Casa abierta al tiempo
UNIVERSIDAD AUTÓNOMA METROPOLITANA
UNIDAD IZTAPALAPA
División de Ciencias Básicas e Ingeniería
Posgrado en Ciencias (Ciencias y Tecnologías de la Información)

“Desarrollo de un Protocolo de Control de Acceso al Medio en Comunicaciones M2M”

TESIS

Que para obtener el grado de

DOCTOR EN CIENCIAS (CIENCIAS Y TECNOLOGÍAS DE LA INFORMACIÓN)

PRESENTA

M. en C. Sergio Javier Alvarez

Matrícula: 2163802704
ORCID: 0009-0002-3371-2883
Correo electrónico: serjaval@xanum.uam.mx

DIRECTOR DE TESIS:

Dr. Miguel López Guerrero

Jurado:

Presidente:	Dr. Enrique Rodríguez de la Colina
Secretario:	Dr. Ricardo Marcelín Jiménez
Vocal:	Dr. Miguel López Guerrero
Vocal:	Dr. Luis Alberto Vásquez Toledo
Vocal:	Dr. Javier Gómez Castellanos

Iztapalapa, Ciudad de México a 4 de agosto de 2025

RESUMEN

En las redes de comunicaciones, los protocolos de control de acceso al medio (MAC) juegan un papel crucial en la gestión de la transmisión de paquetes de datos entre dispositivos. A través de ellos se regula el acceso al canal compartido con el fin de minimizar las colisiones que se presentan y maximizar la eficiencia. Los protocolos MAC se pueden clasificar, dependiendo la forma en que administran el acceso al medio, en protocolos de contienda y protocolos libres de contienda. Los primeros permiten que múltiples dispositivos compitan por el acceso al canal de comunicación, mientras que los segundos asignan el acceso al canal de manera organizada. Sin embargo, con la aparición del Internet de las Cosas (IoT), las comunicaciones Máquina a Máquina (M2M) mostraron que los protocolos MAC presentan diversas deficiencias, ya que fueron diseñados para redes de comunicaciones con interacción de tipo humano. En escenarios de comunicaciones M2M, donde una gran cantidad de dispositivos buscan transmitir simultáneamente pequeños paquetes de datos en intervalos regulares, los mecanismos MAC tradicionales pueden generar congestión y altos tiempos de espera, lo que impacta negativamente en el desempeño y escalabilidad de la red.

Con el fin de paliar las deficiencias de los protocolos MAC en las comunicaciones M2M, distintas organizaciones han desarrollado protocolos que buscan optimizar el desempeño en redes de este tipo. Sin embargo, estos protocolos se basan en estrategias diseñadas para comunicaciones de tipo humano, ya sean de contienda o libres de contienda. Además, en su mayoría están adaptados a los requerimientos específicos de algunas aplicaciones del IoT. Por otra parte, diferentes autores han dedicado sus esfuerzos a diseñar protocolos para comunicaciones M2M conocidos como protocolos híbridos, los cuales combinan las mejores características de los mecanismos de contienda y libres de contienda. No obstante, estos mecanismos también tienen como base algunas variantes de protocolos MAC consolidados que, si bien ofrecen un desempeño adecuado ante ciertos escenarios de las comunicaciones M2M, fueron diseñados para comunicaciones de tipo humano, por lo cual heredan una serie de deficiencias que se exacerban en presencia de tráfico intenso.

Este trabajo de investigación presenta una propuesta de protocolo MAC híbrido para comunicaciones M2M, llamada 2CA-R^2. Este protocolo se basa en el algoritmo de resolución de colisiones conocido como *Adaptive*-2C o de las dos celdas adaptativo. El enfoque utilizado expande las opciones existentes en cuanto a mecanismos de contienda, ya que es distinto a la utilización de una variante de un protocolo MAC consolidado. La idea principal es separar los dispositivos que intentan acceder al canal en dos grupos o celdas, un grupo de transmisión y un grupo de espera. A través de mensajes de retroalimentación, los dispositivos pueden minimizar el tamaño del grupo en la celda de transmisión de forma que se produzca el menor número de colisiones posible. El objetivo de ello es generar mejoras en el desempeño logrado con respecto a un mecanismo de contienda consolidado y ampliamente utilizado en redes inalámbricas de área local, conocido como Función de Coordinación Distribuida (*Distributed Coordination Function* o DCF).

Los resultados obtenidos por la evaluación de desempeño a la que se sometió la propuesta muestran que se obtienen mejoras en algunos escenarios de las comunicaciones M2M, con respecto al desempeño de DCF, en cuanto a medidas como la tasa de transferencia efectiva promedio de la red, retardo promedio de acceso y tiempo de entrega de un conjunto completo de paquetes. Estos resultados, que se obtuvieron por simulación, fueron además validados a través de un modelo matemático basado en una cadena de Markov.

AGRADECIMIENTOS

Mi más sincero agradecimiento:

Por su gran dirección y asesoría a

Dr. Miguel López Guerrero

Por su constante apoyo y motivación a

Dr. Luis Orozco Barbosa

Dr. Luis Alberto Vásquez Toledo

Y a mis padres, hermanos, amigos y compañeros

que con su valiosa cooperación

contribuyeron en alguna forma

TABLA DE CONTENIDO

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

LISTA DE SÍMBOLOS

Símbolo	Concepto
$\%p_d$	Porcentaje promedio de paquetes desechados
2C	Dos Celdas
ACK	Mensaje de confirmación o *Acknowledgment*
AP	Punto de Acceso
BER	Tasa de bit en error
BS	Estación Base
C	Colisión
c_n	Código ortogonal en CDMA
CI	Intervalo de Contienda
CRI	Intervalo de Resolución de Colisiones
CTS	Libre para enviar o *Clear To Send*
CW	Ventana de Contención
D_A	Retardo de acceso promedio
D_Q	Retardo de encolamiento promedio
DCF	Función de Coordinación Distribuida o *Distributed Coordination Function*
DIFS	Espacio Entre Tramas de DCF
DQ	Fila Distribuida o *Distributed Queuing*
DTI	Intervalo de Transmisión de Datos
f_n	Banda de frecuencias en FDMA
f_t	Mensaje de retroalimentación de la ranura t
$H_{A,B}$	Número esperado de *minislots* para alcanzar el estado B partiendo de A
HTC	Comunicaciones de Tipo Humano
IoT	Internet de las Cosas
ms_n	Número promedio de *minislots* para resolver una colisión de tamaño n
M2M	Máquina a Máquina
MAC	Control de Acceso al Medio
MTC	Comunicaciones de Tipo Máquina
N	Multiplicidad de colisión

Símbolo	Concepto
$\widehat{N}$	Estimación de N
NC	No Colisión
p	Probabilidad
$\mathbf{P}$	Matriz de transición
$P_{A,B}$	Probabilidad de ir del estado $A = (a, b)$ al estado $B = (c, d)$ en 2CA-R^2
PER	Tasa de paquetes en error
$p(N)$	Probabilidad en función de N
r_t	Variable de estado en algoritmos 2C y *Adaptive*-2C
RACH	Canal de Acceso Aleatorio
RAW	Ventana de Acceso Restringido
RREQ	Mensaje de solicitud de transmisión en CRI de 2CA-R^2
RTS	Solicitud para Transmitir o *Request To Send*
s	Variable de control de posición para DTI en 2CA-R^2
s_{D_A}	Desviación estándar de retardo de acceso
s_n	Ranura de tiempo en TDMA
SIFS	Espacio Corto Entre Tramas
T_N	*Throughput* promedio de la red
T_S	Tiempo promedio de transferencia de un conjunto completo
$\mathrm{T_x}$	Transmisión
TC	Número de paquetes en celda de transmisión en 2CA-R^2
TIM	Mapa de Indicación de Tráfico
W	Espera
WC	Número de paquetes en celda de espera en 2CA-R^2
λ	Tasa de generación de paquetes del proceso de Poisson

CAPÍTULO 1

Introducción

En la era moderna, el ser humano ha tenido una necesidad creciente de estar comunicado con sus semejantes sin importar que se encuentren en distintos puntos geográficos, situación que originó la invención de aparatos como el teléfono a finales del siglo XIX. Sin embargo, no fue sino hasta la década de los 50 del siglo XX, con la instalación de cables submarinos transatlánticos, que existió una red telefónica mundial confiable y rentable, misma que permitió la comunicación a gran distancia y en tiempo real de miles de personas. Posteriormente, en la década de los 80, junto con el inicio de la digitalización de los servicios de telefonía, se llevaba a cabo también el desarrollo y evolución de la computadora personal. Con ello, surgió la necesidad de interconectar computadoras a mediana y larga distancia con el objetivo de facilitar las labores cotidianas de las personas, principalmente en la realización de tareas colectivas dentro de alguna organización. La solución encontrada fue crear una arquitectura que permitiera el intercambio de información y servicios entre este tipo de dispositivos alrededor del mundo. Así, nació la Internet, una red global descentralizada, formada por múltiples redes de comunicación interconectadas, donde la infraestructura de la red telefónica jugó un papel clave [1]. La Internet evolucionó rápidamente, pasando de ser una herramienta exclusiva de corporaciones, instituciones académicas y gubernamentales a convertirse en un medio fundamental para la comunicación y el acceso a información para millones de usuarios domésticos en todo el mundo.

Con la Internet popularizándose de forma acelerada, en la década de los 90 aparecen las redes inalámbricas de área local, así como las redes de telefonía celular con conexión a la Web, consolidándose ambos casos en la primera década del nuevo milenio. El éxito de este tipo de redes trajo consigo una gran cantidad de aplicaciones que requerían conexión a Internet, permitiendo la comunicación en tiempo real entre personas o entre personas y computadoras. Las características del tráfico generado en la red por este tipo de interacciones contribuyeron a definir el concepto de Comunicaciones de Tipo Humano (*Human Type Communications* o HTC), las cuales se refieren al intercambio de información directamente entre personas o entre personas y máquinas, utilizando medios tecnológicos para facilitar la interacción. Sin embargo, hasta ese momento no se tomaba en

cuenta la importancia de tal concepto ya que la gran mayoría de las comunicaciones existentes eran de este tipo, lo cual significa que las redes que las permitían estaban diseñadas solo para tal propósito [2]. A medida que más personas requerían acceso a telefonía móvil e Internet, los prestadores de servicios expandían sus redes y tecnologías. Como consecuencia, en áreas urbanas densamente pobladas, las redes comenzaron a enfrentar desafíos de congestión, lo que resaltó la importancia de contar con mecanismos de Control de Acceso al Medio (*Media Access Control* o MAC) eficientes para optimizar la comunicación y minimizar interferencias.

Con el paso de los años, el avance de la tecnología permitió la aparición y posterior miniaturización de dispositivos como teléfonos inteligentes, asistentes personales o tabletas, con lo cual Internet dejó de estar limitada a computadoras. En el momento en que se comenzaron a desarrollar programas y aplicaciones que presentaban interacción automática entre dispositivos, es decir, con una mínima intervención humana para el intercambio de información entre estos, fue cuando un hecho se hizo evidente: todas las tecnologías e infraestructuras de red diseñadas y construidas hasta ese punto, las cuales evidentemente incluían un mecanismo MAC, no estaban preparadas para este nuevo tipo de comunicaciones, un problema que se fue acrecentando con el correr del tiempo en espera de soluciones [2].

1.1. Comunicaciones Máquina a Máquina y su relación con el Internet de las Cosas

La implementación de sensores, actuadores y otros dispositivos con conexión a Internet, marcó, en la primera década del siglo XXI, el inicio de la Internet de las Cosas (*Internet of Things* o IoT), un paradigma donde objetos cotidianos, desde electrodomésticos hasta vehículos, se conectan a la red para intercambiar información y ejecutar tareas de manera autónoma. Esta interconexión promete crear ciudades inteligentes que lleven a la transformación de las industrias y mejoren la eficiencia en nuestras vidas, dando lugar a una nueva era tecnológica.

De manera formal, el IoT es el conjunto de dispositivos, también llamados nodos o estaciones, conectados entre sí con aplicaciones o servicios en la nube o en un extremo de la red que permiten un enlace potencial entre la vida real y el mundo virtual [3]. La puesta en marcha del IoT dio lugar a un nuevo tipo de comunicaciones conocidas como Comunicaciones de Tipo Máquina (*Machine Type Communications* o MTC), también llamadas comunicaciones Máquina a Máquina (*Machine to Machine* o M2M), mismas que habilitan la posibilidad de comunicar dos o más dispositivos del IoT para intercambiar información [4].

El término M2M originalmente se utilizó para describir la tecnología de red que permite a diversos dispositivos electrónicos realizar mediciones, comunicarse entre sí, y tomar decisiones, con la menor intervención humana posible [5]. Las MTC se producen en redes en las que pueden coexistir un gran número de dispositivos que recaben datos, realicen procesamiento y ejecuten tareas con base en criterios preestablecidos, dependiendo de la aplicación del IoT para las que sean requeridas; todo lo anterior realizado idealmente en tiempo real, a un bajo costo y con una alta eficiencia energética. Estas características hacen que el tipo de tráfico generado en redes MTC sea muy diferente cuando se compara con el tráfico propiciado por las HTC [2].

En general, las comunicaciones M2M se representan mediante un modelo o arquitectura establecido por el proyecto oneM2M, una asociación global establecida en 2012 y que está constituida por ocho de las principales organizaciones de desarrollo de estándares de telecomunicaciones, incluyendo al Instituto Europeo de Normas de Telecomunicaciones (*European Telecommunications Standards Institute* o ETSI) y a la Asociación de la Industria de Telecomunicaciones (*Telecommunications Industry Association* o TIA). Dicha arquitectura, que fue adoptada de la propuesta publicada por el ETSI en el año 2010 [6], se compone de tres dominios interconectados, tal como se muestra en la Figura 1.1. Estos son el dominio de dispositivo o de área, el dominio de red o de comunicaciones y el dominio de servidor o de aplicación (ver [7] y [8]), mismos que se describen a continuación:

- **Dominio de dispositivo o de área.** Este dominio está formado por una red de dispositivos terminales y puertas de enlace o *gateways*. Los dispositivos, o nodos, recopilan datos, en su mayoría sensoriales, de diferentes partes dentro del dominio del área y los transmiten a un *gateway*, el cual gestiona los paquetes de datos recibidos. Luego, cada *gateway* reenvía los paquetes de datos a través de un dominio de red con destino al servidor del dominio de aplicación. Al igual que los dispositivos, los *gateways* pueden tomar decisiones colaborativas con base en técnicas de procesamiento de la información, tales como fusión de datos y clasificación mediante redes neuronales artificiales, para transmitir sus datos [9]. Típicamente, el *gateway* al que se conecta un grupo de dispositivos en un área común, es el punto de acceso o estación base que los provee de servicios de red.

- **Dominio de red o de comunicaciones.** Este actúa como una interfaz entre el dominio de área y el dominio de aplicación. En este dominio se utilizan redes de largo alcance para proporcionar rutas eficientes constituidas por canales de un solo salto, o de múltiples saltos, que sean rentables, confiables y con una amplia cobertura para transmitir la información hasta el dominio de la aplicación.

- **Dominio de servidor o de aplicación.** Este consta de un servidor y clientes de aplicaciones M2M. El servidor actúa como un punto de integración para almacenar toda la información transmitida desde el dominio del dispositivo. También proporciona los datos en tiempo real a las aplicaciones cliente del IoT para la gestión de tareas de monitorización remota.

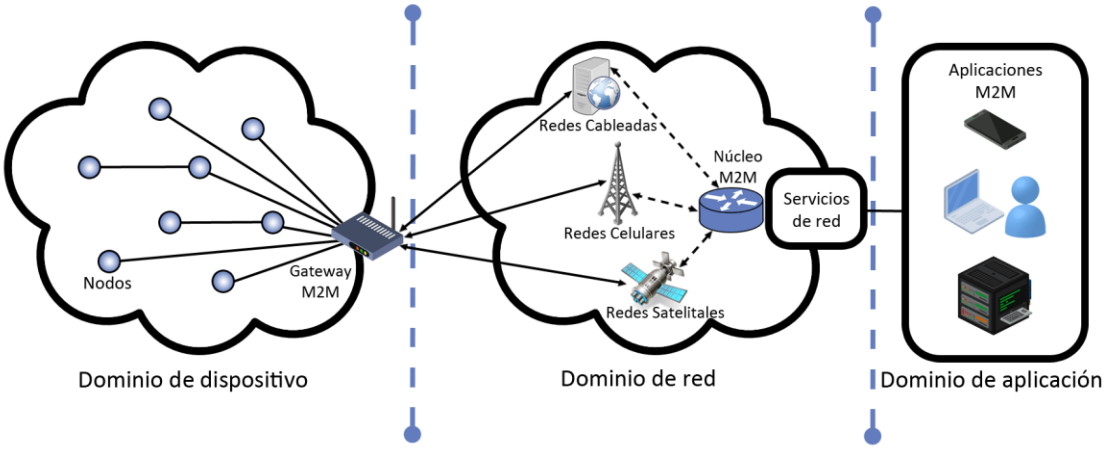

Figura 1.1. Arquitectura de red de las comunicaciones M2M

Una certeza es que en los próximos años el IoT contará con millones de máquinas conectadas a Internet, de las cuales cientos o hasta miles se encontrarán coexistiendo en extensiones geográficas relativamente pequeñas. De hecho, de acuerdo con predicciones hechas en años recientes, para el año 2030 existirán alrededor de 41 mil millones de concentradores o puntos de acceso exclusivos para IoT [10]. Esto implicará la puesta en marcha de un número mucho más elevado de nodos o dispositivos terminales de este tipo, mismos que generarán una enorme cantidad de tráfico exclusivo de comunicaciones M2M, principalmente en el dominio de dispositivo de la arquitectura presentada.

Además, se debe considerar que los dispositivos diseñados para comunicaciones M2M generan datos en función de la aplicación del IoT en la que están involucrados. No obstante, en términos generales, se ha identificado que esta generación ocurre a intervalos regulares (es decir, de manera periódica) o como resultado de la ocurrencia de un evento (es decir, de forma aleatoria). Este patrón de generación de datos conlleva a que el flujo de información en el canal se dirija predominantemente en sentido ascendente, es decir, desde los dispositivos hacia el punto de acceso (ver [11] y [2]). De forma paralela, es conocido que el tamaño de la carga útil en los paquetes de las comunicaciones M2M suele ser del orden de unos pocos cientos de bytes. Un ejemplo de ello se observa en aplicaciones relacionadas con el cuidado de la salud o *healthcare* [11], medición inteligente en hogares o *smart metering* [2], y estacionamientos inteligentes o *smart parking* [2], donde cada paquete generado tiene un tamaño de alrededor de 100 bytes.

Con el crecimiento exponencial del Internet de las Cosas, las características del tráfico generado por las comunicaciones M2M han planteado importantes desafíos que deben abordarse para garantizar un servicio eficiente y eficaz. Estos retos son fundamentales para materializar el escenario proyectado de ciudades inteligentes en el futuro, ya que las tecnologías e infraestructuras inicialmente desarrolladas y construidas para las comunicaciones de tipo humano, no son capaces de dar soporte a este nuevo paradigma que hoy es una realidad. Tal situación exige explorar nuevas alternativas en el diseño de sistemas y mecanismos específicos para las comunicaciones de tipo máquina, asegurando al mismo tiempo su coexistencia con las comunicaciones de tipo humano. Lo anterior incluye, desde luego, a los protocolos MAC.

