El tiempo se asume ranurado, por lo que las estaciones requieren de sincronización para transmitir al inicio de cada ranura; existe una estación central encargada de recibir la transmisión de las demás estaciones y realimentarlas al final de la ranura.

Los nodos de la red se organizan de la siguiente forma. Sea  $T_{ij}$  un subárbol cuyo nodo raíz es  $n_{ij}$ , y sea  $n_{xy}$  uno de los  $y \in \{2^x\}$  nodos a la profundidad  $x$  del árbol, con  $n_{00}$  como nodo raíz. Por ejemplo, el nodo  $n_{20}$ , corresponde al nodo 0 de los  $2^2 = 4$  nodos en la profundidad 2. Cada estación corresponde a una hoja de un árbol binario, y cada estación posee una dirección binaria según el número de hojas en el árbol. En la fig. 2.7 se muestra un árbol binario, en donde las direcciones de cada nodo son de 4 bits, debido a que hay en total 16 hojas. El funcionamiento de este algoritmo es el siguiente:

1. Suponga que no han sucedido colisiones hasta este momento. Sea  $T_{11}$  y  $T_{10}$  dos subárboles.
2. En el siguiente par de ranuras se realizará la transmisión de la siguiente manera: en la primera ranura transmitirán sólo las estaciones asociadas con el subárbol  $T_{10}$  (derecho), mientras que en la segunda ranura lo harán las estaciones del subárbol  $T_{11}$  (izquierdo). Si existe alguna colisión en alguna de las ranuras, entonces:
  - Hasta que la colisión no haya sido resuelta los nuevos paquetes que lleguen no podrán ser transmitidos.
  - Se debe resolver la primera colisión de la primera ranura, antes de resolver la segunda.

Una colisión en  $T_{10}$  (o  $T_{11}$ ) es resuelta dividiendo  $T_{ij}$  en dos subárboles y repitiendo el paso 2. Observe que esto es recurrente, un grupo de paquetes se divide en dos subconjuntos, y cada subconjunto generado se rige por las mismas reglas iniciales.

El algoritmo es muy sencillo pero requiere de una gran coordinación por parte de las estaciones, ya que es necesario que éstas construyan su propio árbol por medio de las realimentaciones (lo cual

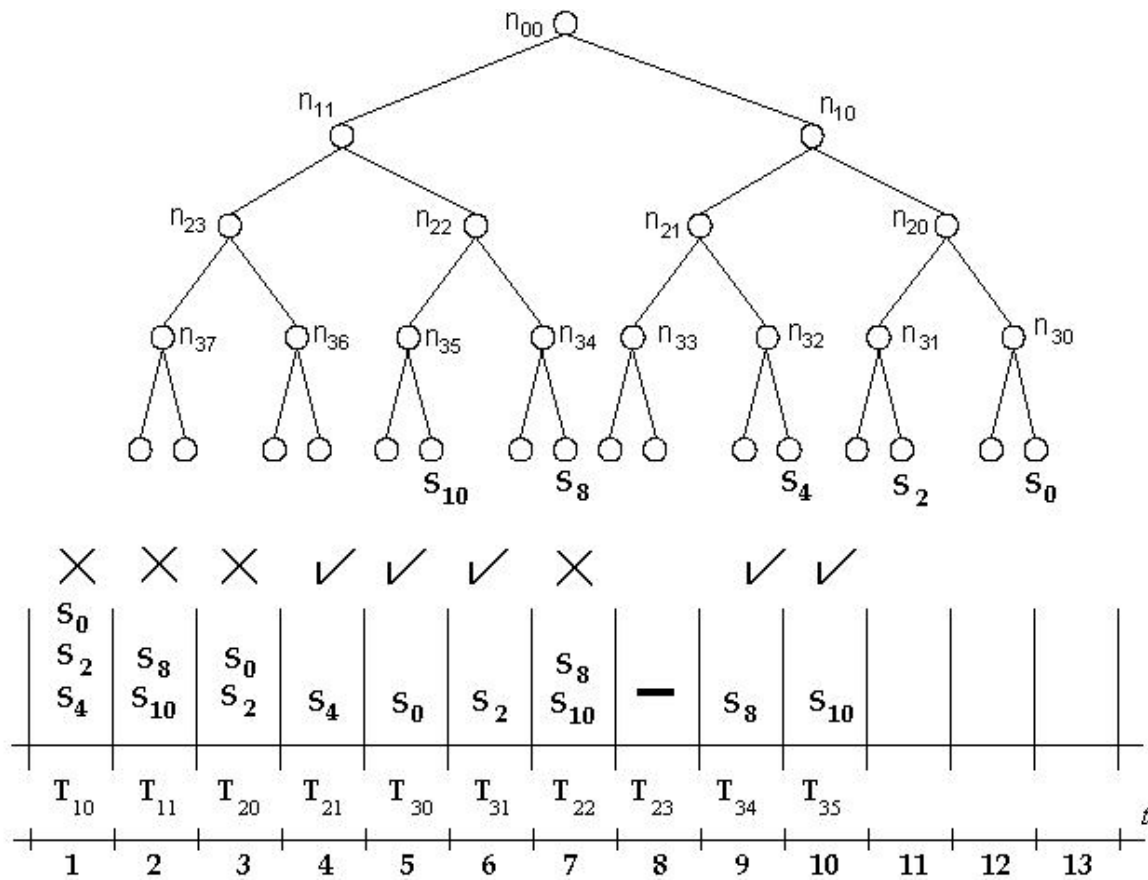


Figura 2.7: Árbol binario de profundidad 4. Control de acceso al medio mediante el algoritmo de árbol.

no es adecuado cuando pueden suceder errores en la realimentación); además cada estación deberá conocer su dirección, para saber su posición en el árbol.

Para entender un poco más el algoritmo, se discute el siguiente ejemplo que fue tomado de [6]. En base a la fig. 2.7, suponga que tienen paquetes por transmitir las estaciones  $S_0$ ,  $S_2$ ,  $S_4$ ,  $S_8$  y  $S_{10}$ . En la primera ranura transmitirán  $S_0$ ,  $S_2$  y  $S_4$  del subárbol  $T_{10}$ , ya que la transmisión inicia con el subárbol derecho; mientras que en la segunda ranura, lo harán  $S_8$  y  $S_{10}$  del subárbol  $T_{11}$ . Obviamente en la primera ranura ocurrirá una colisión, que deberá ser resuelta a partir de la ranura 3, como se muestra en la fig. 2.7. Las estaciones  $S_8$  y  $S_{10}$ , deberán estar conscientes de esto, y esto se logra con la realimentación. En la ranura 3 el subárbol  $T_{10}$  será dividido en los subárboles  $T_{20}$  y  $T_{21}$ , lo que produce que transmitan  $S_0$  y  $S_2$  en la ranura 3, mientras que en la ranura 4 lo hará  $S_4$ , ver fig. 2.7.

Observe que la transmisión en la ranura 4 fue un éxito, mientras que en la ranura 3 ha ocurrido una colisión, por lo que el subárbol  $T_{20}$  tendrá que ser dividido en los subárboles  $T_{30}$  y  $T_{31}$  a partir de la ranura 5. Con esta nueva división, se logra que las estaciones  $S_0$  y  $S_2$ , transmitan su información con éxito en las ranuras 5 y 6 respectivamente. En la ranura 7 toca el turno para el subárbol  $T_{11}$ , el cual debido a que tuvo una colisión en la ranura 2, será dividido en el par de subárboles  $T_{23}$  y  $T_{22}$ . El algoritmo continúa hasta terminar con éxito en la ranura 10.

Note que el algoritmo tiene una cierta desventaja, ya que la división provoca que puedan existir ranuras vacías, lo cual se considera un desperdicio de recursos. El análisis de este protocolo, sugiere que el retardo promedio aumenta conforme aumenta el número de estaciones que intenta transmitir.

Existen versiones mejoradas de este protocolo, por ejemplo J. Massey [12] propone la misma evolución del árbol binario, excepto que trata de eliminar las colisiones evitables. La idea es la siguiente: si en alguna ranura hay una colisión, ésta será dividida en dos ranuras; si en la primera ranura hay un vacío, es obvio que en la segunda habrá una colisión, por lo que la idea plantea dividir esta ranura antes de que suceda la colisión. Sin embargo, se puede ver que existen problemas en esta propuesta; por ejemplo el algoritmo debe entonces reconocer los "vacíos", y ante un error en la realimentación será propenso a fallas.

### 2.2.6 Protocolo de pila (*Stack Protocol*)

Este algoritmo fue propuesto por Tsybakov et al. [24]. El funcionamiento del algoritmo es el siguiente (de igual manera el tiempo está ranurado, sólo es posible transmitir al inicio de cada ranura y existe una estación central encargada de proporcionar los mensajes de realimentación); suponga que a un tiempo  $t$  se han resuelto todas las colisiones y que cada estación maneja una variable llamada *contador de pila* ( $SC$ , *Stack Counter*), entonces:

1. Cuando las estaciones tienen paquetes por transmitir, éstos siempre se envían inmediatamente. Al primer intento de transmisión del paquete, cada estación fija su contador de pila  $SC = 0$ ; básicamente si  $SC = 0$  la estación transmitirá en la ranura de tiempo. Si hay más de dos estaciones involucradas, habrá una colisión; de otra forma la transmisión se habrá llevado con

éxito.

2. Suponiendo que hay una colisión, la estación central sabe que esto ha ocurrido, pero desconoce el número de estaciones involucradas, por lo que mediante un mensaje de realimentación, informa a las estaciones que ha sucedido una colisión.
3. Cada estación sabe que ocurrió una colisión, por lo que realizarán un experimento aleatorio de tipo Bernoulli: con probabilidad  $p$  incrementan  $SC$  a 1 y no intentarán transmitir en la siguiente ranura; por el contrario, con probabilidad  $1 - p$  fijan  $SC = 0$  y transmitirán en la siguiente ranura.
4. Si al intentar de nuevo la retransmisión hay una colisión, se repite el paso 3 para las que colisionaron; pero para las que no transmitieron, incrementan  $SC$  en 1.
5. Si por el contrario, la transmisión es exitosa, todas las estaciones restantes no modifican  $SC$  a menos que la realimentación sea una ranura vacía; en ese caso, decrementan  $SC$  en 1.
6. Cada estación es libre de entrar en la competencia por el medio, siempre y cuando tenga un paquete por transmitir; pero puede ocurrir que se genere una colisión. En este caso, el algoritmo continúa con una nueva estación en la contienda.

Observe que el protocolo es muy semejante al algoritmo de árbol; sin embargo, la esencia de este algoritmo, es que explota la realimentación. El propósito es que todas las estaciones transmitan su paquete después de una colisión, tan pronto y sea posible.

Observe que el tipo de realimentación empleada en el algoritmo es ternaria, de tal suerte que sólo 3 realimentaciones son válidas: *colisión*, *vacío* y *éxito*. Existen otros algoritmos cuya operación se basa en una realimentación binaria, por ejemplo *colisión / éxito*, por lo que *no transmisión* se puede asociar a alguno de los dos eventos antes mencionados, generalmente con un *éxito*.

Algo que es interesante mencionar, es que este algoritmo no es equitativo debido a que una nueva estación con un paquete puede intentar transmitir en cualquier instante. Se puede dar el caso de que a algunas estaciones les tome más tiempo transmitir su paquete, debido a que estaciones más "afortunadas" pudieron hacerlo, aun cuando éstas hayan intentado hacerlo después.

### 2.2.7 Algoritmo de las dos celdas (2C, *Two-Cell Algorithm*)

Al igual que los dos algoritmos anteriores, el algoritmo 2C también trabaja sobre un tiempo ranurado, por tanto debe existir sincronización y una estación central encargada de realimentar a las demás estaciones. El algoritmo 2C guarda una cercana relación con el protocolo de pila de Tsybakov, pero en este caso la pila sólo puede crecer hasta dos niveles; es decir, el algoritmo 2C puede ser modelado por dos celdas (de una pila), una que representa las estaciones en espera (WC, *Waiting Cell*) y otra celda que representa las estaciones en transmisión (TC, *Transmission Cell*). El algoritmo 2C es un caso particular del algoritmo llamado *Z-Cell* [11, 14], en donde la  $Z$  define tamaño de la pila o número de celdas. Sin embargo, el algoritmo 2C fue mencionado en [25] y estudiado a detalle en [15].

A diferencia del protocolo de pila, el algoritmo 2C posee una realimentación binaria ( $C$  colisión, y  $NC$  no colisión), y es de *detección completa*; es decir, las estaciones tienen que sondear constantemente el canal. Cada estación posee un contador, al que llamaremos  $r_i$  (el contador de la estación en la ranura de tiempo  $i$ , y sólo puede tomar los valores 0 o 1). Al final de cada ranura, la estación central envía a las estaciones una realimentación  $x_t$  de lo sucedido en la ranura  $t$ , ( $x_t = C$  representa una colisión y  $x_t = NC$  no colisión). Las estaciones sólo pueden transmitir al inicio de una ranura.

Para explicar el algoritmo suponga que el tiempo se ranura en intervalos de longitud unitaria, en donde el intervalo de la ranura de tiempo  $t$  se representa por  $(t, t + 1]$ , donde  $t \in \mathbb{N}$ . Suponga además que al tiempo  $t_1$ , con  $t_1 < t$ , se han transmitido con éxito todos los paquetes de las estaciones (el intervalo  $(0, t_1]$  se le *denomina intervalo resuelto*). En la ranura  $t$  sólo tomarán parte en la competencia por el medio aquellas estaciones que tengan paquetes por transmitir. Si sólo una estación posee un paquete, éste será transmitido en la ranura  $t$  con éxito. Si hay más estaciones con un paquete, habrá una colisión y su resolución empezará en la ranura de tiempo  $t + 1$ . A este intervalo de tiempo, que inicia en  $t + 1$  y que termina con dos transmisiones exitosas consecutivas, se le conoce como *intervalo de resolución de colisión* (CRI, *Collision Resolution Interval*). Después de la ranura de tiempo  $t$  (una vez que inicia el CRI) no se permite que ninguna otra estación intente transmitir, hasta que se termine el CRI en curso; es decir, es un *algoritmo de acceso bloqueado*<sup>3</sup>.

---

<sup>3</sup>En cambio, cuando se permite que se integren nuevas estaciones aun cuando no se han resuelto las colisiones, se dice que es un *algoritmo de acceso libre* (*Free-Access Algorithm*), un ejemplo de ello es el protocolo de pila [17].

Cuando una estación transmite por primera vez, fija su contador  $r_i = 0$ , y éste se modifica como sigue:

1. La estación transmite al inicio de la ranura  $t$ , si y sólo si,  $r_t = 0$ . Un paquete es transmitido con éxito en  $t$ , sólo si  $r_t = 0$  y  $x_t = NC$ .
2. El contador de cada estación evoluciona en el tiempo como sigue:
  - Si  $x_{t-1} = NC$  y  $r_{t-1} = 1$ , entonces  $r_t = 0$ .
  - Si  $x_{t-1} = C$  y  $r_{t-1} = 1$ , entonces  $r_t = 1$ .
  - Si  $x_{t-1} = C$  y  $r_{t-1} = 0$ , entonces  $r_t = 0$  con probabilidad  $p$  y  $r_t = 1$  con probabilidad  $1 - p$ .

Al final de la ranura la estación central realimenta con  $C$  cuando han colisionado dos o más paquetes, de otra forma emite un  $NC$ . Con este esquema, los vacíos se interpretan como éxitos. A diferencia del protocolo de pila, este algoritmo muestra calidad de servicio en la asignación del canal, debido a que sólo las estaciones que inician un CRI pueden contender por el canal, si alguna otra estación desea utilizar el canal, deberá esperar a que termine el CRI en curso.

Se ha mencionado que el algoritmo en su versión original es de detección completa, lo cual ofrece una gran desventaja en dispositivos que intentan consumir poca energía. El algoritmo 2C es apreciado no sólo por su capacidad de operar en detección completa, sino por que es capaz de operar en *detección limitada*; por lo que en este trabajo se asume que el algoritmo es de detección limitada; es decir, las estaciones sólo sondean el canal cuando necesitan transmitir un paquete. Así, cuando una estación desea transmitir un paquete, ésta deberá primero sincronizarse al recibir dos realimentaciones  $NC$ , ya que esto indica dos cosas: un CRI ha terminado o que el canal está libre.

El algoritmo original emplea una probabilidad  $p = \frac{1}{2}$ , lo cual genera que un CRI no converja rápidamente y los retardos de transmisión de cada nodo se eleven cuando el número de estaciones es grande. Por tanto, es conveniente que sólo algunas estaciones inicien el CRI; de hecho, en [15] se introduce un intervalo de tiempo de captura de estaciones<sup>4</sup>, el cual está en función del tamaño

---

<sup>4</sup>En la siguiente sección se darán más detalles de esto.

del CRI y se define de tal forma que sólo pocas estaciones inicien un CRI con la finalidad de que éste sea corto. Es de suponer que la longitud del CRI depende del número de estaciones que lo inician; así, la propuesta presentada en este trabajo está enfocada a determinar la probabilidad  $p$  como función del número de estaciones que colisionó al inicio o durante la evolución de un CRI. El número de estaciones que colisionó se le conoce como *multiplicidad de la colisión* y lo representamos por la variable  $N$ .

En esta propuesta es necesario que las estaciones conozcan esa multiplicidad de la colisión para así determinar el valor óptimo para  $p$ . Por lo pronto, se asume que por algún mecanismo puede conocerse esta multiplicidad<sup>5</sup>, y nos limitaremos a suponer que sólo la estación central puede predecir (por lo pronto de manera precisa) este valor. No es conveniente que las demás estaciones intenten hacer la estimación de este valor, principalmente por dos razones:

1. Aun con la realimentación las estaciones no conocen de manera precisa la evolución del CRI, debido a que no pueden discernir entre una transmisión exitosa y un vacío (cuando ninguna estación transmitió). Por lo tanto, es adecuado que la estación central ejecute el mecanismo de predicción.
2. Una estación central puede tener más recursos para ejecutar un algoritmo de predicción.

Si se establece que la estación central realiza la estimación, ésta puede realizar este proceso y difundir este resultado usando una de las siguientes dos políticas:

1. En cada ranura. Esto implica que la estación central debe realizar el proceso de predicción al final de la ranura (una vez que las estaciones han enviado sus paquetes), y después difundir este valor a las estaciones en contienda.
2. A intervalos convenientes de tiempo. A diferencia de lo anterior, la estación central debe realizar la predicción después de que transcurre un cierto tiempo.

La primera opción requiere de un mayor tiempo de cómputo por parte de la estación central, ese tiempo de cómputo está en función de la complejidad del proceso de predicción, por lo que es necesario que el algoritmo sea computacionalmente eficiente. En cambio, la segunda opción requiere

---

<sup>5</sup>Más adelante ofreceremos un ejemplo de esto.

un menor tiempo de cómputo, ya que se realiza una predicción después de un cierto número de ranuras. La asignación de este intervalo de tiempo, puede ser fijo o dinámico; sin embargo, tendría que ser dinámico debido a que un intervalo de tiempo fijo puede representar un desperdicio de tiempo de cómputo al hacer predicciones en ranuras que no ofrecerán información relevante<sup>6</sup>. Más adelante se observará que la longitud de este intervalo de tiempo, estará dado por la longitud del CRI; es decir, sólo resulta necesario determinar la multiplicidad de la colisión al inicio de un CRI.

## 2.3 Estado del conocimiento

El algoritmo 2C es considerado uno de los más robustos en presencia de errores en el canal [13], y es apreciado por su capacidad de operar en detección limitada. El principal estudio que se le ha realizado se desarrolla en [15]; en donde el análisis se enfoca en determinar el tamaño de la *ventana de captura* de tal forma que el funcionamiento del algoritmo sea estable<sup>7</sup>.

La ventana de captura se emplea en una gran cantidad de algoritmos MAC, incluso existe una versión del algoritmo de árbol en donde se emplea la ventana de captura; esta ventana permite definir un intervalo de tiempo en el cual se captan las estaciones que competirán en el siguiente CRI; así, el tiempo no sólo está ranurado en la duración de transmisión de un paquete, sino también en ventanas de captura, cuya longitud puede ser fija o dinámica.

En [15] el tamaño de ventana determinado es de 2.33 ranuras, para una caudal de 0.4297 paquetes/ranura para tráfico tipo Poisson. Es de esperar que entre más pequeña sea la ventana de captura, menor será el número de estaciones captadas y por tanto menor será el CRI; sin embargo, la ventana de captura no mejora las prestaciones del algoritmo. El empleo de una ventana de captura sólo permite que el número de estaciones al inicio del CRI sea menor. El objetivo de este trabajo no es optimizar la ventana de captura, sino considerar que para minimizar un CRI, éste depende del número de estaciones que lo inician, por lo que si las estaciones saben este valor, podrían elegir el valor óptimo de  $p$  que minimiza un CRI.

---

<sup>6</sup>Por ejemplo, ranuras en las ninguna o sólo una estación transmitió.

<sup>7</sup>El algoritmo es estable cuando el número de paquetes que se generan para ser transmitidos, es menor a la capacidad del sistema para transmitirlos.

Existen pocas referencias acerca de mejoras a los algoritmos de acceso al medio en función del número de estaciones que colisionan, y menos aun, para el algoritmo 2C. En esta sección describimos algunas de estas propuestas que son aplicadas a alguno de los algoritmos descritos con anterioridad.

Una primera propuesta se ofrece en [8], en donde se propone un algoritmo que estima el número de estaciones que colisionan, para ser implementado en el algoritmo de árbol, esta implementación da como resultado un *algoritmo híbrido* para la resolución de colisiones. El algoritmo presentado en [8] es el siguiente: suponga que colisionan  $n$  estaciones (que es el número a determinar), las estaciones sabrán de la colisión, éxito o vacío (ninguna estación transmitió) por medio de una realimentación; la estrategia es buscar una potencia de 2 con una alta probabilidad de estar lo más cercana de  $n$ . Cada estación involucrada deberá entonces "lanzar una moneda" y con probabilidad  $2^{-i}$  transmitirá, mientras que con probabilidad  $1 - 2^{-i}$  no transmitirá. El algoritmo inicia con el contador  $i = 1$  en cada estación, y realizan el evento de lanzar la moneda para transmitir o no en cada ranura. Después de cada evento se incrementa el contador, y el algoritmo termina cuando no hay colisión. Finalmente, las estaciones sabrán que  $2^i$  es el valor estimado de  $n$ , que se obtiene de la esperanza de la distribución binomial.

El algoritmo es muy sencillo, y realiza la estimación en base a la esperanza de todo el sistema. Este algoritmo híbrido mantiene un 10% mayor grado de estabilidad, en comparación con el algoritmo de árbol; esto es, la razón de arribos permitida es mayor por lo que el caudal es mayor, pero se obtienen ligeras mejoras en cuanto al retardo se refiere. Sin embargo, observe que el algoritmo de estimación propuesto, debe emplear cierta cantidad de tiempo para poder realizar la estimación; lo cual si bien resulta en una mejora, es necesario entonces emplear algún otro mecanismo de estimación a fin de consumir el menor tiempo posible.

Recordemos que en el algoritmo CSMA ranurado, cuando las estaciones colisionan esperan un tiempo aleatorio para volver a intentar transmitir; este proceso de espera se logra mediante el algoritmo *binary exponential backoff*. Este algoritmo se puede considerar como un algoritmo adaptable ante la presencia de colisiones, debido a que el tiempo de espera cambia en función de la observación

de una colisión, lo cual ocasiona que los tiempos de espera tiendan a crecer de forma exponencial. La siguiente propuesta propone adaptar esta ventana de tiempo de espera en función del número de estaciones en colisión, en lugar de permitir la adaptación en función de la presencia de colisiones.

En [3] se propone una modificación al estándar 802.11, en la ventana de contienda adaptable (*Adaptive Contention Window*) del mecanismo de acceso al medio conocido como DCF (*Distributed Coordination Function*) [18], el cual se basa en algoritmo CSMA/CA con retardo exponencial binario (*Binary Exponential Backoff*). El funcionamiento del DCF es el siguiente: cuando una estación tiene un paquete por transmitir, sensa el canal durante un intervalo de tiempo conocido como DIFS (*Distributed InterFrame Space*). Si el canal está ocupado retarda la transmisión durante un intervalo de tiempo aleatorio (*Backoff Time*), en caso de que el canal se haya encontrado libre al finalizar el DIFS, se inicia la transmisión. La duración de este intervalo de tiempo aleatorio será de al menos un DIFS, y éste se decrementa mientras el canal se encuentre libre (el tiempo está ranurado<sup>8</sup>). Para evitar que alguna estación se apropie del canal, deben esperar un tiempo aleatorio entre dos transmisiones consecutivas, aun si el medio se encuentra libre. Para reducir la ocurrencia de colisiones cuando varias detectan libre el canal durante un DIFS, se implementa el mecanismo de evasión de colisión (CA, *Collision Avoidance*).

El tiempo aleatorio es seleccionado de manera uniforme en el intervalo  $(0, CW - 1)$  que se define como *ventana de retardo* (*Backoff Window*). En el primer intento de transmisión  $CW = CW_{min}$  ( $CW$ , *Contention Window*), y es duplicado en cada retransmisión hasta  $CW_{max}$  (la estación destino debe enviar un reconocimiento para indicar si el paquete fue recibido); el estándar define  $CW_{min} = 32$  y  $CW_{max} = 256$  [18].

El análisis de rendimiento de CSMA/CA en [3], muestra que para una óptima operación los parámetros del sistema deben ser seleccionados apropiadamente de acuerdo a las condiciones de tráfico, principalmente el parámetro  $CW_{min}$  que depende del número de estaciones en contienda. Lo anterior sugiere que una mejora en CSMA/CA recae en la selección de  $CW$  como función del número estimado de estaciones en contienda, dicha predicción se obtiene en base a medidas de

---

<sup>8</sup>La longitud de la ranura es igual al tiempo de propagación.

actividad del canal; es decir, en función del número de ranuras ocupadas en un cierto periodo de tiempo (que en un principio es igual al intervalo de retardo). La predicción del número de estaciones en contienda asume que las estaciones son capaces de conocer en todo momento el estado del canal; además de que es necesaria la implementación de un filtro lineal a fin de obtener mejores resultados en la predicción del número de estaciones en contienda.

Sin embargo, el algoritmo DCF posee un problema en medios inalámbricos conocido como *estación oculta*, ya que las estaciones son las encargadas de sondear el canal a falta de una estación central, y no todas las estaciones están al alcance, lo que puede ocasionar información errónea al sondear el canal. Para evitar lo anterior, el DCF emplea mecanismos de envío y confirmación de canal (RTS, *Request To Send*/CTS, *Clear To Send*); es decir, cuando una estación necesita transmitir un paquete, envía un paquete de RTS, y espera un CTS de confirmación antes de usar el canal. En general se obtienen buenos resultados cuando el número de estaciones es grande (alrededor de 50), y los resultados son ligeramente mejores cuando se emplean mecanismos de RTS/CTS [3].

Como se puede observar, para optimizar un protocolo MAC sería conveniente conocer la multiplicidad de la colisión. Una vez que se conoce esto, es necesario realizar ciertas modificaciones al algoritmo, a fin de que pueda aprovechar esta información. Por ejemplo en [9] se asume que la multiplicidad de la colisión se conoce en cada ranura, y se centran en determinar que cambios requeriría el algoritmo, en este caso el algoritmo de árbol<sup>9</sup>. En este caso, el algoritmo existe en dos tipos, cuando la probabilidad de transmisión es  $p = \frac{1}{2}$ , y cuando la probabilidad es variable en función de la multiplicidad de la colisión, el valor óptimo determinado es  $p = \frac{2}{m+1}$ , donde  $m$  es la multiplicidad de la colisión. En el primer caso se envía menos información, ya que en lugar de informar la multiplicidad de la colisión, se envía el estado de cada uno de los dos grupos que divide la probabilidad de transmisión  $p$ . Ambos casos no ofrecen mejoras sustanciales entre ellos, pero si con respecto al algoritmo original de Capetanakis.