1.2. Objetivos

1.2.1. Objetivo general

Desarrollar un protocolo MAC para comunicaciones M2M, dentro del entorno del dominio de área, que mejore las prestaciones del servicio de transferencia de datos respecto de otros protocolos existentes.

1.2.2. Objetivos particulares

- Mejorar la utilización del canal de comunicación debida a las transmisiones de paquetes de los dispositivos en el dominio de área, respecto a las soluciones actualmente disponibles.
- Disminuir el retardo de acceso en las transmisiones de paquetes de los dispositivos en el dominio de área, respecto a las soluciones actualmente disponibles.
- Proporcionar un servicio que garantice condiciones de mayor equidad en las transmisiones de paquetes de los dispositivos en el dominio de área, respecto de las soluciones actualmente disponibles.
- Mejorar la escalabilidad de los dispositivos en el dominio de área, respecto a las soluciones actualmente disponibles.
- Validar el protocolo desarrollado.

1.3. Método de investigación

El proyecto se desarrolló a través del siguiente método:

1. Investigación documental sobre los modelos de tráfico, estándares y protocolos MAC utilizados en el dominio de área en las comunicaciones M2M, con el fin de conformar el estado del conocimiento.
2. Identificación de los nichos de oportunidad alrededor de la información obtenida en la investigación documental.
3. Propuesta de un protocolo de acceso al medio en un entorno de un gran número de dispositivos que desean transmitir pequeñas cantidades de datos de manera simultánea, dentro del dominio de área de una arquitectura de comunicaciones M2M.
4. Implementación de la propuesta en un simulador de eventos discretos.
5. Evaluación de la propuesta mediante simulaciones por computadora.
6. Validación del protocolo MAC desarrollado mediante un modelo matemático.
7. Comunicación de resultados.

1.4. Aportes de la investigación

La principal contribución de este proyecto de investigación se resume en la introducción y evaluación de un protocolo MAC híbrido adaptativo, que se basa en el algoritmo *Adaptive*-2C, y que fue desarrollado para operar en el dominio de área de las comunicaciones M2M. La evaluación del protocolo se llevó a cabo mediante simulaciones por computadora y a través de un modelo matemático basado en una cadena de Markov. A pesar de que el protocolo es simple de implementar, los resultados demuestran que proporciona un acceso más justo al canal y logra un rendimiento superior en comparación con el mecanismo *Distributed Coordination Function* o DCF ampliamente utilizado.

Derivado de lo anterior, se publicó un artículo de revistada listada en el índice *Journal Citation Reports* o JCR [12], así como un capítulo de libro que se encuentra en proceso de revisión. Adicionalmente, se realizó una estancia de investigación en el Instituto de Investigación en Informática (I^3A) de la Universidad de Castilla-La Mancha (UCLM), Campus Albacete, España.

1.5. Estructura del documento

El resto de este documento está estructurado de la siguiente manera:

- En el capítulo 2, se presenta el marco conceptual, así como una descripción general de algunos estándares y trabajos relacionados.
- En el capítulo 3 se presenta la propuesta de un protocolo MAC híbrido para comunicaciones M2M, llamado 2CA-R^2.
- En el capítulo 4 se presenta la evaluación de desempeño de la propuesta y su validación mediante un modelo matemático.
- En el capítulo 5 se presentan las conclusiones de la investigación y se describen las perspectivas de investigación en torno al protocolo propuesto.

CAPÍTULO 2

Marco conceptual y trabajo relacionado

La llegada de la Internet impulsó la implementación de estándares y protocolos específicos para lograr un funcionamiento eficiente. La naturaleza global de Internet requería que los diferentes sistemas y dispositivos, creados por diversas compañías y en distintos países, pudieran utilizar el mismo lenguaje, es decir, comunicarse de manera eficiente sin importar sus diferencias técnicas. La creación de protocolos como TCP/IP permitió el intercambio de datos entre redes heterogéneas, sentando las bases para el crecimiento exponencial de Internet. Sin estos estándares, la interconectividad y la expansión de la red hubieran sido imposibles, ya que cada dispositivo hubiera necesitado su propio método de comunicación, creando un caos por las incompatibilidades.

Como ya se ha mencionado, la evolución acelerada del Internet ha generado escenarios de congestión de red principalmente en áreas densamente pobladas. La aparición del IoT llegó a acentuar este problema llevando a los diseñadores de redes a buscar soluciones, tomando en cuenta que en la actualidad existen dos tipos distintos de tráfico, el propiciado por las HTC y aquel que se debe a las MTC. Al respecto, la coordinación de acceso al canal en una red recae en una subcapa de la capa de enlace de datos, conocida como MAC, la cual forma parte del modelo de referencia OSI para redes de comunicaciones. La subcapa MAC es la responsable de decidir a qué dispositivos, en qué momento y de qué forma se otorga acceso al canal compartido [13]. Debido a que las comunicaciones M2M pueden involucrar a una gran cantidad de dispositivos, con un flujo de tráfico de datos muy distinto al que generan las HTC, los protocolos MAC para MTC deben diseñarse tomando en cuenta un amplio conjunto de requisitos, tales como una tasa de transferencia efectiva alta, gran escalabilidad, eficiencia energética y una baja latencia [14]. Además, también se debe considerar que las características de las aplicaciones que producen tráfico M2M requieren, en general, que el acceso sea inalámbrico. Ejemplo de ello son el *smart metering*, rastreo o *tracking* y *healthcare*. En consecuencia, los dispositivos del IoT normalmente se energizan por medio de una batería y tienen un poder de procesamiento limitado; por lo tanto, los protocolos MAC para comunicaciones M2M deben ser más simples de implementar y tener costos computacionales más bajos en comparación con los protocolos utilizados en HTC. Por esta razón, en años recientes, distintas organizaciones a nivel mundial se han enfocado en diseñar

protocolos inalámbricos MAC que se basan en protocolos de contienda, protocolos libres de contienda y protocolos híbridos (ver [14] y [15]). A continuación, se describe la diferencia entre cada uno de ellos.

2.1. Protocolos de contienda

Los protocolos MAC que se basan en contienda son los más simples en términos de configuración e implementación. En estos protocolos, los dispositivos que forman parte de la red contienden entre ellos para acceder al canal y transmitir datos. Una gran desventaja de estos protocolos es que, al aumentar la cantidad de dispositivos, la probabilidad de colisión también aumenta generando una escalabilidad limitada en la red (ver [14] y [15]). A continuación, se resumen el principio de funcionamiento de algunos protocolos de contienda ampliamente utilizados.

2.1.1. ALOHA

El protocolo ALOHA fue uno de los primeros protocolos MAC en redes de comunicaciones. Este protocolo permite que un dispositivo envíe datos en cualquier momento, sin coordinación previa. Tan pronto como este tiene disponible un paquete de datos, lo transmite sin escuchar si el canal está ocupado o no. Si la transmisión es exitosa, es decir, no se interfiere con otras transmisiones, el dispositivo continúa con su operación. Si ocurre una colisión, es decir, si se detecta que dos o más dispositivos transmiten al mismo tiempo, estos deben retransmitir después de un tiempo aleatorio. A pesar de su simplicidad de operación, ALOHA tiene una utilización máxima del canal de aproximadamente 18.39%, asumiendo que el tráfico sigue un proceso de Poisson [16]. Esto se debe a que el número de colisiones crece rápidamente conforme aumenta el número de dispositivos. La Figura 2.1 muestra un ejemplo del funcionamiento del protocolo ALOHA.

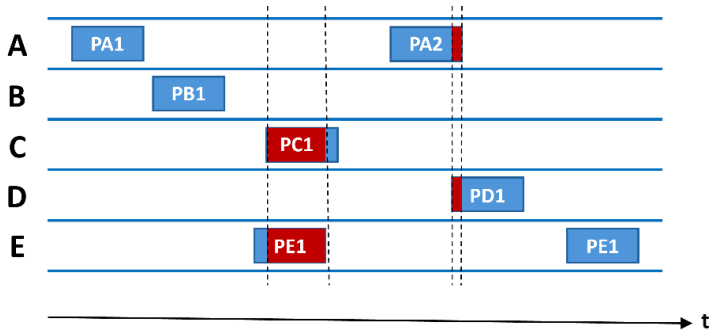

Figura 2.1. Ejemplo de operación del protocolo ALOHA

En este ejemplo se pueden observar cinco dispositivos denotados como A, B, C, D y E. El eje horizontal representa el tiempo y los bloques en colores representan los paquetes de datos transmitidos por cada dispositivo a través del canal compartido. Cuando dos o más paquetes colisionan en el medio, como en el caso de los paquetes PC1 y PE1 o en el caso de los paquetes PA2 y PD1, estos deben ser retransmitidos después de un tiempo aleatorio.

2.1.2. ALOHA ranurado (*slotted* ALOHA)

Es una mejora del protocolo ALOHA original diseñada para reducir la probabilidad de colisiones en un canal compartido. Esta variante introduce una división del tiempo en intervalos discretos o «ranuras», de modo que los dispositivos solo pueden transmitir al inicio de una ranura de tiempo. Un dispositivo con datos para enviar debe esperar al inicio de la siguiente ranura antes de transmitir. Si un único dispositivo transmite en una ranura, la transmisión es exitosa. Si dos o más dispositivos intentan transmitir en la misma ranura, ocurre una colisión, y los dispositivos involucrados deben esperar un número aleatorio de ranuras antes de intentar retransmitir. El protocolo *slotted* ALOHA, también llamado s-ALOHA, presenta una mayor eficiencia, con respecto a ALOHA original, con una máxima utilización del canal del 36.79%, asumiendo que el tráfico sigue un proceso de Poisson [16]. Sin embargo, requiere una sincronización precisa de las estaciones para garantizar que todas compartan el mismo esquema de ranuras de tiempo [13]. La Figura 2.2 muestra un ejemplo similar al de la Figura 2.1, pero esta vez empleando el protocolo *slotted* ALOHA.

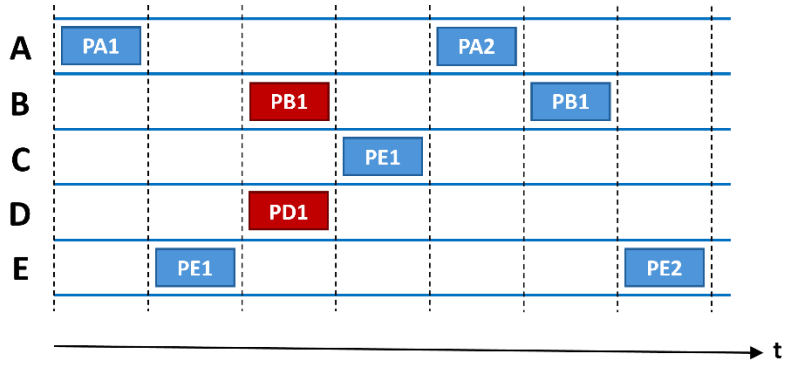

Figura 2.2. Ejemplo de operación del protocolo s-ALOHA

En este ejemplo se puede observar que, cuando dos o más dispositivos eligen la misma ranura de transmisión, sus paquetes colisionan en el medio, como en el caso de PB1 y PD1. Bajo tal situación, cada uno de estos paquetes deben retransmitirse después de un número aleatorio de ranuras.

2.1.3. CSMA (*Carrier Sense Multiple Access*)

La idea básica del protocolo de Acceso Múltiple por Detección de Portadora, o CSMA, es que los dispositivos en la red deben monitorizar el canal compartido antes de transmitir, con el fin de disminuir las colisiones con las transmisiones de otros dispositivos [17]. El protocolo CSMA tiene tres variantes que se diferencian en el comportamiento de un dispositivo cuando desea transmitir y el medio está ocupado [13]:

- **CSMA No Persistente.** Los dispositivos no monitorizan continuamente el estado del canal. Si un dispositivo desea transmitir y detecta que el canal está ocupado, espera un tiempo aleatorio antes de volver a verificar si está libre. En cambio, si el dispositivo detecta que el canal está libre, transmite inmediatamente.
- **CSMA 1-Persistente.** Los dispositivos monitorizan constantemente el canal. Si el canal está libre, el dispositivo transmite inmediatamente, en caso contrario sigue escuchando hasta que se libere para transmitir de inmediato.
- **CSMA p-Persistente.** Los dispositivos monitorizan constantemente el canal. Cuando el canal está libre, un dispositivo transmite con una probabilidad p o espera una unidad de tiempo predefinida con una probabilidad $(1 - p)$ antes de intentar transmitir. Si el canal está ocupado, el dispositivo espera hasta que esté libre.

Una variante más reciente del protocolo CSMA, llamada CSMA/CA (con Evasión de Colisiones o *with Collision Avoidance*), es una de las más utilizadas en redes inalámbricas, misma que implementa tiempos de espera a fin de evitar colisiones. En el protocolo CSMA/CA, un dispositivo con un paquete de datos a transmitir (dispositivo fuente) debe seguir el procedimiento que se representa en la Figura 2.3, el cual se describe a continuación [17]:

1. Si el canal está libre, el dispositivo fuente transmite su paquete de datos después de un periodo de tiempo, conocido como Espacio Entre Tramas de DCF o DIFS.
2. Si el canal está ocupado, lo cual no se refleja en la Figura 2.3, el dispositivo fuente selecciona aleatoriamente un valor de espera (*backoff*) que se encuentra dentro de un intervalo de tiempo conocido como Ventana de Contención o CW. Luego, efectúa una cuenta decreciente desde este valor hasta cero siempre que detecte que el canal se encuentre libre. En caso de detectar que el canal se ocupa, el valor del contador entra en pausa.
3. Una vez que el contador llega a cero, el dispositivo fuente transmite su paquete de datos al receptor (dispositivo destino).
4. El dispositivo fuente queda a espera de un mensaje ACK o *Acknowledgement* por parte del receptor, mismo que será enviado después de un periodo de tiempo conocido como Espaciado Corto Entre Tramas o SIFS. Si el ACK es recibido, se confirma que la transmisión fue exitosa. En caso contrario, incrementa el tamaño de la CW y reinicia el proceso desde el paso 2.

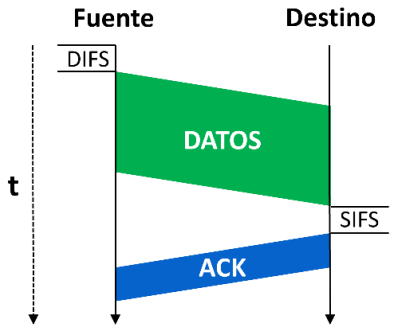

Figura 2.3. Ejemplo de operación de CSMA/CA con canal libre

Es importante mencionar que el intervalo inicial de la CW se encuentra en un rango entre 0 y un valor preestablecido $CW_{min}-1$. Cada vez que se produce una transmisión no exitosa, el valor de la CW se duplica, lo que se conoce como *Binary Exponential Backoff*. Si las transmisiones no exitosas persisten, el valor de la CW llegará hasta un máximo permitido CW_{max}.

De forma general, la evolución de la CW se lleva a cabo de acuerdo con la ecuación (2.1),

$$CW = \min(2^k \cdot CWmin,\ CWmax), \tag{2.1}$$

donde $k =$ número de reintentos de transmisión, *CWmin* es la ventana de contienda mínima y *CWmax* es la ventana de contienda máxima. Una vez que la transmisión es exitosa, la ventana de contienda se restablece a su valor mínimo.

El protocolo CSMA/CA cuenta con un esquema opcional de reserva del canal compartido llamado Solicitud para Enviar/Libre para Enviar o *Request to Send*/*Clear to send* (RTS/CTS). Este esquema favorece ampliamente a la evasión de colisiones cuando se presenta un problema conocido como nodo oculto o terminal oculta. Este problema surge dado un escenario del cual se muestra un ejemplo en la Figura 2.4. Como se aprecia, dos o más dispositivos fuente, denotados como A y B respectivamente, pueden tener en común el mismo receptor y por consiguiente caer en el rango de alcance o área de cobertura de tal receptor, lo cual se representa con el circulo en color coral. Sin embargo, el área de cobertura de cada dispositivo fuente puede o no incluir al resto de ellos, lo cual se representa con los círculos en color verde para los dispositivos A y B. En otras palabras, algunos dispositivos fuente pueden estar ocultos para otro, pero ningún dispositivo está oculto para el receptor [17], tal como ocurre para los dispositivos A y B del ejemplo.

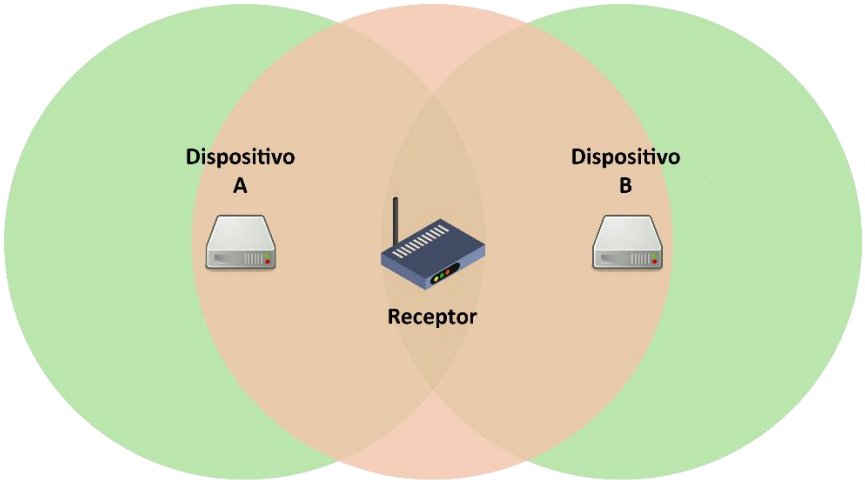

Figura 2.4. Ejemplo de escenario de nodo oculto

Un escenario de nodo oculto es problemático debido a que un primer dispositivo fuente, que se encuentra oculto para un segundo dispositivo, puede estar transmitiendo datos en el instante en que el segundo dispositivo intentará transmitir, asumiendo que el canal se encuentra libre ya que no escucha la transmisión del primero. Esto producirá una colisión en el canal, generando a su vez, desperdicio de tiempo. Para evitar este fenómeno, el protocolo CSMA/CA puede habilitar un esquema de intercambio de mensajes, donde un dispositivo fuente con un paquete de datos para transmitir inicialmente deberá esperar un periodo DIFS para después enviar una trama corta de solicitud RTS. Si el receptor recibe exitosamente la trama RTS, esperará un periodo SIFS y luego responderá difundiendo a todos los dispositivos una trama CTS. Esta trama sirve para informar al dispositivo emisor de la trama RTS que puede transmitir su paquete de datos y también para informar al resto de dispositivos que no deben transmitir durante el periodo de tiempo reservado. Finalmente, si el receptor recibe exitosamente el paquete de datos, esperará un periodo SIFS y luego difundirá un mensaje ACK, indicando a todos los dispositivos que el canal ha quedado libre [17]. La Figura 2.5 muestra un ejemplo del procedimiento de envío de un paquete de datos cuando se usa el protocolo CSMA/CA con el esquema RTS/CTS activo.

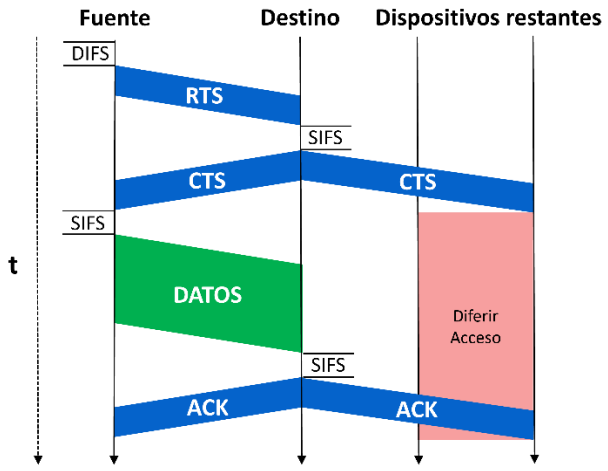

Figura 2.5. Ejemplo de operación de CSMA/CA con esquema RTS/CTS

El protocolo CSMA/CA tiene una implementación específica que se utiliza como mecanismo de acceso al medio por defecto en el estándar IEEE 802.11, y que se conoce como Función de Coordinación Distribuida (*Distributed Coordination Function* o DCF). DCF incluye características adicionales para garantizar la transmisión ordenada tal como el uso obligatorio de ACK, ya que CSMA/CA puede no requerirlo en todas sus implementaciones. Además, DCF incluye un soporte más definido para RTS/CTS

donde existe un componente clave conocido como Límite de Reintentos de transmisión o *Retry Limit*, el cual es un parámetro que impide que un mismo paquete de datos se pueda intentar reenviar sin éxito más allá de un número predeterminado de veces [18].

En general, la principal desventaja de los protocolos MAC basados en contienda es que, a medida que aumenta el número de dispositivos en la red, también aumenta la probabilidad de colisión. Aunque se pueden tomar algunas medidas de control para paliar esta situación, este factor conduce a una escalabilidad limitada (ver [14] y [15]). Otros efectos que surgen en tales circunstancias son un mayor desperdicio de energía proveniente tanto de colisiones como de escuchas inactivas, así como una mayor sobrecarga debido a la transmisión de paquetes de control, que pueden consumir más energía que las transmisiones de paquetes de datos de carga útil [14]. Todos estos efectos se exacerban aún más cuando se producen intentos de acceso simultáneo en gran número.

2.2. Protocolos libres de contienda o de reserva

Los protocolos libres de contienda, también llamados protocolos de reserva, eliminan o minimizan las colisiones al organizar previamente el uso del canal. Estos protocolos aseguran que los dispositivos o estaciones no transmitan simultáneamente utilizando el mismo recurso, lo que evita interferencias y mejora la eficiencia del sistema [14]. A continuación, se describen algunos de los protocolos MAC de reserva más utilizados.

2.2.1. TDMA (*Time División Multiple Access*)

El protocolo TDMA, o de Acceso Múltiple por División de Tiempo, permite que múltiples dispositivos compartan un mismo canal de comunicación dividiendo el tiempo en intervalos llamados ranuras de tiempo. Cada dispositivo tiene asignada una ranura específica durante la cual puede transmitir datos asegurando que no haya interferencia con otros dispositivos.

El funcionamiento de TDMA se basa en una sincronización precisa entre dispositivos. El canal se divide en tramas (*frames*), que se subdividen en varias ranuras. Cada ranura es asignada a un dispositivo, quien puede transmitir únicamente en su turno. Los dispositivos deben estar sincronizados con un reloj común para evitar interferencias. Cuando un dispositivo necesita enviar datos, espera su ranura asignada dentro de la trama actual. Durante su tiempo asignado, el dispositivo utiliza todo el ancho de « disponible del canal. Después de completar su transmisión, el siguiente dispositivo ocupa su ranura correspondiente (ver [19] y [20]). La Figura 2.6 muestra un esquema conceptual de TDMA donde cada

ranura de tiempo se representa como s_n. TDMA es eficiente en sistemas con tráfico constante, como redes celulares y satelitales. Sin embargo, su principal desventaja es que las ranuras ociosas, es decir, ranuras asignadas a dispositivos sin datos para transmitir, no pueden ser utilizadas por otros dispositivos, lo que puede disminuir la eficiencia en sistemas con tráfico variable.

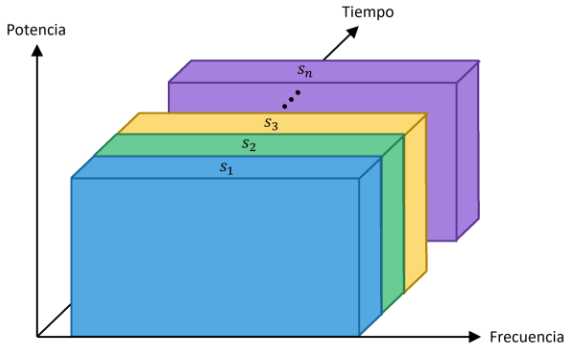

Figura 2.6. Representación conceptual de TDMA

2.2.2. FDMA (*Frequency División Multiple Access*)

El protocolo FDMA, o de Acceso Múltiple por División de Frecuencia, permite a múltiples dispositivos compartir un canal de comunicación dividiéndolo en diferentes bandas[1] de frecuencia. Cada dispositivo recibe una banda exclusiva dentro del espectro total disponible, lo que garantiza que las transmisiones sean independientes y no interfieran entre sí. En FDMA, cada banda de frecuencia es fija y se asigna permanentemente a un dispositivo durante la duración de la comunicación. Los dispositivos transmisores y receptores están sintonizados para operar dentro de sus frecuencias asignadas. Se utilizan filtros para evitar interferencias entre las bandas adyacentes (ver [19] y [20]). La Figura 2.7 muestra un esquema conceptual de FDMA donde cada banda de frecuencia se representa como f_n. Es importante notar que entre cada banda de frecuencias existe una pequeña franja no utilizada, conocida como banda de guarda, cuya función principal es minimizar la interferencia entre bandas adyacentes.

[1] Entiéndase «banda» como un intervalo de frecuencias donde se transmite o recibe una señal.

El funcionamiento es continuo, lo que significa que, una vez asignada la frecuencia, el dispositivo tiene acceso garantizado al canal, incluso si no está transmitiendo. Sin embargo, esto puede llevar a un uso ineficiente del espectro cuando los usuarios tienen tráfico intermitente, ya que la banda de frecuencia permanece reservada pero inactiva (ver [19] y [20]). FDMA se utiliza en sistemas de telecomunicaciones como redes satelitales y de radiofrecuencia. Su principal ventaja es la simplicidad y la ausencia de colisiones. Sin embargo, su asignación estática limita la flexibilidad y la eficiencia en comparación con técnicas más avanzadas como TDMA o CDMA.

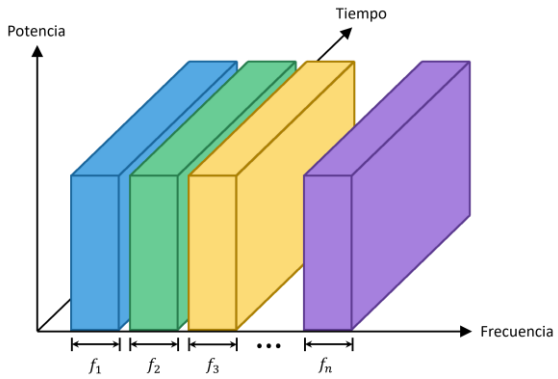

Figura 2.7. Representación conceptual de FDMA

2.2.3. CDMA (*Code Division Multiple Access*)

El protocolo CDMA, o de Acceso Múltiple por División de Código, permite a múltiples dispositivos compartir un mismo canal de comunicación utilizando códigos únicos para diferenciar las transmisiones. A diferencia de FDMA o TDMA, en CDMA todos los dispositivos transmiten simultáneamente en el mismo rango de frecuencias y durante el mismo tiempo. Sin embargo, las señales de cada dispositivo son separadas mediante códigos pseudoaleatorios ortogonales o de baja correlación. El funcionamiento de CDMA se basa en un proceso de dispersión espectral (*spread spectrum*). Cada dispositivo utiliza un código único para codificar sus datos antes de la transmisión (ver [19] y [20]). En el receptor, el mismo código se utiliza para descodificar y recuperar los datos del dispositivo correspondiente, mientras que las señales de otros dispositivos aparecen como ruido de fondo. La Figura 2.8 muestra un esquema conceptual de CDMA donde cada código ortogonal se representa como c_n. CDMA ofrece varias ventajas, como una mayor resistencia al ruido e interferencias, mayor capacidad del canal en comparación con otros protocolos de acceso múltiple, y flexibilidad para soportar distintos tipos de tráfico. CDMA se utiliza en redes móviles, GPS y sistemas

militares debido a su eficiencia espectral y seguridad inherente al utilizar códigos únicos. Sin embargo, requiere sincronización precisa y algoritmos complejos para manejar la codificación y descodificación [17]. Además, requiere ecualización o balanceo de la intensidad de potencia de todos los usuarios ya que su rendimiento puede degradarse si hay demasiados accesos simultáneos.

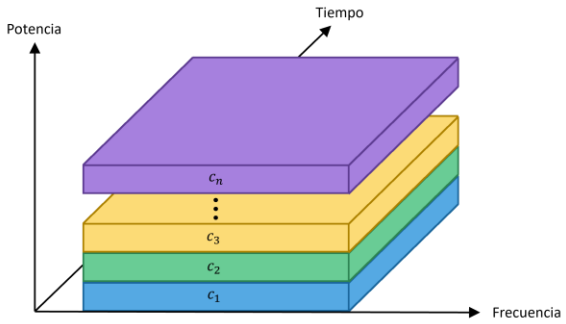

Figura 2.8. Representación conceptual de CDMA

La principal ventaja de los protocolos libres de contienda es su mejor utilización del canal con cargas elevadas. Sin embargo, un problema típico asociado a este tipo de protocolos es que presentan una adaptabilidad limitada a las condiciones cambiantes en el tráfico de la red. Un ejemplo de ello se presenta cuando en TDMA o FDMA un dispositivo se queda sin datos para transmitir, provocando un desperdicio de recursos al no utilizar sus ranuras de tiempo o bandas de frecuencia reservadas. Otro problema que se debe tener en cuenta es la sobrecarga generada por el procedimiento de asignación de recursos. Además, los protocolos de reserva tienen una serie de desafíos técnicos que deben abordarse. Por ejemplo, TDMA requiere una estricta sincronización de reloj para evitar fallas de comunicación debido a interferencias entre los dispositivos [15]. Esta característica implica un alto consumo de energía ya que requiere la transmisión y detección regular de una señal de reloj [21]. A su vez, FDMA no es adecuado para el funcionamiento con dispositivos IoT de bajo costo, ya que requieren circuitos complejos para comunicarse y cambiar entre diferentes canales de radio. Por otra parte, CDMA requiere operaciones computacionalmente costosas para codificar y descodificar mensajes, lo que lo hace menos apropiado para redes donde los dispositivos carecen de *hardware* especial [15].

2.3. Protocolos híbridos

Los protocolos híbridos pertenecen a una categoría que pretende incorporar las mejores características, tanto de los protocolos de contienda como de los de reserva, proporcionando un control de acceso al medio más eficiente. Estos protocolos consideran las características de la red para lograr condiciones de equilibrio. Un enfoque posible para implementar un protocolo MAC híbrido se muestra en la Figura 2.9. Como se puede apreciar, este enfoque consiste en dividir un ciclo de transmisión en al menos dos etapas, un Intervalo de Contención (*Contention Interval* o CI) y un Intervalo de Transmisión de Datos (*Data Transmission Interval* o DTI). Durante el CI, los dispositivos o nodos con datos para enviar compiten, mediante un paquete de solicitud, por una oportunidad para transmitirlos, utilizando un protocolo de contienda para ello. Típicamente este protocolo suele ser alguna variante de ALOHA o CSMA. Los nodos exitosos durante el CI recibirán la asignación de recursos (por ejemplo, un intervalo de tiempo o una frecuencia determinada) para transmitir sus datos de manera ordenada durante el DTI, lo cual se representa en la Figura 2.9 con las ranuras s_j. Al finalizar ambas etapas, CI y DTI, se completa un periodo conocido como ciclo. En términos generales, este enfoque proporciona una mayor flexibilidad que los protocolos puramente de reserva y una mayor escalabilidad que los protocolos puramente de contienda [22].

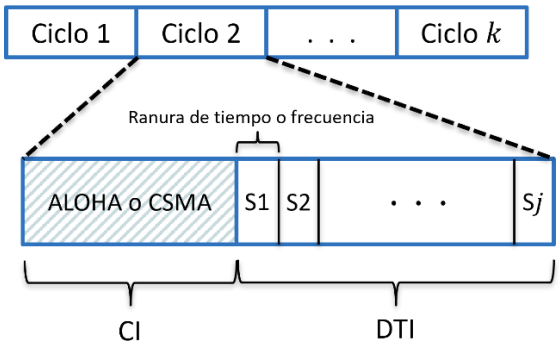

Figura 2.9. Estructura de un ciclo de protocolo MAC híbrido

Como se podrá apreciar en la sección 2.4, varios estándares MAC para comunicaciones M2M, considerados por diferentes tecnologías destinadas a respaldar las aplicaciones del IoT, emplean mecanismos que se basan en alguno de los protocolos de contienda o de reserva mencionados hasta este punto, o bien en un protocolo híbrido conformado por los mismos.

2.4. Protocolos MAC en estándares de comunicaciones M2M

Considerando el modelo propuesto por el ETSI [7], que define la arquitectura de las comunicaciones M2M, el dominio de dispositivo o de área es aquél que está formado por una amplia red de dispositivos que envían datos a sus respectivos puntos de acceso (*Access Point* o AP) o estación base (*Base Station* o BS). En este dominio, los protocolos empleados para proporcionar servicio a los dispositivos, o nodos de la red, se clasifican en tecnologías de largo o corto alcance, dependiendo de si el rango existente, entre ellos y el AP o BS con el que se comunican, es superior, o no, a 1 km de distancia. A continuación, se describen algunos de los estándares más utilizados en comunicaciones M2M, haciendo énfasis en su protocolo MAC. Estos fueron seleccionados debido a que se consideran candidatos para soportar la mayoría de las aplicaciones del creciente IoT (ver [23] y [24]).

2.4.1. Estándares de tecnologías de corto alcance

2.4.1.1. Bluetooth *Low Energy* (BLE)

BLE es una tecnología inalámbrica de corto alcance (hasta 70 m), que se basa en el estándar IEEE 802.15.1, desarrollada para dar servicio a una gran cantidad de dispositivos (hasta 5,917 por AP). BLE define una capa MAC liviana que, en particular, emplea una arquitectura de red maestro/esclavo, llamada *piconet*, donde un nodo maestro (AP) gestiona numerosas conexiones con múltiples nodos esclavos y cada nodo esclavo está asociado con un solo maestro. Los nodos esclavos están por defecto en modo de suspensión y se despiertan periódicamente para escuchar posibles paquetes transmitidos desde el maestro. A su vez, el maestro regula el acceso al medio usando un esquema TDMA para asignar los intervalos de tiempo cuando los esclavos necesitan escuchar. Tras establecer la conexión, el maestro proporciona información al nodo esclavo para la frecuencia del canal seleccionado y el tiempo para el intercambio de datos. BLE ha sido implementada en aplicaciones de datos multimedia y de monitorización y control, tal como se menciona en [23] y [25].

2.4.1.2. Zigbee

Zigbee se basa en el estándar IEEE 802.15.4 y opera en bandas sin licencia, con un alcance entre 10 y 100 m. Utiliza dos tipos de mecanismos de acceso al canal, según la configuración de la red [26]:

- Modo con baliza o *beacon*. Utiliza un mecanismo MAC híbrido, donde todos los nodos con datos a transmitir comienzan un período de espera sincronizado al recibir la transmisión del *beacon* del AP. Luego compiten por el canal en un Período de Acceso a Contienda (*Contention Access Period* o CAP) donde se utiliza un mecanismo que se basa en CSMA/CA. Cuando un dispositivo desea transmitir, espera un período de tiempo aleatorio. Una vez transcurrido este período, si el canal se encuentra inactivo, el dispositivo envía sus datos, de lo contrario, el dispositivo espera un período de tiempo aleatorio hasta que verifica nuevamente la disponibilidad del canal.

- Modo sin *beacon* Utiliza un mecanismo basado en CSMA/CA simple sin ranuras, es decir, detección de canal y tiempos de espera exponenciales aleatorios para la resolución de contiendas.

A pesar de contar con un modo híbrido, la gran mayoría de las implementaciones de Zigbee solo utilizan el modo sin *beacon*, en aplicaciones como redes de sensores ad-hoc, debido al bajo rendimiento de esta tecnología en redes con un elevado número de dispositivos [23].

2.4.1.3. Wireless-HART (*Highway Addressable Remote Transducer Protocol*)

Wireless-HART es una variación del estándar IEEE 802.15.4 para redes inalámbricas centralizadas con un alcance de entre 10 y 600 m. Está diseñado para cumplir con los requisitos de las aplicaciones inalámbricas industriales con restricciones estrictas de sincronización, problemas de seguridad críticos e interferencias severas por obstáculos. El enfoque de Wireless-HART es la comunicación de un solo salto. Utiliza TDMA combinado con saltos de frecuencia en su capa MAC, lo que permite que varios dispositivos transmitan datos al mismo tiempo a lo largo de diferentes canales. Durante el proceso de conexión de los dispositivos a la red, el AP distribuye los enlaces de comunicación y los patrones de salto de canal a los dispositivos [24].

2.4.1.4. Z-Wave

Z-Wave es una tecnología que se basa en el estándar ITU G.9959 y fue desarrollada por la empresa Zensys para proporcionar comunicación inalámbrica entre dispositivos, a una distancia máxima de 100 m, con un enfoque en la automatización residencial a través de sensores y actuadores. Los dispositivos Z-Wave están organizados en una topología de red en malla. Opera en las bandas de radiofrecuencia industrial, científica y médica (*Industrial, Scientific and Medical* o ISM). La capa MAC utiliza CSMA/CA

y tiene capacidad para 232 identificadores de red únicos que permite que la misma cantidad de nodos se conecten a la red simultáneamente. Establece un tiempo de espera cuando ocurre una colisión y garantiza la fiabilidad de los datos mediante recepción de reconocimientos y mecanismos de retransmisión. Además, cuenta con un mecanismo de ahorro de energía mediante un modo de suspensión con un patrón de activación para cada dispositivo [24].