La mejora al algoritmo 2C asume que es posible aplicar algún algoritmo de predicción para determinar la multiplicidad de la colisión, por lo que resta determinar qué cambios se le harán al

---

<sup>9</sup>Sin embargo, se puede ver mejor desde el punto de vista del algoritmo de pila, ya que en general ambos algoritmos son similares.

algoritmo con la finalidad de mejorar sus prestaciones.

Existen algunas otras propuestas de acceso al medio en donde no es esencial conocer el número de estaciones en contienda; por ejemplo, lo que se conoce como *algoritmos de captura* [4], en donde las estaciones transmiten y la estación central se encarga de decodificar los paquetes que son enviados. En este caso, la estación central debe informar cuál paquete llegó con éxito, pero sólo a la estación correspondiente. La forma de capturar los paquetes puede llevarse de dos formas posibles [7]:

- Con ayuda de receptores múltiples, en donde la estación central puede recibir varios paquetes de forma simultánea.
- Empleando niveles de potencia, en función de la proximidad con la estación central, o por medio de alguna técnica de codificación o multiplexión.

Dado lo anterior, no es difícil imaginarse algunas otras técnicas de acceso al medio; aunque en este caso dichas técnicas están orientadas no a mejorar los protocolos MAC, sino a hacerles el trabajo más fácil. Por ejemplo mediante el empleo de procesamiento de señales; es decir, permitir que la colisión ocurra y recuperar, cuando es posible, estos paquetes mediante algún procesamiento de señales [22], si bien la colisión puede eliminarse, en algunos casos no será posible y será necesario emplear mecanismos para evitarlas o controlarlas. El protocolo MAC y la capa física tendrían que interactuar por medio alguna subcapa, entre las capas de enlace de datos y física.

# Capítulo 3

## El algoritmo 2C-adaptable y su análisis de desempeño

---

La propuesta de mejora del algoritmo 2C, requiere que cada estación conozca la multiplicidad de la colisión  $N$  para así calcular el valor óptimo para la probabilidad  $p$  de transmisión. Esta propuesta la denominamos *algoritmo 2C-adaptable*. En principio cada estación podría ser capaz de realizar la estimación de  $N$ , pero ello resulta poco práctico debido a que en general se emplea información que a la estación central le sería más fácil conocer; así, resulta más adecuado que la estación central sea quien realice la estimación de la multiplicidad de la colisión.

Si se asume que la estación central es capaz de determinar la multiplicidad de la colisión  $N$ , se debe entonces implementar algún mecanismo de difusión de esta información para que el resto de las estaciones pueda conocer  $N$ . Básicamente este mecanismo de difusión recae en la realimentación que proporciona la estación central, por ello se proponen dos métodos descritos a continuación.

El primer método consiste en una *política de realimentación por ranura*, la cual requiere que la estación central estime la multiplicidad de la colisión  $N$  en cada ranura de transmisión (exista o no la colisión); así, al final de cada ranura la estación central se encargaría de informar la multiplicidad de la colisión a todas las estaciones. Note que es necesario que el algoritmo 2C sufra una notable modificación, ya que pasa de una realimentación binaria a una realimentación *N-aria*. Es de esperar que dicha política necesitará un mayor tiempo de cómputo por parte de la estación central al estimar

$N$ , así como en las demás estaciones al calcular  $p = p(N)$  en cada ranura.

El segundo método, el cual se ha denominado *política de realimentación por fase* y que dió como resultado al algoritmo 2C-adaptable, requiere que la estación central estime la multiplicidad de la colisión cada cierto intervalo de tiempo durante el desarrollo del CRI. Con este método, existe el inconveniente de que hay que determinar la longitud de dicho intervalo. Ésta no es una tarea sencilla, pero es posible hacer uso de la realimentación del algoritmo; así, se define como *fase* al número de ranuras que le toma al sistema lograr una transmisión exitosa, partiendo de una colisión de multiplicidad<sup>1</sup>  $M$ . Obviamente una fase siempre inicia después de una transmisión exitosa, ya que de otra forma no tendría sentido la definición; es decir, una realimentación de *No-Colisión* define los límites de una fase. Al definir de esta forma la fase, se permite que la estación central estime la multiplicidad de la colisión al inicio del CRI o al inicio de cada fase, por lo que el algoritmo 2C sufriría ligeros cambios. Se puede proponer otra forma de definir la longitud del ciclo (por ejemplo un número fijo), pero esto puede ocasionar que la estación central realice la estimación en ranuras que no son adecuadas, por ejemplo en una ranura en donde no se transmitió.

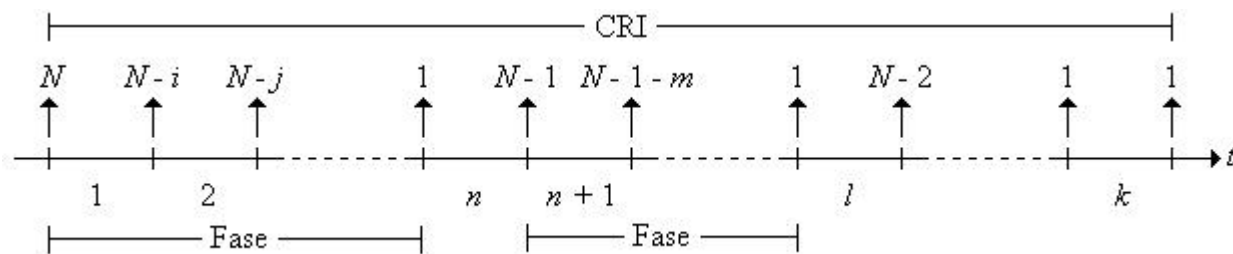


Figura 3.1: CRI de multiplicidad de la colisión  $N$ , las flechas indican el número de estaciones que transmitieron al finalizar la ranura de tiempo. En la ranura 1,  $N - i$  estaciones colisionaron, en la ranura 2 colisionaron  $N - j$  (con  $j > i$ ), eventualmente una estación será la que transmita (o ninguna), lo cual determina el fin de una fase, en este caso la fase tuvo una longitud de  $n - 1$  ranuras. En la ranura  $n$  intentaran transmitir  $N - 1$  estaciones (o  $N$  si en la ranura  $n - 1$  ninguna estación transmitió), una fase dará inicio y terminara en la ranura  $l - 1$ . Observe que al finalizar cada fase, el número de estaciones en contienda se reduce en una unidad (o en todo caso permanece igual), el CRI evoluciona hasta que este termina con dos transmisiones exitosas consecutivas hasta la ranura  $k$ ; un CRI se compone de al menos  $N - 1$  fases.

En términos generales, un CRI se compone de fases y éstas a su vez se componen de ranuras

<sup>1</sup>Es importante hacer notar que un CRI inicia con una colisión de multiplicidad  $N$ , mientras que la fase inicia con una colisión de multiplicidad  $M$ , en donde  $M \leq N$ .

de tiempo. En la Fig. 3.1 se muestra un CRI, note que éste inicia con una colisión de multiplicidad  $N$  y finaliza con dos éxitos consecutivos hasta la ranura  $k$ ; al igual que el CRI, la fase también inicia con una colisión de multiplicidad mayor o igual a 2, pero finaliza con un éxito de acuerdo a la realimentación proporcionada por la estación central<sup>2</sup>. En el algoritmo 2C el tiempo se encuentra ranurado, por lo cual la ranura se considera como la unidad mínima de tiempo.

Lo que resta del capítulo, se presenta el análisis de mejora y resultados de simulación para ambas propuestas asumiendo que la estación central puede determinar la multiplicidad de la colisión  $N$  de forma exacta; sin embargo, también se dan detalles de lo que ocurre si se predice la multiplicidad de la colisión, que es lo que en la realidad sucedería. Con esta información cada estación puede calcular el valor óptimo de la probabilidad de transmisión  $p$  como función de la multiplicidad de la colisión  $N$ . Se asume que no existen errores en el canal, y que la realimentación se produce de acuerdo a las políticas mencionadas.

### 3.1 Política de realimentación de la multiplicidad de la colisión por ranura

#### 3.1.1 Análisis

En el algoritmo 2C original cada ranura puede tener dos posibles resultados: *NC-No colisión* o *C-Colisión*, en donde la realimentación  $C$  sólo aparece cuando 2 o más estaciones transmiten al mismo tiempo. Cuando sucede una colisión, las estaciones involucradas se dividen dos celdas o grupos: una celda de estaciones en transmisión (TC) y otra celda de estaciones en espera (WC). En este caso, supongamos que la realimentación es *N-aria*; es decir, al final de la ranura la estación central retorna el número de estaciones que transmitieron en la ranura a través del mismo canal de comunicación, la realimentación puede ser  $0, 1, \dots$ , o  $N$  (0 y 1 indican no transmisión y éxito respectivamente, y  $N$  es la multiplicidad de la colisión).

Para iniciar el análisis, suponga que en la ranura  $T$  transmitieron  $N$  estaciones<sup>3</sup> ( $N \geq 2$ ,  $N \in \mathbb{N}$ ).

<sup>2</sup>Note que si ninguna estación transmite, se considera un éxito y por lo tanto el final de la fase.

<sup>3</sup>Sin pérdida de generalidad, se puede asumir que no existen estaciones en espera, ello se debe a que aun cuando están activas, no participan en la contienda por el canal de transmisión.

La estación central informará de la colisión al final de la ranura  $T$ , y cada estación realizará un experimento de tipo Bernoulli; después del cual irá a la celda de transmisión con probabilidad  $p$  o a la celda de espera con probabilidad  $1 - p$ , por lo que en en la ranura  $T + 1$ ,  $N - i$  estaciones estarán en la celda de transmisión, e  $i$  en la celda de espera, con  $0 \leq i \leq N$ , e  $i \in \mathbb{N}$ . Al final de la ranura  $T + 1$ , la estación central informará a todas las  $N$  estaciones en contienda, que  $N - i$  han colisionado. El proceso continúa hasta que se logre una transmisión con éxito, el cual sucede después de  $k$  ranuras; es decir, cuando  $i = N - 1$ , ver fig. 3.2.



Figura 3.2: Política de realimentación por ranura.  $k$  representa el número de ranuras que le toma al sistema llegar a una transmisión con éxito después de una colisión de multiplicidad  $N$ . La estación central informa al final de cada ranura (representado por la flechas verticales), el número de estaciones en la celda de transmisión (que representa la multiplicidad de la colisión).

Es importante notar que en cada ranura, las estaciones buscarán el valor óptimo de  $p$  como función de  $N$  que permita minimizar el número de ranuras para lograr una transmisión exitosa; desde el punto de vista de la fig. 3.2, se intenta minimizar el valor de  $k$ .

Definimos el estado del sistema<sup>4</sup> como el número de estaciones que transmite en una ranura de tiempo (que es igual al número de estaciones que permanecen en la celda de transmisión o a la multiplicidad de la colisión en cada ranura); por lo tanto, el estado deseable en cada ranura es 1, que presenta una estación transmitiendo en cada ranura. Así, se puede modelar la transición entre estados del sistema por medio de una distribución binomial, para lo cual se tiene:

$$P_{N,K}(p) = \binom{N}{K} p^K (1-p)^{N-K}, \quad 0 < p < 1, \quad 0 \leq K \leq N. \quad (3.1)$$

En donde  $p$  es la probabilidad de que la estación transmita o vaya a la celda de transmisión, y  $N$  es el valor de la multiplicidad de la colisión. Note que  $K$  es una variable aleatoria (v.a.) que representa el estado del sistema al final de una ranura de tiempo; así, podemos ver a  $P_{N,K}(p)$  como la proba-

<sup>4</sup>De aquí en adelante nos referiremos al *sistema* como al conjunto de estaciones (incluyendo la estación central), que forma nuestro sistema de comunicación.

bilidad de que  $K$  estaciones se dirijan a la celda de transmisión de un conjunto de  $N$ ; es decir, la ec. 3.1 es la probabilidad de que el sistema transite del estado  $N$  al estado  $K$  ( $K \leq N$ ) en una sola transición o de una ranura a la siguiente.

Obviamente el mejor de los estados al final de una ranura tiempo es 1, ya que implica una transmisión exitosa. Los demás estados permiten una evolución favorable del sistema, salvo el estado 0 que implica un desperdicio de tiempo, ya que a la siguiente ranura, de nuevo  $N$  estaciones competirán por el canal de comunicación. Así, la probabilidad de pasar de un estado a otro está dado por la ec. 3.1; y más aun, no importa el número de estaciones en la celda de espera, sólo es importante tomar el número de estaciones que permanece en la celda de transmisión, ya que es el principal parámetro en la ec. 3.1 dado por  $N$ .

Minimizar el tiempo que le toma a una estación transmitir con éxito implica llevar al sistema lo más rápido posible al estado 1. El objetivo es determinar  $p = p(N)$  de la ec. 3.1, y para lograr lo anterior debido a que en cada ranura las estaciones conocen la multiplicidad de la colisión  $N$ , se intenta conducir al sistema a transitar al estado 1 en cada ranura de tiempo; así, para alcanzar el estado  $K = 1$  se proponen cuatro criterios que a continuación se describen.

### Criterio 1

Parece razonable suponer que  $p$  está determinada por el máximo de la ec. 3.1 cuando se busca transitar al estado  $K = 1$ . Entonces utilizando los métodos usuales de cálculo, se encontrará el valor de  $p$  que maximiza la ec. 3.1.

Si se deriva la ecuación 3.1 se tiene:

$$\frac{dP_{N,K}}{dp} = \frac{d}{dp} \left[ \binom{N}{K} p^K (1-p)^{N-K} \right] = P_{N,K} \left[ \frac{K}{p} - \frac{N-K}{1-p} \right] = P_{N,K} \left[ \frac{K - Np}{p(1-p)} \right]. \quad (3.2)$$

Con este resultado es posible calcular el máximo de la ec. 3.1 cuando  $K = 1$ , el cual se alcanza cuando

$$p = p_M(N) \triangleq \frac{1}{N}. \quad (3.3)$$

Recordando que la esperanza de la distribución binomial está dado por  $E[K] = Np$ , cuando  $p = p_M(N)$  la esperanza es  $E[K] = 1$ . Es importante observar que el estado más probable al que el sistema transita, cuando se repite un número grande de veces el experimento, es el estado 1. Se ha obtenido un resultado que de alguna manera resulta obvio, si el estado 1 es el más probable, entonces la mayor parte de las veces el sistema transitará hacia ese estado. Sin embargo, la esperanza no refleja lo que sucede con los otros estados; es decir, qué tan probable es llegar a los estados 2 o 4 en comparación con el estado 1. A fin de explorar estos casos, definimos la *razón de probabilidades*  $\alpha_{1,K}$  como sigue:

$$\alpha_{1,K}(p) \triangleq \frac{P_{N,1}(p)}{P_{N,K}(p)} = N \binom{N}{K}^{-1} \left[ \frac{1-p}{p} \right]^{K-1}. \quad (3.4)$$

El siguiente paso es emplear la ec. 3.4 y determinar qué sucede con  $P_{N,K}(p)$  conforme  $p$  y  $N$  cambian. Nos interesa examinar la razón cuando  $K = 0$  y  $K = 2$ , pero es posible examinar otras razones, por ejemplo cuando  $K \geq 3$ ; sin embargo, como se explica en el apéndice A.1, debemos asegurarnos de que la probabilidad  $p \in \left( \frac{1}{N+1}, \frac{2}{N+1} \right)$ , para evitar  $P_{N,1}(p) < P_{N,K}(p)$ , donde  $K = 0, 2, 3, 4, \dots, N$ . A lo anterior podemos agregar que se espera que  $p \xrightarrow{N \rightarrow \infty} 0$ , lo cual se infiere al suponer que la probabilidad de transmisión  $p$  debe ser cada vez menor cuando el número de estaciones es muy grande, para permitir que sólo unas cuantas intenten transmitir en la siguiente ranura. Tomando la ec. 3.4 y sustituyendo  $p$  por  $p_M(N)$  se tiene:

$$\alpha_{1,0}(p_M(N)) = \frac{Np}{1-p} = \frac{N}{N-1} \xrightarrow{N \rightarrow \infty} 1; \quad (3.5)$$

$$\alpha_{1,2}(p_M(N)) = \frac{2}{N-1} \left[ \frac{1}{p} - 1 \right] = 2. \quad (3.6)$$

A pesar de que el número de estaciones no es infinito, nos interesa el comportamiento de  $N$  cuando es grande, lo cual ocurre al inicio del CRI. El punto interesante de la ec. 3.5, es que la probabilidad de que el sistema llegue al estado 0 o al estado 1 al inicio del CRI es casi la misma. La esperanza indica que el estado más probable al que el sistema transita, en promedio es 1, lo que a final de cuentas es conveniente. Sin embargo, la esperanza no indica que hay otros estados que tienen una alta probabilidad de aparecer, de acuerdo con la ec. 3.5 el estado 0 tiene una probabilidad semejante a la del estado 1, y esto puede resultar en el desperdicio de ranuras de tiempo. La ec. 3.6 por el contrario, indica que es el doble de probable de ir al estado 1 que al estado 2, lo cual en nuestro

caso resulta adecuado; sin embargo, lo anterior indica que el sistema estaría inclinado a transitar a estados menores al estado 2, lo cual no es bueno ya que dentro de esos estados esta el 0.

## Criterio 2

De lo anterior resulta que no es conveniente tomar el máximo de  $P_{N,1}(N)$ . Parece razonable tratar de maximizar la distancia entre  $P_{N,1}(p)$  y  $P_{N,0}(p)$ ; es decir, alejar  $P_{N,1}(p)$  de  $P_{N,0}(p)$ . Definimos la distancia entre las probabilidades de los estados 0 y 1 como:

$$D \triangleq P_{N,1}(p) - P_{N,0}(p) = (1-p)^N \left[ \frac{Np}{1-p} - 1 \right], \quad \text{con } D > 0. \quad (3.7)$$

Para que la ecuación anterior sea siempre positiva, se debe cumplir que  $p > \frac{1}{N+1}$ , lo cual será tomado en cuenta para lo siguiente: si se iguala a cero la derivada de la ecuación anterior, encontramos que el máximo de  $D$  se obtiene cuando

$$p(N) = p_D(N) \triangleq \frac{2}{N+1} > \frac{1}{N+1}. \quad (3.8)$$

Debido a que el cálculo de la esperanza ya no resulta atractivo, haremos uso de las razones de probabilidad, las cuales cuando  $p = p_D(N)$  se tiene

$$\alpha_{1,0} = \frac{2N}{N-1} \xrightarrow{N \rightarrow \infty} 2 \quad (3.9)$$

$$\alpha_{1,2} = \frac{N-1}{N-1} = 1 \quad (3.10)$$

Como se puede observar en las ecuaciones 3.9 y 3.10, es el doble de probable llegar al estado 1 que al estado 0 cuando  $N$  es grande, y resulta que la probabilidad de llegar al estado 1 o 2 es la misma. Esto no es del todo indeseable, pero debido a este resultado surge la siguiente pregunta ¿Existirá un mejor valor para  $p$ ?

## Criterio 3

Una respuesta rápida a la pregunta anterior proviene de la fig. 3.3, la idea es tomar los puntos intermedios entre las curvas de  $p_M(N)$  y  $p_D(N)$ , ello se debe a que ambas curvas son iguales en cuanto a comportamiento (decrecen de la misma forma al aumentar  $N$ , y ambos tienden a 0 cuando

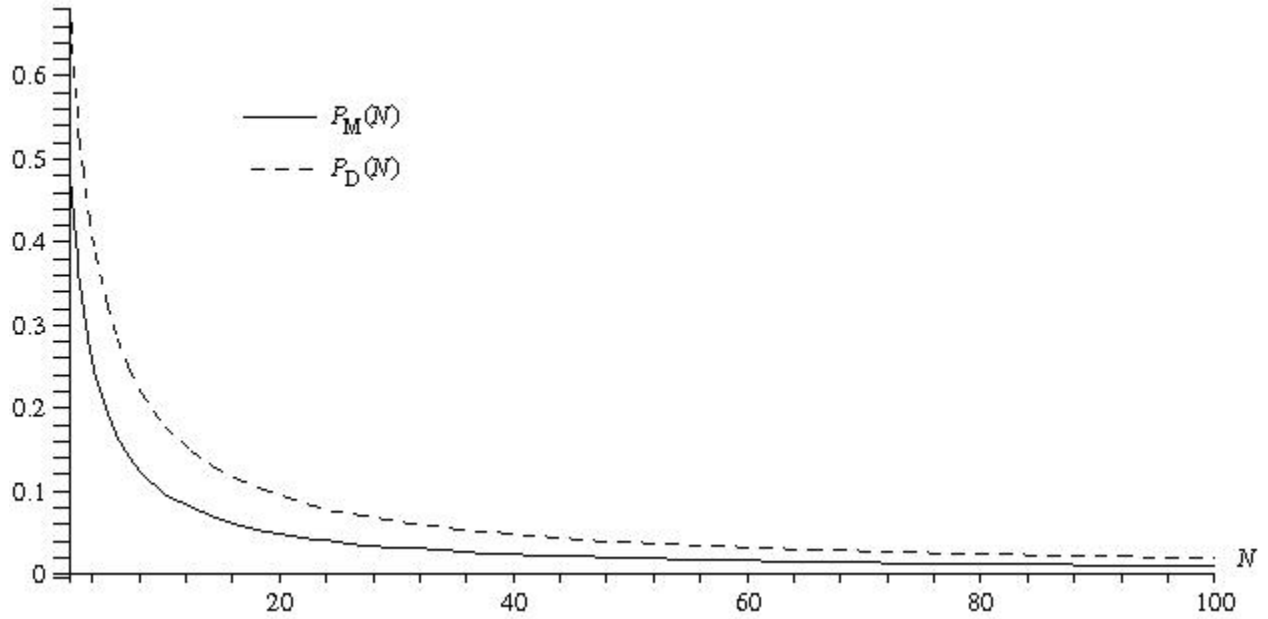


Figura 3.3: Curvas de  $p_M(N)$  y  $p_D(N)$ .

$N$  aumenta). Así el punto medio entre ambas curvas es

$$p_m(N) \triangleq \frac{p_M + p_D}{2} = \frac{1}{2} \left[ \frac{3N + 1}{N(N + 1)} \right]. \quad (3.11)$$

Sigue ahora determinar las razones, las cuales resultan al sustituir  $p = p_m(N)$ :

$$\alpha_{1,0} = \frac{3N^2 + N}{2N^2 - N - 1} \xrightarrow{N \rightarrow \infty} \frac{3}{2} \quad (3.12)$$

$$\alpha_{1,2} = \frac{4N^2 - 2N - 2}{3N^2 - 2N - 1} \xrightarrow{N \rightarrow \infty} \frac{4}{3} \quad (3.13)$$

Resulta que las razones casi tienen el mismo valor, pero aún hay un ligero sesgo hacia el estado 2, lo cual resulta deseable. Para verlo, se calcula el valor de la esperanza usando 3.11, la cual es

$$E[K] = \frac{1}{2} \left[ \frac{2N}{N + 1} + 1 \right] \xrightarrow{N \rightarrow \infty} \frac{3}{2}.$$

#### Criterio 4

Con lo anterior, la siguiente propuesta parte del criterio de que las razones tomen valores similares, lo que implica que  $\alpha_{1,0} = \alpha_{1,2}$ ; es decir, que  $P_{N,0}(N) = P_{N,2}(N)$ . Así la probabilidad  $p$  se determina al igualar la ec. 3.1, con  $K = 0$  y  $K = 2$

$$\begin{aligned}(1-p)^N &= \frac{N(N-1)}{2}p^2(1-p)^{N-1}, \\ [N(N-1) - 2]p^2 + 4p - 2 &= 0, \\ p &= \frac{\pm\sqrt{2N(N-1)-2}}{(N-2)(N+1)}.\end{aligned}$$

Obviamente estamos interesados en la solución positiva, por lo que  $p(N) = \frac{\sqrt{2N(N-1)-2}}{(N-2)(N+1)}$ . Note que la solución posee una singularidad en  $N = 2$ ; sin embargo, al calcular el límite se tiene (por L'Hopital)

$$\lim_{N \rightarrow 2} \frac{\sqrt{2N(N-1)-2}}{(N-2)(N+1)} = \lim_{N \rightarrow 2} \frac{\frac{1}{2}(2N(N-1))^{-\frac{1}{2}}[4N-2]}{(N-2) + (N+1)} = \frac{1}{2}.$$

Así, se tiene que

$$p = p_O(N) \triangleq \begin{cases} \frac{1}{2} & \text{para } N = 2, \\ \frac{\sqrt{2N(N-1)-2}}{(N-2)(N+1)} & \text{para } N > 2. \end{cases} \quad (3.14)$$

Con lo anterior es posible obtener las razones, las cuales son

$$\alpha_{1,0} = \alpha_{1,2} = \begin{cases} 2 & \text{para } N = 2, \\ \sqrt{2} & \text{para } N > 2. \end{cases}$$

Se ha obtenido un resultado para el cual la probabilidad de que el sistema llegue al estado 1 está en igual proporción a la probabilidad de llegar a los estados 0 y 2. La fig. 3.4 presenta las curvas de las  $p$ 's encontradas, observe que el comportamiento de todas las curvas es semejante; sin embargo, se espera que las curvas centrales que representan  $p_O(N)$  y  $p_m(N)$ , tengan resultados muy semejantes.

El siguiente paso es generar una simulación por cada criterio, esto permitirá comparar y determinar cuál probabilidad ofrece mejores resultados. Dado el análisis anterior, las opciones más factibles son  $p = p_m(N)$  y  $p = p_O(N)$ ; aunque un punto importante, es determinar cual probabilidad es más

fácil de calcular, ya que cada estación es la encargada de hacer esto<sup>5</sup>. El parámetro de comparación a emplear será la longitud promedio de un CRI, ya que al disminuir la longitud del CRI se eleva la calidad de servicio del sistema, al permitir que las estaciones puedan transmitir sus paquetes de forma rápida.

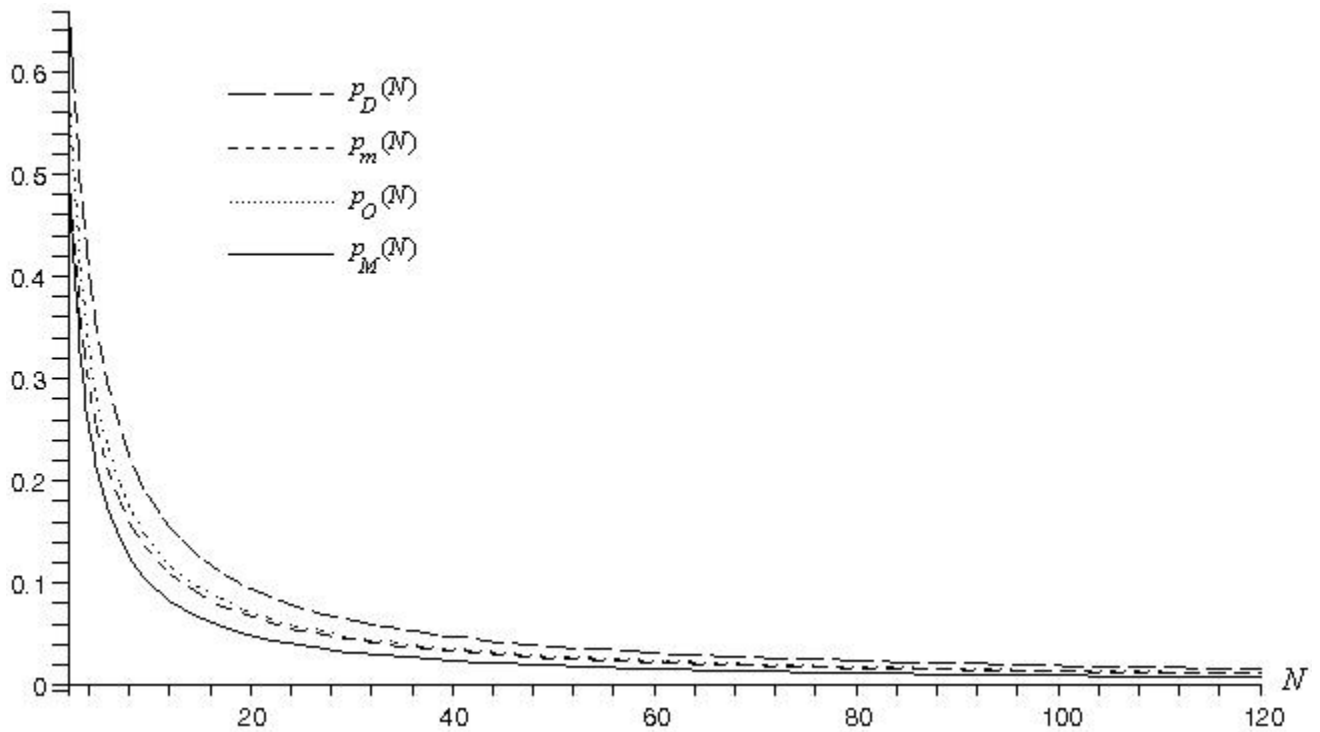


Figura 3.4: Curvas de  $p_M(N)$ ,  $p_D(N)$ ,  $p_O(N)$  y  $p_m(N)$ .