2.4.1.5. Wifi HaLow

Wifi HaLow es una tecnología que se basa en el estándar IEEE 802.11ah y fue propuesta para admitir un gran número de nodos asociados con un AP (más de 8,000), con tasas de datos de hasta 100 kbps. Opera en la banda sub-1 GHz y alcanza rangos de transmisión de hasta 1 km. Introduce tres tipos de estaciones, cada una de ellas asociada a diferentes mecanismos MAC:

- Estaciones de Mapa de Indicación de Tránsito (*Traffic Indication Map* o TIM). Utilizan un método de acceso habilitado por *beacons* con reservas de franjas horarias, denominado Ventana de Acceso Restringido (*Restricted Access Window* o RAW). RAW constituye un período de tiempo entre *beacons* de señalización y consta de uno o varios intervalos de tiempo. El AP es responsable de asignar cada intervalo de tiempo a un grupo de estaciones TIM y transmite esta información dentro de las tramas de los *beacons*. A su vez, las estaciones TIM, al recibir la información de RAW, identifican si se les permite competir, o no, por el acceso al medio en un intervalo de tiempo, mediante DCF. Esta técnica asegura un acceso al espectro entre un gran número de nodos, reduce el número de intentos de acceso simultáneos y maximiza la utilización del canal.
- Estaciones no TIM. A diferencia de las estaciones TIM, no requieren de la escucha de *beacons* y negocian directamente con el AP para obtener un tiempo de transmisión asignado en una RAW periódica (*Periodic* RAW o PRAW) donde el acceso para estaciones TIM está prohibido.
- Estaciones no programadas. No requieren de la escucha de ningún *beacon* antes de la transmisión y el AP asigna intervalos de tiempo fuera de ambas ventanas restringidas para su acceso esporádico al canal.

Al lector interesado se le sugiere revisar las referencias [23] y [27] para detalles adicionales.

2.4.2. Estándares de tecnologías de largo alcance

2.4.2.1. Long Term Evolution (LTE)

LTE es un estándar de acceso por radiofrecuencias que forma parte del *3rd Generation Partnership Project* o 3GPP, un organismo de estandarización internacional que define las características técnicas para las tecnologías móviles [28]. LTE está diseñado para comunicaciones inalámbricas móviles de alta velocidad sobre la infraestructura de telefonía celular. Debido a las limitaciones de LTE, en cuanto al número de dispositivos conectados simultáneamente, el 3GPP desarrolló mejoras destinadas al soporte de las comunicaciones M2M. Este conjunto de mejoras se denomina *LTE for Machines* o LTE-M. Como método de acceso, los nodos en la red utilizan el Canal de Acceso Aleatorio (*Random Access Channel* o RACH) para realizar la asociación de red inicial, solicitar recursos de transmisión y restablecer una conexión al AP. El modo de acceso puede ser de contienda o libre de contienda [23]:

- En el modo libre de contienda, el AP asigna recursos de acceso específicos para las solicitudes que requieren una alta probabilidad de éxito (acceso restringido por retraso) mediante FDMA.
- El modo de acceso de contienda utiliza *slotted* ALOHA multicanal, es decir, los preámbulos mutuamente ortogonales son utilizados por cada nodo para competir en las ranuras de acceso aleatorio disponibles.

LTE-M tiene una tasa de datos máxima de 1 Mbps tanto en el enlace ascendente como en el enlace descendente y un rango máximo de alcance de 5 km [24]. Vale la pena mencionar que LTE-M está diseñado para operar en redes 4G LTE y es compatible con redes 5G a través de la infraestructura heredada (ver [29] y [30]).

2.4.2.2. Narrow Band IoT (NB-IoT)

NB-IoT es un estándar de acceso por radio, promovido por el 3GPP, que opera dentro de la gama de recursos de LTE. Explota las bandas de guarda entre canales LTE o utiliza recursos dedicados en la frecuencia. NB-IoT tiene en cuenta la información de intensidad de la señal con el fin de reducir el costo del dispositivo y el consumo de energía, admitiendo una gran cantidad de dispositivos de bajo rendimiento. Sin embargo, utiliza los mismos métodos MAC de LTE, añadiendo la capacidad de resolver colisiones en el canal mediante CDMA [23].

Esta tecnología tiene tasas de transmisión de datos de enlace descendente y ascendente de hasta 200 kbps, operando en *Half-Duplex*. La tecnología NB-IoT presenta pérdidas menores, respecto a LTE, en la transmisión, lo que otorga un radio de cobertura de hasta 20 km con línea de vista [24]. Por esta razón NB-IoT puede operar en redes 4G y es compatible con redes 5G [30].

2.4.2.3. Weightless

Weightless es un estándar abierto de tecnología de Red de Área Amplia de Bajo Consumo (*Low Power Wide Area Network* o LPWAN). Fue diseñada por el Weightless *Special Interest Group* (Weightless SIG) para proporcionar comunicaciones M2M de costo bajo, utilizando espectro de baja frecuencia (sub-GHz) y técnicas que permiten las comunicaciones en un alcance de hasta 5 km [24]. Opera en un sistema típico de topología en estrella compuesto por los dispositivos finales (*End Devices* o ED) y su estación base (BS). El método MAC consta de una parte de enlace descendente seguida de una de enlace ascendente. La BS asigna oportunidades de transmisión de enlace ascendente a los ED. Esta asignación se transmite en las ranuras de enlace descendente, mientras que las transmisiones se producen en las ranuras de enlace ascendente. El acceso múltiple de enlace ascendente se produce mediante una combinación de FDMA y TDMA. En el caso de una asociación de red inicial o de transmisiones de mensajes no programadas, el acceso al canal se lleva a cabo aprovechando una variación de *slotted* ALOHA. A su vez, Weightless emplea mecanismos para reducir el número de colisiones, los cuales incluyen la configuración dinámica del número de ranuras de acceso que se basan en contienda y la priorización de dispositivos para la restricción de acceso a ciertas clases [23]. Cabe señalar que Weightless tiene tres variaciones, Weightless-P, Weightless-N y Weightless-W, cuya diferencia principal está en la banda de frecuencias en la que opera cada una de ellas [24].

2.4.2.4. Long Range Protocol WAN (LoRaWAN)

LoRa es una tecnología inalámbrica LPWAN, desarrollada por la empresa Cycleo, que define la capa física para utilizar el espectro expandido con un rango de transmisión de hasta 15 km, proporcionando un protocolo de enlace simple. A su vez, LoRaWAN es una especificación de red propuesta por LoRa Alliance que ofrece una capa MAC ligera que se basa en la capa física de LoRa, la cual permite tener más de 10,000 dispositivos conectados en un mismo AP. LoRaWAN define tres clases diferentes de dispositivos, que, aunque permiten transmisiones bidireccionales, proporcionan enfoques diferentes:

- Clase A: El nodo inicia la comunicación con su AP, solo cuando tiene datos para transmitir, mediante ALOHA. Para recibir mensajes del AP, el nodo abre una ventana de recepción después de cada transmisión. Este enfoque es adecuado para aplicaciones que requieren una respuesta de servidor poco después de la transmisión favoreciendo el consumo de energía en el nodo.
- Clase B: El nodo inicia la comunicación, pero utiliza un esquema de comunicación, con intervalos de tiempo habilitados por mensajes *beacon*, que permite ventanas programadas de recepción de mensajes. También usan ALOHA para transmitir, sin embargo, requiere sincronización entre el AP y cada nodo.
- Clase C: El nodo permanece en el modo de escucha listo para la recepción, excepto cuando se encuentra transmitiendo. Esta operación de escucha siempre activa conduce a latencias extremadamente bajas a costa de un mayor consumo de energía en el dispositivo.

2.4.2.5. Sigfox

La tecnología Sigfox adopta una implementación de banda ultra estrecha, que utiliza bandas de frecuencia sub-GHz para permitir la comunicación de largo alcance (hasta 15 km) para comunicaciones M2M, con velocidades de datos de aproximadamente 100 bps [23]. Con respecto al acceso al medio, Sigfox emplea una técnica que se basa en el protocolo ALOHA, denominada Acceso Múltiple por División de Tiempo y Frecuencia Aleatoria (*Random Frequency* TDMA o RFTDMA). Esta técnica MAC no tiene un método de control de colisiones, lo que permite que los dispositivos accedan al medio en cualquier instante, a cualquier frecuencia (dentro de la banda operativa), sin verificar la ocupación del medio. RFTDMA favorece el bajo consumo de energía al no sondear el medio antes de transmitir y al no utilizar ningún tipo de paquete de sincronización o técnica de mensaje *beacon* [24]. Es importante señalar que RFTDMA elige al azar las frecuencias de transmisión dentro de un intervalo continuo. Sin embargo, tal aleatoriedad de tiempo y frecuencia hace que este esquema sea propenso a colisiones e interferencias. Para hacer frente a este problema, se aplican técnicas de radio definidas por software en el lado del receptor para garantizar un rendimiento general adecuado, a costa de un aumento considerable en la complejidad, el costo computacional y el consumo energético [23].

2.4.3. Comparación entre estándares

En la Tabla 2.1 y Tabla 2.2 se resumen las características de los estándares empleados en las comunicaciones M2M, descritos en las secciones 2.4.1 y 2.4.2.

Tabla 2.1. Resumen comparativo de las tecnologías empleadas en MTC de corto alcance

Standard	BLE (802.15.1)	ZigBee (802.15.4)	Wireless-HART (802.15.4)	Z-Wave (G.9959)	Wifi (802.11 ah)
Rango Típico	70 m	10 – 100 m	10 – 600 m	100 m	100 – 1000 m
Método MAC	TDMA	CSMA/CA con Reserva de Tiempo	TDMA	CSMA/CA	CSMA/CA con Reserva de Tiempo
Escalabilidad	5,917	ND*	ND*	232	8,191

*ND - No disponible

Tabla 2.2. Resumen comparativo de las tecnologías empleadas en MTC de largo alcance

Tecnología	LTE	NB-IoT	Weightless	LoRa	Sigfox
Rango Típico	5 km	20 km (LOS)*	2 - 5 km	5 – 15 km (LOS)*	15 km (LOS)*
Método MAC	s-ALOHA o FDMA con Reserva de Tiempo	s-ALOHA o FDMA o CDMA	s-ALOHA o FDMA con TDMA	CDMA o ALOHA con o sin Reserva de Tiempo	ALOHA
Escalabilidad	Determinada por la Implementación de Red	Determinada por la Implementación de Red	Alta	> 10,000	> 10,000

*LOS – Línea de vista o *line of sight*

2.5. Propuestas de protocolos MAC híbridos para comunicaciones M2M

En términos generales, los protocolos híbridos ofrecen mayor flexibilidad que los protocolos puramente de reserva y escalan mejor que los protocolos puramente de contienda. Debido a estas características, no es sorprendente que en los últimos años hayan aparecido, en la literatura, una serie de protocolos híbridos destinados a ser utilizados en comunicaciones M2M. En esta sección se revisan algunos trabajos relevantes sobre este tema.

El análisis de los trabajos presentados en esta sección es importante por varias razones. Por un lado, para identificar aquellas propuestas de protocolos MAC híbridos que se han presentado con una

estructura de ciclo de transmisión (también llamado *frame* por algunos autores) similar al usado en esta investigación. Es decir, que se encuentren divididas en al menos dos etapas, un intervalo de contienda y un intervalo de transmisión de datos. Adicionalmente, esta revisión permite a) identificar las condiciones en las que usualmente se prueban estos protocolos y tomarlas en cuenta en el diseño de los experimentos; b) identificar las medidas de desempeño usualmente utilizadas por la comunidad y c) identificar posibles herramientas de software con las que se puede llevar a cabo la evaluación de desempeño de la propuesta.

La identificación de las propuestas existentes en la literatura se llevó a cabo utilizando la combinación del término «MAC» con los términos «*hybrid*», «M2M», «MTC» e «IoT». Para la búsqueda se emplearon las bases de datos IEEE Xplore, Scopus, Springer Journals y Web of Science. Es importante destacar que se excluyeron propuestas de protocolos MAC para redes submarinas o multisalto, así como aquellas propuestas basadas exclusivamente en mecanismos de contienda o de reserva, con movilidad en los nodos o que contemplaran más de un punto de acceso o estación base en su análisis de desempeño. La búsqueda abarcó los trabajos propuestos desde el año 2013 hasta 2025.

A continuación, se enlistan los trabajos identificados, con la información esencial requerida para interpretar correctamente los resultados de cada propuesta e identificar en qué condiciones cada solución fue efectiva o limitada.

En [22], Liu *et al*. proponen un protocolo MAC híbrido para redes M2M. En esta propuesta hay una etapa de contienda, llamada Período de Solo Contienda (*Contention Only period* o COP), y una etapa de transmisión de datos, llamada Período de Solo Transmisión (*Transmission Only Period* o TOP). Durante el COP, que se basa en CSMA p-persistente, los dispositivos compiten por oportunidades de transmisión. Los identificadores de los dispositivos que resultan exitosos durante el COP, así como el orden en el cual transmitirán, se envían por la estación base durante una etapa llamada Período de Anuncio (*Announcement Period* o AP). Luego, los dispositivos exitosos pueden transmitir sus datos durante el TOP utilizando TDMA. Los resultados analíticos y de simulación demuestran las ventajas de este protocolo MAC híbrido en comparación con ALOHA, s-ALOHA y TDMA.

El *frame* implementado en el protocolo MAC híbrido propuesto por Verma *et al*. en [31] combina CSMA, DCF y TDMA. Esta propuesta consta de cuatro etapas diferentes: un Período de Notificación (*Notification period* o NP), un Período de Contienda (*Contention Period* o CP), un Período de Anuncio (*Announcement Period* o AP) y un Período de Transmisión (*Transmission Period* o TP). Primero, el punto de acceso transmite una notificación a todos los dispositivos durante el NP para señalar el inicio del CP. Durante el CP, los dispositivos, que tienen datos para transmitir, envían solicitudes de transmisión al punto de acceso utilizando técnicas basadas en el protocolo CSMA. Luego, durante el AP, el punto de acceso señala el final del CP y el comienzo del TP a todos los dispositivos que resultaron exitosos en el período de contienda. Finalmente, durante el TP, los dispositivos transmiten sus paquetes de datos utilizando el modo DCF de IEEE 802.11 dentro de una ranura TDMA para hacer frente a las fallas de comunicación, debido a la falta de sincronización de reloj. A pesar de la sobrecarga de señalización, los autores informan que su propuesta logra un mejor rendimiento en comparación con s-ALOHA y TDMA en términos de tasa de transferencia de datos y retardo de transmisión promedio.

En el protocolo MAC híbrido propuesto para redes M2M por Hegazy *et al*. [32] cada *frame* consta de tres partes principales: Periodo de Solo Contienda (*Contention Only Period* o COP), Periodo de Solo Notificación (*Notification Only period* o NOP) y Periodo de Solo Transmisión (*Transmission Only Period* o TOP). Durante el COP, diferentes dispositivos compiten por intervalos de tiempo de transmisión utilizando «CSMA No Persistente» como estrategia de contienda. En el NOP, el punto de acceso anuncia las respectivas reservas de intervalos de tiempo que se utilizarán en el TOP para transmitir paquetes de datos. El método utilizado para la transmisión es TDMA. Si a un dispositivo que compite con éxito no se le puede conceder un intervalo de tiempo durante el TOP, tendrá la posibilidad de enviar sus datos dentro del intervalo de transmisión de otro dispositivo, sujeto a la condición de que no se supere un valor objetivo para la Relación Señal a Interferencia más Ruido (*Signal to Interference plus Noise Ratio* o SINR). De esta manera, el mecanismo aumenta el número de paquetes transmitidos. El protocolo propuesto logra un *throughput* más alto, un menor tiempo de retardo promedio y una menor tasa de colisión de paquetes en comparación con CSMA No Persistente y *slotted* ALOHA.

Yang *et al*. propusieron en [33] un *frame* MAC escalable asistido por aprendizaje automático o *machine learning*. En este trabajo, cada ciclo, llamado *superframe*, consta de cuatro partes: un Periodo de Rendezvous (*Rendezvous Period* o RP), un Período de Acceso a Contienda (*Contention Access Period* o CAP), un Período de Notificación (*Notification Period* o NP) y un Período Libre de Contienda (*Contention Free Period* o CFP). Una característica particular de esta propuesta es que, mediante el uso de técnicas

de *machine learning*, en el RP un *gateway* detecta dinámicamente la cantidad de dispositivos activos a partir de las señales que estos envían. Según la cantidad de dispositivos activos que se han detectado, *el gateway* puede determinar la longitud óptima (M) del CAP. Luego, cada dispositivo activo selecciona aleatoriamente una de las M ranuras de tiempo y envía un mensaje de solicitud para competir por una ranura de datos que se podrá utilizar en el CFP. Si dos o más dispositivos seleccionan la misma ranura de tiempo en el CAP, se produce una colisión. En tal caso, tienen que esperar un nuevo *superframe* para volver a intentar la transmisión. Una vez que se han recibido los mensajes de solicitud, el *gateway* asigna ranuras de datos a los dispositivos exitosos y los anuncia mediante un mensaje de *broadcast* en el NP. Durante el CFP, los dispositivos a los que se les han asignado ranuras de datos pueden transmitir paquetes de datos utilizando TDMA. A pesar de la mayor complejidad de este *frame*, las simulaciones exhaustivas informadas en [34] demostraron que logra un rendimiento superior en comparación con soluciones como s-ALOHA y protocolos MAC basados en reserva, como TR-MAC [35].

En el protocolo MAC híbrido sugerido por Saad *et al*. en [36], cada *frame* consta de cuatro periodos: Periodo de Notificación (*Notification Period* o NP), Periodo de Contienda (*Contention Period* o CP), Periodo de Anuncio (Announcement *Period* o AP) y Periodo de Transmisión (*Transmission Period* o TP). Al comienzo de cada *frame*, en el NP, todos los dispositivos reciben un mensaje de difusión o *broadcast* desde la Estación Base (BS) que pretende indicar el inicio del CP. Durante el CP, los dispositivos que tienen un paquete de datos para transmitir compiten enviando un mensaje de solicitud de transmisión (REQ) a la BS para reservar una ranura de transmisión utilizando el protocolo s-ALOHA. Si no se produce ninguna colisión, la BS reservará una ranura de tiempo válida en el TP para el dispositivo cuyo REQ fue exitoso, confirmándolo con un mensaje ACK. Una vez finalizado el CP, se inicia el AP. En él, la BS envía un mensaje a todos los dispositivos exitosos con el número de sus ranuras de tiempo reservadas. Después de eso, estos dispositivos transmiten sus paquetes durante el TP utilizando TDMA. Saad et al. informan que su propuesta muestra un rendimiento superior en términos de tasa de transferencia del sistema, retardo promedio de paquetes, tasa de acceso exitoso y tasa de reservación, en comparación con algunos otros esquemas híbridos, tales como CSMA No Persistente/TDMA, CSMA *p*-Persistente/TDMA y s-ALOHA/TDMA).

Un protocolo MAC híbrido sugerido por Lachtar *et al*. en [37] clasifica los dispositivos en diferentes clases dependiendo de su nivel de prioridad. También divide el tiempo en ciclos, donde cada uno de ellos consta de tres períodos: un Período de Notificación (*Notification period* o NP), un Período de Contienda (*Contention Period* o COP) y un Período de Transmisión (*Transmission Period* o TOP). Cada ciclo se inicia con un NP, donde una Estación Base (BS) anuncia el inicio del COP a todos los dispositivos independientemente de su clase. A lo largo de la etapa de COP, los dispositivos, con datos para transmitir, utilizan CSMA *p*-Persistente para enviar solicitudes de transmisión a la BS. Aquí, los dispositivos pertenecientes a una clase con mayor prioridad tendrán una mayor probabilidad de contienda exitosa que los dispositivos pertenecientes a una clase con una prioridad menor. A los dispositivos exitosos se les otorga una ranura para transmitir datos utilizando TDMA en el TOP. Esta arquitectura resulta lograr una tasa de transferencia de datos mayor, un menor retardo promedio, así como un menor consumo de energía en comparación con TDMA y CSMA *p*-persistente [37].

Otras propuestas relevantes fueron introducidas por Olatinwo *et al*. en [38] y [39]. Sin embargo, es importante mencionar que fueron diseñadas para utilizarse en escenarios de Redes Inalámbricas de Área Corporal (*Wireless Body Area Network* o WBAN). En [38] introducen un protocolo MAC híbrido multiclase cuyo *frame* consta de las siguientes cuatro etapas: Fase de Notificación (*Notification Period* o NP), Fase de Contienda (*Contention Period* o CP), Fase de Anuncio (*Announcement Period* o AP) y Fase de Transmisión (*Transmission Period* o TP). Al comienzo de un *frame*, en la NP, todos los dispositivos de la red reciben un mensaje del punto de acceso notificándoles sobre el comienzo de la CP. En la CP, solo los dispositivos que están listos para transmitir compiten por oportunidades de transmisión utilizando el esquema s-ALOHA, y los dispositivos que compiten con éxito envían sus paquetes empleando un esquema TDMA en la TP. Como particularidad de este protocolo, los dispositivos se agrupan en dos clases, donde los dispositivos de clase 1 generan datos críticos que requieren alta confiabilidad y bajo retardo, mientras que los dispositivos de clase 2 generan paquetes que son menos críticos. En función de la clase de cada dispositivo y del tamaño de sus paquetes de datos, el coordinador (es decir, el punto de acceso) tomará las decisiones necesarias para asignar recursos durante el TP. A partir de los resultados de la simulación, se concluye que el protocolo propuesto tiene un mejor rendimiento que un protocolo basado en la combinación de CSMA No Persistente y TDMA, en términos de la tasa de transferencia total del sistema y retardo promedio de paquetes.

Utilizando un enfoque diferente, Olatinwo *et al*. en [39] proponen otro *frame* MAC híbrido para redes WBAN que consta de las siguientes dos etapas: un periodo de contienda, llamado «periodo CSMA/CA», y un periodo de transmisión, llamado «periodo TDMA». Al comienzo del periodo CSMA/CA, el punto de acceso (AP) envía un mensaje a todos los dispositivos de la red. A partir de ese momento, los dispositivos que tienen datos para transmitir envían un mensaje de solicitud de transmisión (REQ-T) de forma aleatoria, pero en función del tamaño de su propia ventana de contienda. Si dos o más dispositivos envían un REQ-T simultáneamente, se producirá una colisión. Los dispositivos que no tienen paquetes para transmitir entrarán en un estado de suspensión para ahorrar energía. Para finalizar el periodo CSMA/CA, el AP transmite un mensaje de retroalimentación general (OACK) a todos los dispositivos para informar cuáles mensajes REQ-T tuvieron éxito, en lugar de enviar mensajes de reconocimiento individuales cada vez que recibe con éxito un mensaje REQ-T. Además, el mensaje OACK contiene el orden de transmisión de manera que a cada dispositivo con un REQ-T exitoso se le asigna una ranura de tiempo específico para la transmisión de datos. Durante el periodo TDMA, la transmisión de paquetes de datos se realiza dependiendo del orden establecido en el periodo CSMA/CA. En caso de una transmisión exitosa de un paquete de datos, el AP retorna un mensaje ACK. En caso contrario, este mensaje no se envía. De esta manera, el dispositivo cuyo paquete no fue entregado envía un mensaje de intención de retransmisión con el cual el AP le asignará una nueva ranura. Los resultados reportados en [39] muestran que el uso de un único OACK conduce a retardos más cortos cuando se compara con el ACK convencional usado en la mayor parte de la literatura. Además, concluyen que su propuesta logra un mejor desempeño que los protocolos HyMAC [40] y CPMAC [41]. A pesar de estos resultados prometedores, y probablemente debido a su aplicación prevista, la propuesta fue probada con un máximo de 15 dispositivos intentando transmitir simultáneamente. Por lo tanto, sus propiedades de escalamiento siguen siendo desconocidas.

En el protocolo MAC híbrido propuesto por Fan *et al*. en [42],cada *superframe* se compone de cinco periodos: Periodo de Beacon (*Beacon Period* o BP), Periodo de Contención (*Contention Period* o CP), Periodo de Asignación (*Assignment Period* o AP), Periodo de Energía (*Energy Period* o EP) y Periodo Variable de Subida (*Variable Upload Period* o vUP). Durante el BP, el nodo coordinador transmite un paquete de sincronización que contiene parámetros esenciales como la duración y el número total de miniranuras, así como la longitud máxima de los paquetes de datos. En el CP, los dispositivos que tienen datos para transmitir dividen su carga útil en paquetes más pequeños (si es necesario) y compiten por el acceso al canal utilizando un algoritmo basado en CSMA/CA. Cada dispositivo puede transmitir un solo paquete que incluye información sobre cuántos paquetes le quedan por enviar.

Luego, durante el AP, el nodo coordinador transmite una lista de asignación con los *slots* reservados para cada dispositivo que logró transmitir correctamente durante el CP. Esta asignación está basada en el número restante de paquetes indicados por cada dispositivo. En el EP, el nodo coordinador envía paquetes de energía utilizando tecnología de transferencia por radiofrecuencia para recargar las baterías de los dispositivos, mismos que pueden cosechar esta energía en estado de reposo. Finalmente, durante el vUP, los dispositivos transmiten sus paquetes restantes utilizando los *slots* asignados. Fan *et al*. informan que su propuesta, llamada PAH-MAC, muestra ventajas significativas en comparación con protocolos como CSMA/CA y AEE-MAC [43]. Cuando el tamaño de los paquetes de datos es pequeño, se comporta como un protocolo de contienda, mientras que, al encontrarse con paquetes de tamaño grande, se asemeja a un esquema basado en asignación de ranuras, logrando una mayor tasa de transferencia, menor tasa de colisiones y menor consumo de energía por byte transmitido.

La Tabla 2.3 resume las características más importantes de los trabajos descritos anteriormente. Podemos observar que, a excepción de Yang et al. [33], el resto de las propuestas utilizan, en sus respectivas etapas de contienda, variantes de los protocolos CSMA/CA y ALOHA, que exhiben un desempeño adecuado cuando el número de usuarios contendientes simultáneos es bajo y la carga de tráfico general es baja. Sin embargo, se espera que sufran de congestión a medida que la carga de tráfico y el número de dispositivos aumentan [23]. Incluso en el trabajo de Yang et al., donde la longitud del período de contienda es óptima para cada *frame*, la selección aleatoria de ranuras por parte de los dispositivos contendientes podría llevar a un alto número de colisiones en un escenario de acceso múltiple.

A pesar de sus ventajas, los protocolos MAC híbridos no están exentos de desafíos técnicos, ya que también heredan las limitaciones de sus protocolos constituyentes. Un problema que surge en algunos protocolos MAC basados en contienda (por ejemplo, la implementación DCF), independientemente de si se utilizan de forma aislada o como parte de un protocolo híbrido, es un cierto grado de injusticia en la forma en que se otorga acceso a los nodos de la red. En condiciones de tráfico intenso, un nodo puede acumular una serie de intentos de transmisión fallidos (es decir, colisiones) antes de que su

Tabla 2.3. Resumen comparativo de protocolos MAC híbridos utilizados en MTC

Año	Autores	Mecanismo de Reserva	Mecanismo de Acceso	No. Dispositivos	Tasa de datos	Tamaño de Paquete
2013	Y. Liu *et al*.	CSMA *p*-persistente	TDMA	1 – 500	1.7 Gbps	1,728 bytes
2014	K. Verma *et al*.	CSMA	DCF dentro de ranuras TDMA	1 – 500	1.5 Gbps	1,500 bytes
2017	E. Hegazy *et al*.	CSMA No Persistente	TDMA	100, 200, 500	ND*	ND*
2018	B. Yang *et al*.	Reservación aleatoria de ranuras	TDMA	1 – 300	11 Mbps	1000 bytes
2018	W. Saad *et al*.	s-ALOHA	TDMA	1 – 1200	256 kbps	128 bits
2020	A. Lachtar *et al*.	CSMA *p*-persistente	TDMA	500, 800, 1200	1.7 Gbps	1728 bytes
2020	D. Olatinwo *et al*.	s-ALOHA	TDMA	1000	256 kbps	128 bits
2022	D. Olatinwo *et al*.	CSMA	TDMA	3 - 15	5 Mbps	ND*
2024	X. Fan *et al*.	CSMA/CA	TDMA (si es necesario)	1-100	256 kbps	50 - 500 bytes

*ND - No disponible

paquete se transmita exitosamente. Al mismo tiempo, otro nodo que, súbitamente, se une a la contienda, puede transmitir con éxito su paquete en su primer intento. Este efecto no es deseable en muchas aplicaciones, especialmente en las que requieren la entrega de datos de alta prioridad. Una alternativa, que proporciona un acceso justo al canal y un bajo retraso de acceso cuando ocurren colisiones en el medio, es la implementación de algoritmos relacionados con el algoritmo de árbol (ver [44] y [23]). Un ejemplo de esto es *Distributed Queuing* (DQ), que se introdujo en [45] y se presentó una variante en [5]. Otra variante del algoritmo de árbol, que se conoce como el algoritmo de 2 celdas (2C) [46], aporta cierto nivel de equidad con la siguiente regla. Cuando ocurre una colisión, no se permite que nuevos nodos se unan a la contienda hasta que todos los involucrados en la colisión anterior transmitan con éxito.

En contraste con los trabajos descritos previamente, en este trabajo de tesis se presenta un protocolo MAC híbrido que tiene como base una variante del algoritmo 2C durante la etapa de contienda. Dicha estrategia se describe detalladamente en el capítulo 3.

CAPÍTULO 3

Propuesta del protocolo 2CA-R^2

En la revisión presentada en la sección 2.4, se puede observar que un amplio número de protocolos de control de acceso al medio empleados en los estándares para comunicaciones M2M, se basan en protocolos de contienda, específicamente alguno de los descritos en la sección 2.1, en protocolos de reserva, detallados en la sección 2.2, o en una combinación de ambos tipos. Asimismo, en la sección 2.5 podemos apreciar que los protocolos híbridos analizados están conformados por alguno de los protocolos de contienda mencionados en la sección 2.1, en conjunto con TDMA como protocolo de reserva. Es importante destacar que, en cualquiera de los dos casos, el uso de protocolos de contienda se ha limitado a variantes de los protocolos ALOHA y CSMA/CA, heredando las limitaciones técnicas inherentes a estas soluciones. Este hecho incluye a la implementación DCF, misma que se utiliza como mecanismo de acceso al medio por defecto en el estándar IEEE 802.11, el cual se encuentra presente en casi cualquier red inalámbrica de área local en el mundo, tal como se menciona en [17] y [47]. Sin embargo, no se debe perder de vista que tanto el protocolo ALOHA como el protocolo CSMA/CA, así como sus variantes, fueron diseñados para HTC, por lo cual no es de extrañar que sufran un deterioro importante en su rendimiento cuando se les somete a tráfico intenso de tipo M2M. Esto se refleja en la degradación que se genera en medidas de desempeño como *throughput* y retardo de acceso conforme aumenta el número de dispositivos contendientes, situación que además incrementa las condiciones de inequidad y limita de forma importante la escalabilidad de la red. Por esta razón, es importante explorar soluciones, en cuanto a protocolos de control de acceso al medio, que se basen en enfoques distintos a los ya utilizados, considerando las características del tráfico M2M, en busca de mejorar el rendimiento y optimizar el proceso de contienda de los dispositivos que buscan transmitir sus paquetes de datos.

La propuesta de un protocolo MAC híbrido para comunicaciones M2M, que se presenta en este capítulo, tiene sus bases en una versión adaptable del algoritmo 2C, el cual se describe detalladamente a continuación, previa exposición de los antecedentes correspondientes.

3.1. Antecedentes

3.1.1. El algoritmo 2C

El algoritmo 2C fue introducido por Paterakis et al. en [46], con el objetivo de resolver colisiones mediante acceso aleatorio. En este, se considera que el tiempo está dividido en ranuras y que los dispositivos pueden transmitir solo al comienzo de una ranura de tiempo. Una ranura equivale al intervalo de tiempo que toma transmitir un paquete por parte de un dispositivo y recibir un mensaje de retroalimentación por parte de una estación central. El mensaje de retroalimentación es binario, es decir, es un mensaje C (colisión) cuando se detecta una colisión en la estación central y un mensaje NC (no colisión) en caso contrario. Si solo transmite un dispositivo, el paquete correspondiente se transmitirá con éxito. Por otro lado, si hay varios intentos de transmisión en la misma ranura, se producirá una colisión y su resolución comenzará en la siguiente ranura. El procedimiento de resolución de colisiones finalizará hasta que todos los dispositivos involucrados en la colisión inicial transmitan con éxito sus paquetes. El intervalo de tiempo transcurrido desde la colisión inicial hasta el final de la resolución de la colisión se conoce como Intervalo de Resolución de Colisiones (*Collision Resolution Interval* o CRI). Un dispositivo que genera una nueva solicitud de transmisión, cuando un CRI está en curso, tiene que esperar hasta que finalice el CRI actual. De esta manera, el algoritmo 2C proporciona algunas garantías de acceso justo al canal, ya que no se atenderá a ningún dispositivo recién llegado antes que a los que ya estén en disputa.

Cada dispositivo que participa en un CRI mantiene una variable de estado que controla su acceso al canal alternando entre transmisión ($\mathrm{T_x}$) y espera (W). Denotemos por r_t el valor de esta variable al comienzo de una ranura arbitraria t. Un dispositivo puede intentar acceder al canal en la ranura t solo cuando r_t es igual a $\mathrm{T_x}$. Sea f_t el mensaje de retroalimentación correspondiente a la transmisión en la ranura t y recibido al final de la misma ranura. Si la transmisión resultó en una colisión, entonces $f_t = \mathrm{C}$; de lo contrario, $f_t = \mathrm{NC}$.

Supongamos que, hasta la ranura actual, se han transmitido todos los paquetes (es decir, no hay ningún CRI en curso). Los dispositivos con nuevos paquetes por transmitir establecerán sus variables de estado en el estado $\mathrm{T_x}$ e intentarán acceder al canal en la siguiente ranura. Dependiendo de la secuencia de mensajes de retroalimentación recibidos desde la estación central, cada dispositivo actualiza su variable de acuerdo con la ecuación (3.1).

$$
\begin{aligned}
&1) \quad \text{Si } r_t = \mathrm{W} \text{ y } f_t = \mathrm{NC}, \text{ entonces } r_{t+1} = \mathrm{Tx} \\
&2) \quad \text{Si } r_t = \mathrm{W} \text{ y } f_t = \mathrm{C}, \quad \text{entonces } r_{t+1} = \mathrm{W} \\
&3) \quad \text{Si } r_t = \mathrm{Tx} \text{ y } f_t = \mathrm{C}, \quad r_{t+1} = \begin{cases} \mathrm{Tx} \text{ con probabilidad } & p = 1/2 \\ \mathrm{W} \text{ con probabilidad } & 1-p = 1/2 \end{cases}
\end{aligned} \tag{3.1}
$$

En la siguiente ranura, cada dispositivo se comportará conforme al valor actualizado de su variable de estado. Esto se repite hasta que todos los dispositivos hayan podido transmitir sus paquetes. Nótese que a medida que ocurren colisiones, cada vez más dispositivos abandonan el estado de transmisión y se unen al grupo de dispositivos en espera. Este procedimiento continúa hasta que solo quede un dispositivo en el estado de transmisión y logre una transmisión exitosa. En caso de que ocurra una transmisión exitosa, o todos los dispositivos abandonen el estado de transmisión, la estación central transmite un mensaje de retroalimentación NC. Nótese también que, después de un mensaje de retroalimentación NC, todos los dispositivos con $r_t = \mathrm{W}$ cambiarán su estado a Tx e intentarán transmitir en la siguiente ranura, lo que provocará una nueva colisión. Por lo tanto, solo pueden ocurrir dos mensajes de retroalimentación NC consecutivos al final de un CRI. Este algoritmo se llama 2C (*two cells*) porque los dispositivos en contienda pueden estar transmitiendo o esperando, y esta situación se puede representar utilizando dos celdas (i.e. dos grupos). Una celda contiene el grupo de dispositivos en estado de transmisión (es decir, $r_t = \mathrm{Tx}$) y la segunda celda contiene los que se encuentran en espera (es decir, $r_t = \mathrm{W}$).

Para ilustrar el funcionamiento del algoritmo 2C, considérese, como ejemplo, la red que se muestra en la Figura 3.1, que consta de siete dispositivos (es decir, A-G). En un momento determinado, cuatro de ellos (es decir, A, B, C y D) tienen paquetes que transmitir, por lo que esperan al siguiente CRI para competir.

Figura 3.1. Ejemplo de una red inalámbrica

En la Figura 3.2, se representa una posible secuencia de eventos, misma que se explica a continuación:

- Primera ranura. Al comienzo de la primera ranura, los cuatro dispositivos establecen sus variables de estado en Tx (es decir, $r_1 = \mathrm{Tx}$) y transmiten. Como consecuencia, se produce una colisión y la estación central devuelve el mensaje $f_1 = \mathrm{C}$.
- Segunda ranura. En respuesta al mensaje de retroalimentación por parte de la estación central, los dispositivos en disputa actualizan sus variables de estado siguiendo las reglas mencionadas anteriormente (ecuación (3.1)). Es decir, permanecen en la celda de transmisión con probabilidad $p = 1/2$ o se unen al grupo en la celda de espera (también con probabilidad $1/2$). En el ejemplo, los dispositivos A y C eligen la primera opción (es decir, $r_2 = \mathrm{Tx}$), en cambio, B y D eligen la segunda (es decir, $r_2 = \mathrm{W}$). Como ahora dos dispositivos transmiten, se produce otra colisión y la estación central devuelve el mensaje $f_2 = \mathrm{C}$.
- Tercera ranura. Los dispositivos A y C reaccionan al mensaje de retroalimentación y deciden aleatoriamente si continúan transmitiendo (con probabilidad p = 1/2) o no. En el ejemplo, el dispositivo A pasa a la celda de espera (es decir, $r_3 = \mathrm{W}$). Por el contrario, el dispositivo C permanece en la celda de transmisión, por lo tanto, $r_3 = \mathrm{Tx}$. Como en esta ranura solo transmite el dispositivo C, se produce una transmisión exitosa y la estación central devuelve la $f_3 = \mathrm{NC}$. Es importante destacar que este evento señala el final de un intervalo de tiempo que comenzó con una colisión de multiplicidad 4 en la primera ranura y finalizó cuando se logró una transmisión exitosa. Este es el concepto de lo que se considera una fase en esta propuesta y que difiere del de otros trabajos enfocados en protocolos MAC híbridos. Una fase comienza con la colisión inicial de un CRI, o bien, con la colisión que sigue después de una transmisión exitosa. Para fines de identificación, consideremos que la fase N comienza con una colisión de multiplicidad N. En todos los casos, una fase termina cuando hay una ranura que contiene una transmisión exitosa. En este ejemplo, la fase 4 comienza en la ranura 1 y termina en la tercera ranura.
- Cuarta ranura. Cuando los dispositivos en la celda de espera (es decir, A, B y D) reciben la retroalimentación $f_3 = \mathrm{NC}$, cambian su estado de espera a transmisión ($r_4 = \mathrm{Tx}$). Por lo tanto, cuando transmiten se produce una nueva colisión y la estación central devuelve el mensaje $f_4 = \mathrm{C}$. Este evento inicia la fase 3.