### 3.1.2 Resultados de simulación para la política de realimentación por ranura

La simulación consistió en reproducir el algoritmo, con la premisa de que la estación central es capaz de informar el número de estaciones en colisión al final de la ranura de transmisión; es decir, en lugar de tener una realimentación binaria, se tiene una realimentación  $N$ -aria; salvo el argumento anterior, el algoritmo es el mismo. En el apéndice B se muestra el funcionamiento básico del simulador, por lo que sólo se adaptó según el criterio de cálculo de  $p$  para el algoritmo.

<sup>5</sup>Es posible aplicar otros métodos para evitar calcular  $p$  a partir de la ecuación, por ejemplo el empleo de una tabla con los valores ya calculados.

Dada una multiplicidad de colisión  $N$ , desde 2 hasta 200 estaciones en contienda, se ejecutó el algoritmo y se determinó (sin considerar ventana de captura empleada en [15]): longitud promedio de un CRI, número de éxitos (debe ser igual a  $N$ ), número de ranuras sin transmisión (vacíos) y número de ranuras con una colisión. Cada conjunto de simulaciones se realizó con las diferentes propuestas para  $p$ , explicadas anteriormente en esta sección, las cuales en resumen son:

$$\begin{aligned}
 p &= \frac{1}{2}; \\
 p &= p_M(N) = \frac{1}{N}; \\
 p &= p_D(N) = \frac{2}{N+1}; \\
 p &= p_m(N) = \frac{3N+1}{2N(N+1)}; \\
 p &= p_O(N) = \begin{cases} \frac{1}{2} & \text{para } N = 2, \\ \frac{\sqrt{2N(N-1)}-2}{(N-2)(N+1)} & \text{para } N > 2. \end{cases}
 \end{aligned}$$

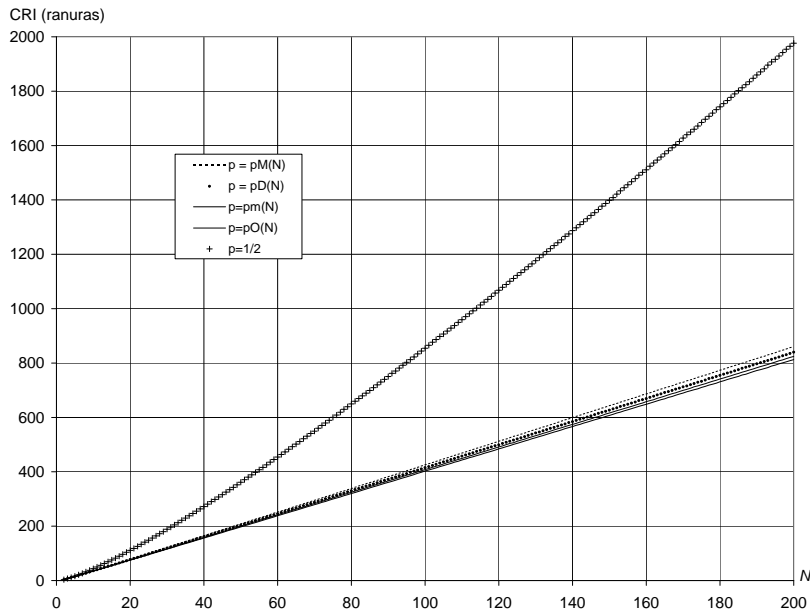


Figura 3.5: Longitud promedio de un CRI vs.  $N$ , para cada valor de  $p$  propuesto, incluyendo el valor utilizado en el algoritmo original.

Para cada valor de  $N$  ( $2 \leq N \leq 200$ ,  $N \in \mathbb{N}$ ) se realizó un conjunto de simulaciones, cada conjunto consistió de 10000 experimentos, cada uno con semilla diferente para el generador de números aleatorios y se obtuvieron promedios del total de las simulaciones. Se desprecia el tiempo de propagación de la señal, se asume que la realimentación llega al mismo tiempo a todas las estaciones, y como se mencionó antes, no hay errores en el canal de transmisión.

Los resultados de la simulación se pueden observar en la fig. 3.5. La figura muestra la longitud promedio de un CRI como función de la multiplicidad de la colisión  $N$ . Note que las propuestas para  $p$  ofrecen un mejor desempeño en comparación con los resultados del algoritmo original; por ejemplo para una multiplicidad  $N = 50$ , la longitud promedio de un CRI es de aproximadamente 360 ranuras en el algoritmo original, mientras que las propuestas ofrecen una longitud promedio de alrededor de 200 ranuras. Por otra parte, observe que la longitud promedio de un CRI, aparentemente crece de forma exponencial cuando  $N$  crece en el algoritmo original, en cambio las propuestas muestran al parecer un crecimiento lineal.

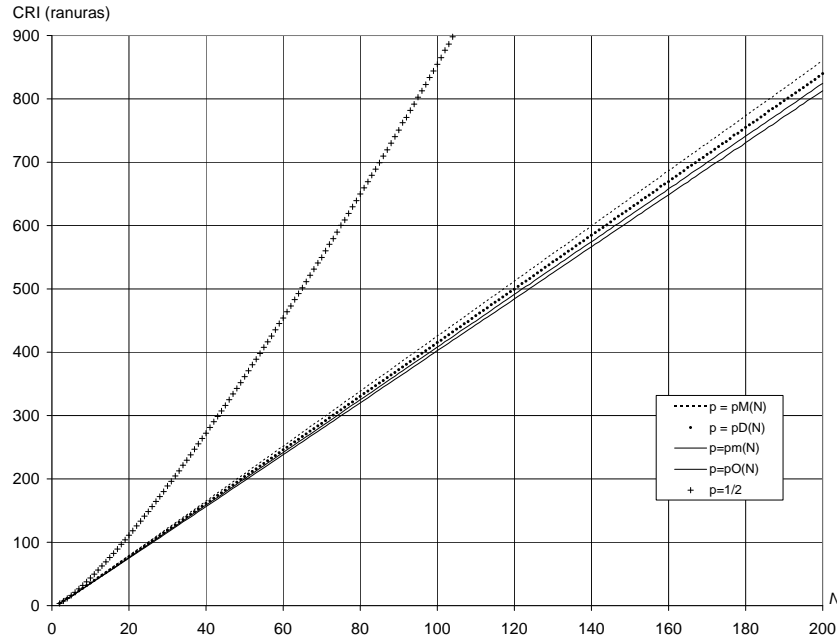


Figura 3.6: Longitud promedio de un CRI vs.  $N$ , la escala vertical ha sido modificada.

En la fig. 3.6 se muestran de nuevo todos los valores para  $p$  propuestos, pero en una escala vertical

menor; observe que quien ofrece el mejor desempeño es  $p_m(N)$ , dado que produce CRIs más cortos. La opción  $p_O(N)$  permite ligeramente mejores resultados cuando  $N$  es relativamente pequeño (alrededor de 40 estaciones), sin embargo, les resultará más fácil a las estaciones realizar cálculos con  $p_m(N)$ , por lo que está será nuestra mejor propuesta.

La fig. 3.7 muestra que todos los valores para  $p$  presentan resultados similares cuando  $N = 2$  (esto será importante para la siguiente propuesta), pero  $p_m(N)$  ofrece mejores resultados cuando  $N > 2$ . Sin embargo, aun en  $N = 2$ ,  $p_m(N)$  permite resultados ligeramente mejores a  $p = \frac{1}{2}$ , ello se puede apreciar en la tabla 3.1. Note que la longitud promedio de un CRI cuando  $N = 2$  es de 3.5167 ranuras para el algoritmo original, lo cual ofrece un caudal<sup>6</sup> promedio de 0.568715 paquetes/ranura; en cambio, para  $p = p_m(N)$  el caudal promedio es de 0.580029 paquetes/ranura. En [15] se determina un caudal de 0.4297 paquetes/ranura con una ventana de captura de 2.330 ranuras. La ventana de captura permite que en promedio un par de estaciones inicien un CRI. Si se promedia el caudal de los datos presentados en la tabla 3.1, se tiene un valor de 0.4420 paquetes/ranura para el algoritmo original, mientras que para la propuesta  $p_m(N)$  se tiene un caudal de 0.4567 paquetes/ranura; esto hace que la propuesta sea ligeramente más atractiva que el algoritmo original para un número reducido de estaciones en contienda.

Tabla 3.1: Resultados numéricos de la simulación para política de realimentación por ranura de la simulación. Sólo se muestran resultados para el algoritmo original dado por  $p = \frac{1}{2}$ , y una de las propuestas dada por  $p = p_m(N)$ . Todos los valores, salvo la multiplicidad de la colisión  $N$ , tienen como unidad la ranura de tiempo.

$p = \frac{1}{2}$					$p = p_m(N)$				
$N$	Éxitos	Vacios	Colisiones	Longitud promedio CRI	$N$	Éxitos	Vacios	Colisiones	Longitud promedio CRI
2	2	0.5079	1.0088	3.5167	2	2	0.3691	1.0790	3.4481
3	3	0.9104	3.4081	7.3185	3	3	0.7950	3.3098	7.1048
4	4	1.2846	6.2325	11.5171	4	4	1.2444	5.6155	10.8599

Finalmente, el número de experimentos por simulación fue de 10000 para cada valor de  $N$ ; sin embargo, ese número de experimentos resulta grande para darle confianza a los resultados de la

<sup>6</sup>El caudal del sistema se determina en paquetes/unidad de tiempo, en este caso se han calculado como la razón  $\frac{N}{\text{Longitud promedio CRI}}$ . Se emplea la multiplicidad de la colisión, ya que no sólo representa el número de estaciones en conflicto, sino también el número de paquetes que ha colisionado, ya que cada estación aporta un paquete.

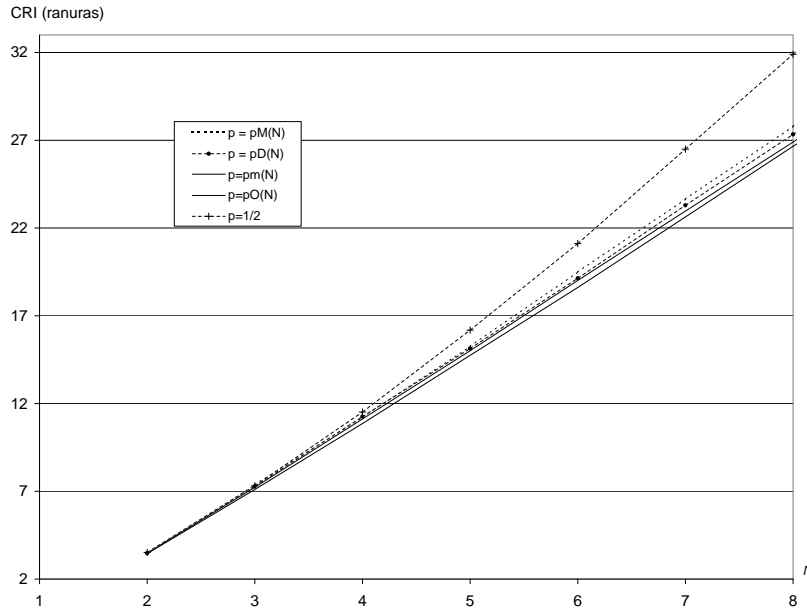


Figura 3.7: Longitud promedio de un CRI *vs.*  $N$ , para valores pequeños de  $N$ .

simulación, como lo demuestra la fig. 3.8. El análisis que se siguió para determinar el número de simulaciones, se basa en observar el promedio temporal de la longitud de un CRI en cada simulación. En total hubo 199 simulaciones con 10000 experimentos para cada uno de los cinco valores de probabilidad  $p$ ; por tanto, hay 995 gráficas que presentan el promedio de la longitud de un CRI *vs.* el número de simulaciones. Debido a la cantidad de cómputo necesario para su elaboración y a limitaciones de espacio, sólo se tomo una muestra de ese conjunto de 995 gráficas, en este caso la fig. 3.8 representa el peor caso observado en esa muestra.

Según la fig. 3.8 es necesario considerar únicamente hasta 4000 experimentos por simulación, para que el promedio de la longitud del CRI se estabilice; sin embargo, en general un valor entre 4000 y 5000 experimentos por simulación eran suficientes, ya que en la mayoría de las simulaciones el promedio de la longitud del CRI comienza a estabilizarse a partir de 3000 experimentos.

### 3.1.3 Margen de error permitido para la política de realimentación por ranura

En la sección anterior se propuso una mejora al algoritmo bajo el principio de que todas las estaciones en proceso de transmitir un paquete son capaces de saber el número de estaciones en contienda; la

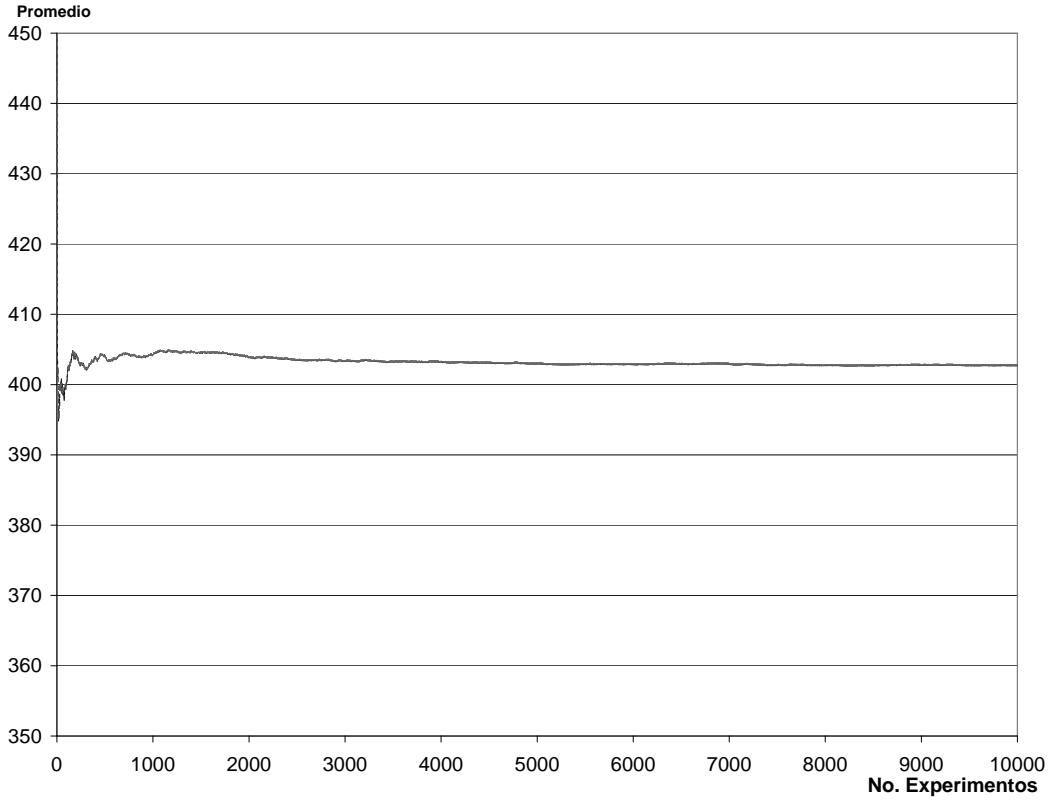


Figura 3.8: Promedio de la longitud de un CRI en función del número de simulaciones; en este caso se uso  $p = p_m(N)$  y  $N = 100$ .

simulación y el análisis determinaron que  $p_m(N)$  ofrece un buen desempeño en comparación con el algoritmo original, dado que produce CRIs más cortos. Lo que sigue ahora es determinar cuál es el error permitido, para no beneficiar otros estados diferentes al estado 1; es decir, la multiplicidad de la colisión  $N$  será en realidad un valor estimado, por lo que sería adecuado saber cuál es la máxima desviación del valor  $N$ , con la finalidad de no entorpecer el desarrollo del algoritmo. El punto de partida lo da el intervalo de  $p$  (ver apéndice A.1). Sea  $N$  el valor real de la multiplicidad de la colisión, y  $\hat{N}$  su valor estimado, si  $p$  es función de  $N$ , entonces es necesario determinar el intervalo de  $N$  tal que  $p$  se mantenga dentro del intervalo  $(\frac{1}{N+1}, \frac{2}{N+1})$ ; de esta forma, se asegura que la probabilidad de llegar al estado 1 es mayor que para cualquier otro estado. Lo anterior se traduce en determinar  $\hat{N}$  para  $p_m(N)$  tal que

$$\frac{1}{N+1} < p_m(\hat{N}) < \frac{2}{N+1},$$

$$\frac{1}{N+1} < \frac{3\widehat{N}+1}{2\widehat{N}(\widehat{N}+1)} < \frac{2}{N+1}.$$

Después de algunas manipulaciones algebraicas, se obtiene

$$2\widehat{N}(\widehat{N}+1) < (3\widehat{N}+1)(N+1) < 4\widehat{N}(\widehat{N}+1),$$

$$2\widehat{N}^2 - (3N+1)\widehat{N} - (N+1) < 0 < 4\widehat{N}^2 - (3N-1)\widehat{N} - (N+1).$$

Se tiene dos ecuaciones que son sencillas de resolver, cuyos "ceros" ofrecen el intervalo para la variable  $\widehat{N}$ . Ambas ecuaciones poseen un mínimo, por lo que los valores de  $\widehat{N}$  deberán estar entre

$$\begin{aligned} \frac{(3N-1)+\sqrt{(3N-1)^2+16(N+1)}}{8} < \widehat{N} < \frac{(3N+1)+\sqrt{(3N+1)^2+8(N+1)}}{4}, \\ \frac{(3N-1)}{8} \left[ 1 + \sqrt{1 + \frac{16(N+1)}{(3N-1)^2}} \right] < \widehat{N} < \frac{(3N+1)}{4} \left[ 1 + \sqrt{1 + \frac{8(N+1)}{(3N+1)^2}} \right]. \end{aligned} \quad (3.15)$$

La ecuación anterior representa la desviación que se puede permitir al predecir la multiplicidad de la colisión. Es posible aproximar a primer orden la ecuación anterior y obtener algo como

$$N - \Delta_1(N) < \widehat{N} < N + \Delta_2(N);$$

de tal forma que  $\Delta_1(N)$  determina subestimación permitida de  $N$ , y  $\Delta_2(N)$  la sobrestimación. En lugar de lo anterior, dividiremos los extremos de la desigualdad 3.15 entre  $N$  para obtener el porcentaje permitido de error para cada valor de  $N$ . La fig. 3.9 muestra la razón de error que se acepta al estimar  $N$ , observe que cuando  $N$  crece se permite sobrestimar su valor hasta un 50%; en cambio, sólo se permite hasta un 25% cuando se subestima. Por ejemplo, si el valor original para la multiplicidad de la colisión es  $N = 40$ , entonces es posible sobrestimar  $N$  hasta 1.5 veces de su valor original; en cambio para subestimarlo, es posible una razón de error de hasta 0.25 veces del valor real. Lo anterior permite establecer un criterio de selección, para escoger algún algoritmo de estimación en la estación central que cumpla estas condiciones.

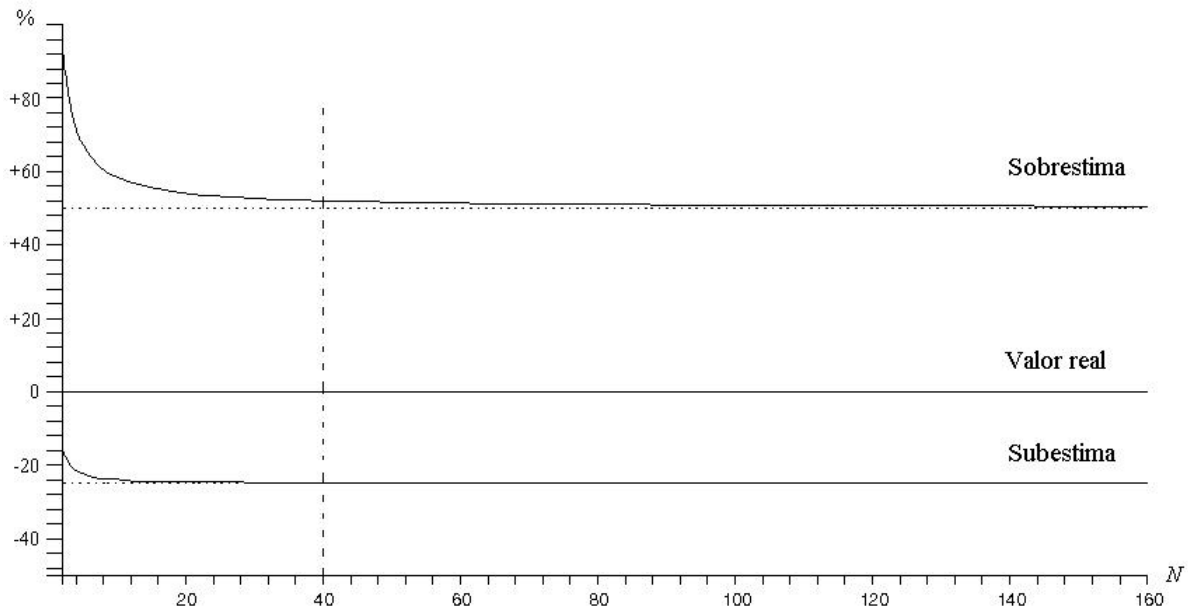


Figura 3.9: Razón de error al estimar la multiplicidad de la colisión  $N$ . La figura muestra la razón error *vs*  $N$ ; la línea horizontal continua, representa el valor de  $N$  real, las demás curvas representan la desviación que puede sufrir éste valor real de  $N$ .

## 3.2 Política de realimentación de la multiplicidad de la colisión por fase

### 3.2.1 Análisis

Recordemos que un CRI se compone de fases, y las fases se componen de ranuras en colisión, incluyendo la ranura donde sucedió el éxito que determina el final de la fase<sup>7</sup>. Así, una fase está limitada por dos éxitos, entre los cuales hay al menos una colisión y la fase tendrá al menos una ranura de longitud. Para esta política, la estación central será la encargada de realimentar a las demás estaciones únicamente al inicio de una fase. Así, la estimación de la multiplicidad de la colisión se podrá realizar de dos formas:

1. En cada fase. En este caso la estación central deberá predecir del valor de la multiplicidad de la colisión que inicia la fase, y realimentar con este valor a las estaciones en contienda.
2. Al inicio de un CRI. Una vez que la estación central determina la multiplicidad de la colisión,

<sup>7</sup>Recuerde que un vacío también finaliza un fase.

es capaz de actualizar este valor en cada fase, debido a que al finalizar una fase la estación central sabrá dos cosas: una estación transmitió con éxito o ninguna estación transmitió. Al contabilizar el número de transmisiones exitosas, la estación central será capaz de determinar cuántas estaciones iniciarán una nueva fase, e informar de ello a las estaciones en contienda.

El proceso de estimar la multiplicidad de la colisión depende en gran medida del algoritmo propuesto para ello. Por ejemplo, si el algoritmo de estimación no tiene un buen grado de confianza, entonces será adecuado que la estimación se realice en cada fase, ya que si la estimación se realiza al inicio del CRI se corre el riesgo de propagar el error a lo largo de todo el CRI. En cambio un buen algoritmo de predicción podría emplearse al inicio del CRI.

A diferencia de la política de realimentación por ranura, en esta política las estaciones no pueden saber como evoluciona la multiplicidad de la colisión en cada ranura, debido a que la estación central informa el valor de la multiplicidad de la colisión al inicio de la fase. Por lo tanto, las estaciones tendrán que seleccionar el valor óptimo para  $p$  en función de la multiplicidad de la colisión al inicio de la fase, que permita minimizar la longitud de la fase.

El algoritmo propuesto sugiere ligeros cambios al algoritmo original, debido a que sólo al inicio de una fase la realimentación será  $M$ -aria; en donde  $M$  representa la multiplicidad de la colisión al inicio de una fase. En las demás ranuras, la realimentación será binaria, debido a que no será posible conocer la evolución de la multiplicidad de la colisión.

El análisis de esta política no se ve afectado por las observaciones anteriores, ello se debe a que la estación central informa la multiplicidad de la colisión al inicio de cada fase, independiente de cuando realice la estimación. Para el análisis que sigue, se asume que la estación central informa de manera exacta el valor de la multiplicidad de la colisión al inicio de la fase; más adelante discutiremos qué sucede cuando la estación predice este valor.

Al igual que en la propuesta de realimentación por ranura, el estado del sistema está representado por el número de estaciones que existe en la celda de transmisión al final de la ranura. Sea  $N$

la multiplicidad de la colisión al inicio de la fase<sup>8</sup>, así, si  $i$  es el estado del sistema definido como el número de estaciones en transmisión, entonces  $i \in \{0, 1, 2, \dots, N\}$ . El cambio de un estado a otro, lo que llamamos transición o salto, está determinado por la ecuación (3.1); es decir, la probabilidad de llegar al estado  $K$  dado  $N$  está dada por la ec. 3.1 ya que sistema continúa comportándose de acuerdo una distribución binomial.

Para esta política la transición entre estados se ve condicionada por el estado anterior; es decir, al inicio de una fase el estado del sistema es  $N$ , después del cual hay una serie de transiciones a otros estados hasta llegar al estado 0 o 1, que determinan el final de una fase. No es posible conocer por qué estados transitará el sistema, pero si se sabe que el siguiente estado es menor o igual al estado anterior, de ahí que cada transición a un cierto estado está condicionada al estado anterior.

Debido a estas dependencias, el método de análisis que seguiremos empleará un *árbol de transiciones*, en donde cada rama representa un camino que el sistema puede seguir desde la raíz o estado  $N$ , hasta un estado  $i$ . La profundidad del árbol está representada por el número de saltos o transiciones que hay desde la raíz hasta el estado  $i$ . En la fig. 3.10 se muestra un árbol de profundidad 2; note que los estados 0 y 1 no dan origen a nuevas transiciones, en cambio los demás estados pueden generar transiciones a otros estados. El estado 0 produce que se tenga una nueva fase con multiplicidad de la colisión  $N$ , en cambio el estado 1, produce que la siguiente fase tenga multiplicidad de la colisión igual a  $N - 1$ .