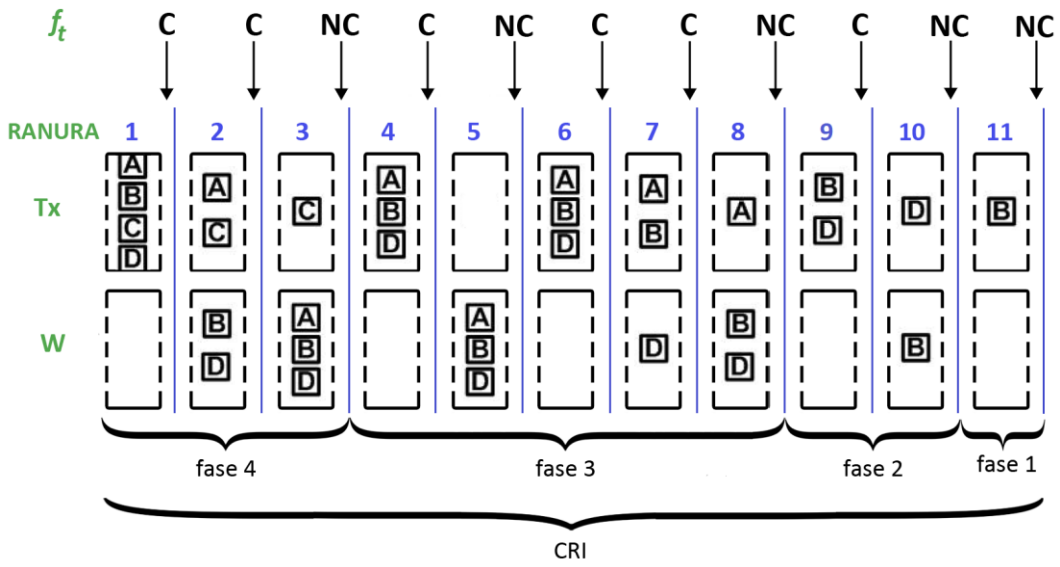

Figura 3.2. Ejemplo de operación del algoritmo 2C

- Quinta ranura. En respuesta al mensaje de retroalimentación, los dispositivos en la celda de transmisión actualizan sus variables de estado de acuerdo con la ecuación (3.1). Es decir, permanecen en la celda de transmisión o se unen a la celda de espera, ambos casos con probabilidad $1/2$. En el ejemplo, los tres dispositivos A, B y D eligen pasar a la celda de espera (es decir, $r_5 = \mathrm{W}$). Como consecuencia, se produce una ranura vacía y la estación central devuelve el mensaje $f_5 = \mathrm{NC}$.
- Ranuras restantes. Al recibir la retroalimentación, los dispositivos A, B y D, que estaban en la celda de espera, cambian su estado a transmisión, es decir, $r_6 = \mathrm{Tx}$. Cuando transmiten, nuevamente se produce una colisión y la estación central devuelve el mensaje $f_6 = \mathrm{C}$. A partir de este punto, el procedimiento anteriormente descrito se repite en cada una de las siguientes fases hasta que el último dispositivo involucrado en la colisión inicial del CRI transmite con éxito. Nótese que las últimas transmisiones derivan en dos mensajes de retroalimentación NC consecutivos, lo que señala el final del CRI.

3.1.2. El algoritmo 2C Adaptable

En el algoritmo 2C original, cuando el número de dispositivos en contienda es grande, la longitud del CRI resultante se vuelve extremadamente larga, lo que puede considerarse como un inconveniente importante. Esto se debe a que la probabilidad de permanecer en el estado de transmisión, después de una colisión, es un valor fijo (es decir, la probabilidad $p = 1/2$), independientemente del número de dispositivos contendientes. El algoritmo 2C Adaptable o *Adaptive*-2C, que se presenta en [48], propone que p debe ser una función del número de dispositivos en contienda N (es decir, $p = p(N)$). Por ejemplo, cuando N es grande, p debe ser pequeño de modo que la mayoría de estos dispositivos pasen a la celda de espera en un solo paso del algoritmo. Sin embargo, no se debe hacer demasiado pequeño o el sistema podría transitar a un estado donde ninguno, de los N dispositivos activos, permanezca en transmisión, lo que conduce a ranuras vacías y desperdicio de recursos. En el caso óptimo, después de una colisión de multiplicidad N, solo un dispositivo debe permanecer en transmisión. Para ello, en [48] se dedujeron algunas alternativas para lograr este objetivo, encontrando la forma óptima de calcular $p(N)$, misma que se muestra en la ecuación (3.2).

$$p = p(N) = \begin{cases} \frac{1}{2} & si\ \ N = 2 \\ \frac{\sqrt{2N(N-1)}-2}{(N-2)(N+1)} & si\ N > 2. \end{cases} \tag{3.2}$$

De esta manera, el algoritmo *Adaptive*-2C logra una resolución de colisiones más rápida que el algoritmo 2C. Para poder utilizar la ecuación (3.2), la estación central debe estimar N y este valor debe ser comunicado a todos los dispositivos de la red en los puntos de tiempo apropiados. En [48] se propusieron dos estrategias para realizar esta tarea, llamadas «actualización por ranura» y «actualización por fase».

3.1.2.1. Estrategia de actualización por ranura

En la estrategia de actualización por ranura, el valor de p varía en función de la última multiplicidad de colisión. Esta estrategia necesita que la estación central estime e informe, después de cada ranura de transmisión, el número de dispositivos que colisionaron. Por lo tanto, la retroalimentación ya no es binaria. La Figura 3.3 muestra un ejemplo del algoritmo *Adaptive*-2C con la política de actualización por ranura, que es equivalente a la que se mostró anteriormente en la Figura 3.2 para el algoritmo 2C. Nótese que, al final de cada ranura en la Figura 3.3, el mensaje de retroalimentación f_t indica el

número de dispositivos en contienda (en caso de colisión) o NC (en caso de no colisión). Como resultado, el número de ranuras en el CRI se puede reducir, en comparación con el algoritmo 2C.

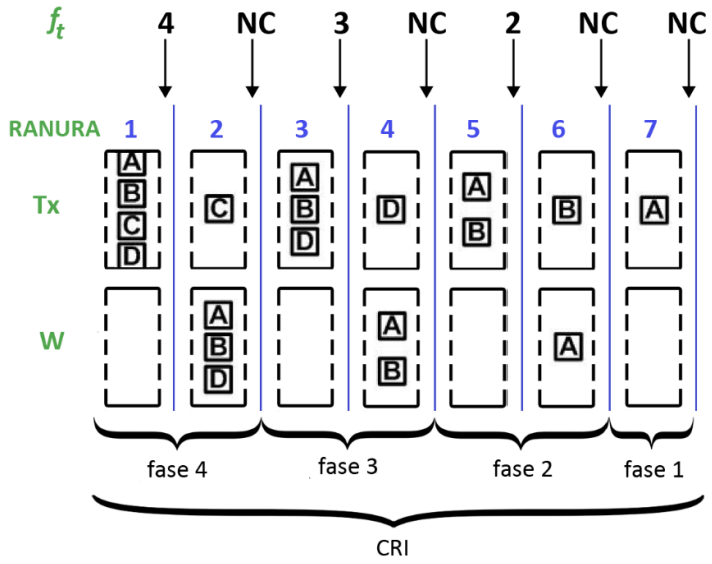

Figura 3.3. Ejemplo de operación del algoritmo *Adaptive*-2C con actualización por ranura

3.1.2.2. Estrategia de actualización por fase

Como ya se mencionó, la estrategia de actualización por ranura requiere que la estación central estime el número de dispositivos colisionados en cada ranura de transmisión. El problema con este enfoque es que esta información puede ser muy difícil de obtener en condiciones reales. Sin embargo, nótese que incluso si no es posible saber cuántos dispositivos colisionaron al comienzo de un CRI, esta información puede conocerse *a posteriori*, cuando dicho CRI termina. Esto se debe a que el número de dispositivos involucrados en la primera colisión de un CRI es igual al número total de transmisiones exitosas observadas durante dicho CRI. De esta manera, la estación central puede realizar un seguimiento de cuántos dispositivos compitieron en los últimos CRI y puede usar esta información para estimar el número de dispositivos contendientes en el siguiente CRI. Para ello, se pueden aplicar varios métodos de estimación, como la predicción lineal.

En la estrategia de actualización por fase, cuando ocurre la primera colisión de un CRI, la estación central aplica un método de predicción para estimar la multiplicidad de dicha colisión (la cual se denota como $\widehat{N}$) y proporciona este valor como mensaje de retroalimentación a la red. Esta información es utilizada por los dispositivos de la red para calcular el valor apropiado de p (mediante el uso de la ecuación (3.2)), es decir, $p = p(\widehat{N})$. Si se observa una segunda colisión, los dispositivos fijarán el valor

de p en un valor fijo de $1/2$ y el sistema evolucionará de acuerdo con las reglas del algoritmo 2C hasta que se logre una transmisión exitosa. Es importante recordar que dicho éxito es notificado por la estación central mediante la difusión de un mensaje de retroalimentación NC y que este evento marca el final de la fase N y el comienzo de la fase $(N - 1)$[2]. Este procedimiento básico se repite en cada una de las siguientes fases. Es decir, cuando ocurre la primera colisión de una fase, los dispositivos de la red utilizan una estimación de la multiplicidad de colisiones para calcular el valor apropiado de p (es decir, $p = p(\widehat{N})$). Esta probabilidad determina si continúan o no en el estado de transmisión. Si ocurre otra colisión, los dispositivos utilizarán el valor fijo $p = 1/2$ hasta que se logre la siguiente transmisión exitosa. Vale la pena destacar que la estación central tiene que comunicar una estimación de la multiplicidad de colisiones solo una vez (es decir, al comienzo de un CRI), ya que cada vez que hay una transmisión exitosa, esta estimación se puede actualizar localmente en los dispositivos restando 1 de su valor actual.

El procedimiento descrito anteriormente constituye la base de la estrategia de actualización por fase, sin embargo, conviene hacer un par de comentarios:

- Supongamos que hay un CRI en curso. Dado que el cambio del estado de transmisión al de espera (es decir, el paso de la celda de transmisión a la de espera) está controlado por una probabilidad, puede producirse una ranura vacía, ya que todos los dispositivos en contienda pueden haber decidido pasar al estado de espera. Como se detallará en el próximo ejemplo, es necesario distinguir entre una ranura vacía y una transmisión exitosa. Por lo tanto, en lugar de utilizar un mensaje NC para ambos casos, cuando ocurra una ranura vacía, la estación central transmitirá un mensaje E (vacío) forzando a todos los dispositivos a volver a la celda de transmisión y produciendo una colisión como consecuencia. Dado que todos los dispositivos activos estarán involucrados en dicha colisión, se producirá una situación idéntica a la del comienzo de la fase actual y se tratará como tal. Esto es, después de dicha colisión, utilizarán el valor adaptativo de p (es decir, $p = p(\widehat{N})$ de la fase en curso calculado mediante la ecuación (3.2)) y después utilizarán $p = 1/2$, si es necesario.

[2] Nótese que la variable N se utiliza para denotar la fase y también para denotar el número de dispositivos en contienda ya que, idealmente, este valor debe coincidir en ambos casos.

- En $t = 0$, es decir, al inicio del sistema no hay suficiente historial para calcular una estimación del número de dispositivos en contienda al comienzo del primer CRI. Por lo tanto, dependiendo del orden del predictor, se debe completar un cierto número de CRI con el fin de generar datos suficientes para utilizar en la predicción. Mientras tanto, el sistema debe evolucionar utilizando las reglas del algoritmo 2C estándar (es decir, $p = 1/2$).

La Figura 3.4 muestra una posible evolución de un CRI donde el acceso al canal está controlado por el algoritmo *Adaptive*-2C con la estrategia de actualización por fase. Este caso es similar a los descritos anteriormente, es decir, al inicio de un CRI cuatro de siete dispositivos contienden por el medio de transmisión (ver la topología de red mostrada en la Figura 3.1). Supongamos que el sistema ha estado en operación durante un tiempo para que un algoritmo de predicción tenga datos suficientes para operar. La siguiente descripción ilustra varios casos de interés.

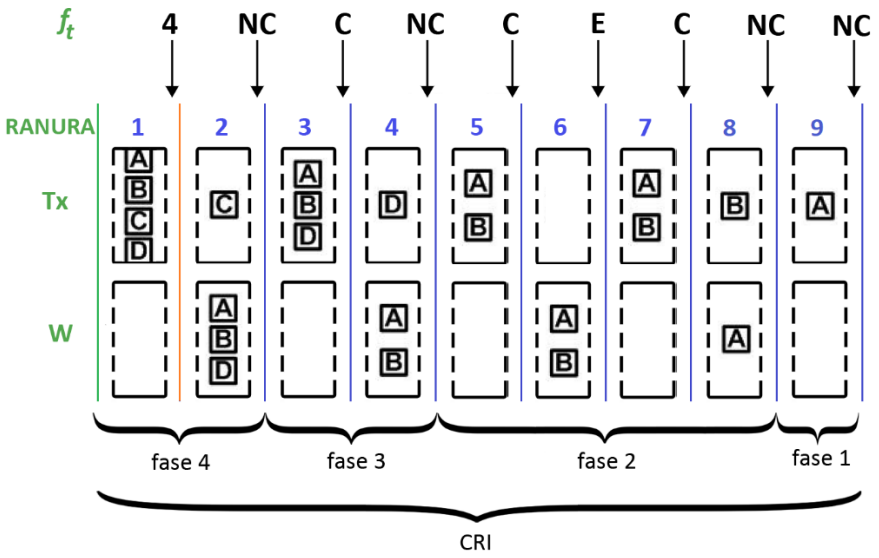

Figura 3.4. Ejemplo de operación del algoritmo *Adaptive*-2C con actualización por fase

- Primera ranura. En el ejemplo, los dispositivos A, B, C y D transmiten simultáneamente y, mediante un método de predicción, la estación central calcula el número de dispositivos que colisionaron (es decir, $\widehat{N}$). Este valor se devuelve como mensaje de retroalimentación al final de la primera ranura. De forma ideal, $\widehat{N}$ será igual a 4 para este caso.
- Segunda ranura. Con el mensaje de retroalimentación recibido, los dispositivos en contienda calculan $p = p(\widehat{N})$. De acuerdo con esta probabilidad, sólo uno de ellos (es decir, C) permanece en transmisión, mientras que el resto se mueve a la celda de espera. Por lo tanto, el dispositivo C transmite con éxito y la estación central devuelve un mensaje NC al final de esta ranura. Este evento marca el final de la fase 4 (nótese que comenzó con una colisión de multiplicidad 4 y terminó con una transmisión exitosa). Se puede ver que al final de esta ranura, el número estimado de dispositivos que regresarán a contienda se puede actualizar como $\widehat{N} - 1$.
- Tercera ranura. Todos los dispositivos en la celda de espera se mueven a la celda de transmisión, de modo que se observa una colisión en el medio y la estación central devuelve un mensaje C.
- Cuarta ranura. Cada dispositivo utiliza su estimación local del número de dispositivos que permanecen en contienda (es decir, $\widehat{N} - 1$) y calcula la probabilidad de continuar en el estado de transmisión (es decir, $p = p(\widehat{N} - 1)$). En el ejemplo, solo uno de los dispositivos (denotado por D) permanece en transmisión y el resto se mueve a la celda de espera. Por lo tanto, se logra una segunda transmisión exitosa y la estación central devuelve un mensaje de retroalimentación NC. Este evento marca el final de la fase 3. Al final de esta ranura, el número estimado de dispositivos en contienda se convierte en $\widehat{N} - 2$.
- Quinta ranura. Los dispositivos en la celda de espera (es decir, A y B) se mueven a la celda de transmisión, lo que provoca otra colisión. La estación central devuelve un mensaje C.
- Sexta ranura. Al utilizar $p = p(\widehat{N} - 2)$, los dispositivos deciden si continúan transmitiendo o no. Por casualidad, todos deciden moverse a la celda de espera, por lo que se produce una ranura vacía. Este evento lo informa la estación central al enviar un mensaje de retroalimentación E.

- Séptima ranura. De acuerdo con las reglas del algoritmo *Adaptive*-2C, después de una ranura vacía, todos los dispositivos en la celda de espera cambian su estado y transmiten. Al producirse otra colisión, la estación central devuelve un mensaje C.
- Octava ranura. Nótese que la colisión contenida en la séptima ranura involucró a todos los dispositivos activos. Sin embargo, tal colisión se produjo después de una ranura vacía, por lo que la estimación del número de dispositivos en contienda no debe actualizarse de la misma forma que en el caso de una transmisión exitosa. En cambio, los dispositivos utilizan el último valor conocido de p (es decir, $p = p(\hat{N} - 2)$) para determinar si continúan transmitiendo o no. En el ejemplo, el dispositivo B continúa transmitiendo y A pasa a la celda de espera, derivando en una transmisión exitosa.
- Novena ranura. El proceso continúa según las reglas del algoritmo y el último dispositivo en contienda transmite exitosamente su paquete de datos. Cuando la estación central envía la retroalimentación correspondiente, el CRI finaliza. Nótese que este evento se señala con dos mensajes NC consecutivos, como también ocurre en el caso del algoritmo 2C estándar.

Como se puede apreciar, durante la evolución de este CRI, hubo transmisiones exitosas en las ranuras 2, 4, 8 y 9. Por lo tanto, al final del CRI, se sabe que la colisión inicial (en la primera ranura) tuvo una multiplicidad de 4. Esta información ayuda a la estación central a estimar la multiplicidad de colisión inicial en los siguientes CRI.

Con el fin de brindar una comprensión más profunda, en la Figura 3.5 se muestra el diagrama de tiempo de un CRI correspondiente a la estrategia de actualización por fase del algoritmo *Adaptive*-2C en una red que consta de dos dispositivos, denotados como «Disp. A» y «Disp. B», así como de una estación central. En este diagrama se ilustran los diferentes eventos que pueden producirse en un CRI, así como los cambios en la probabilidad de transmisión que se producen en consecuencia de estos. La ranura 1, o *slot* 1, contiene la primera colisión. El *slot* 2 corresponde a un caso en el que todas las estaciones abandonan el estado de transmisión y se produce una ranura vacía. El *slot* 3 contiene una colisión después de una ranura vacía. El *slot* 4 contiene una colisión que no es ni la primera colisión del CRI ni una colisión después de una ranura vacía. Por último, las ranuras 5 y 6 contienen las dos transmisiones exitosas.

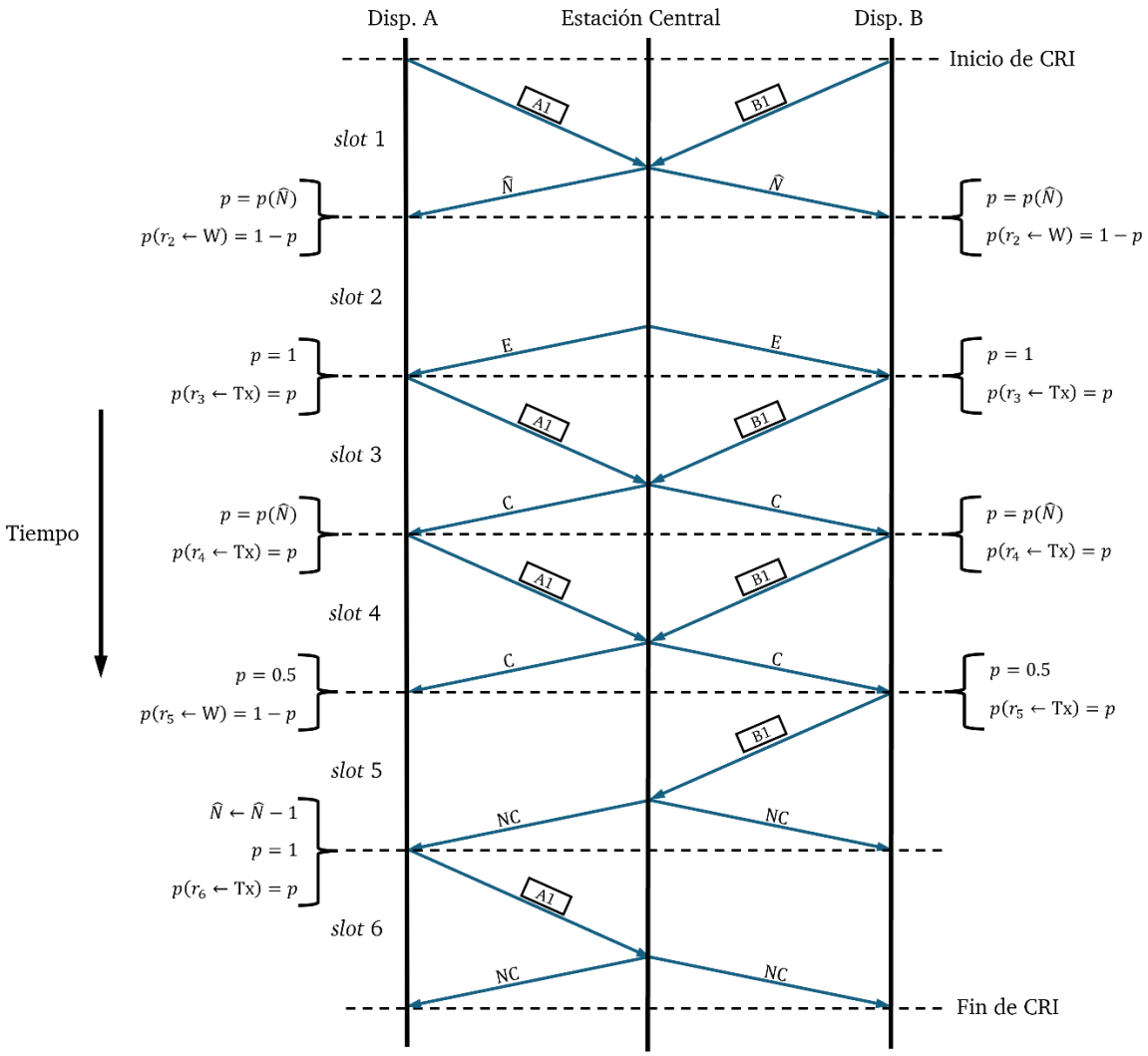

Figura 3.5. Diagrama de tiempo del algoritmo *Adaptive*-2C con actualización por fase

3.2. Propuesta del protocolo MAC híbrido 2CA-R^2

En este trabajo, se propone un protocolo MAC híbrido para comunicaciones M2M llamado 2C Adaptable con Reservación de Recursos o 2CA-R^2. En este protocolo, la base del intervalo de contienda es el algoritmo *Adaptive*-2C con estrategia de actualización por fase. Es importante destacar que, en justo reconocimiento a un trabajo previo, el protocolo que aquí se presenta tuvo como inspiración las ideas del trabajo de P. Hernández, reportado en [49]. En este trabajo se propuso, a manera de una investigación inicial exploratoria, un protocolo MAC híbrido en donde se usó el algoritmo 2C estándar como mecanismo de contienda. Sin embargo, las principales diferencias con aquel trabajo radican en que, en la presente propuesta, se incorporó la versión adaptable del algoritmo 2C en la etapa de

contienda, además de realizar la evaluación de desempeño en condiciones más realistas, bajo el contexto de las comunicaciones M2M y en comparación con la implementación DCF del protocolo CSMA/CA. Adicionalmente, la validación del protocolo 2CA-R^2 se realizó mediante un modelo matemático desarrollado considerando las características específicas del algoritmo empleado en la etapa de contienda.

Como protocolo MAC híbrido, en 2CA-R^2 hay dos etapas operativas, el Intervalo de Resolución de Colisiones o CRI y el Intervalo de Transmisión de Datos o DTI. En el CRI, en donde las ranuras o *slots* ahora se denominan «*minislots*», todos los dispositivos activos (es decir, dispositivos con paquetes para transmitir) contienden para lograr la reserva de un turno. Cuando todos los dispositivos que participaron en la colisión inicial obtienen la reserva de su turno, el CRI finaliza. Estas reservas se utilizarán durante el siguiente DTI. En el DTI, el orden de los turnos en el que los dispositivos transmitirán su paquete de datos corresponde al mismo orden en el que lograron su reserva en el CRI.

Vale la pena enfatizar que estas características se respaldan naturalmente ya que los mensajes de retroalimentación proporcionados por la estación central permiten que todos los dispositivos conozcan el orden en el que reservaron el canal con éxito. Cada dispositivo activo debe monitorizar el medio de transmisión desde el inicio del CRI hasta que transmite su paquete. Los dispositivos que se activaron después del inicio de un CRI deben esperar hasta que la estación central indique el inicio de un nuevo CRI. A continuación, se describen con más detalle ambas etapas del protocolo 2CA-R^2.

3.2.1. Intervalo de Resolución de Colisiones (CRI)

Considérese una red con una estación central o estación base y varios dispositivos cuyo acceso al canal está controlado por el protocolo 2CA-R^2. El inicio de un CRI es señalado por la estación base con un mensaje de inicio de reserva o RB. Posteriormente, la estación base se encarga de examinar todos los *minislots* y proporcionar la retroalimentación adecuada en función del evento detectado. Esta retroalimentación puede ser uno de los siguientes cuatro mensajes: $\widehat{N}$, (valor de estimado de N), C o colisión, E o vacío y NC o no colisión. Se denota como f_t el mensaje de retroalimentación correspondiente al *minislot* t.

Durante un CRI, cada dispositivo mantiene una variable de estado r_t que puede tomar los valores Tx y W y se actualiza de acuerdo con los mensajes de retroalimentación.

Cada dispositivo también utiliza una variable s para llevar un registro de su ranura de transmisión reservada y la variable N para mantener el número estimado de dispositivos en contienda. Cuando un dispositivo tiene un paquete de datos para transmitir, establece su variable de estado r_t en Tx, lo que significa que debe esperar al siguiente CRI. También establece s en 0 y N en 1. Un dispositivo con r_t establecido en Tx enviará un paquete de Solicitud de Reservación o RREQ en el primer *minislot* del siguiente CRI. El RREQ tendrá éxito solo si $r_t = \mathrm{Tx}$ y $f_t = \mathrm{NC}$. La longitud del CRI será de un *minislot* solo si un dispositivo envía un RREQ (es decir, acceso aleatorio inmediato). Sin embargo, si dos o más dispositivos transmiten, se producirá una colisión y el conflicto tendrá que resolverse como lo indica el algoritmo *Adaptive*-2C con estrategia de actualización por fase (es decir, se aplica el acceso controlado). El propósito del CRI es asignar una ranura de transmisión a cada dispositivo en contienda. El CRI evoluciona de acuerdo con las siguientes reglas (tenga en cuenta que el protocolo utiliza cuatro mensajes de retroalimentación en lugar de la retroalimentación binaria utilizada por el algoritmo 2C estándar).

- Si ($r_t = \mathrm{Tx}$ y $f_t = \mathrm{NC}$) entonces $s = s + 1$. Esta regla corresponde a un caso en el que un dispositivo transmitió una RREQ que fue exitosa. Como consecuencia, el dispositivo aumenta su posición de reserva y abandona la contienda. Después de este punto, el dispositivo no modificará más el valor de la variable s durante el resto del CRI, ya que esta indica su posición en la trama de transmisión correspondiente.

- Si ($r_t = \mathrm{Tx}$ y $f_t = \widehat{N}$) entonces $N \leftarrow \widehat{N}$. Esta regla se aplica solo cuando, en el *minislot* t, ocurre la primera colisión del CRI. En este caso, la estación base devuelve el valor estimado de la multiplicidad de colisión ($\widehat{N}$), el cual se copia en la variable local N que se utiliza para calcular la probabilidad $p = p(N)$, de acuerdo con la ecuación (3.2). Como consecuencia, el dispositivo mantiene $r_{t+1} = \mathrm{Tx}$ (es decir, permanece en la celda de transmisión) con probabilidad p o cambia a $r_{t+1} \leftarrow \mathrm{W}$ (es decir, pasa a la celda de espera) con probabilidad $1 - p$.

- Si ($r_t = \mathrm{W}$ y $f_t = \mathrm{E}$) entonces $r_{t+1} \leftarrow \mathrm{Tx}$. El dispositivo está en la celda de espera y nadie ha transmitido. Por lo tanto, establece su estado en Tx para poder transmitir en el siguiente *minislot*.

- Si ($r_t = \mathrm{Tx}$ y $f_t = \mathrm{C}$) entonces el dispositivo mantiene $r_{t+1} = \mathrm{Tx}$, con probabilidad p, o cambia su valor a $r_{t+1} \leftarrow \mathrm{W}$ con probabilidad $1 - p$. Esta regla determina qué hacer cuando el dispositivo está en la celda de transmisión y se detecta una colisión en el medio. Ante esta situación, son posibles los dos casos siguientes. Si t es el primer *minislot* de una fase o el que sigue a un mensaje de retroalimentación E, el dispositivo utiliza el valor actual de N para calcular $p = p(N)$, de acuerdo con la ecuación (3.2). En caso contrario, la estación utiliza $p = 0.5$.

- Si ($r_t = \mathrm{W}$ y $f_t = \mathrm{NC}$) entonces $r_{t+1} \leftarrow \mathrm{Tx}$, $s \leftarrow s + 1$ y $N \leftarrow N - 1$. El dispositivo está en la celda de espera y otro dispositivo logró transmitir exitosamente una RREQ. Por lo tanto, pasa a la celda de transmisión, aumenta su posición de reserva y disminuye en uno el número estimado de dispositivos en contienda.

- Si ($r_t = \mathrm{W}$ y $f_t = \mathrm{C}$) entonces el dispositivo mantiene $r_{t+1} = \mathrm{W}$. El dispositivo está en la celda de espera y hubo una colisión en el medio. Por lo tanto, permanece en estado de espera.

Es importante notar que el valor $\widehat{N}$ lo proporciona la estación base solo en el mensaje de retroalimentación del primer *minislot* colisionado del CRI. Cada dispositivo contendiente copia dicho valor en su variable N y lo actualiza localmente al decrementarlo en 1 cuando recibe un mensaje de retroalimentación NC, lo cual significa una transmisión exitosa de un RREQ. Además, los dispositivos son capaces de identificar el inicio de una fase y un *minislot* vacío cuando reciben un mensaje de retroalimentación NC o E, respectivamente. El CRI finaliza con dos mensajes de retroalimentación NC consecutivos (nótese que esta situación solo puede ocurrir al final de un CRI). Como resultado de este algoritmo, al final del CRI cada dispositivo conoce su turno de transmisión en el DTI, gracias a su variable s.

3.2.2. Intervalo de Transmisión de Datos (DTI)

Una vez finalizado el CRI, comienza el DTI. En esta etapa los dispositivos transmiten sus paquetes turnándose según el orden en que consiguieron su reservación, lo que típicamente podría considerarse TDMA. La estación base notifica el final del DTI en curso a través del mensaje de inicio de reserva RB.

3.2.3. Ejemplo

En la Figura 3.6, se muestra un ejemplo del funcionamiento de 2CA-R^2. Considerando la red de la Figura 3.1 y la celda de transmisión de la Figura 3.6 (a), se puede observar que la evolución del CRI es la misma que en el ejemplo de la Figura 3.4. Por lo tanto, se puede asumir el mismo procedimiento del algoritmo *Adaptive*-2C con estrategia de actualización por fase descrito en la sección 3.1.2.2. Sin embargo, esta vez, cada dispositivo contiende mediante un paquete RREQ para obtener una reservación de un turno en la DTI, es decir, reservar una ranura de transmisión de datos. Como consecuencia, una vez que la estación base indica el final del CRI, los dispositivos transmitirán su paquete de datos en el mismo orden en que obtuvieron su reservación durante el CRI, lo cual se muestra en la Figura 3.6 (b).

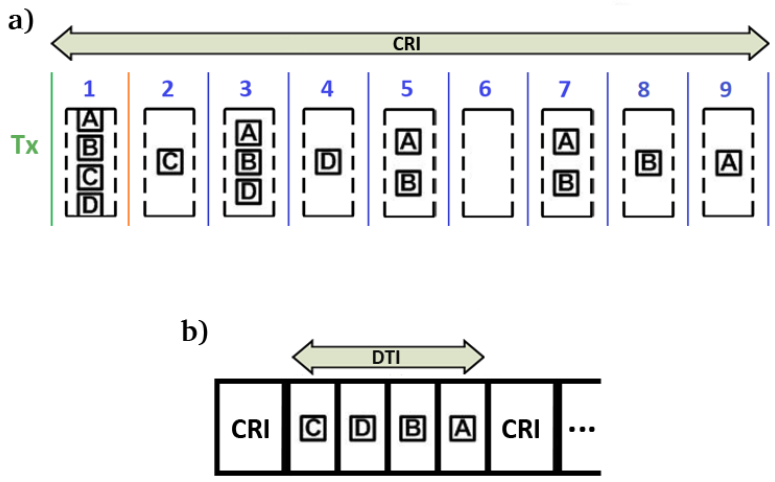

Figura 3.6. Ejemplo de funcionamiento de 2CA-R^2. a) Evolución del CRI; b) Evolución del DTI

CAPÍTULO 4

Evaluación del protocolo 2CA-R^2

4.1. Elementos para la evaluación de desempeño

El desempeño del protocolo propuesto fue evaluado mediante la simulación de eventos discretos. Para ello, se implementó el modelo del protocolo 2CA-R^2 en el simulador OMNeT++. OMNeT++ es un simulador modular de licencia pública académica basado en lenguaje C++, empleado habitualmente para modelar redes de comunicación cableadas e inalámbricas, que incorpora un entorno de ejecución gráfico. OMNeT++ posee una arquitectura de componentes para modelos gratuitos que proporcionan compatibilidad con redes de sensores, redes ad-hoc redes inalámbricas, protocolos de Internet, etc. Los componentes o módulos se programan y luego se ensamblan en modelos más grandes utilizando un lenguaje de alto nivel [50]. Uno de los modelos disponibles para OMNeT++, llamado INET *Framework*, contiene módulos para la pila de Internet, además de protocolos cableados e inalámbricos entre los cuales se encuentran algunas versiones de la serie de estándares MAC IEEE 802.11, así como sus especificaciones de capa física [51].

El modelo del protocolo 2CA-R^2 se construyó como proyecto independiente, ajeno a cualquier modelo existente en OMNeT++. Por otra parte, y con fines comparativos, se hizo uso de la implementación DCF disponible en INET *Framework*. Vale la pena destacar que la razón de elegir DCF como referencia radica en que esta implementación del protocolo CSMA/CA es el mecanismo MAC fundamental de la serie de estándares IEEE 802.11, entre los que se encuentra 802.11ah, tecnología que es una de las principales candidatas para soportar aplicaciones IoT en redes inalámbricas de área local (WLAN) en los próximos años [52].

La implementación DCF, utilizada en el simulador, sigue el comportamiento estándar descrito en la especificación 802.11 (incluidos los valores de *CWmin*, *CWmax*, el límite de reintentos de transmisión o *Retry Limit*, y el mecanismo RTS/CTS).

Todas las simulaciones se realizaron en OMNeT++ versión 5.7, con INET *Framework* versión 4.2.10, bajo el sistema operativo Linux Ubuntu 20.04.6 LTS.

Es importante destacar que al referirse a la estación base, se utiliza el término «punto de acceso» de forma indistinta, situación que es común en el contexto de las redes Wi-Fi.