Estamos interesados en aquellas ramas cuyo estado final es  $i = 1$ . Es posible encontrar caminos muy largos hacia el estado 1, los cuales pueden tener una gran cantidad de transiciones, por lo que nuestro análisis se limita a elegir el valor de  $p$  que cumpla con las siguientes dos características:

- Incrementa la probabilidad de tomar aquellas ramas cuyo estado final sea 1.
- Deberá producir el menor número de transiciones hacia el estado 1.

---

<sup>8</sup>De aquí en adelante denotaremos por  $N$  la multiplicidad de la colisión al inicio de la fase.

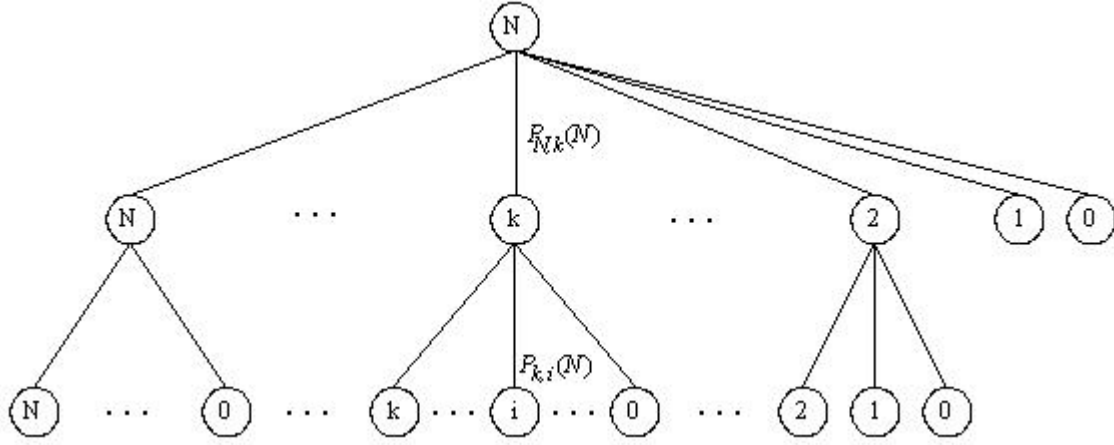


Figura 3.10: Árbol de transiciones de profundidad 2; en general el árbol puede tener una profundidad mayor.

En el apéndice A.3 se demuestra que si se parte de un estado  $N$ , la probabilidad de llegar al estado  $i$  después de  $n$  transiciones o saltos está dada por la siguiente ecuación

$$Q_{N,i}^{(n)}(p) = \binom{N}{i} p^{ni} \begin{cases} [(1-p^n)^{N-i} - (1-p^{n-1})^{N-i} - \delta(i)Np^{n-1}q(1-p^{n-1})^{N-1}], & i < 2, \\ (1-p^n)^{N-i}, & i \geq 2; \end{cases} \quad (3.16)$$

en donde  $n \geq 1$ ,  $0 \leq i \leq N$ ,  $N \geq 2$ ,  $q = 1 - p$  y  $\delta(i)$  es una función indicador definida por la ec. A.7 del apéndice. Una primera propuesta de solución, es maximizar  $Q_{N,1}^{(n)}(p)$  para valores de  $n$  cercanos a 1 y así tener fases de duración corta.

En  $n = 1$  se tiene el caso de la política de realimentación por ranura, el cual ya fue analizado; sin embargo, recordemos que ese valor de  $p$  es adecuado cuando se conoce para cada ranura el valor de la multiplicidad de la colisión, lo cual no es el caso. Dicho análisis de realimentación por ranura arrojó que un buen valor de  $p$  puede ser  $p_m(N)$  (ec. 3.11), además de que  $p$  puede aceptar una desviación (o error) en  $N$  siempre y cuando dicho error no ocasione que  $p$  salga del intervalo  $(\frac{1}{N+1}, \frac{2}{N+1})$ ; por este motivo sería fácil suponer que usar  $p = p_m(N)$  durante toda la fase es adecuado.

Sin embargo, hay que analizar la ec. 3.16 a fin de determinar la opción óptima para este nuevo sistema. Un caso interesante en la ec. 3.16 sucede cuando  $i = 0$ , lo que permite eliminar el término  $p^{ni}$ . Este término produce que 3.16 tenga un crecimiento lento en función de  $p$  y que tienda a 0

conforme  $i$  y  $n$  crecen. Lo anterior predispone a  $Q_{N,0}^{(n)}(p)$  a tender a 1 cuando  $i = 0$  (debido a términos de la forma  $(1 - p^n)^{N-i}$ ), lo cual desde luego no es deseable. Esto resulta obvio ya que si en la primera transición (o en subsecuentes) no se logra el éxito, en la siguiente transición es muy probable que se llegue al estado 0, debido a que el número de estaciones restantes emplean el mismo valor para  $p$ , el cual es adecuado para el número de estaciones que hay al inicio de la fase. En general es muy complicado determinar el máximo de  $Q_{N,0}^{(n)}(p)$  como función de  $p$  y ver que efectivamente hay un crecimiento a 1, ya que se tiene el problema de determinar los ceros de un polinomio de grado  $(nN - 1)$ . Sin embargo, se puede examinar una transición particular; por ejemplo la segunda transición ( $n = 2$ ). Si se obtiene el máximo de  $Q_{N,0}^{(2)}(p)$  como función de  $p$ , resulta

$$\frac{dQ_{N,0}^{(2)}(p)}{dp} = \frac{d}{dp} \left[ (1 - p^2)^N - (1 + Np)(1 - p)^N \right] = 0,$$

donde el máximo de  $Q_{N,0}^{(2)}(p)$  que definimos como  $p_{M20}(N)$  resulta ser

$$p = p_{M20}(N) \triangleq \left( \frac{N+1}{2} \right)^{\frac{1}{N-1}} - 1. \quad (3.17)$$

No es fácil determinar a primera vista cuál es el comportamiento de la ec. 3.17, pero  $p_{M20}(N) \rightarrow 0$  cuando  $N \rightarrow \infty$ ; para ello sea  $p_{M20} = \gamma - 1$ , en donde  $\gamma = \left( \frac{N+1}{2} \right)^{\frac{1}{N-1}}$ . Se tiene que  $\gamma \geq 1$ , por lo que podemos expresar a  $\gamma$  como  $\gamma = 1 + h(N)$ , en donde  $h(N) \geq 0$ , así

$$\frac{N+1}{2} = (1 + h(N))^{N-1} \geq 1 + (N-1)h(N) + \frac{(N-1)(N-2)}{2}h(N)^2 \geq \frac{(N-1)(N-2)}{2}h(N)^2,$$

$$\frac{N+1}{(N-1)(N-2)} \geq h(N)^2 \geq 0.$$

De lo anterior,  $h(N) \rightarrow 0$  cuando  $N \rightarrow \infty$ , por lo que  $\gamma \rightarrow 1$  y por lo tanto  $p_{M20}(N) \rightarrow 0$ . El problema es que el máximo de  $Q_{N,0}^{(2)}(p)$  tiene el mismo comportamiento que  $p_m(N)$  o al menos ambos tienden a 0 conforme aumenta el valor de  $N$ , por lo que es posible que  $Q_{N,0}^{(2)}(p)$  se vea beneficiado dentro del intervalo definido para  $p$  en el apéndice A.1; y en efecto, por ejemplo cuando  $p = \frac{2}{N+1}$

(un extremo del intervalo para  $p$ ) se tiene

$$Q_{1,1}^{(2)}\left(\frac{2}{N+1}\right) = \frac{2N}{N+1} \left(1 - \frac{2}{N+1}\right)^{N-1} \xrightarrow{N \rightarrow \infty} 2e^{-2},$$

$$Q_{2,0}^{(2)}\left(\frac{2}{N+1}\right) = \left(1 - \frac{2}{N+1}\right)^N \left[\left(1 + \frac{2}{N+1}\right)^N - 1 - \frac{2N}{N+1}\right] \xrightarrow{N \rightarrow \infty} 1 - 3e^{-2};$$

es decir, cuando  $p = \frac{2}{N+1}$  la probabilidad  $Q_{N,0}^{(2)}(p)$  es prácticamente el doble que  $Q_{N,1}^{(1)}(p)$  cuando  $N$  es grande, lo cual refuerza la hipótesis de que a la segunda transición es muy probable que el sistema llegue al estado 0. Sin embargo, sólo para ciertos valores de  $N$  se beneficia  $Q_{N,0}^{(2)}(p)$ , ello se puede observar estudiando para qué valores se cumple la desigualdad  $Q_{N,1}^{(1)}(p) > Q_{N,0}^{(2)}(p)$  cuando  $p = \frac{2}{N+1}$ , así se tiene

$$Q_{N,1}^{(1)}(p) > Q_{N,0}^{(2)}(p),$$

$$Np(1-p)^{N-1} > (1-p^2)^N - (1+Np)(1-p)^N,$$

$$\frac{Np}{1-p} + (1+Np) > (1+p)^N,$$

sustituyendo  $p = \frac{2}{N+1}$  (el extremo superior para el intervalo de  $p$ ) en la desigualdad se tiene

$$\frac{5N^2 - 1}{N^2 - 1} > \left[\frac{N+3}{N+1}\right]^N; \quad (3.18)$$

en general es difícil resolver algebraicamente la ecuación anterior para  $N$ , pero es posible obtener numéricamente el valor para el cual la desigualdad no se cumple; entonces,  $Q_{N,1}^{(1)}(p) > Q_{N,0}^{(2)}(p)$  dentro del intervalo  $(\frac{1}{N+1}, \frac{2}{N+1})$  sólo si  $N \leq 9$ , para valores mayores a este límite no es posible asegurar que la desigualdad se cumpla para el intervalo antes mencionado. Para el otro extremo del intervalo  $(\frac{1}{N+1})$ , la desigualdad siempre se cumple, por lo que el problema radica en el extremo superior del intervalo  $(\frac{1}{N+1}, \frac{2}{N+1})$ .

Proponer  $p_m(N)$  durante todo el desarrollo de la fase (dado que resultó la mejor opción en la política de realimentación por fase), pudiese ser válido sólo si la multiplicidad de la colisión  $N \leq 9$ , y sólo resultaría adecuado en la primera transición. Un valor mayor a 9 propiciaría que el sistema tuviera cierta tendencia a ir al estado 0 en la segunda transición. Por este motivo, el proponer una

razón para determinar esta tendencia, resulta muy complicado debido a la forma de la ec. 3.16. De lo anterior se puede concluir lo siguiente:

1. Si se elige  $p = p_m(N)$ , como la probabilidad que se debe usar en la fase, se corre el riesgo de incrementar la probabilidad de transitar al estado 0 en la segunda transición, después de un determinado valor de  $N$ . En términos más generales, cuando  $N > 9$  la amplitud del intervalo definido en el apéndice A.1 para la probabilidad  $p$  tiende a decrecer, lo cual no es deseable dado que el error permitido para la estimación será cada vez menor.
2. Otra opción es emplear  $p = p_M(N)$  (ec. 3.3), ya que en este caso se asegura que  $Q_{N,1}^{(1)}(p) > Q_{N,0}^{(2)}(p)$ . Se maximiza la probabilidad de que el sistema llegue al estado 1 en una transición, pero también se beneficia la probabilidad de llegar al estado 0 en una transición cuando  $N$  es grande. De los dos puntos anteriores surge la idea de que resulta conveniente llevar al sistema a un estado  $i > 1$ , después de la primera transición en caso de no lograr llegar al estado 1.
3. Es de esperar que  $Q_{N,0}^{(2)}(p)$  tienda a crecer debido a que si el sistema evoluciona favorablemente (no va al estado 0), cada vez un menor número de estaciones intentará hacer el experimento de tipo Bernoulli con la misma probabilidad, la cual fue adecuada para el número grande de estaciones al inicio de la fase, pero puede resultar no adecuada cuando disminuye el número de estaciones contendientes; así, es muy probable que en dos o más transiciones se llegue al estado 0.

En base a estas observaciones, se propone el siguiente algoritmo el cual consiste en emplear dos probabilidades por fase en lugar de una:

---

#### Algoritmo 2C adaptable

---

1. **En la primera ranura de la fase el sistema empleará por probabilidad  $p = p_m(N)$ ; se considera  $p = p_m(N)$  debido a que arrojó mejores resultados en la política de realimentación por ranura.**
  2. **En las subsecuentes ranuras y hasta el final de la fase, se empleara  $p = \frac{1}{2}$ . Con lo anterior, se espera tener en promedio  $\frac{3}{2}$  estaciones compitiendo en la segunda ranura (ver sec. 3.1.1).**
-

La propuesta tiene la finalidad de conducir al sistema al llevarlo de un número  $N$  de estaciones al inicio de la fase, a un número pequeño en la siguiente ranura y emplear a partir de ahí el algoritmo original; lo anterior es el resultado de observar la fig. 3.7, ya que cuando  $N$  es pequeño ( $N < 8$ ), todas las propuestas para  $p$  tienen el mismo rendimiento que el algoritmo original, lo que permite emplear  $p = \frac{1}{2}$  para el resto de la fase. Note además que la propuesta sólo requiere de un ligero cambio al algoritmo original.

Por otra parte, es posible emplear los resultados anteriores para el caso de un algoritmo de realimentación ternaria; en el cual, además de *colisión* ( $C$ ) y *no colisión* ( $NC$ ), se tiene *no transmisión o vacío* ( $E$ ). En este caso particular, la estación central es capaz de informar cuando ninguna estación empleó el canal para transmitir; la fase queda ahora definida exclusivamente en términos de transmisiones exitosas. Sin embargo existen un par de problemas con esta propuesta, ya que el considerar que la fase termina exclusivamente con el primer éxito, produce un efecto recursivo cuando ninguna estación transmite; es decir, es posible tener una multiplicidad de la colisión  $N$  dentro de una fase; si esto sucede, el algoritmo no debe continuar empleando  $p = \frac{1}{2}$ , en lugar de ello las estaciones deberán "recordar" el último valor de la multiplicidad de la colisión y emplearlo con  $p = p_m(N)$  para después regresar al valor  $p = \frac{1}{2}$ .

El segundo problema surge fuera del CRI, ya que un conjunto de realimentaciones  $E$  no podrán interpretarse como fuera o dentro del CRI; un mecanismo mencionado en [14] para evitar esto, es colocar un contador que permita esperar un cierto número de vacíos antes de que la estación decida transmitir un paquete.

### 3.2.2 Resultados de simulación para la política de realimentación por fase

Al igual que en el caso de la política de realimentación por fase, se asume que: no existen errores en el canal, se desprecia el tiempo de propagación de la señal y se considera que todas las estaciones reciben la realimentación al mismo tiempo. La simulación constó de 10000 experimentos para cada multiplicidad de la colisión desde 2 hasta 200. De los resultados se obtiene la longitud promedio del CRI como función de la multiplicidad  $N$ , y nuevamente éste será el parámetro de comparación para determinar cuál propopuesta es mejor.

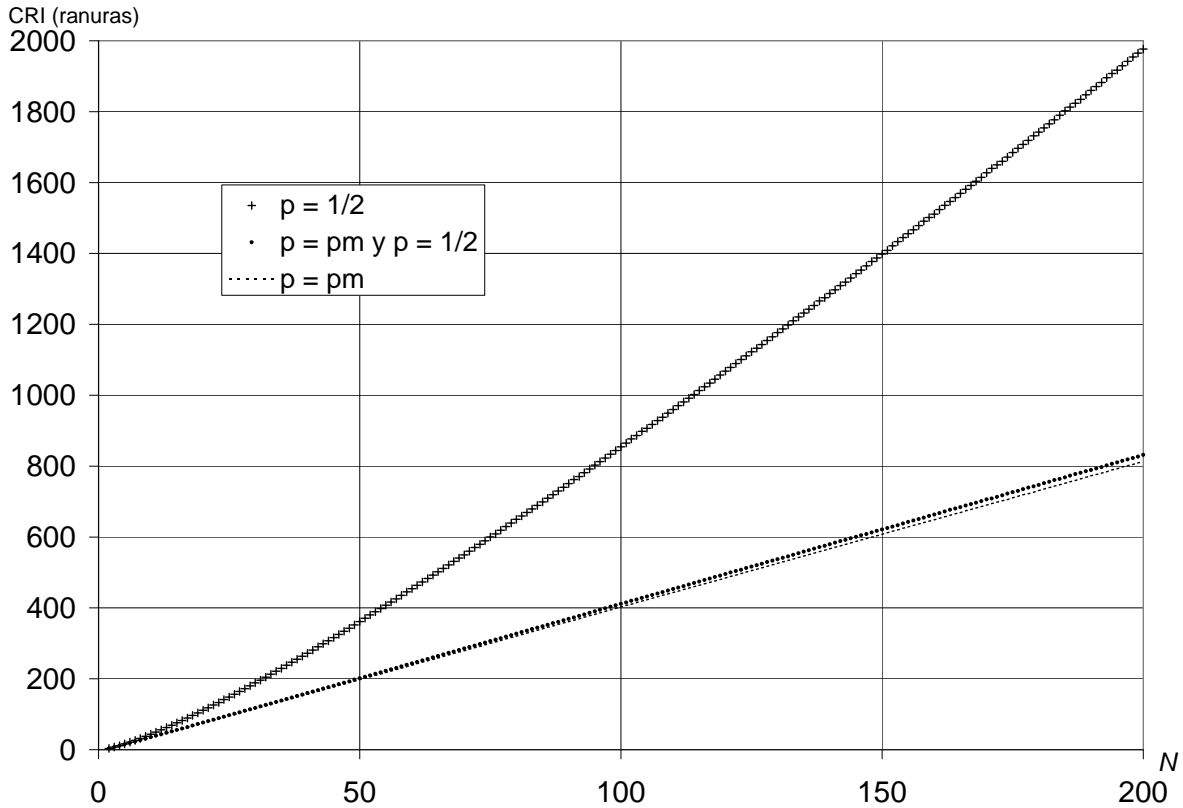


Figura 3.11: Resultados de la simulación para la política de realimentación por fase. En la gráfica se muestra también los resultados del algoritmo original, y de la política de realimentación por ranura.

En la fig. 3.11 se muestra los resultados de la simulación, observe que los resultados prácticamente concuerdan con los obtenidos en la política de realimentación por ranura, por lo que estimar la multiplicidad de la colisión en cada fase o al inicio del CRI resulta adecuado, además de que el algoritmo no requiere de grandes cambios.

Al igual que en la política de realimentación por ranura, surge la pregunta de si 10000 experimentos por simulación son suficientes. Nuevamente este número es grande en comparación con el requerido. La fig. 3.12 muestra uno de los peores casos observados en donde se requieren al menos 6000 experimentos por simulación; el análisis que se siguió para obtener la gráfica, es el mismo que se describió en la política de realimentación por ranura. En la mayoría de los casos, son suficientes unos 5000 experimentos por simulación.

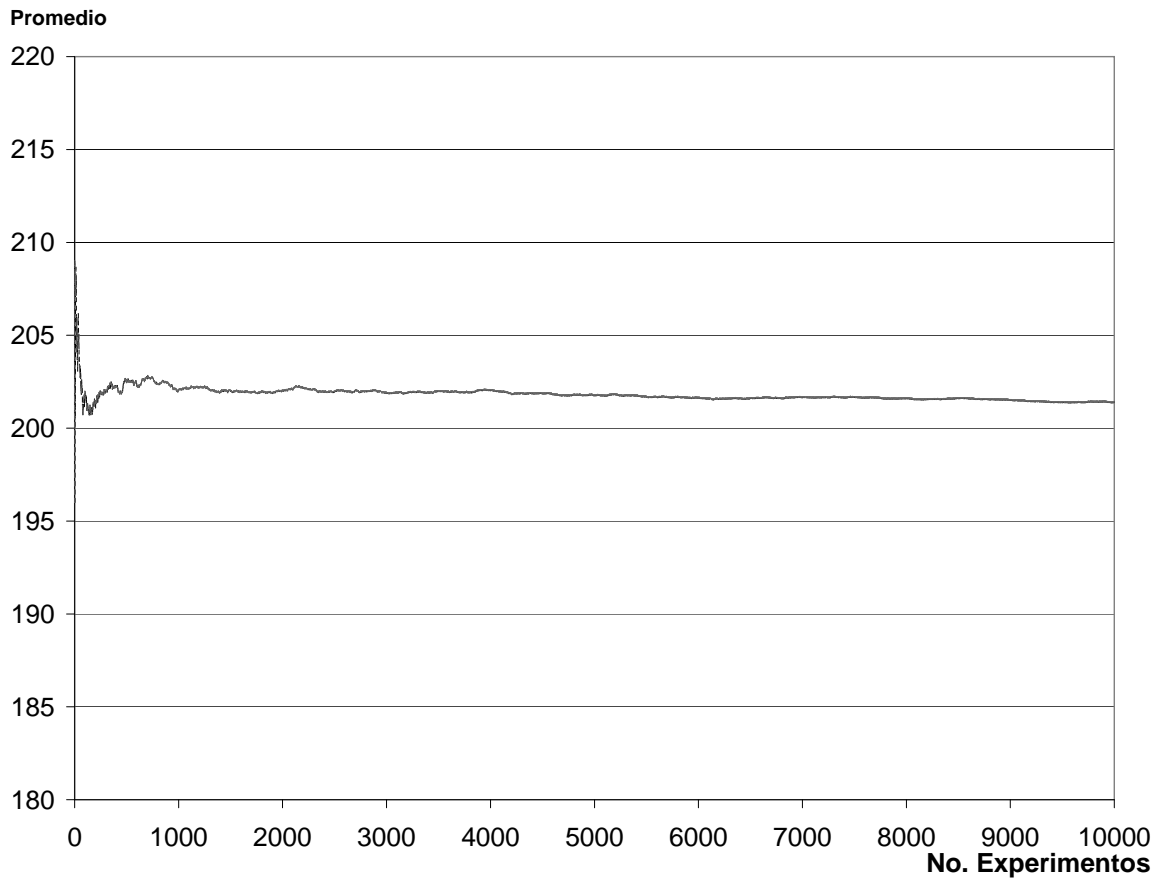


Figura 3.12: Promedio de la longitud de un CRI en función del número de simulaciones; en este caso se usó el algoritmo 2C adaptable con  $N = 50$ .

### 3.2.3 Margen de error permitido para la política de realimentación por fase

El error que permite esta propuesta depende básicamente de lo que sucede en la primera ranura; es decir, el error permitido para la primera ranura es el mismo que el que se tiene en la propuesta de realimentación por ranura. El hecho de delimitar un intervalo para la probabilidad  $p$  asegura que en promedio el sistema llegará al estado 1 o al estado 2 (la esperanza es igual a  $\frac{3}{2}$ , ver sec. 3.1.1), lo cual es muy deseable ya que la probabilidad  $p$  en la segunda ranura se establece en base a este criterio. Sin embargo, ¿qué sucede si en la segunda ranura existen más de dos estaciones? Si se

toma el intervalo obtenido en el apéndice A.1 se tiene

$$\frac{1}{M+1} < \frac{1}{2} < \frac{2}{M+1},$$

$$1 < M < 3;$$

en donde  $M$  es el número de estaciones que colisionarán en la segunda ranura de la fase. En otras palabras, en la segunda ranura el número de estaciones restantes no debería ser mayor a 3, para permitir que la fase termine en la siguiente ranura. Esta restricción no es tan grave, ya que si el número de estaciones es mayor, el sistema evoluciona al permitir que un número cada vez menor de estaciones compita por el canal. Elegir  $p = \frac{1}{2}$  cuando el número de estaciones restantes es mayor a 3, disminuye la probabilidad de que el sistema llegue al estado 0, ya que esta probabilidad está a la derecha del intervalo propuesto para  $p$ , en donde quienes se benefician son los estados mayores a 1.

### 3.2.4 Comparación de los datos simulados *vs.* análisis

En los apéndices A.5 y A.6, se determina de forma analítica la longitud promedio de un CRI para el algoritmo original y la propuesta respectivamente. El comportamiento de estas ecuaciones se comparó con los resultados de las simulaciones, en donde se obtuvieron los siguientes resultados mostrados en la figuras 3.13 y 3.14.

La fig. 3.13 muestra los resultados de la simulación, junto con los resultados obtenidos al sustituir  $N$  en la ec. A.42 para el algoritmo 2C original. En este caso se puede observar que existe una gran similitud, lo cual permite suponer que los datos obtenidos mediante simulación son correctos. Asimismo, la ec. A.42 demuestra que el comportamiento del algoritmo 2C original es logarítmico en función del factorial de  $N$ , que en términos prácticos crece más lento que el comportamiento exponencial como se había especulado con anterioridad. Otro punto importante en la ec. A.42, es que cuando  $N$  es pequeño el comportamiento del algoritmo 2C original es casi lineal; de ahí que la política de realimentación por fase, lleve al sistema a un estado mucho menor que el estado dado por la multiplicidad de la colisión al inicio de la fase.

La fig. 3.14 muestra los resultados simulados y de análisis teórico para el algoritmo 2C adapta-

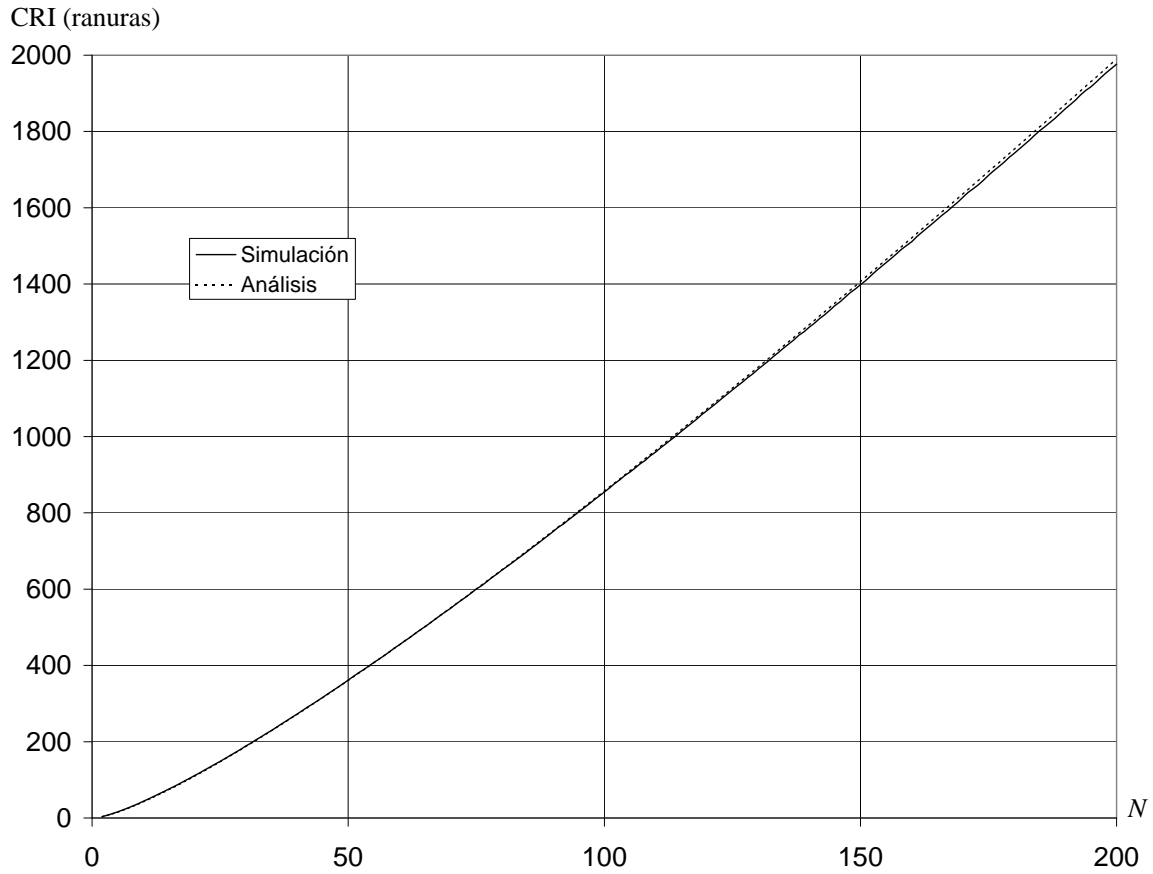


Figura 3.13: Resultados de la simulación y análisis para el algoritmo 2C original. La gráfica muestra la longitud de un CRI *vs*  $N$ .

ble; se puede observar que existe una ligera discrepancia entre los datos simulados y los obtenidos mediante el análisis de la ec. A.50, pero manteniendo el mismo comportamiento. Esto es debido a que el análisis de esta ecuación no resulta sencillo, por lo que la ec. A.50 es en realidad una aproximación. Aun así, el análisis muestra que la longitud de un CRI es lineal, lo cual permite una rápida convergencia del algoritmo.

### 3.3 Ejemplo de estimación de la multiplicidad de la colisión en el algoritmo 2C-adaptable

En esta sección se presenta, a manera de ejemplo, una evaluación de la mejora al algoritmo 2C incluyendo una propuesta para la estimación de la multiplicidad de la colisión. Para este propósito, se

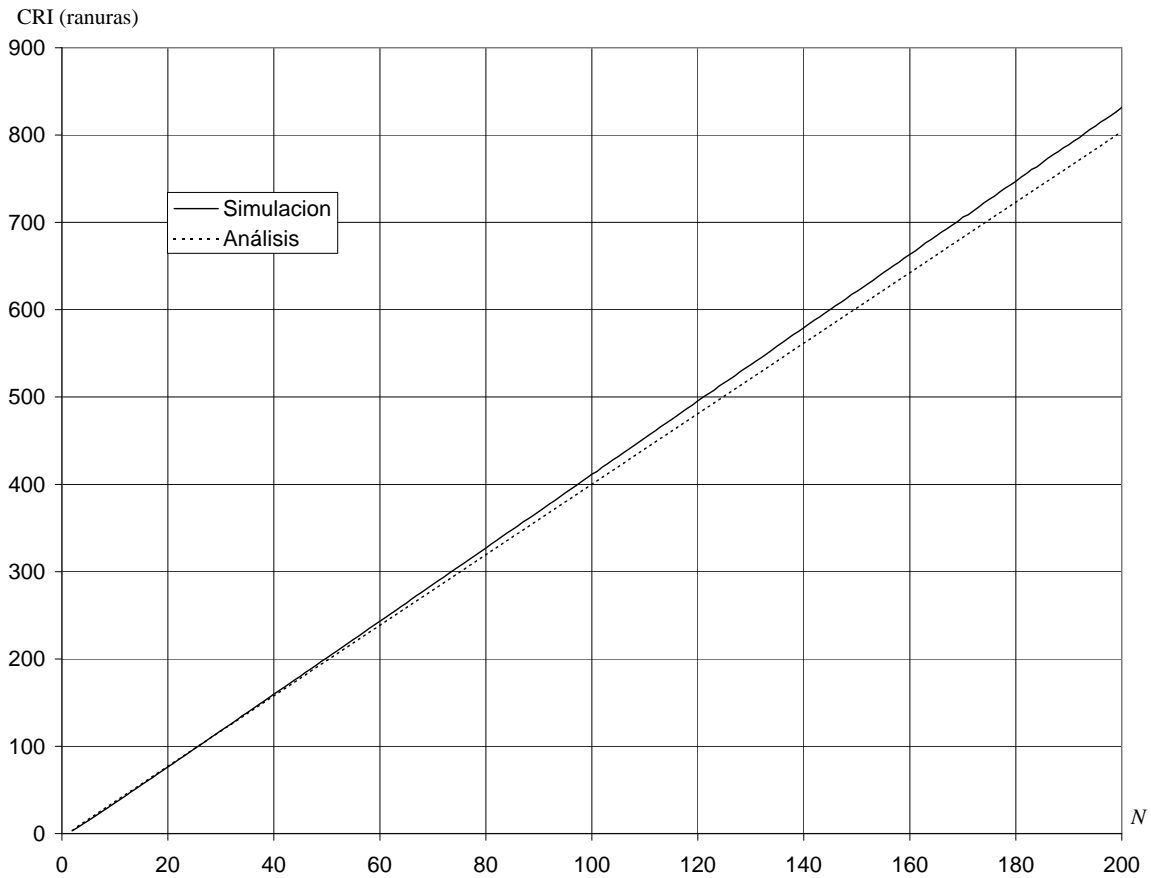


Figura 3.14: Resultados de la simulación y análisis para la política de realimentación por fase (algoritmo 2C adaptable). La gráfica muestra la longitud de un CRI *vs*  $N$ .

emplea un fuente generadora de tráfico junto con la política de realimentación por fase. El método aquí empleado hará uso de un conjunto de muestras pasadas que es capaz de almacenar la estación central, y se usará un proceso de estimación lineal.

Cada muestra del proceso está representada por el número de estaciones que iniciaron un CRI, y que la estación central conoce al final del mismo<sup>9</sup>. El método de estimación propuesto, se basa en estimar el número de estaciones que iniciará un CRI en base al conjunto de muestras pasadas; es decir, el número de estaciones que han iniciado anteriores CRIs servirá para estimar el número de estaciones con que iniciará el próximo CRI.

<sup>9</sup>La estación central simplemente cuenta el número de transmisiones exitosas. Por otra parte, se asume que no existe ruido que pudiese alterar las muestras.

La propuesta resulta bastante sencilla de implementar, ya que la estación central sólo tiene que almacenar un conjunto de muestras (el número de estas muestras lo denotaremos por  $M_{CRI}$ ), y con ellas determinará el número aproximado de estaciones con que iniciará el siguiente CRI.

### 3.3.1 Fuente de tráfico

La fuente de tráfico a usar será del tipo *On-Off* que se emplea para modelar tráfico y usualmente para modelar los paquetes de voz<sup>10</sup>. Para explicar esta fuente, tomemos como ejemplo la voz; ésta se modela por dos estados: uno activo que implica que el hablante ha generado un sonido (*Talk Spurts*), y uno inactivo que corresponde al silencio entre los sonidos que ha generado. Podemos ver esto como conjuntos de paquetes separados por vacíos, ver fig. 3.15.

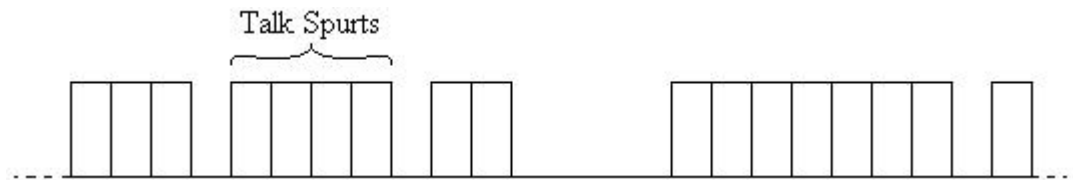


Figura 3.15: Cada conjunto de paquetes se denomina Talk Spurts y corresponden al periodo activo de la voz.

Para el ejemplo se plantea un modelo similar, en este caso las estaciones no generan un solo paquete por arribo, sino un conjunto de paquetes en cada arribo<sup>11</sup>, y éstos deberán formarse en un *buffer de transmisión*. Así los estados de inactividad representan las ranuras en las cuales no hay un arribos, y los estados de actividad representan los paquetes que se han generado por el arribo. El tiempo de permanencia en cada estado está modelado por una distribución geométrica.

La distribución geométrica determina la probabilidad de que suceda un cierto evento  $B$ , después un número  $k$  de experimentos. Esto es, si  $X_B$  es la variable aleatoria (v.a.) que representa el número de experimentos necesarios para que suceda el evento  $B$ , entonces la probabilidad de que la v.a.

<sup>10</sup>Ver [20] pp. 21-58.

<sup>11</sup>De aquí en adelante, un arribo implica que alguna estación ha generado uno o más paquetes para su transmisión.

tome el valor  $k$  está dado por

$$P(X_B = k) = r^{k-1}(1 - r), \quad k \geq 1;$$

en donde  $r$  es la probabilidad de que no suceda el evento  $B$ , y el complemento  $1 - r$  de que suceda.

La esperanza de la v.a.  $X_B$  es

$$E[X_B] = \sum_{k=1}^{\infty} kP(X_B = k) = \frac{1}{1 - r}. \quad (3.19)$$

Las ecuaciones anteriores permitirán fijar los parámetros del modelo. Así, sean  $X_a$  y  $X_p$  dos v.a. que se distribuyen geoméricamente y que representan el número de ranuras para que suceda un arribo, y el número de paquetes que se generan en cada arribo, respectivamente. Sea  $1 - u$  la probabilidad de que se genere un arribo, y  $1 - v$  la probabilidad generar un paquete.

De la ec. 3.19 se tiene que en promedio hay un arribo por cada  $E[X_a] = \frac{1}{1-u}$  ranuras. Por otra parte, en promedio se generan  $E[X_p] = \frac{1}{1-v}$  paquetes en cada arribo. Por lo tanto, estos datos serán de ayuda para modelar la fuente de tráfico, en este caso los parámetros que se emplean son

$$\lambda_a = 1 - u, \quad (3.20)$$

que representa el número promedio de arribos por ranura, y

$$\lambda_p = \frac{1}{1 - v}, \quad (3.21)$$

que representa el número promedio de paquetes por arribo. La definición de estos parámetros, permite determinar las probabilidades  $u$  y  $v$  que se emplearán en cada distribución. Además note que existe una fuente por cada estación, lo que permitirá modelar el tráfico que genera cada una de ellas. Por ejemplo, si en promedio una estación genera un arribo por cada 100 ranuras, y 5 paquetes por arribo, entonces los respectivos parámetros de esa estación son  $\lambda_a = 0.01$  y  $\lambda_p = 5$ ; en donde las respectivas probabilidades serán  $u = 0.99$  y  $v = 0.8$ .

Los parámetros dados por las ecuaciones 3.20 y 3.21, permitirán en todo caso determinar el número promedio de paquetes que se generan por cada ranura; esto es,

$$\lambda_e = \lambda_a \lambda_p = \frac{1 - u}{1 - v}, \quad (3.22)$$

representa el número promedio de paquetes por ranura que genera cada estación. Con la ecuación anterior es posible obtener el promedio de paquetes que se generan y que entran al sistema; es decir, en base al número de estaciones que hay en el sistema (definido por  $N_E$ ), es posible determinar el número promedio de paquetes que se generan por ranura ( $\lambda$ ), y que entrarán a competir por el canal dado por

$$\lambda = N_E \lambda_a \lambda_p = N_E \left( \frac{1 - u}{1 - v} \right). \quad (3.23)$$

Regresando al ejemplo anterior, si hay 10 estaciones ( $N_E = 10$ ) entonces en promedio las 10 fuentes de cada estación le generan al sistema una razón de  $\lambda = 0.5$  paquetes/ranura.

Cada conjunto de paquetes de cada arribo, se genera en instantes arbitrarios de tiempo; es decir, en cualquier ranura puede existir un arribo representado por un conjunto de paquetes listos a transmitirse.

### 3.3.2 Estimador de la multiplicidad de la colisión

Existen muchos métodos para realizar una estimación en base a un conjunto de datos, en este caso proponemos un estimador lineal, que se basa en *minimizar el error cuadrático medio*. Básicamente se busca dar solución al sistema de ecuaciones Yule-Walker [16]; es decir, encontrar los coeficientes  $a_i$  de las ecuaciones 3.24 y 3.25, que permitiría estimar la multiplicidad de la colisión como una combinación lineal de muestras pasadas. Así, si se conocen un conjunto de muestras pasadas  $\{Y[n -$

$i]$ ,  $i = 1, 2, \dots, M_{CRI}$  y se desea estimar la v.a  $Y$  al  $n$ -ésimo CRI ( $\hat{Y}[n]$ ) se tiene

$$\begin{bmatrix} r[0] & r[-1] & \dots & r[-M_{CRI} + 1] \\ r[1] & r[0] & \dots & r[-M_{CRI} + 2] \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ r[M_{CRI} - 1] & r[M_{CRI} - 2] & \dots & r[0] \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \cdot \\ \cdot \\ \cdot \\ a_{M_{CRI}} \end{bmatrix} = - \begin{bmatrix} r[0] \\ r[1] \\ \cdot \\ \cdot \\ \cdot \\ r[M_{CRI}] \end{bmatrix}; \quad (3.24)$$

en donde  $M_{CRI}$  se le conoce como el *orden del estimador*, y  $r[k]$  es la autocorrelación de la señal  $Y[i]$ , la cual se define como

$$r[k] = E[(Y[t+k])(Y[k])].$$

La combinación lineal para estimar la muestra  $\hat{Y}[n]$  es

$$\hat{Y}[n] = -a_1 Y[n-1] - a_2 Y[n-2] - \dots - a_{M_{CRI}} Y[n-M_{CRI}], \quad (3.25)$$

En general existen varios algoritmos para dar solución al sistema 3.24; sin embargo, en la simulación simplemente obtendremos la *matriz diagonal escalonada* del sistema para así determinar los coeficientes.

Uno de los inconvenientes al emplear este estimador es que éste es válido para señales *estocásticas estacionarias* en sentido amplio; es decir, señales  $Y[n]$  que cumplen [16]:

1. La media  $\mu_Y[x] = E(Y[n])$  es constante.
2. La autocorrelación es función solamente de la diferencia de tiempo entre las dos muestras del proceso.

Recordemos que cada muestra  $Y[i]$  representa el número de estaciones que inician el CRI  $i$ , y esta v.a. toma valores entre 0 y  $N_E$ , por lo que la media de la v.a. es un número comprendido en el intervalo  $[0, N_E]$ . Cada muestra  $Y[i]$  está en función del número de estaciones que tienen al menos un paquete por transmitir, y de los arribos de nuevos paquetes. Es posible asumir que la v.a.  $Y[i]$  posee una media constante, ello se debe a que el número de estaciones en el sistema es constante

y que sus distribuciones son de parámetros constantes. Lo anterior resulta válido a intervalos de tiempo, ya que pueden ocurrir varios arribos a lo largo de un CRI, lo que resulta en un incremento en el valor de la v.a. para el siguiente CRI.

Lo que no es tan evidente de determinar es que la v.a  $Y[i]$  posea autocorrelación invariante en el tiempo. De igual forma, es posible que la v.a. tenga un comportamiento similar a lo largo de ciertos intervalos de tiempo, lo que permita que la v.a posea en esos intervalos una autocorrelación que sólo dependa de la diferencia de tiempos.

Finalmente, un punto importante es que también es necesario estimar la autocorrelación en base al conjunto de muestras que se tiene. Para ello se utiliza la autocorrelación muestral [16]

$$r[m] = \frac{1}{M} \sum_{i=0}^{M-1-|m|} Y[i]Y[i+|m|], \quad 0 \leq m \leq M_{CRI};$$

en donde  $M$  representa el conjunto total de muestras que se tiene.

### 3.3.3 Resultados de la simulación con el estimador de la multiplicidad de la colisión

Para la simulación es necesario proponer un valor para las siguientes variables:

- $\lambda_a$     Número de arribos por ranura,
- $\lambda_p$     Número de paquetes por arribo,
- $M_{CRI}$     Orden del estimador, y
- $M$     Número total de muestras pasadas.

Por lo tanto se usó un total de  $N_E = 100$  estaciones, las cuales generan en promedio  $\lambda_p = 3$  paquetes por arribo. Con estos dos datos y con  $\lambda$ , estamos en posibilidades de seleccionar los demás parámetros; así, si la razón de paquetes por ranura puede tomar valores entre  $(0.02, 0.40)$ , entonces,  $\lambda_a$  tomará los valores que se muestran en la tabla 3.2 para cada valor de  $\lambda$  propuesto.

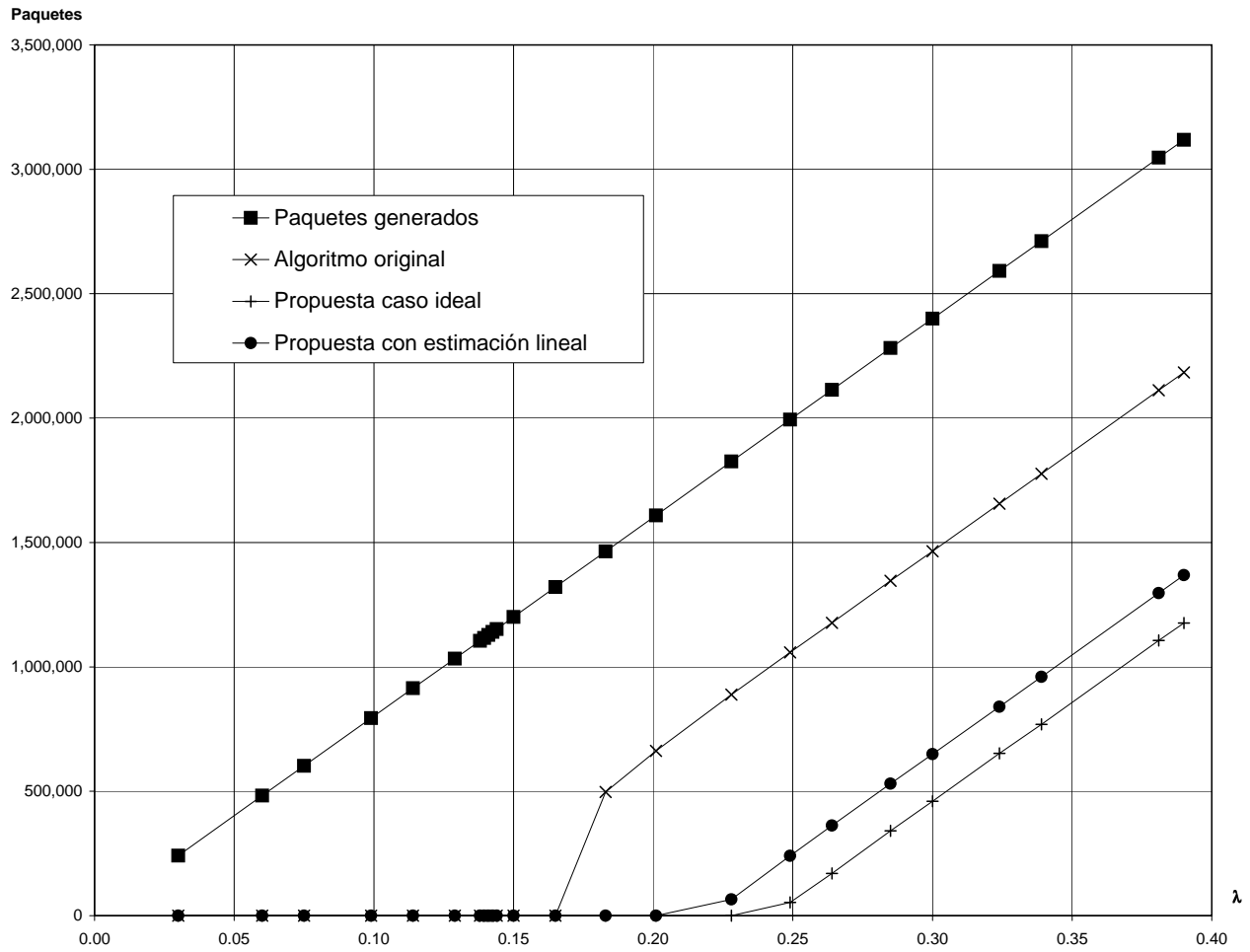


Figura 3.16: Número de paquetes generados y paquetes en espera de ser transmitidos en la simulación *vs.*  $\lambda$ .

El intervalo elegido para  $\lambda$ , se seleccionó en base a lo empleado en [15], en donde la razón de paquetes por ranura alcanzaba un máximo valor de 0.40, esto proporcionó un punto de partida para realizar una comparación de la propuesta con el algoritmo original.

Por otra parte, es necesario fijar el orden del estimador, y el total de muestras pasadas. En este caso, emplearemos un estimador de orden 3. Estos valores se eligieron en base a resultados de simulaciones, ya que al incrementarlos no se apreciaron cambios significativos en los resultados; sin embargo, esta observación depende de la fuente de tráfico propuesta, por lo que no necesariamente pueden ser los adecuados para otra fuente. Finalmente, la simulación tuvo una duración de 8,000,000 de ranuras; de igual forma se hicieron las modificaciones apropiadas al simulador descrito

Tabla 3.2: Parámetros de la simulación.

$\lambda$	0.0300	0.0600	0.0750	0.0990	0.1140	0.1290	0.1380	0.1395	0.1410	0.1425
$\lambda_a \times 10^{-3}$	0.100	0.200	0.250	0.330	0.380	0.430	0.460	0.465	0.470	0.475

$\lambda$	0.1440	0.1500	0.2010	0.2280	0.2490	0.2640	0.2850	0.3000	0.3240	0.3390
$\lambda_a \times 10^{-3}$	0.480	0.500	0.670	0.760	0.830	0.880	0.950	1.000	1.080	1.130

$\lambda$	0.3810	0.3900
$\lambda_a \times 10^{-3}$	1.270	1.300

en el apéndice B, a fin de ejecutar una simulación continua con 100 estaciones en donde cada una intenta transmitir con una fuente de tráfico *On-Off*.

En este caso la longitud promedio de un CRI no fue nuestro parámetro comparativo, en lugar de ello se empleó el número de paquetes no transmitidos durante la simulación. La fig. 3.16 muestra el número de paquetes generados durante la simulación, y el número de paquetes no transmitidos por cada propuesta. Por ejemplo cuando  $\lambda = 0.25$  paquetes/ranura (el sistema recibe en promedio un paquete cada cuatro ranuras), el número de paquetes generados durante toda la simulación fue de alrededor 2,000,000; el algoritmo original no pudo transmitir alrededor de 1,000,000 paquetes; mientras que la propuesta 2C adaptable con estimación perfecta (caso ideal en donde se conoce de manera exacta la multiplicidad de la colisión), no logró transmitir 100,000 paquetes; finalmente la propuesta 2C adaptable con estimación de la multiplicidad de la colisión no logró transmitir 300,000 paquetes.

Como se puede observar del a fig. 3.16 a partir de  $\lambda = 0.18$  paquetes/ranura, el algoritmo original no logra transmitir el 35% de los paquetes generados durante la simulación. En cambio el algoritmo 2C adaptable es capaz de transmitir el 100% de sus paquetes a una razón de  $\lambda = 0.20$  paquetes/ranura.

Una característica importante de la propuesta, es que en caso de un ráfaga de paquetes, el tiempo de respuesta ante esta eventualidad sería más corto en comparación con el algoritmo original; es decir, si en algún instante de tiempo un gran número de estaciones intenta transmitir, entonces la longitud

de un CRI en el algoritmo original requeriría casi del doble de tiempo que el que se emplea en las propuestas. La fig. 3.17 muestra el retardo promedio de cada paquete, observe que cuando  $\lambda > 0.16$  paquetes/ranura al algoritmo original le toma en promedio 480 ranuras transmitir un paquete; desde luego, a partir de este punto el algoritmo ha dejado de ser estable. Las propuestas tienen un retardo promedio de alrededor de 220 ranuras; así en una ráfaga de paquetes, las propuestas se estabilizarán más rápidamente que en el algoritmo original.

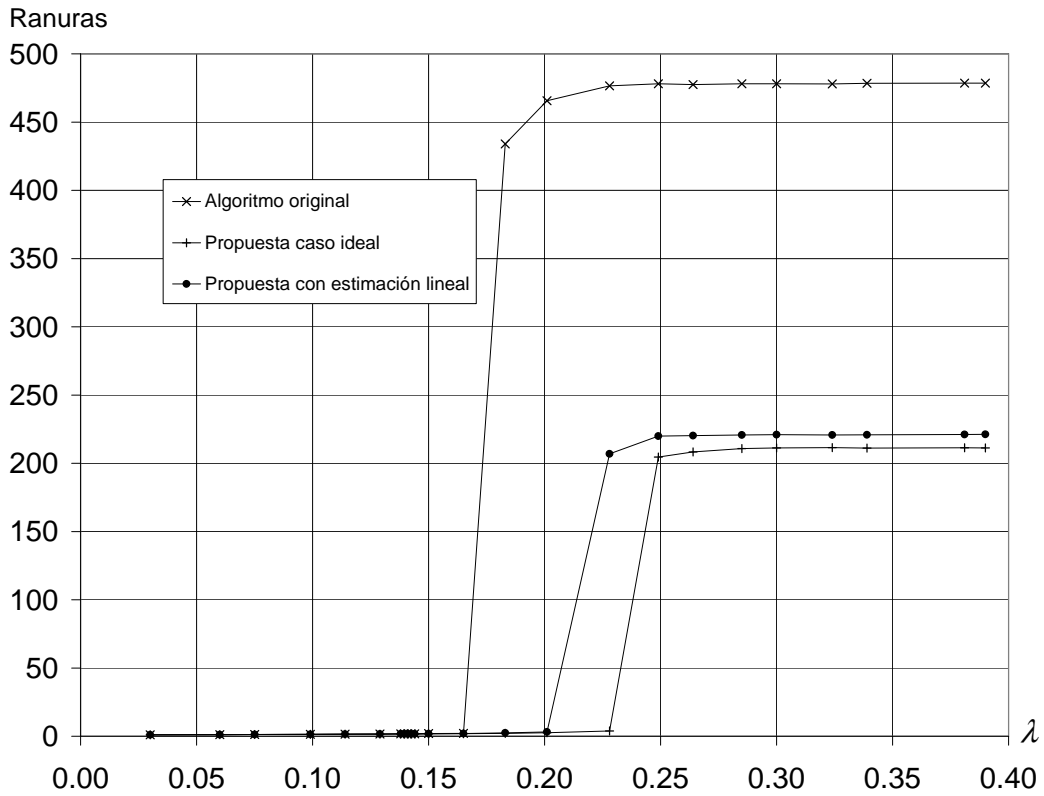


Figura 3.17: Retardo promedio de cada paquete vs.  $\lambda$ .



# Capítulo 4

## Conclusiones y recomendaciones para trabajo futuro

---

Se presentó una propuesta de mejora al algoritmo de control de acceso al medio 2C. Se denominó a la propuesta con el nombre de 2C-adaptable. El algoritmo 2C emplea una probabilidad de transmisión o de cambio de estado  $p$  fija y de valor  $\frac{1}{2}$ , lo que produce que la resolución de colisiones tome un largo tiempo. Se demostró que una mejora sustancial al algoritmo se puede dar si se elige una probabilidad  $p$  óptima en función del número de estaciones en contienda, como se mostró en la fig. 3.5. Sin embargo, es adecuado emplear el valor de la probabilidad en  $p = \frac{1}{2}$ , cuando las multiplicidades de colisión son pequeñas (de alrededor de 3 estaciones), lo cual se determinó de la fig 3.7 y la ec. A.42. Dada las características de la ec. A.42, esta predice que para valores pequeños de  $N$ , el comportamiento es casi lineal.

La mejora al algoritmo se basó en considerar que la longitud de un CRI está en función de la multiplicidad de la colisión, y dado que esta longitud está en función de ese número, es necesario conocerlo. La mejora asume que es posible conocer la multiplicidad de la colisión por algún medio, y que la estación central es capaz de transmitir este valor a las estaciones en contienda. Este problema permitió proponer dos políticas que permitieron mejorar el algoritmo.

Debido a que es adecuado emplear  $p = \frac{1}{2}$  cuando el número de estaciones es pequeño, entonces la propuesta principal denominada *algoritmo 2C adaptable* (política de realimentación por fase), se

basó en disminuir el número de estaciones a la siguiente ranura donde sucedía la colisión; para ello un CRI se fragmentó en intervalos de tiempo denominados *fases*, y al inicio de una fase, la estación central difunde la multiplicidad de la colisión para que cada estación en contienda emplee  $p = p_m(N)$  (ec. 3.11); esto permite disminuir el número de estaciones en contienda a la siguiente ranura de la fase. De esta forma los cambios al algoritmo original resultarían mínimos, y se emplearía la misma infraestructura de comunicaciones.

El empleo de la probabilidad  $p = p_m(N)$ , no sólo tiene la ventaja de llevar al sistema a que sólo una estación transmita en la primera ranura, sino que además permite un amplio intervalo de error, lo cual es importante al momento de definir el algoritmo de predicción para la multiplicidad de la colisión. El error se discutió en la sección 3.1.3, y se concluyó que subestimar de la multiplicidad de la colisión de hasta un 25%, mientras que se puede sobrestimar alrededor del 50%, del valor real de la multiplicidad de la colisión. Sin embargo, no sólo este criterio es aplicado al momento de seleccionar el algoritmo predicción, al dividir un CRI en fases la estación central podrá determinar la multiplicidad de la colisión al inicio del CRI o al inicio de cada fase, ello dependerá del grado de confianza que tenga el predictor empleado.

Debido a la forma en que se planteó la propuesta, se especula que esta idea puede tener un rendimiento similar a la utilización de una ventana de captura óptima como se plantea en [15], con la diferencia de que la ventana de captura no se ajusta al número de estaciones en contienda; no fue posible hacer una comparación entre el algoritmo 2C-adaptable y el algoritmo 2C con ventana de captura, debido a la dificultad que presenta la doble sincronización en cada estación, ya que ahora hay dos ranuras de tiempo debido a la introducción de la ventana de captura. La propuesta 2C adaptable permite atender un número grande de estaciones en menos tiempo, que el que toma el algoritmo original.

Aun cuando se pueda considerar que el algoritmo 2C con ventana de captura y la propuesta 2C adaptable tienen similitudes, el 2C-adaptable integra su propia "ventana de captura adaptable" dentro del algoritmo (lo cual se logra con un pequeño cambio), y no así para el 2C en [15], en donde es necesario implementar la ventana de captura que en principio es de longitud fija, y no en función

de la carga que se aplica al sistema; además, de que hay que especificar la forma en que serán seleccionadas las estaciones que tendrá derecho a competir por el canal, ya que esto no es una tarea fácil. La propuesta permite esto último, al emplear la propia realimentación que genera la estación central.

Por otra parte, a manera de ejemplo se propuso un algoritmo de predicción lineal, para estimar la multiplicidad de la colisión. Las simulaciones del algoritmo 2c-adaptable con estimación lineal, confirmó que los errores debidos a la estimación no afectaron en gran medida el desempeño de la propuesta, al mostrar un comportamiento semejante al 2C-adaptable ideal en donde se asume que se conoce de manera exacta la multiplicidad de la colisión. Esto no es un argumento contundente, pero muestra las capacidades de la propuesta ante los errores introducidos al estimar la multiplicidad de la colisión.

Como trabajo futuro, se presentan un par de recomendaciones las cuales se mencionan a continuación:

- La propuesta asume que la estación central conoce la multiplicidad de la colisión, el trabajo muestra un ejemplo de como determinar este número con ayuda de un algoritmo de predicción lineal, pero es posible explorar otros mecanismos que permitan conocer este valor.
- Es conveniente evaluar el algoritmo 2C adaptable en un escenario real, y determinar con mayor precisión las prestaciones del mismo, al compararlo con algoritmos que se emplean en la actualidad, especialmente con CSMA.

Finalmente, el presente trabajo dio lugar a la publicación y presentación oral de un artículo de investigación [2] en Canadadá, en el congreso: "*Canadian Conference on Electrical and Computer Engineering*", llevada a cabo en el mes de abril del presente año en Vancouver.



# Apéndice A

## Desarrollo y análisis de las ecuaciones para las políticas de realimentación por ranura y fase

---

### A.1 Intervalo de $p$ para el análisis de realimentación por ranura

El objetivo es determinar un intervalo para  $p$  que permita garantizar  $P_{N,1}(p) > P_{N,K}(p)$ , dados por la ec. 3.1, para  $K = \{0, 2, 3, \dots, N\}$ . Este intervalo asegura que la probabilidad de llegar al estado 1 deber ser mayor que la probabilidad de llegar a cualquier otro estado; además de que permitiría fijar un intervalo para el cual es posible admitir un determinado grado de error, ya que es imposible conocer la multiplicidad de la colisión con exactitud.

El primer paso es determinar qué valor de  $p$  cumple la desigualdad  $P_{N,K}(p) > P_{N,(K+1)}(p)$ , para  $0 \leq K \leq N$ ; de la ec. 3.1 se tiene

$$P_{N,K}(p) > P_{N,(K+1)}(p),$$
$$\binom{N}{K} p^K (1-p)^{N-K} > \binom{N}{K+1} p^{K+1} (1-p)^{N-K-1},$$

$$\frac{K+1}{N-K} > \frac{p}{1-p},$$

$$p < \frac{K+1}{N+1}.$$

De lo anterior se concluye lo siguiente

$$\text{Si } K = 0 \text{ entonces } P_{N,0}(p) > P_{N,1}(p) \text{ cuando } p < \frac{1}{N+1},$$

$$\text{si } K = 1 \text{ entonces } P_{N,1}(p) > P_{N,2}(p) \text{ cuando } p < \frac{2}{N+1}.$$

Lo anterior fija una intervalo para  $p$ ; es decir,  $P_{N,1}(p) > P_{N,2}(p)$  y  $P_{N,1}(p) > P_{N,0}(p)$  sólo si  $p \in (\frac{1}{N+1}, \frac{2}{N+1})$ . Con respecto a las probabilidades de los demás estados, no existe problema alguno, ya que  $\frac{2}{N+1} < \frac{K+1}{N+1}$ , con  $K > 1$ ; es decir,  $P_{N,1}(p) > P_{N,m}(p)$  para  $m \geq 2$  en el intervalo antes propuesto.

Finalmente, debemos agregar también que se cumple lo siguiente

$$\frac{1}{N+1} < p_M(N) < \frac{2}{N+1},$$

$$\frac{1}{N+1} < p_m(N) < \frac{2}{N+1},$$

$$\frac{1}{N+1} < p_O(N) < \frac{2}{N+1}.$$

El caso excepcional es  $p_D(N)$ , ya que es igual a uno de los extremos; sin embargo, podemos llevar más allá el intervalo al tomarlo cerrado, en este caso se tendría que  $p \in [\frac{1}{N+1}, \frac{2}{N+1}]$ .

Observe que el intervalo antes definido, es una medida de la desviación permitida en caso de que se tenga que estimar el valor de  $N$ ; además note que esa desviación, aparentemente disminuye conforme  $N$  aumenta, pero de igual forma la probabilidad disminuye al aumentar  $N$ , por lo que se espera que el error permitido se estabilice al aumentar  $N$ .

El intervalo antes calculado, determina los casos extremos que la probabilidad puede tomar cuando se estima  $N$ ; es decir, cuando la probabilidad  $p$  esta cerca del valor de  $\frac{1}{N+1}$ , la probabilidad de llegar al estado 0 será casi la misma que la de llegar al estado 1; en cambio cuando  $p$  esta más cerca del valor  $\frac{2}{N+1}$ , entonces la probabilidad de llegar al estado 2 será la misma que la probabilidad de llegar al estado 1. La opción más conveniente es la última, ya que permite que el sistema evolucione favorablemente al ir a un estado mayor a 1, y no al estado 0 que involucraría repetir el proceso.

## A.2 Probabilidad por rama para el árbol de transiciones

Estamos interesados en determinar la probabilidad de que el sistema tome una cierta rama, del árbol de transiciones mostrado en la fig. 3.10. En términos generales, la probabilidad de que el sistema llegue al estado  $i$ , a partir de un estado cualquiera  $j$  en  $n$  transiciones. Sin embargo, es posible manipular sólo dos transiciones, y extender el proceso a una mayor cantidad.

Sea  $A_m$  el evento de que el sistema tome el estado  $m$ <sup>1</sup>, en donde  $m \in \mathbb{N}$ , con  $m \geq 2$ . Suponga que el sistema llega al estado  $j$  con probabilidad  $P(A_j)$ , ¿cuál es la probabilidad de que el sistema llegue al estado  $i$ , pasando por los estados  $j$  y  $k$ , decir  $P(A_j, A_k, A_i)$  (ver fig. A.1)? El problema involucra probabilidad condicional, debido a que es necesario que el sistema haya pasado por un determinado estado, así entonces se tiene

$$\begin{aligned} P(A_i/A_j, A_k) &= \frac{P(A_j, A_k, A_i)}{P(A_j, A_k)}; \\ P(A_k/A_j) &= \frac{P(A_j, A_k)}{P(A_j)}. \end{aligned}$$

Los términos de la izquierda de ambas ecuaciones, están dados por la ec. 3.1, esto es  $P(A_k/A_j) = P_{k,j}(p)$  y  $P(A_i/A_j, A_k) = P_{j,i}(p)$ , debido a que esa es la probabilidad de transitar a partir de un estado dado. Sustituyendo ambas ecuaciones se tiene

$$P(A_j, A_k, A_i) = P_{k,j}(p)P_{j,i}(p)P(A_j). \quad (\text{A.1})$$

---

<sup>1</sup>En principio este evento debería contemplar el estado de procedencia, pero ello no es necesario, ya que se considera que los estados están dados.

Si el estado  $j$  fuera la raíz del árbol, entonces su probabilidad sería igual a 1, ya que el sistema parte de un estado determinado; si por el contrario  $j$  es un estado intermedio, entonces habrá que determinar la probabilidad de que el sistema llegue al estado  $j$  a partir de estados anteriores. Independientemente de lo anterior, lo que establece la ec. A.1 es que la probabilidad de que el sistema pase por  $n$  estados, es igual al producto de las probabilidad de transición entre dichos estados.

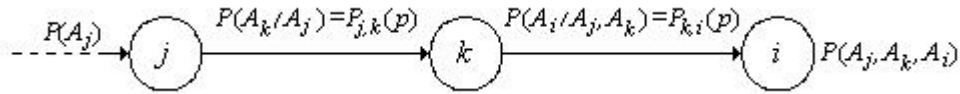


Figura A.1: Probabilidad para cada rama del árbol de transiciones.

Por otra parte, cada rama forma parte de un evento que es ajeno a otro; es decir, el sistema puede llegar al mismo estado por ramas diferentes de igual número de transiciones. Cada evento  $A_m$  es ajeno a otros eventos (i. e.  $A_m \cap A_l = \emptyset$ ,  $m \neq l$ ), el sistema no puede tomar dos estados al mismo tiempo, de ahí que cada rama sea única. Así entonces, la probabilidad de que el sistema llegue al estado  $i$  en  $n$  transiciones, es la suma de la probabilidad de cada rama que lleva al sistema al estado  $i$  en  $n$  transiciones. La ec. A.1 sólo toma en cuenta una rama, por lo que la probabilidad total es la suma sobre los índices de la ecuación; esto será tomado en cuenta en la siguiente sección.

### A.3 Probabilidad de estados en un árbol de transiciones

Sea  $Q_{N,i}^{(n)}(p)$  la probabilidad de llegar al estado  $i$  en  $n$  transiciones a partir del estado  $N$  en un árbol de transiciones; para  $i, n \in \mathbb{N}$ ,  $0 \leq i \leq N$  y  $n \geq 1$ ;  $N$  es la multiplicidad de la colisión al inicio de la fase. Para este caso, se presenta el árbol de transiciones en la fig. A.2, en donde los estados 0 y 1 son terminales. De acuerdo con la sec. A.2, la probabilidad de cada rama está dada por el producto de las probabilidades de transición; así pues, determinaremos  $Q_{N,i}^{(n)}(p)$  para algunos valores de  $n$ , ofreceremos una ecuación general, y la demostraremos empleando inducción en la siguiente sección del presente Apéndice. Para  $n = 1$  es fácil ver que  $Q_{N,k}^{(1)}(p) = P_{N,k}(p)$ , para  $0 \leq k \leq N$ ; es decir, como sólo hay una transición se resume al caso de la política de difusión por ranura, dado por la ec. 3.1.

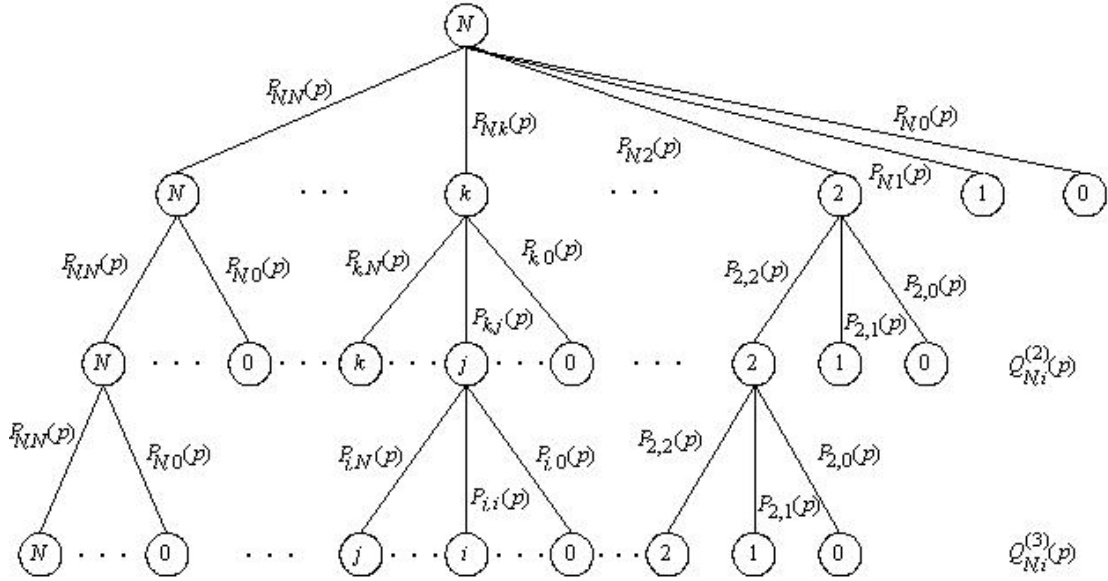


Figura A.2: Árbol de transiciones de nivel  $n = 3$ , que representa la profundidad del árbol; cada rama tiene asociada su probabilidad de transición dada por la ec. 3.1.

Para  $n = 2$  existen dos transiciones: de  $N$  a  $k$  y de  $k$  a  $i$ , para  $2 \leq k \leq N$  y  $0 \leq i \leq k$ . Cada transición tiene restricciones, ya que no es posible llegar a un estado  $i$  tal que  $i > k$  (el número de estaciones en la celda de transmisión siempre es decreciente o igual), lo cual es válido para todo  $k \geq 2$  ya que los estados 0 y 1 son terminales. Con lo anterior, la probabilidad de transitar al estado  $i$  en 2 transiciones desde el estado  $N$  por una sóla rama es

$$\bar{Q}_{N,i}^2 = P_{N,k}(p)P_{k,i}(p), \quad \text{en donde } \text{Max}(2, i) \leq k \leq N \quad \text{y} \quad 0 \leq i \leq N. \quad (\text{A.2})$$

Por lo que la probabilidad  $Q_{N,i}^{(2)}(p)$  es la suma de todas las ramas que llevan al estado  $i$  en 2 saltos (ver sec. A.2), en donde el índice de la suma es controlado por  $k$ .

Definimos la función  $S(i)$  como:

$$S(i) \triangleq \text{Max}(2, i). \quad (\text{A.3})$$

La función  $S(i)$  permite expresar las restricciones que produce el valor de  $i$ , por lo tanto  $Q_{N,i}^{(2)}(p)$  queda como

$$Q_{N,i}^{(2)}(p) = \sum_{k=S(i)}^N P_{N,k}(p)P_{k,i}(p), \quad \text{para } 0 \leq i \leq N. \quad (\text{A.4})$$

Si expresaremos en otra forma el término dentro de la suma al colocar la ec. 3.1, se obtiene

$$\begin{aligned}
P_{N,k}(p)P_{k,i}(p) &= \binom{N}{k} \binom{k}{i} p^{k+1} (1-p)^{N-k} (1-p)^{k-i} \\
&= \frac{\binom{N}{k} \binom{k}{i}}{\binom{N}{i}} p^k P_{N,i}(p) \\
&= \binom{N-i}{N-k} p^k P_{N,i}(p) \\
&= \binom{N-i}{k-i} p^k P_{N,i}(p)
\end{aligned} \tag{A.5}$$

Sustituyendo A.5 en A.4 se tiene

$$Q_{N,i}^{(2)}(p) = P_{N,i}(p) \sum_{k=S(i)}^N \binom{N-i}{k-i} p^k.$$

Se tiene algo muy semejante al binomio de Newton, por lo que con el cambio de variable  $m = k - i$  se obtiene

$$Q_{N,i}^{(2)}(p) = p^i P_{N,i}(p) \sum_{m=S(i)-i}^{N-i} \binom{N-i}{m} p^m. \tag{A.6}$$

Manipularemos un poco la ecuación anterior, con la finalidad de obtener términos más sencillos.

Sea  $i < 2$ , por lo que la función  $S(i)$  toma el valor de 2 y por tanto

$$Q_{N,i}^{(2)}(p) = p^i P_{N,i}(p) \sum_{m=2-i}^{N-i} \binom{N-i}{m} p^m;$$

en este caso  $i$  puede tomar dos valores 0 o 1, ello afectará a los términos que hagan falta para completar el binomio de Newton. Definimos la función indicadora  $\delta(i)$  como

$$\delta(i) = \begin{cases} 0 & \text{para } i \neq 0, \\ 1 & \text{para } i = 0. \end{cases} \tag{A.7}$$

La función indicadora permite seleccionar los términos que hacen falta para completar el binomio de Newton, de esta forma se tiene

$$\begin{aligned} Q_{N,i}^{(2)}(p) &= p^i P_{N,i}(p) \left[ \sum_{m=0}^{N-i} \binom{N-i}{m} p^m - 1 - \delta(i) Np \right] \\ &= p^i P_{N,i}(p) \left[ (1+p)^{N-i} - 1 - \delta(i) Np \right]. \end{aligned} \quad (\text{A.8})$$

Para el caso  $i \geq 2$ , la función  $S(i)$  toma el valor de  $i$  dando como resultado el binomio de Newton por lo que la ec. A.6 es ahora

$$Q_{n,i}^{(2)}(p) = p^i P_{N,i}(p) \sum_{m=0}^{N-i} \binom{N-i}{m} p^m = p^i P_{N,i}(p) (1+p)^{N-i}. \quad (\text{A.9})$$

Finalmente, al unir las ecuaciones A.8 y A.9 se tiene:

$$Q_{N,i}^{(2)}(p) = p^i P_{N,i}(p) \begin{cases} [(1+p)^{N-i} - 1 - \delta(i) Np] & \text{para } i < 2, \\ (1+p)^{N-i} & \text{para } i \geq 2; \end{cases} \quad 0 \leq i \leq N. \quad (\text{A.10})$$

Para  $n = 3$  el procedimiento es el mismo que antes; tomemos el caso de  $\bar{Q}_{N,i}^{(3)}(p)$  que se muestra en la fig. A.2, en donde los saltos que puede dar el sistema son: de  $N \rightarrow k \rightarrow j \rightarrow i$ . Al igual que en el caso de  $n = 2$ , existen algunas restricciones sobre  $i$ , ya que no es posible llegar a  $i$  si  $i > j$ , y de igual forma no es posible si  $i > k$ , salvo en el caso en que  $i = 0$  o  $i = 1$ . Lo anterior son los límites inferiores de los índices, como límites superiores se tiene que  $k \leq N$ ,  $j \leq K$  e  $i \leq j$ . De lo anterior la ecuación para  $\bar{Q}_{N,i}^{(3)}(p)$  es

$$\bar{Q}_{N,i}^{(3)}(p) = P_{N,k}(p) P_{k,j}(p) P_{j,i}(p), \quad \text{con } S(i) \leq k \leq N, \quad S(i) \leq j \leq k \quad \text{y} \quad 0 \leq i \leq N. \quad (\text{A.11})$$

De igual manera, la ecuación anterior es válida para una rama. Para tomar en cuenta todas las ramas se tiene una doble sumatoria, por lo tanto la ec. A.11 queda como

$$Q_{N,i}^{(3)}(p) = \sum_{k=S(i)}^N \sum_{j=S(i)}^k P_{N,k}(p) P_{k,j}(p) P_{j,i}(p), \quad \text{con } 0 \leq i \leq N. \quad (\text{A.12})$$

Al reacomodar la ecuación anterior, se observa que hay un termino similar a  $Q_{N,i}^{(2)}$ , excepto que se tiene  $k$  en lugar de  $N$ ; sustituyendo A.10 en la ec. A.12 se tiene

$$Q_{N,i}^{(3)}(p) = p^i \sum_{k=S(i)}^N P_{N,k}(p) P_{k,i}(p) \begin{cases} [(1+p)^{k-i} - 1 - \delta(i)kp] & \text{para } i < 2, \\ (1+p)^{k-i} & \text{para } i \geq 2; \end{cases} \quad 0 \leq i \leq N. \quad (\text{A.13})$$

Sustituyendo la ec. A.5 y haciendo el cambio de variable  $m = k - i$  se tiene

$$Q_{N,i}^{(3)}(p) = p^{2i} P_{N,i}(p) \sum_{m=S(i)-i}^{N-i} \binom{N-i}{m} p^m \begin{cases} [(1+p)^m - 1 - \delta(i)mp] & \text{para } i < 2, \\ (1+p)^m & \text{para } i \geq 2; \end{cases} \quad 0 \leq i \leq N. \quad (\text{A.14})$$

Tenemos dos casos según el valor de  $i$ . Sea  $i < 2$ , por lo que la ecuación anterior queda como

$$\begin{aligned} Q_{N,i}^{(3)}(p) &= p^{2i} P_{N,i}(p) \sum_{m=2-i}^{N-i} \binom{N-i}{m} p^m \{(1+p)^m - 1 - \delta(i)mp\} \\ &= p^{2i} P_{N,i}(p) \sum_{m=2-i}^{N-i} \binom{N-i}{m} \{(p+p^2)^m - p^m - \delta(i)mp^{m+1}\}. \end{aligned} \quad (\text{A.15})$$

Para los dos primeros términos de la suma, es posible completar el binomio de Newton, pero para el tercer término no es fácil hacer lo mismo, además de que sólo es válido cuando  $i = 0$ ; expresemos el último término de la siguiente forma

$$\sum_{m=2}^N \binom{N}{m} mp^{m+1} = N \sum_{m=2}^N \binom{N-1}{m-1} p^{m+1} = Np^2 \sum_{l=1}^{N-1} \binom{N-1}{l} p^l = Np^2 [(1+p)^{N-1} - 1].$$

Con lo anterior, la ec. A.15 queda como

$$\begin{aligned} Q_{N,i}^{(3)}(p) &= p^{2i} P_{N,i}(p) \left[ (1+p+p^2)^{N-i} - 1 - \delta(i)(N-i)(p+p^2) - (1+p)^{N-i} + 1 + \delta(i)(N-i)p \right. \\ &\quad \left. - \delta(i)Np^2[(1+p)^{N-1} - 1] \right] \\ &= p^{2i} P_{N,i}(p) \left[ (1+p+p^2)^{N-i} - (1+p)^{N-i} - \delta(i)Np^2(1+p)^{N-1} \right], \quad i < 2. \end{aligned} \quad (\text{A.16})$$

Cuando  $i \geq 2$  se tiene

$$\begin{aligned} Q_{N,i}^{(3)}(p) &= p^{2i} P_{N,i}(p) \sum_{m=0}^{N-i} \binom{N-i}{m} p^m (1+p)^m \\ &= p^{2i} P_{N,i}(p) (1+p+p^2)^{N-i}, \quad i \geq 2. \end{aligned} \quad (\text{A.17})$$

Así entonces,  $Q_{N,i}^{(3)}(p)$  esta dado por

$$Q_{N,i}^{(3)}(p) = p^{2i} P_{N,i}(p) \begin{cases} [(1+p+p^2)^{N-i} - (1+p)^{N-i} - \delta(i)Np^2(1+p)^{N-1}], & i < 2, \\ (1+p+p^2)^{N-i}, & i \geq 2; \end{cases} \quad 0 \leq i \leq N. \quad (\text{A.18})$$

Podemos seguir con el análisis, y encontrar que por ejemplo para  $n = 4$  se tiene

$$Q_{N,i}^{(4)}(p) = p^{3i} P_{N,i}(p) \begin{cases} [(1+p+p^2+p^3)^{N-i} - (1+p+p^2)^{N-i} - \delta(i)Np^3(1+p+p^2)^{N-1}], & i < 2, \\ (1+p+p^2+p^3)^{N-i}, & i \geq 2. \end{cases} \quad (\text{A.19})$$

Con las ecuaciones A.10, A.18 y A.19, es posible determinar una formula general que más adelante demostraremos por inducción. La ecuación para  $Q_{N,i}^n$  es entonces

$$\begin{aligned} Q_{N,i}^{(n)} &= p^{(n-1)i} P_{N,i}(p) \begin{cases} [(\sum_{k=0}^{n-1} p^k)^{N-i} - (\sum_{k=0}^{n-2} p^k)^{N-i} - \delta(i)Np^{n-1}(\sum_{k=0}^{n-2} p^k)^{N-1}], & i < 2, \\ (\sum_{k=0}^{n-1} p^k)^{N-i}, & i \geq 2. \end{cases} \\ &= \binom{N}{i} p^{ni} \begin{cases} [(1-p^n)^{N-i} - (1-p^{n-1})^{N-i} - \delta(i)Np^{n-1}q(1-p^{n-1})^{N-1}], & i < 2, \\ (1-p^n)^{N-i}, & i \geq 2; \end{cases} \end{aligned} \quad (\text{A.20})$$

en donde  $0 \leq i \leq N$ ,  $n \geq 1$  y  $q = 1 - p$ . La ec. A.20 representa la probabilidad de llegar al estado  $i$  en  $n$  saltos o transiciones a partir de un estado  $N$ . Desde luego,  $N$  puede ser cualquier estado tal que  $N \geq 2$ ; en la siguiente sección se demostrara la validez de la ec. A.20.

## A.4 Demostración de la ecuación A.20

El método que se empleara para demostrar la ec. A.20 es por inducción. Sea  $n = 1$ , sustituyendo en A.20 se tiene

$$\begin{aligned} Q_{N,i}^{(1)}(p) &= \binom{N}{i} p^i \begin{cases} [(1-p^1)^{N-i} - (1-p^{1-1})^{N-i} - \delta(i)Np^{1-1}(1-p)(1-p^{1-1})^{N-1}], & i < 2, \\ (1-p^1)^{N-i}, & i \geq 2; \end{cases} \\ &= \binom{N}{i} p^i (1-p)^{N-i}, \quad 0 \leq i \leq N. \end{aligned}$$

Para  $n = 1$  se cumple, ya que se tiene obtiene la probabilidad de transición entre estados. Suponga válido para  $n = M$  donde  $M > 1$ , debemos demostrar que la ec. A.20 se cumple para  $M + 1$ , por lo que el proceso que seguiremos es similar al que se realizó en la sec. A.3. Suponga que se parte de una multiplicidad de colisión  $N$ , en donde la probabilidad de elegir la rama cuyo estado final es  $i$  después de  $M + 1$  transiciones es

$$\bar{Q}_{N,i}^{(M+1)}(p) = \prod_{l=0}^M P_{a_l, a_{l+1}}(p), \quad \text{con } a_0 = N, \quad a_{M+1} = i \quad \text{y} \quad S(i) \leq a_{l+1} \leq a_l, \quad \text{para } l = 0 \dots M.$$

Lo anterior es para una sola rama, para la probabilidad total hay que sumar las probabilidades de las demás ramas que terminan en el estado  $i$  después de  $M + 1$  transiciones, entonces

$$\begin{aligned} Q_{N,i}^{(M+1)}(p) &= \underbrace{\sum_{a_1=S(i)}^{a_0} \sum_{a_2=S(i)}^{a_1} \dots \sum_{a_M=S(i)}^{a_{M-1}} \prod_{l=0}^M P_{a_l, a_{l+1}}(p)}_{M+1 \text{ sumas}} \\ &= \sum_{a_1=S(i)}^N P_{a_0 a_1} \sum_{a_2=S(i)}^{a_1} \dots \sum_{a_M=S(i)}^{a_{M-1}} \prod_{l=1}^M P_{a_l, a_{l+1}}(p). \end{aligned} \quad (\text{A.21})$$

El segundo término de la primera suma está determinado por la ec. A.20, pero con  $M$  en lugar de  $n$ , y  $a_1$  en lugar de  $N$ . Para facilitar la manipulación, suponga que  $a_1 = K$ ; así entonces sustituyendo

A.20 se tiene

$$Q_{N,i}^{(M+1)}(p) = \sum_{K=2}^N P_{N,K}(p) p^{(M-1)i} P_{K,i}(p) \left[ \left( \sum_{k=0}^{M-1} p^k \right)^{K-i} - \left( \sum_{k=0}^{M-2} p^k \right)^{K-i} - \delta(i) K p^{M-1} \left( \sum_{k=0}^{M-2} p^k \right)^{K-1} \right]; \quad (\text{A.22})$$

para  $i < 2$ , y para  $i \geq 2$  tenemos

$$Q_{N,i}^{(M+1)}(p) = \sum_{K=i}^N P_{N,K}(p) p^{(M-1)i} P_{K,i}(p) \left( \sum_{k=0}^{M-1} p^k \right)^{K-i}. \quad (\text{A.23})$$

Al sustituir la ec. A.5 en las ecuaciones anteriores se tiene

$$Q_{N,i}^{(M+1)}(p) = p^{(M-1)i} P_{N,i}(p) \sum_{K=2}^N \binom{N-i}{K-i} p^K \left[ \left( \sum_{k=0}^{M-1} p^k \right)^{K-i} - \left( \sum_{k=0}^{M-2} p^k \right)^{K-i} - \delta(i) K p^{M-1} \left( \sum_{k=0}^{M-2} p^k \right)^{K-1} \right], \quad (\text{A.24})$$

$$Q_{N,i}^{(M+1)}(p) = p^{(M-1)i} P_{N,i}(p) \sum_{K=i}^N \binom{N-i}{K-i} p^K \left( \sum_{k=0}^{M-1} p^k \right)^{K-i}; \quad (\text{A.25})$$

para  $i < 2$  e  $i \geq 2$  respectivamente. Al manipular por separado cada ecuación, sea  $i < 2$  y  $j = K - i$ , por lo que la ec. A.24 resulta

$$Q_{N,i}^{(M+1)}(p) = p^{Mi} P_{N,i}(p) \left[ \sum_{j=2-i}^{N-i} \binom{N-i}{j} \left( p \sum_{k=0}^{M-1} p^k \right)^j - \sum_{j=2-i}^{N-i} \binom{N-i}{j} \left( p \sum_{k=0}^{M-2} p^k \right)^j - \delta(i) p^{M-1} \left( \sum_{k=0}^{M-2} p^k \right)^{-1} \sum_{j=2}^{N-i} \binom{N-i}{j} j \left( p \sum_{k=0}^{M-2} p^k \right)^j \right]. \quad (\text{A.26})$$

Los dos primeros términos pueden ser completados y obtener el binomio de Newton; el tercer término es el mismo que se tenía para la ec. A.15, y recordando que sólo es válido cuando  $i = 0$ , la ec. A.26

queda como

$$\begin{aligned}
Q_{N,i}^{(M+1)}(p) &= p^{Mi} P_{N,i}(p) \left[ \left( 1 + p \sum_{k=0}^{M-1} p^k \right)^{N-i} - 1 - \delta(i) N p \sum_{k=0}^{M-1} p^k \right. \\
&\quad - \left( 1 + p \sum_{k=0}^{M-2} p^k \right)^{N-i} + 1 + \delta(i) N p \sum_{k=0}^{M-2} p^k \\
&\quad \left. - \delta(i) p^M N \left( \sum_{k=0}^{M-1} p^k \right)^{N-1} + \delta(i) p^M N \right] \\
&= p^{Mi} P_{N,i}(p) \left[ \left( \sum_{k=0}^M p^k \right)^{N-i} - \left( \sum_{k=0}^{M-1} p^k \right)^{N-i} - \delta(i) p^M N \left( \sum_{k=0}^{M-1} p^k \right)^{N-1} \right]. \quad (\text{A.27})
\end{aligned}$$

Cuando  $i > 2$  tenemos, por la ec. A.25 y sustituyendo  $j = K - i$

$$\begin{aligned}
Q_{N,i}^{(M+1)}(p) &= p^{(M-1)i} P_{N,i}(p) \sum_{j=0}^N \binom{N-i}{j} p^{j+i} \left( \sum_{k=0}^{M-1} p^k \right)^j \\
&= p^{Mi} P_{N,i}(p) \sum_{j=0}^N \binom{N-i}{j} \left( p \sum_{k=0}^{M-1} p^k \right)^j \\
&= p^{Mi} P_{N,i}(p) \left( 1 + p \sum_{k=0}^{M-1} p^k \right)^{N-i} \\
&= p^{Mi} P_{N,i}(p) \left( \sum_{k=0}^M p^k \right)^{N-i}. \quad (\text{A.28})
\end{aligned}$$

Finalmente, uniendo las ecuaciones A.27 y A.28 tenemos

$$Q_{N,i}^{(M+1)}(p) = p^{Mi} P_{N,i}(p) \begin{cases} \left[ \left( \sum_{k=0}^M p^k \right)^{N-i} - \left( \sum_{k=0}^{M-1} p^k \right)^{N-i} - \delta(i) p^M N \left( \sum_{k=0}^{M-1} p^k \right)^{N-1} \right], & i < 2; \\ \left( \sum_{k=0}^M p^k \right)^{N-i}, & i \geq 2. \end{cases}$$

Sustituyendo  $\sum_{k=0}^n p^k = \frac{1-p^{n+1}}{1-p}$  para  $p > 0$  y la ec. 3.1, se tiene

$$Q_{N,i}^{(M+1)}(p) = \binom{N}{i} p^{(M+1)i} \begin{cases} \left[ (1-p^{M+1})^{N-i} - (1-p^M)^{N-i} - \delta(i) p^M N q (1-p^M)^{N-1} \right], & i < 2; \\ (1-p^{M+1})^{N-i}, & i \geq 2; \end{cases} \quad (\text{A.29})$$

donde  $q = 1 - p$  y  $N \geq 2$ . Esto demuestra que la ec. A.20 es válida para cualquier  $n > 1$ .

## A.5 Longitud promedio de un CRI para el algoritmo 2C

La ec. 3.16 puede proveer información acerca del comportamiento de la fase para poder estimar la longitud promedio de un CRI, tanto para el algoritmo 2C original, como para la propuesta 2C-adaptable; para ello es conveniente determinar el estado  $i$  esperado dado un número de transiciones  $n$ . Se tiene

$$E[i/n] = \sum_{i=0}^N i Q_{N,i}^{(n)}(p);$$

es decir, cuál es el estado  $i$  que en promedio el sistema tomará una mayor cantidad de veces, después de  $n$  transiciones a partir de un estado  $N$  (multiplicidad de la colisión al inicio de la fase). Para resolver la ecuación anterior, se sustituye 3.16, con lo cual se tiene

$$\begin{aligned} E[i/n] &= Np^n(1-p^n)^{N-1} - Np^n(1-p^{n-1})^{N-1} + \sum_{i=2}^N i \binom{N}{i} p^{ni}(1-p^n)^{N-i} \\ &= -Np^n(1-p^{n-1})^{n-1} + (1-p^n)^N \sum_{i=1}^N i \binom{N}{i} \left[ \frac{p^n}{1-p^n} \right]^i. \end{aligned}$$

Al completar el binomio de Newton en el segundo término, se obtiene

$$E[i/n] = Np^n \left[ 1 - (1-p^{n-1})^{N-1} \right], \text{ para } N \geq 2 \text{ y } n \geq 1. \quad (\text{A.30})$$

La ecuación anterior también permite determinar el número de transiciones  $n$ , en el caso de que un estado  $i$  sea el de mayor ocurrencia; es decir, si  $E$  es el estado que en promedio el sistema toma un mayor número de veces, entonces el número de transiciones promedio para que eso suceda estará dado por  $n$ . Dado lo anterior, resulta importante determinar  $n$  como función  $E$ ,  $N$  y  $p$  para así determinar la longitud de una fase; es decir

$$Np^n \left[ 1 - (1-p^{n-1})^{N-1} \right] = E, \quad 0 < E \leq N, \quad N \geq 2 \text{ y } n \geq 1. \quad (\text{A.31})$$

Sin embargo, sólo nos interesa el valor de  $n$  cuando  $E = 1$ , ya que se busca el número de transiciones promedio que una fase tendrá, para que éste termine en la mayoría de las veces en el estado 1; por otra parte, en principio  $p = \frac{1}{2}$  ya que la fase es la misma tanto para el algoritmo original, como

para la propuesta<sup>2</sup>. El problema con la ec. A.31, es que las soluciones para  $n(E, N)$  se determinan al resolver un polinomio de grado  $N$ ; es posible resolver la ecuación para los primeros dos casos, es decir, en  $N = 2$  se tiene

$$\begin{aligned} E &= \frac{2}{2^n} \left[ 1 - \left( 1 - \frac{2}{2^n} \right) \right], \\ 2^{2n} &= \frac{4}{E}. \end{aligned}$$

En donde  $n = 1$  para  $E = 1$ . El caso para  $N = 3$ , corresponde a resolver una ecuación de 3er grado, es decir

$$\begin{aligned} E &= \frac{3}{2^n} \left[ 1 - \left( 1 - \frac{2}{2^n} \right)^2 \right], \\ 0 &= (2^n)^3 - 12(2^n) + 12. \end{aligned}$$

En donde la solución es  $n = 1.47$ . El proceso puede continuar para seguir obteniendo soluciones numéricas, sin embargo, esto no resulta práctico para cualquier valor de  $N$ , por lo que se propone una aproximación.

Conviene entonces analizar el término  $(1 - \frac{2}{2^n})^{N-1}$ , en donde una primera aproximación es expandir la expresión en su forma de binomio de Newton; sin embargo, cada término en la expansión está multiplicado por coeficientes binomiales de  $N - 1$ , además de que el signo conmuta en cada término, lo que produce que no sea claro cuantos términos son necesarios tomar para una buena aproximación. Se sugiere entonces otro método, tomando en consideración que  $0 < \frac{2}{2^n} \leq 1$  para  $n \geq 1$ .

Sea  $\delta = (1 - \frac{2}{2^n})^{N-1}$ , debido a que

$$0 < \frac{2}{2^n} \leq 1 \implies 0 \leq \left( 1 - \frac{2}{2^n} \right) < 1 \implies 0 \leq \left( 1 - \frac{2}{2^n} \right)^{N-1} < 1;$$

---

<sup>2</sup>En realidad no son iguales, en la propuesta la primera ranura de la fase se emplea  $p = p_m(N)$ , pero de ahí en adelante es como una nueva fase de multiplicidad menor o igual a  $N$ , más adelante daremos mayor detalle.

para  $n \geq 1$  se tiene que  $0 \leq \delta < 1$ . Es decir

$$E = \frac{N}{2^n}(1 - \delta) \implies E \leq \frac{N}{2^n} \implies n \leq \log_2\left(\frac{N}{E}\right). \quad (\text{A.32})$$

Lo anterior permite tener una mejor acotación para  $\delta$ ; es decir,

$$2^n \leq \frac{N}{E} \implies \frac{1}{2^n} \geq \frac{E}{N} \implies 1 - \frac{2}{2^n} \leq 1 - \frac{2E}{N} \implies \left(1 - \frac{2}{2^n}\right)^{N-1} \leq \left(1 - \frac{2E}{N}\right)^{N-1}.$$

Debido a que

$$\left(1 - \frac{2E}{N}\right)^{N-1} \leq e^{-2E}.$$

Por lo tanto se tiene

$$\left(1 - \frac{2}{2^n}\right)^{N-1} \leq e^{-2E} \implies 0 \leq \delta \leq e^{-2E}.$$

Sustituyendo la primera ecuación de A.32 se tiene

$$1 \geq 1 - \delta \geq 1 - e^{-2E} \implies 1 \geq \frac{E2^n}{N} \geq 1 - e^{-2E} \implies E \leq \frac{N}{2^n} \leq \frac{E}{1 - e^{-2E}}. \quad (\text{A.33})$$

Mediante la ecuación anterior, conviene examinar varios casos para  $E$ , por lo que se define  $\gamma = \frac{N}{2^n}$ . Esta variable será de ayuda al momento de hacer una aproximación, sólo en términos de  $N$ .

**Caso  $E < 1$**

No es posible examinar el caso  $E = 0$ , debido a que ello implica que  $n \rightarrow \infty$ ; sin embargo, es posible conocer aun más sobre  $\gamma$ , es decir, por L'Hopital se tiene

$$\lim_{E \rightarrow 0} \frac{E}{1 - e^{-2E}} = \frac{1}{2};$$

por lo tanto,

$$0 \leq \gamma \leq \frac{1}{2}.$$

Es decir, tomando sólo el límite superior, se puede estimar que para cuando  $E \rightarrow 0$ , en promedio el número de ranuras necesarias ( $\hat{n}(N)$ ) para que una fase termine en el estado 0 debe cumplir

$$N \leq \frac{2^n}{2} \implies \hat{n}(N) \geq \log_2(N) + 1. \quad (\text{A.34})$$

### Caso $E=1$

Para  $E = 1$  se tiene lo siguiente

$$1 \leq \gamma \leq \frac{1}{1 - e^{-2}} \approx 1.15652;$$

por lo que es posible asumir que  $\gamma$  es casi constante, y permitir una aproximación en función de  $N$  del término  $(1 - \frac{2}{2^n})^{N-1}$  en la ec. A.31.

Al aproximar  $\gamma$  como función de  $N$ , se emplea la ec A.31

$$E = 1 = \gamma \left[ 1 - \left( 1 - \frac{2\gamma}{N} \right)^{(N-1)} \right].$$

Para  $N$  grande, el término dentro del paréntesis se puede aproximar como  $e^{-2\gamma}$ , por lo que sustituyendo se tiene

$$\frac{1}{\gamma} = 1 - e^{-2\gamma}.$$

La ecuación anterior puede ser resuelta numéricamente, por lo que el resultado es  $\gamma = 1.119323$ .

Finalmente, de la definición de  $\gamma$  se tiene que

$$n(N) = \log_2 \left( \frac{N}{\gamma} \right). \quad (\text{A.35})$$

Es decir,  $n$  representa la longitud promedio de una fase exitosa<sup>3</sup>. Sin embargo, recordemos que para  $N = 2$  de manera exacta se encontró que  $n = 1$ , mientras que por la ec. A.35 el valor es

---

<sup>3</sup>Una *fase exitosa* indica que una estación logró transmitir con éxito; mientras que en una *fase no exitosa*, ninguna estación transmitió. En ambos casos, la fase termina.

$n = 0.837374$ . En cambio para  $N = 3$  de manera exacta se obtuvo que  $n = 1.47$ , y la ec. A.35 indica que  $n = 1.422336$ . Aun para dos valores esto resulta lógico, ya que la ec. A.35 resulta mejor cuando  $N$  es más grande. Por lo tanto, definimos la longitud promedio de una fase exitosa como

$$\hat{n}(N) = \begin{cases} 1 & \text{para } N = 2, \\ \log_2\left(\frac{N}{\gamma}\right) & \text{para } N > 2. \end{cases} \quad (\text{A.36})$$

### Caso $E > 1$

Este caso resulta interesante, ya que si  $E$  es grande, entonces el término  $e^{-2E} \rightarrow 0$  y los límites de la desigualdad A.33 tienden a  $E$ , por lo que

$$\gamma = \frac{N}{2^n} \approx E.$$

Por lo tanto, la longitud promedio para llegar al estado  $E$  (en este caso la fase no ha terminado), es

$$\hat{n}(N) \approx \log_2\left(\frac{N}{E}\right). \quad (\text{A.37})$$

Observe que la ecuación anterior indica que se requieren menos ranuras para llegar a un estado  $E > 1$ . Lo cual resulta lógico y demuestra la lenta convergencia del algoritmo cuando  $p = \frac{1}{2}$ .

En este caso nos interesa la longitud promedio de una fase cuando  $E = 1$ , lo cual está dado por la ec. A.36. El siguiente paso es determinar la probabilidad de que una fase sea exitosa, dado que termino en  $n$  transiciones dado por la ec. A.36. Para ello es necesario considerar dos probabilidades: de que una fase exitosa termine en  $n$  transiciones, la cual está dada por la ec. 3.16 cuando  $i = 1$  y  $p = \frac{1}{2}$ ; y la probabilidad de que una fase termine en  $n$  transiciones, de igual manera dada por la ec. 3.16 para  $i = 0$  e  $i = 1$  con  $p = \frac{1}{2}$ .

Sea  $F_n(p, N)$  la probabilidad de que una fase sea exitosa dado que terminó en  $n$  transiciones, entonces,  $F_n(p, N)$  está dado por

$$F_n\left(p = \frac{1}{2}, N\right) = \frac{Q_{N,1}^n\left(\frac{1}{2}\right)}{Q_{N,0}^n\left(\frac{1}{2}\right) + Q_{N,1}^n\left(\frac{1}{2}\right)} \quad (\text{A.38})$$

$$\begin{aligned}
&= \frac{\frac{N}{2^n} \left[ \left(1 - \frac{1}{2^n}\right)^{N-1} - \left(1 - \frac{2}{2^n}\right)^{N-1} \right]}{\left[1 + \frac{N-1}{2^n}\right] \left(1 - \frac{1}{2^n}\right)^{N-1} - \left[1 + \frac{2(N-1)}{2^n}\right] \left(1 - \frac{2}{2^n}\right)^{N-1}} \\
&= \frac{\gamma \left[ \left(1 - \frac{\gamma}{N}\right)^{N-1} - \left(1 - \frac{2\gamma}{N}\right)^{N-1} \right]}{\left[1 + \frac{N-1}{N}\gamma\right] \left(1 - \frac{\gamma}{N}\right)^{N-1} - \left[1 + \frac{2(N-1)}{N}\gamma\right] \left(1 - \frac{2\gamma}{N}\right)^{N-1}}
\end{aligned}$$

Para  $N$  grande y sustituyendo A.36, es posible aproximar la ecuación anterior como<sup>4</sup>

$$\begin{aligned}
F_n(N) &= \frac{\gamma(e^{-\gamma} - e^{-2\gamma})}{(1 + \gamma)e^{-\gamma} - (1 + 2\gamma)e^{-2\gamma}} \\
&= \frac{\gamma(e^\gamma - 1)}{(1 + \gamma)e^\gamma - (1 + 2\gamma)} = 0.7099;
\end{aligned} \tag{A.39}$$

lo anterior para  $N > 2$ , y para  $N = 2$ ,  $F_n(N) = \frac{2}{3}$ . Por lo tanto se tiene

$$F_n(N) = \begin{cases} \frac{2}{3} & \text{para } N = 2, \\ 0.7099 & \text{para } N > 2. \end{cases} \tag{A.40}$$

La probabilidad de que una fase sea exitosa dado que terminó en  $\hat{n}(N)$  transiciones o ranuras para  $E = 1$  es  $F_n(N)$ , por lo que surge la siguiente pregunta: ¿cuántos intentos son necesarios para que una fase sea exitosa? Se tiene entonces, una distribución geométrica en donde  $F_n(N)$  representa la probabilidad de éxito. Recordemos que la esperanza del número de intentos en la distribución geométrica, es inversamente proporcional a esa probabilidad; entonces, el número de intentos necesarios  $Ni(N)$  es

$$Ni(N) = \begin{cases} \frac{3}{2}, & \text{para } N = 2, \\ 1.4086, & \text{para } N > 2. \end{cases} \tag{A.41}$$

La longitud promedio de un CRI para el algoritmo original, estaría dado por la longitud de cada una de las fases que hay en un CRI, que en total son  $N - 1$ , más la longitud de todas las fases no exitosas. Sin embargo, como se observó en la ec. A.34, no es posible determinar una longitud promedio para una fase no exitosa, de ahí que se tome el número de intentos necesarios para lograr

---

<sup>4</sup>A menos que se especifique otra cosa, se empleara  $F_n(N) = F_n(\frac{1}{2}, N)$ .

una fase exitosa. Así, este número de intentos contempla tanto el número de transiciones para lograr el éxito como de no lograrlo.

Con lo anterior, se tiene que la longitud de un CRI de multiplicidad  $N$  está dado por: la suma de las longitudes de cada fase, más las ranuras de colisión que inician las  $N - 1$  fases, más una ranura que representa el éxito de la última estación, menos la primera ranura, que no pertenece al CRI. Sin embargo, hay que considerar que la ec. A.41 indica el número de intentos o el número de veces que hay que repetir la fase, para que ésta sea exitosa. Por lo tanto se tiene que la longitud promedio de un CRI de multiplicidad  $N$  en el algoritmo original es

$$\begin{aligned}\bar{L}_{CRI}(N) &= \sum_{k=2}^N Ni(k)\hat{n}(k) + Ni(N)(N - 1) + 1 - 1 \\ &= \sum_{k=3}^N Ni(k)\hat{n}(k) + Ni(N)N\end{aligned}$$

Sustituyendo la ec. A.36 en la ecuación anterior se tiene

$$\begin{aligned}\bar{L}_{CRI}(N) &= \sum_{k=2}^N Ni(k)\hat{n}(k) + Ni(N)(N - 1) \\ &= \sum_{k=3}^N Ni(k)\log_2(\lambda k) + Ni(N)N.\end{aligned}$$

Lo anterior para  $N > 2$ , y para  $N = 2$  se tiene que  $\bar{L}_{CRI}(N) = 3$ . Después de algunas manipulaciones algebraicas se tiene

$$\bar{L}_{CRI}(N) = \begin{cases} 3, & \text{para } N = 2, \\ 1.4086 \left[ \log_2 \left( \frac{N!}{2\gamma^{N-3}} \right) + N \right], & \text{para } N > 2; \end{cases} \quad (\text{A.42})$$

con  $\gamma = 1.119323$ .

## A.6 Longitud promedio de un CRI para el algoritmo 2C-adaptable

En la sec. 3.2 se ha determinado que la propuesta de mejora del algoritmo 2C, se basa en disminuir el número de estaciones que compiten a partir de la segunda ranura de una fase; para ello, en la primer ranura cada estación determina el valor de la probabilidad de transmisión  $p$  mediante la ec. 3.11, si a la ranura siguiente ocurre una colisión, entonces la probabilidad de transmisión se supone fija en  $p = \frac{1}{2}$ , y se continua tal y como sucede en el algoritmo original.

Es posible, mediante la misma idea de la sección anterior, obtener la longitud promedio de un CRI para la propuesta; sin embargo, el calculo no resulta sencillo debido a que en este caso la fase posee dos probabilidades: en la primera ranura de la fase se emplea la ec. 3.11, y en las subsecuentes, hasta el final de la fase, se emplea  $p = \frac{1}{2}$ . Para éste último caso, se puede considerar que una fase inicia en la segunda ranura, pero ya no con una multiplicidad  $N$ , sino con una multiplicidad menor o igual a  $N$ , por lo que es necesario conocer este valor. A lo anterior, una primera aproximación es asumir que en promedio  $\frac{3}{2}$  estaciones iniciarán en la segunda ranura<sup>5</sup>, pero ello no resulta adecuado, ya que este valor asume que el sistema puede tomar cualquier estado, lo cual ya no resulta cierto debido a que ahora, está condicionado a la evolución que una fase pueda tomar.

Cuando una fase inicia, ésta puede tener dos posibles caminos: la fase termina en una ranura (no importando si es éxito o no, y que denotaremos por T), o puede no terminar (denotado por NT), en el cual, a la siguiente ranura se puede considerar que se "inicia" una fase de multiplicidad  $J$  (donde en principio,  $2 \leq J \leq N$ ), ver fig. A.3, por lo que el primer problema es determinar el valor de  $J$  como función de  $N$ .

Con lo anterior, se propone un análisis mediante un árbol de transiciones, para determinar principalmente dos cosas:

1. La probabilidad de que la fase termine en éxito, y con ello determinar el número de intentos que son necesarios, para lograrlo;
2. y el número de ranuras necesario para lograr el éxito en el número de intentos determinado.

---

<sup>5</sup>Este valor es la esperanza, y fue determinado en la sección 3.1.1.

Si la fase se desarrolla por la opción T, se tienen nuevamente dos posibles caminos: que la fase transite al estado 0 (fase no exitosa) o que vaya al estado 1 (la fase es exitosa). La probabilidad de que una fase sea exitosa dado que termino en la primera ranura, está determinada por la ec. A.38 con  $p = p_m(N)$  y  $n = 1$ ; es decir

$$\begin{aligned} F_1(p_m(N), N) &= \frac{Q_{N,1}^n(p_m(N))}{Q_{N,0}^n(p_m(N)) + Q_{N,1}^n(p_m(N))} \\ &= \frac{3N^2 + N}{5N^2 - 1}. \end{aligned} \quad (\text{A.43})$$

Y la probabilidad de que la fase tome la opción T, la cual definimos por  $pT(N)$ , esta dada por la probabilidad de que el sistema tome el estado 0 o el estado 1 en la primera ranura, con  $p = p_m(N)$ ; es decir

$$\begin{aligned} pT(N) &= Q_{N,0}^n(p_m(N)) + Q_{N,1}^n(p_m(N)) \\ &= [1 + (N - 1)p_m(N)](1 - p_m(N))^{(N-1)}. \end{aligned} \quad (\text{A.44})$$

Así, el que una fase termine en éxito dado que terminó en una ranura, se determina por el producto de las ecuaciones A.43 y A.44 (ver fig. A.3); es decir, la probabilidad de que el sistema llegue al estado 1 a partir de la rama T, y la cual definimos como  $T_1(N)$ , es

$$T_1(N) = pT(N)F_1(p_m(N), N) = Q_{N,1}^n(p_m(N)). \quad (\text{A.45})$$

Por otra parte, si la fase se desarrolla por la rama NT, entonces es necesario determinar el número de estaciones que en promedio accederán al canal, dado que la fase no terminó en la primera ranura. Así, la probabilidad de que el sistema tome el estado  $K$ , dado que no terminó en la primera ranura (o dado que tomó la rama NT), y que definimos por  $nF$ , es

$$\begin{aligned} nF(p_m(N), N, K) &= \frac{Q_{N,K}^1(p_m(N))}{1 - pT(N)} \\ &= \frac{\binom{N}{K} p_m(N)^K (1 - p_m(N))^{N-1}}{1 - pT(N)}. \end{aligned} \quad (\text{A.46})$$

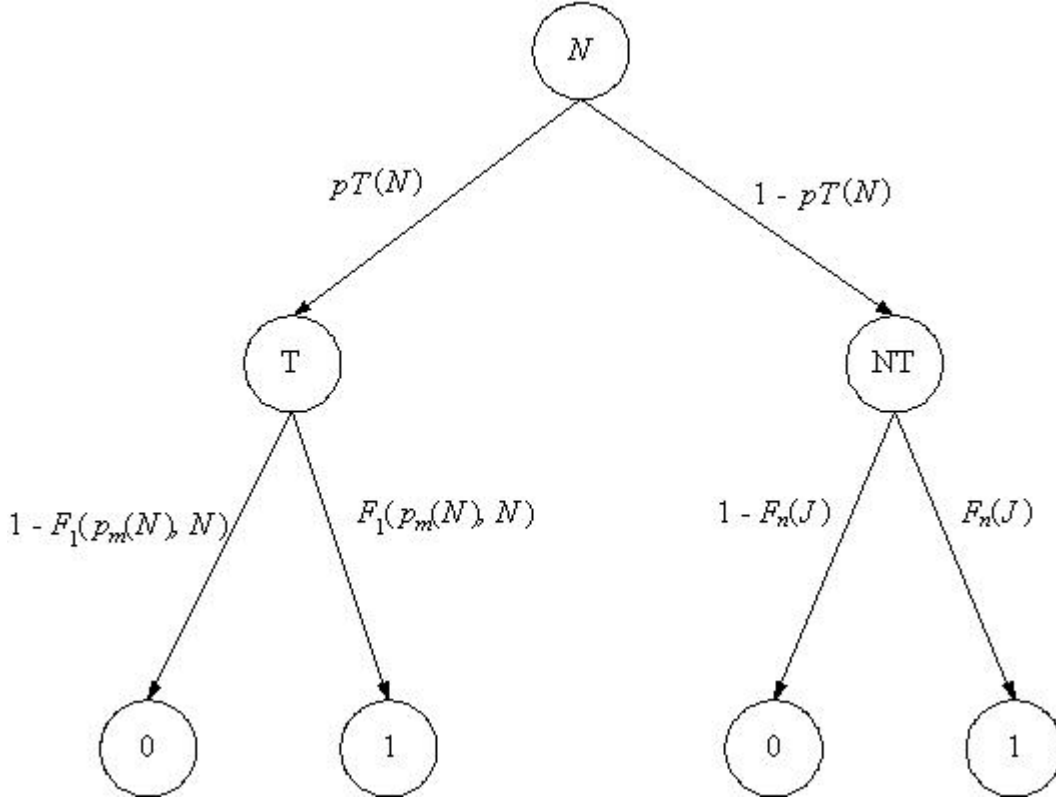


Figura A.3: Árbol de transiciones para longitud promedio de un CRI en el algoritmo 2C-Adaptable. Se muestra la probabilidad de transiciones entre cada nodo. Las hojas del árbol representan los estados posibles que determinan el fin de una fase. La probabilidad de llegar a cada hoja, es el producto de las probabilidades entre los respectivos nodos, ver sección A.2.

Con la ecuación anterior, estamos en posibilidades de determinar el estado (o el número de estaciones) esperado, que definimos como  $J(N)$ , es decir, el estado que más veces tomara el sistema cuando la fase no es exitosa en la primera ranura, para ello se tiene que

$$\begin{aligned}
 J(N) = E[K/NT] &= \frac{1}{1 - pT(N)} \sum_{K=2}^N K \binom{N}{K} p_m(N)^K (1 - p_m(N))^{N-K} \\
 &= \frac{N}{1 - pT(N)} \sum_{K=2}^N \binom{N-1}{K-1} p_m(N)^K (1 - p_m(N))^{N-K} \\
 &= \frac{N p_m(N)}{1 - pT(N)} \sum_{m=1}^{N-1} \binom{N-1}{m} p_m(N)^m (1 - p_m(N))^{N-1-m} \\
 &= \frac{N p_m(N)}{1 - pT(N)} \left[ 1 - (1 - p_m(N))^{N-1} \right] \tag{A.47}
 \end{aligned}$$

La ecuación anterior determina el número de estaciones promedio, que iniciarán en las segunda ranura de la fase, dado que esta no fue exitosa en la primera. Así, se espera que la longitud de la "fase que se forma" a partir de la segunda ranura está determinada por la ec. A.31, al sustituir  $J(N)$ , para  $p = \frac{1}{2}$  y  $E = 1$ . Lo anterior debido a que  $2 \leq J(N) \leq 2.6354$ , para  $N \rightarrow \infty$ . Así se tiene que

$$\hat{n}(J(N)) \in [1, 1.3008].$$

Finalmente, es necesario determinar la probabilidad de que esta nueva "fase" sea exitosa, dado que hay  $J(N)$  estaciones en contienda; esta probabilidad ya se determinó, y se está dada por la ec. A.38; así, para este caso se tiene<sup>6</sup>

$$F_n(J(N)) = \begin{cases} \frac{2}{3} & \text{para } J(N) = 2, \\ 0.6927 & \text{para } J(N) > 2. \end{cases}$$

Examinando cada una de las ramas, para el camino T el número de intentos necesarios para lograr el éxito está dado por una distribución geométrica, y el resultado es  $\frac{1}{F_1(p_m(N), N)}$ ; mientras que para el camino NT la distribución es la misma y el resultado  $\frac{1}{F_n(J(N))}$ . Por lo tanto, el valor esperado del número de intentos entre cada rama es

$$I(N) = \frac{1}{F_1(p_m(N), N)} pT(N) + \frac{1}{F_n(J(N))} (1 - pT(N)). \quad (\text{A.48})$$

Con esto, estamos en posibilidades de determinar la longitud de un fase para el algoritmo 2C-adaptable; sin embargo, es necesario calcular el número de ranuras que en promedio toma la fase. Del árbol de transiciones, si la fase toma el camino T, entonces le toma un ranura; en cambio, si toma el camino NT, entonces le tomará  $\hat{n}(J(N)) + 1$  ranuras<sup>7</sup>. Así la longitud promedio de una fase está dado por

$$\hat{n}_f(N) = pT(N) + (\hat{n}(J(N)) + 1)(1 - pT(N)), \quad (\text{A.49})$$

que no es otra cosa que la esperanza. Por lo tanto, la longitud promedio de un CRI para el algoritmo 2C-adaptable, es el producto de la longitud promedio de cada fase (ec. A.49, y en total hay  $N - 1$  fases para un CRI de multiplicidad  $N$ ) por el número de intentos necesarios (ec. A.48), más el

<sup>6</sup>En este caso no es necesario emplear toda la ec. A.38, debido a que  $J(N)$  tiene su valor alrededor de 2.

<sup>7</sup>Se agrega 1, por la primer ranura que no terminó en éxito, y que dió origen al camino NT.

número de ranuras que inician cada fase, más la última ranura que emplea la última estación en transmitir su paquete, menos la primera ranura, que no pertenece al CRI; es decir,

$$\begin{aligned}\bar{L}'_{CRI}(N) &= \hat{n}_f(N)I(N)(N-1) + I(N)(N-1) + 1 - 1 \\ &= \frac{[pT(N) + (\hat{n}(J(N)) + 1)(1 - pT(N))](N-1) + (N-1)}{pT(N)[F_1(p_m(N), N) - F_n(J(N))] + F_n(J(N))}.\end{aligned}$$

En general es complicado usar todas las ecuaciones y sustituirlas en la anterior, por lo que se propone una aproximación, excepto para  $N = 2$ . El motivo de esta aproximación es que en general todas las ecuaciones convergen a un valor determinado, y dadas sus características, no es necesario considerar todos los valores para  $N$  ya que la convergencia se puede apreciar desde los primeros valores  $N$ . Así se tiene que la longitud promedio de un CRI para el algoritmo 2C-adaptable es

$$\bar{L}'_{CRI}(N) = \begin{cases} 3.2898 & \text{para } N = 2, \\ 4.0380(N-1) & \text{para } N > 2. \end{cases} \quad (\text{A.50})$$

En donde, cuando  $N$  es grande se tiene

$$\begin{aligned}pT(N) &\rightarrow \frac{5}{2}e^{-3/2}, \\ F_1(p_m(N), N) &\rightarrow \frac{3}{5}, \\ J(N) &\rightarrow 2.6354, \\ \hat{n}(J(N)) + 1 &\rightarrow 2.3008, \\ F_n(J(N)) &\rightarrow 0.6926.\end{aligned}$$

# Apéndice B

## Simulador del algoritmo 2C-adaptable

---

Para realizar una comparación entre el algoritmo 2C original y las modificaciones propuestas para su mejora, se construyó un simulador de eventos discretos en lenguaje C++. El algoritmo es muy sencillo de implementar, por lo que no es necesario considerar algunos mecanismos extra para su implementación, por lo que no se consideró emplear algún simulador conocido; por ejemplo, NS2.

El simulador tenía la misión de reproducir íntegramente el algoritmo 2C bajo ciertas condiciones. Cada ejecución simulaba un *experimento* que representa un CRI; es decir, dada una multiplicidad de la colisión  $N$ , se permitía el desarrollo del algoritmo 2C hasta que todas las estaciones transmitieran con éxito sus paquetes. Los resultados obtenidos en cada experimento son: el número de colisiones, vacíos, éxitos y la duración del CRI, todo en número de ranuras<sup>1</sup>. En general se simuló de forma exhaustiva, por lo que en realidad lo que se tiene son los promedios de un conjunto experimentos.

En cada experimento se asume lo siguiente:

1. La simulación del experimento inicia con una colisión de multiplicidad  $N$  y termina con dos éxitos consecutivos.
2. Todas las estaciones recibían las transmisiones de la estación central al mismo tiempo, lo mismo ocurre con las transmisiones de las estaciones a la estación central; es decir, se desprecia el retardo en la propagación.

---

<sup>1</sup>Obviamente la suma de colisiones, vacíos y éxitos debe ser igual al tamaño del CRI.

3. No existen errores en la transmisión, incluyendo los mensajes de realimentación.
4. No se genera tráfico de ninguna especie. Salvo en el caso de la simulación con el algoritmo de predicción.
5. La estación central es capaz de saber con exactitud (excepto al emplear un algoritmo de predicción), el número de estaciones que colisionó después de cada ranura o al menos al inicio de un CRI. En el caso del algoritmo original, no es posible que la estación central determine el número de estaciones en colisión.
6. Una vez que la estación haya transmitido su paquete, termina y ya no genera ningún otro paquete más.
7. Todos los nodos están sincronizados.

El presente capítulo no intenta explorar a detalle el simulador; se pretende dar una idea del funcionamiento del simulador, y ofrecer sólo una perspectiva de la forma en que obtuvieron los resultados entre el algoritmo 2C original y la propuesta.

## B.1 Descripción del Simulador

El simulador de eventos discretos, fue elaborado en lenguaje C++ y consta principalmente de 3 clases:

1. *Simulador*: La clase Simulador se encarga de llevar el control de la simulación; los métodos que ahí radican, permiten la evolución de la simulación al calendarizar los mensajes y llevar el control del reloj que existe en cada nodo, por medio de *eventos*. En realidad los nodos no transmiten paquetes de información, en su lugar transmiten cortos mensajes que simulan un paquete, por lo que cuando la estación central recibe dos o más mensajes se produce una colisión
2. *Mensaje*: Contiene las propiedades y métodos para el manejo del objeto Mensaje; generalmente un *evento* o una *acción* dispara un mensaje, y un mensaje lleva una acción a realizar; cada mensaje se compone de nodo destino, nodo origen, tiempo de envío (relativo al origen), tiempo de arribo (relativo al destino), un evento y datos. En este caso los *datos* pueden ser el número

de estaciones que colisionaron, el paquete a transmitir o cualquier otro. Es necesario especificar que tipo de evento está asociado al mensaje, para de esta forma determinar como va actuar el nodo ante el mensaje recibido.

3. *Nodo*: La instanciación de esta clase es el objeto *Nodo*, cada nodo posee los métodos que modifican su comportamiento, y que permiten el envío de mensajes entre nodos. Las propiedades básicas de cada nodo son: el identificador de nodo y su reloj. Cada nodo sólo puede actuar ante un evento, el cual puede provenir de si mismo, de algún otro nodo o del objeto *Simulador*.

El simulador permite el envío de mensajes de dos formas: punto a punto o por difusión; ambos esquemas son necesarios para la simulación del algoritmo 2C, ya que en principio cada estación envía mensajes a la estación central (punto a punto), y la estación central a su vez, difunde mensajes (realimentación) a todos los nodos involucrados.

Existen otras clases y funciones que permiten son necesarias para el funcionamiento del simulador, pero éstas no son fundamentales para la implantación del algoritmo en el simulador. Básicamente no se menciona la clase *Reloj*, y algunas de propósito general.

## B.2 Funcionamiento del Simulador

El simulador posee un *reloj global*, que es actualizado cada cierto intervalo de tiempo, ello depende de la precisión del reloj (todos los relojes tienen la misma precisión); por ejemplo, si el reloj tiene una precisión de centésimas, entonces el reloj se actualiza cada 0.01 unidades, y por lo tanto es necesario que pasen 100 "actualizaciones" para que el reloj se incremente en una unidad. Este reloj global es independiente del que posee la computadora, por lo que una unidad de tiempo en el simulador no siempre será un número fijo de ciclos de reloj en la computadora. La función principal del reloj global, es actualizar todos los relojes de cada uno de los nodos por medio de eventos. Cuando un nodo recibe el evento *Actualiza reloj*, procede a incrementar su reloj y a ejecutar el código que el programador haya dispuesto.

Una vez que se han actualizado todos los relojes, cada nodo verifica su agenda y con base a su reloj, determina si el instante es el planificado para ejecutar el evento que se encuentra en su agenda.

Cuando un mensaje es enviado por algún nodo, el objeto Simulador se encarga de despacharlo al nodo destino (o destinos); esto es, calcula el tiempo de llegada al nodo destino y se lo envía; por lo tanto, cada nodo inserta en su agenda los mensajes que recibe, y los toma en cuenta cuando el tiempo de arribo del mensaje concuerda con su reloj. Es importante mencionar que un evento puede disparar un mensaje, y un mensaje siempre debe desencadenar un evento.

Cuando todos los mensajes han sido procesados (de cada uno de los nodos), en ese instante el objeto Simulador se encarga de actualizar el reloj global, y por lo tanto el reloj de cada nodo. Por otra parte, así como este objeto envía mensajes a los nodos, también es posible que un nodo envíe un mensaje al objeto Simulador; el número de mensajes es limitado y en nuestro caso es de ayuda cuando un nodo desea detener la simulación. La fig. B.1 Muestra el diagrama a bloques del simulador.

### B.3 Implantación del algoritmo 2C en el Simulador

Con lo anterior se tienen las bases para la implantación del algoritmo en el simulador. En este caso el tiempo se asume ranurado y de longitud fija, cada ranura de tiempo equivale a una unidad de tiempo en el simulador. Como las estaciones están sincronizadas, no es necesario fijar una determinada precisión al reloj, en este caso la precisión empleada es de  $1/3$  de unidad. Así entonces, cada nodo transmite en el tiempo  $X.1$  (inicio de la ranura  $X$ ), y la realimentación les es enviada en el instante de tiempo  $(X + 1).0$  (fin de la ranura  $X$ ), donde  $X \in \mathbb{N}$ , tal que  $X \geq 0$  y representa la parte entera del reloj.

Para la simulación se generaron  $N$  nodos (multiplicidad de la colisión), y un nodo central cuyo identificador es 0 (el identificador de los demás nodos no es importante); cada nodo conoce el identificador del nodo central, pero no así el identificador de los demás nodos, de hecho los nodos no saben cuál es el número total de nodos en el sistema, a menos que la estación central se los informe.

Al tiempo 0 a cada nodo se le envía el evento *Despierta*, que los obliga a transmitir un paquete hacia el nodo 0 en el instante 0.1 (todas las transmisiones se hacen en el instante  $X.1$ , que marca el

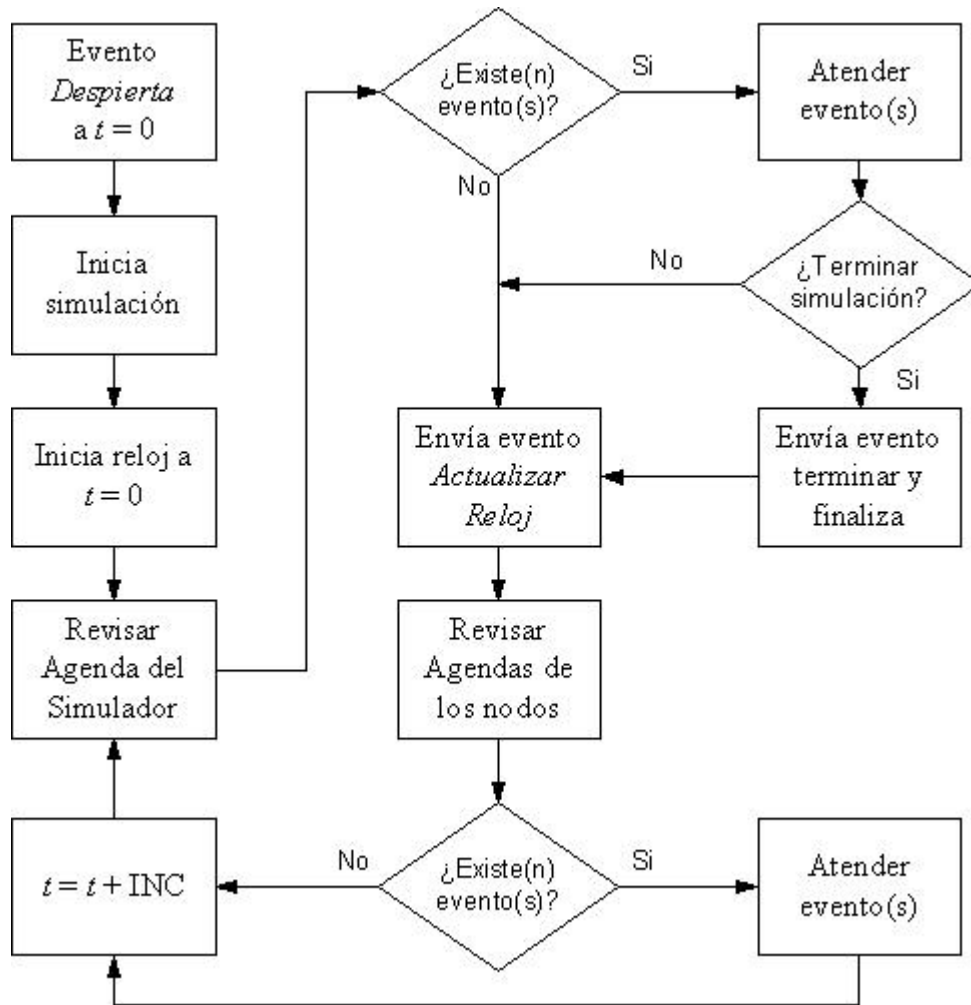


Figura B.1: Funcionamiento del simulador. La constante INC supone el valor del incremento en el tiempo; para el caso de la simulación, el valor es igual a  $1/3$ .

inicio de una ranura). El nodo central cuenta el número de paquetes que le son enviados (cuando así sea necesario), y emite la respectiva realimentación a todos los nodos en el instante 1.0; la realimentación se envía mediante los eventos *Colisión* y *No Colisión*, y dependiendo de la implementación de algoritmo 2C, el evento *Colisión* llevará el número de nodos involucrados. Con esta información los nodos realizan un experimento de tipo Bernoulli, y sólo transmiten aquellos a los que les fue favorable el experimento; en otro caso, también pueden optar por transmitir sin hacer el experimento, ello depende del valor de la realimentación. Como la transmisión se realiza en los instantes  $X.1$  y la realimentación llega en instantes  $(X + 1).0$ , es necesario que cada nodo se envíe a sí mismo un mensaje con el evento *Transmite*, con la finalidad de calendarizar la transmisión del

paquete al tiempo  $X.1$ ; el algoritmo continua hasta que todas las estaciones puedan transmitir su paquete; la Fig. B.2 muestra el desarrollo del algoritmo en un nodo con el simulador.

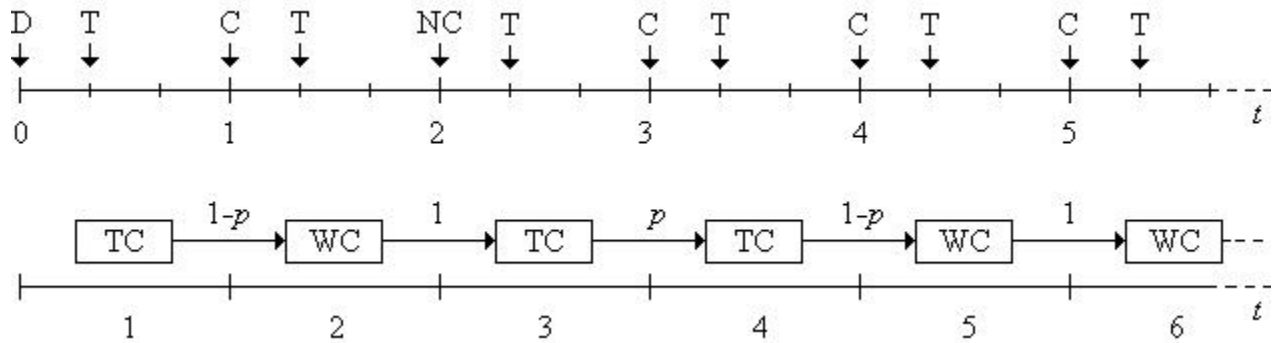


Figura B.2: Comportamiento de un nodo en el simulador contra el algoritmo 2C. La primera recta muestra el reloj del nodo en el simulador, mientras que la segunda recta muestra el tiempo ranurado en el algoritmo. La simbología es la siguiente:  $D$  - evento *Despierta*,  $T$  - evento *Transmite*,  $C$  - evento *Colisión* y  $NC$  - evento *No Colisión*; mientras que  $WC$  - celda de espera,  $TC$  celda de transmisión y  $p$  la probabilidad de ir a la celda de transmisión. Observe que en los instantes  $X.0$ , los nodos realizan el experimento de tipo Bernoulli, y en base al resultado deciden si van a la celda de transmisión o espera. La mejora en el algoritmo, propone  $p(N)$  en donde  $N$  representa la multiplicidad de la colisión.

Cuando el nodo central detecta que todos los nodos han transmitido su respectivo paquete, solicita al objeto Simulador que finalice la ejecución del experimento. Una vez terminada la simulación del experimento, se inicia uno nuevo con la misma multiplicidad; este proceso se realizó 10000 veces para cada valor de  $N$ ; es decir, 10000 experimentos para cada valor de la multiplicidad de la colisión  $N$ . El nodo central se encarga de calcular las estadísticas durante todo el proceso de simulación, por lo que al final de todo el proceso (10000 experimentos), se obtienen valores promedio de: duración del CRI, número éxitos, colisiones y vacíos para cada valor de  $N$  propuesto en número de ranuras.

Es importante mencionar que cada nodo emplea una semilla diferente en su generador de números pseudoaleatorios; de hecho es imposible usar la misma semilla, ya que esto puede provocar que el proceso de simulación se bloquee. Elegir apropiadamente las semillas para los generadores, evitó que en un determinado punto en la simulación, dos o más nodos posean el mismo resultado en el experimento Bernoulli (debido a que se generan ciclos de números aleatorios), esto último impide que la ejecución del algoritmo evolucione y que este no llegue a un resultado satisfactorio. De hecho esto último sería un problema, si al implementar el algoritmo se usaran generadores pseudoaleatorios de ciclo corto.

# Referencias

---

- [1] N. Abramson, "The ALOHA system-another alternative for computer communications," in Fall Joint Computer Conference, pp. 281-285, 1970.
- [2] L. Alarcon-Ramos, M. Lopez-Guerrero y D. Makrakis. "Adaptive 2C: a Novel Access Control for Fair and Efficient Channel Sharing." IEEE, 2007 Canadian Conference on Electrical and Computer Engineering, abril 2007.
- [3] G. Bianchi, L. Fratta, and M. Oliveri, "Performance evaluation and enhancement of the CSMA/CA MAC protocol for 802.11 wireless LANs," Personal, Indoor and Mobile Radio Communications, 1996. PIMRCapros; 96., Seventh IEEE International Symposium on Volume 2, Issue , 15-18 octubre 1996 pp. 392-396 vol. 2
- [4] C. C. Bisdikian, "The  $n$ -ary stack algorithm for the wireless random access channel," Mobile Networks and Applications 2 (1997), pp. 89-99.
- [5] J. Capetanakis "The Multiple Access Broadcast Channel: Protocol and Capacity Considerations." PhD thesis, MIT, Dept. of Electrical Engineering and Computer Science, 1977.
- [6] J. I. Capetanakis. "Tree algorithms for packet broadcast channel," IEEE Trans. Inform. Theory, vol. IT-25, pp. 505-515, septiembre 1979.
- [7] G. Domic, N. D. Sidiropoulos and R. Zhang, "Medium Access Control-Physical Cross-Layer Design," IEEE Signal Processing Magazine pp. 40-50 (septiembre 2004).
- [8] A. G. Greenberg, P. Flajolet and R. E. Ladner, "Estimating the Multiplicities of Conflicts to Speed Their Resolution in Multiple Access Channels," Journal of the Association for Computing Machinery, Vol. 34, No. 2, abril 1987, pp. 289-325.

- [9] E. Gulko - Kazakos, " *Tree-Based Multiaccess Protocols Where Collision Multiplicities Are Known*," IEEE Transactions on Communications, Vol. COMM - 33, No. 9, septiembre 1985, pp. 999-1001.
- [10] J. Hayes, " *An Adaptive Technique for Local Distribution*," IEEE Trans. on communication, COM-26(8) pp. 1178-1186 (agosto 1978).
- [11] D. Makrakis, R. S. Mander and L. Orozco Barbosa, " *Integrated traffic protocol for personal communication networks*," Communications, Computers, and Signal Processing, 1995. Proceedings. IEEE Pacific Rim Conference on mayo 1995, pp. 20-25
- [12] 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.
- [13] M. L. Molle and G. C. Polyzos, " *Conflict Resolution Algorithms and their Performance Analysis*," Technical Report, CS93-300, Dept. of Computer Science and Engineering, University of California at San Diego, LaJolla, julio 1993.
- [14] M. Paterakis and P. Papantoni-Kazakos: " *A Limited Sensing Random-Access Algorithm with Binary Success-Failure Feedback*," IEEE Transactions on Communications, Vol. 37, No. 5, mayo 1989, pp. 526-530.
- [15] M. Paterakis and P. Papantoni-Kazakos: " *A Simple Window Random Access Algorithm with Advantageous Properties*," IEEE Transactions on Information Theory, Vol. IT - 35, No. 5, septiembre 1989, pp. 1124-1130.
- [16] B. Porat, " *A Course in Digital Signal Processing*," First Edition, John Willey & Sons, USA 1997.
- [17] C. Po-Ning, W. Chiung-Shien, M. Gin-Kou " *An efficient collision resolution scheme for wireless multipleaccess*," Vehicular Technology Conference, 1998. VTC 98. 48th IEEE Volume 2, Issue, 18-21 mayo 1998 pp. 1341-1345 vol.2
- [18] P802.11, " *IEEE Draft Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*," D2.0, julio 1995.

- [19] R. Rom y M. Sidi, *Multiple Access Protocols; Performance and Analysis*, Springer Verlag, 1990.
- [20] M. Schwartz, "*Broadband Integrated Networks*," First Edition, Prentice Hall, USA 1996.
- [21] A. Tanenbaum, "*Computer Networks*." Third Edition, Prentice Hall.
- [22] L. Tong, V. Naware and P. Venkitasubramaniam, "*Signal Processing in Random Access*," IEEE Signal Processing Magazine pp. 29-39 (septiembre 2004).
- [23] B.S. Tsybakov and V.A. Mikhailov, "*Free Synchronous Packet Access in a Broadcast Channel with Feedback*," Probl. Information Transmission, 14(4) pp. 259-280 (octubre - diciembre 1978).
- [24] B. S. Tsybakov and N. D. Vvedenskaya, "*Random multiple access stack algorithms*," Probl. Inform. Transm., vol. 16, no. 3, pp. 230-243, 1980.
- [25] B. S. Tsybakov and N. B. Likhanov, "*Some New Random Multiple-Access Algorithms*," Problemy Peredachi Informatsii 21(2), pp. 69-89 (abril - junio 1985). (También en Proc. International Symposium on Information Theory, Brighton, UK, p. 102, junio 24-28, 1985.).