4.2. Medidas de desempeño

Para medir la eficiencia del protocolo propuesto 2CA-R^2 y verificar su rendimiento, en comparación con DCF, se utilizaron las siguientes medidas de desempeño: la tasa de transferencia efectiva promedio de la red o *mean network throughput*, el retardo de acceso promedio global o *global mean access delay*, el retado de encolamiento promedio global o *global mean queuing delay* y el porcentaje promedio de paquetes desechados.

a) Para la i-ésima simulación, el *network throughput* (${T_N}^{(i)}$) se calcula como el número total de bits correspondientes a paquetes de carga útil, recibidos en el punto de acceso ($\#bits$), dividido entre el tiempo total de esa simulación (t_s). Se define conforme a la ecuación (4.1).

$$ {T_N}^{(i)} = \frac{\#bits}{t_s}. \tag{4.1} $$

A su vez, el *mean network throughput* (T_N) se calcula como el promedio del *network throughput* de un conjunto de K simulaciones. Se define conforme a la ecuación (4.2).

$$ T_N = \frac{1}{K}\sum_{i=1}^{K} {T_N}^{(i)}. \tag{4.2} $$

b) El *access delay* del *j*-ésimo paquete de datos (D_{A_j}) es el tiempo que transcurre desde el instante en que se genera en su respectivo dispositivo (t_{g_j}), hasta el instante en que se recibe el acuse de transmisión exitosa correspondiente (devuelto por el punto de acceso) (t_{r_j}). Se define conforme a la ecuación (4.3).

$$ D_{A\,j} = t_{r_j} - t_{g_j}. \tag{4.3} $$

Luego, el *mean access delay* de la *i*-ésima simulación ($D_A{}^{(i)}$), se calcula como el promedio del *access delay* de los J paquetes recibidos exitosamente durante el tiempo total de la simulación. Se define conforme a la ecuación (4.4).

$$D_A{}^{(i)} = \frac{1}{J}\sum_{j=1}^{J} D_{A_j}. \tag{4.4}$$

A su vez, el *global mean access delay* (D_A) se calcula como el promedio del *mean access delay* de un conjunto de K simulaciones. Se define conforme a la ecuación (4.5).

$$D_A = \frac{1}{K}\sum_{i=1}^{K} D_A{}^{(i)}. \tag{4.5}$$

c) El *queuing delay* del *j*-ésimo paquete de datos (D_{Q_j}) es el tiempo que transcurre desde el instante en que se genera en su respectivo dispositivo (t_{g_j}), hasta el instante en que se comienza a intentar su transmisión (t_{a_j}); es decir, hasta que no hay otro paquete por delante de él en la cola de espera. Se define conforme a la ecuación (4.6).

$$D_{Q_j} = t_{a_j} - t_{g_j}. \tag{4.6}$$

Luego, el *mean queuing delay* de la *i*-ésima simulación ($D_Q{}^{(i)}$), se calcula como el promedio del *queuing delay* de los J paquetes recibidos exitosamente durante el tiempo total de la simulación. Se define conforme a la ecuación (4.7).

$$D_Q{}^{(i)} = \frac{1}{J}\sum_{j=1}^{J} D_{Q_j}. \tag{4.7}$$

A su vez, el *global mean queuing delay* (D_Q) se calcula como el promedio del *mean queuing delay* de un conjunto de K simulaciones. Se define conforme a la ecuación (4.8).

$$D_Q = \frac{1}{K}\sum_{i=1}^{K} D_Q{}^{(i)}. \tag{4.8}$$

d) Para la i-ésima simulación, el porcentaje de paquetes desechados ($\%\ {p_d}^{(i)}$) se calcula como número total de paquetes desechados ($\#p_d$), dividido entre el número total de paquetes transmitidos en esa simulación ($\#p_{Tx}$). Se define conforme a la ecuación (4.9).

$$\%\ {p_d}^{(i)} = \frac{\#p_d}{\#p_{Tx}}(100)\ \ [\%]. \tag{4.9}$$

A su vez, el porcentaje promedio de paquetes desechados ($\%p_d$) se calcula como el promedio del porcentaje de paquetes desechados de un conjunto de K simulaciones. Se define conforme a la ecuación (4.10).

$$(\%p_d) = \frac{1}{K}\sum_{i=1}^{K} \%\ {p_d}^{(i)}\ \ [\%]. \tag{4.10}$$

4.3. Parámetros de simulación

Cada uno de los escenarios de simulación consistió en un área cuadrada de 100 m × 100 m, con un único punto de acceso ubicado en el centro de la superficie y un número fijo de dispositivos o nodos distribuidos uniformemente. Este número se ajustó, para cada escenario, entre 1 y 256 dispositivos[3]. Tanto el punto de acceso como cada dispositivo se configuraron con un rango de alcance máximo de 71 m, de forma que estos fueran mutuamente alcanzables, pero no necesariamente todos los nodos fueran alcanzables entre sí. Es decir, el problema del nodo oculto era latente cuando había más de un dispositivo en el escenario. La Figura 4.1 muestra un ejemplo de un escenario de simulación con 5 dispositivos, en el cual los dispositivos denotados como $\mathrm{device}[3]$ y $\mathrm{device}[4]$ no son alcanzables entre sí; misma situación que ocurre entre $\mathrm{device}[1]$ y $\mathrm{device}[4]$.

[3] Para efectos comparativos con DCF, se estableció un máximo de 256 dispositivos. El protocolo DCF, utilizado en el estándar IEEE 802.11ah, restringe a dicho número la cantidad de estaciones que pueden contender simultáneamente.

Figura 4.1. Ejemplo de escenario de simulación con 5 dispositivos

Para propiciar escenarios con una gran cantidad de intentos de acceso simultáneos, se consideraron condiciones de generación de tráfico *greedy*. Es decir, cada dispositivo generaba un paquete de datos en el instante en que se iniciaba la simulación. A partir de ese momento, tan pronto como recibía un acuse de envío exitoso, generaba un nuevo paquete para ser transmitido. Este patrón de generación de tráfico también se conoce con frecuencia como condiciones de saturación y asemeja las condiciones esperadas en el contexto de las comunicaciones M2M. Vale la pena mencionar que tales condiciones siguen un comportamiento que no se puede modelar mediante el proceso de Poisson estándar [53].

Es importante señalar que, en el protocolo 2CA-R^2 se utilizó el número de RREQ exitosos del CRI anterior como estimación de la multiplicidad de colisión ($\widehat{N}$) del CRI en curso. Es decir, se utilizó un predictor lineal de orden 1 para estimar $\widehat{N}$. Esta estrategia de estimación se consideró adecuada en condiciones de generación de tráfico *greedy* ya que, una vez finalizado un DTI, idealmente todos los dispositivos en el escenario habrían generado nuevamente un paquete de datos para transmitir. Así, el número de dispositivos contendientes en un CRI actual, sería igual o similar al del CRI anterior.

Todas las simulaciones se ejecutaron con una tasa de transmisión de datos de 1 Mbps y una carga útil de paquete de datos de 65 bytes. Estos valores de tasa de transmisión y tamaño de paquete se encuentran comúnmente en diferentes tipos de aplicaciones del IoT (ver [11] y [53]). Para la tasa de transmisión y el tamaño de paquete elegidos, se configuró cada escenario de simulación con un valor fijo de Tasa de Bit en Error (*Bit Error Rate* o BER) para producir una Tasa de Pérdida de Paquetes de

datos (*Packet Error Rate* o PER) en el canal de transmisión. En caso de error utilizando el protocolo 2CA-R^2, el punto de acceso envía un mensaje de error y se intenta retransmitir cada paquete afectado en el siguiente CRI.

Para determinar el BER necesario en función del PER deseado, se utilizó la relación que se muestra en la ecuación (4.11).

$$\mathrm{BER} = 1 - (1 - \mathrm{PER})^{\frac{1}{N_b}} \tag{4.11}$$

donde, N_b es el número de bits en el paquete de datos. En esta ecuación se asume que al menos un bit en error del paquete ocasiona la pérdida del paquete completo.

Adicionalmente y con fines de comparación, el tamaño elegido del RREQ fue de 20 bytes, que es la misma longitud que utiliza DCF para los mensajes RTS. Vale la pena aclarar que, el tamaño del RREQ puede reducirse, ya que lo único que se requiere es el ID del dispositivo transmisor. La operación de cada protocolo se simuló con 10 diferentes semillas, para cada cantidad fija de dispositivos, de cada escenario. El objetivo de ello fue obtener valores medios y proporcionar sus intervalos de confianza con un nivel de confianza del 95%. El resumen de los valores de los parámetros utilizados se muestra en la Tabla 4.1.

Tabla 4.1. Resumen de los parámetros de simulación

Nombre del Parámetro	Valor
Área de simulación	100 m × 100 m
Número de dispositivos	1, 2, 3, 5, 10, 15, 20, 25, 50, 100, 150, 200, 256
Número de CRI para calcular $\widehat{N}$ en 2CA-R^2	1
Máximo rango de transmisión (d_{max})	71 m
Tasa de transmisión de datos (R)	1 Mbps
Tamaño de paquete de datos	65 bytes
Tamaño de RTS / Tamaño de RREQ	20 bytes
Tamaño de mensaje de Retroalimentación en 2CA-R^2	1 byte
Retry Limit en DCF	7
Valores de *CWmin*, *CWmax* en DCF	31, 1023
Duración de ranura en DTI en 2CA-R^2	520 μs
Duración de *minislot* en 2CA-R^2	160 μs
Duración de cada simulación	60 s

4.4. Escenarios de simulación

4.4.1. Canal libre de errores

El primer conjunto de simulaciones se llevó a cabo bajo un escenario con un canal de transmisión ideal, es decir, libre de errores. Para ello, el BER se estableció con un valor igual a 0, dando como resultado un PER igual a 0, de acuerdo con la ecuación (4.11). Asimismo, se incluyó el protocolo 2C-R^2, descrito en [49], en la evaluación de desempeño, cuyo modelo también fue implementado en OMNeT++. El objetivo de considerar el protocolo 2C-R^2 en las simulaciones de este escenario fue analizar su desempeño y examinar las ventajas de una probabilidad adaptable, en lugar de una fija, en la solución de conflictos, todo ello en el contexto de las comunicaciones M2M. Es oportuno tener en cuenta que, tal como se menciona en la sección 3.2, el protocolo 2C-R^2 utiliza el algoritmo 2C estándar en la etapa de contienda, mientras que el protocolo 2CA-R^2 emplea el algoritmo *Adaptive*-2C, con estrategia de actualización por fase, para el mismo propósito. El protocolo 2C-R^2 se configuró con solicitudes RREQ de 20 bytes y con mensajes de retroalimentación binarios de tamaño despreciable.

Las figuras 4.2 y 4.3 muestran los resultados obtenidos, en términos de *mean network throughput* y de *global mean access delay*, respectivamente. Cabe destacar que, en todas las gráficas de esta tesis, los resultados correspondientes a los protocolos 2C-R^2 y 2CA-R^2 están etiquetados como «2C-R2» y «2CA-R2» respectivamente.

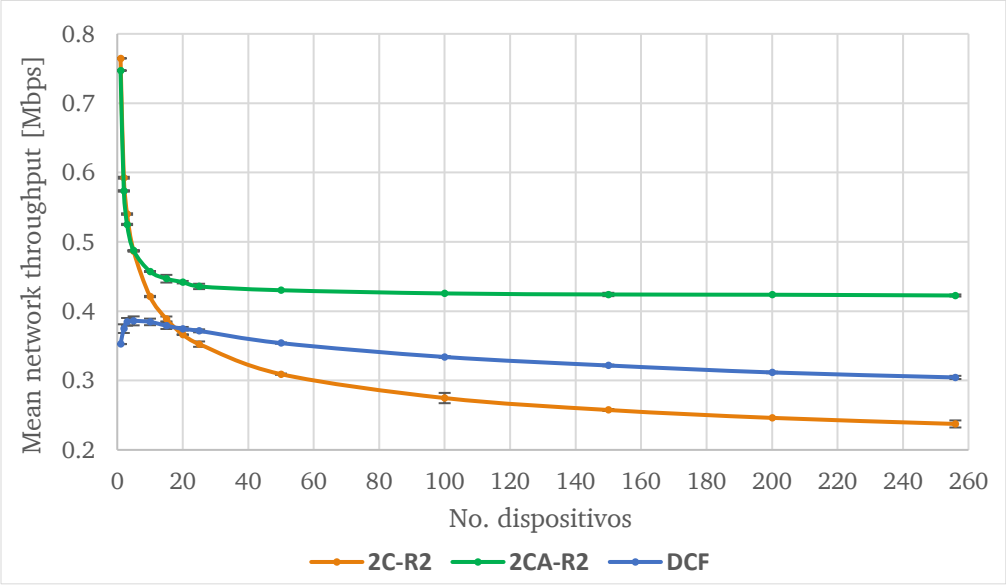

Figura 4.2. *Mean network throughput* con canal libre de errores

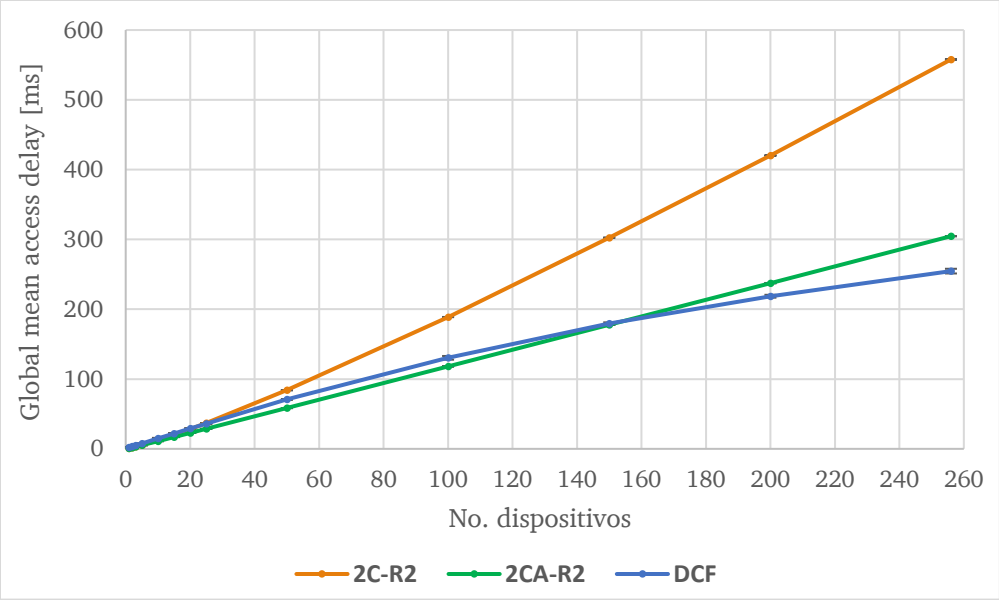

Figura 4.3. *Global mean access delay* con canal libre de errores

En la Figura 4.2 se puede observar que, cuando hay entre 1 y 20 dispositivos contendientes en el escenario, el desempeño de 2CA-R^2 disminuye a medida que aumenta el número de dispositivos. Sin embargo, el *mean network throughput* sigue siendo mayor que el alcanzado por DCF y 2C-R^2. Además, se observa que, cuando hay más de 20 dispositivos contendientes, el desempeño de 2CA-R^2 se mantiene estable a medida que aumenta el número de dispositivos y es mejor comparado con el de DCF y 2C-R^2, que disminuyen su desempeño a medida que aumenta el número de dispositivos.

En la Figura 4.3 vale la pena notar que, en términos de *global mean access delay*, el desempeño de 2C-R^2 es inferior, comparado con el de DCF y 2CA-R^2, cuando hay más de 30 dispositivos en contienda. Por otra parte, cuando el número de dispositivos en contienda es mayor a 160, ocurre un cambio de tendencia en dónde el *global mean access delay* es mejor para DCF que para 2CA-R^2. Sin embargo, esto se debe a la cantidad de paquetes que son desechados por DCF tras exceder el límite de reintentos de transmisión, mientras que para 2CA-R^2 y 2C-R^2 no se descartan paquetes por esta situación ya que no existe un límite de reintentos. Para clarificar este punto, la Figura 4.4 muestra el porcentaje promedio de paquetes que exceden el parámetro *Retry Limit* de DCF, mismos que son desechados. Se añaden además intervalos de confianza con un nivel de confianza del 95%. Se puede observar que, en el peor caso de este escenario, es decir, cuando 256 dispositivos se encuentran transmitiendo simultáneamente, alrededor del 16% de sus paquetes son desechados. Es importante destacar que estos paquetes no contribuyen en el cálculo del *global mean access delay*, ya que no son entregados

exitosamente. De esta manera, el parámetro *Retry Limit* en DCF ayuda a mantener el *mean access delay* bajo control, ya que, el tiempo de retardo depende del valor elegido para dicho parámetro a costa de desechar algunos paquetes de datos provenientes de los dispositivos.

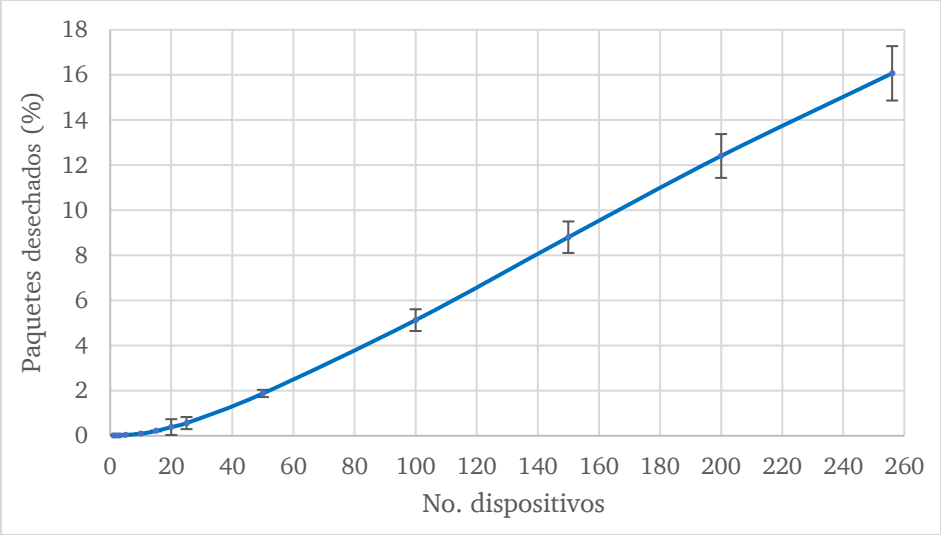

Figura 4.4. Porcentaje promedio de paquetes desechados por *Retry Limit* en DCF con canal libre de errores

4.4.2. Canal con pérdida de paquetes de 1%

Un segundo conjunto de simulaciones se llevó a cabo bajo un escenario con un canal de transmisión con una pérdida de paquetes de datos aproximada al 1%. Para ello, se estableció un valor de BER $= 1.9327 \times 10^{-5}$, dando como resultado un PER ≈ 0.01, de acuerdo con la ecuación (4.11). En este escenario de nueva cuenta se incluyó el protocolo 2C-R^2 en la evaluación de desempeño con la misma configuración que en la sección 4.4.1. Es importante precisar que todos los escenarios en los que se estableció un BER con un valor mayor a 0, este parámetro se configuró de tal forma que tuviera efecto solo en los paquetes de datos, en los mensajes RTS en DCF y mensajes RREQ en 2C-R^2 y 2CA-R^2. Es decir, el BER solo afectó a los paquetes destinados al punto de acceso. Sin embargo, para los mensajes RTS y RREQ cuyo tamaño era de 20 bytes, el $PER_R \approx 0.0031$, de acuerdo con la ecuación (4.11).

Las figuras 4.5 y 4.6 muestran los resultados obtenidos, en términos de *mean network throughput* y de *global mean access delay*, respectivamente.

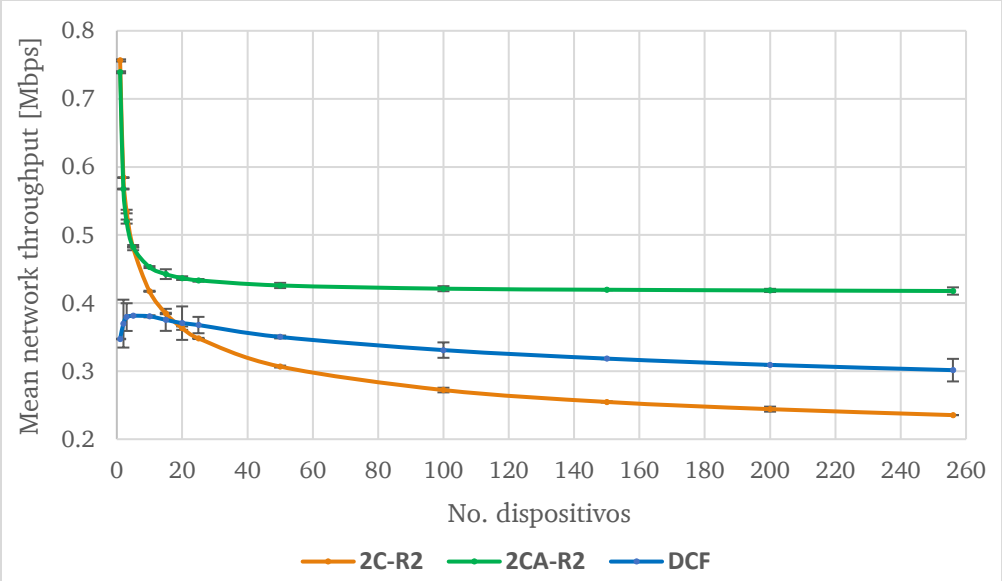

Figura 4.5. *Mean network throughput* con canal con pérdida de paquetes de 1%

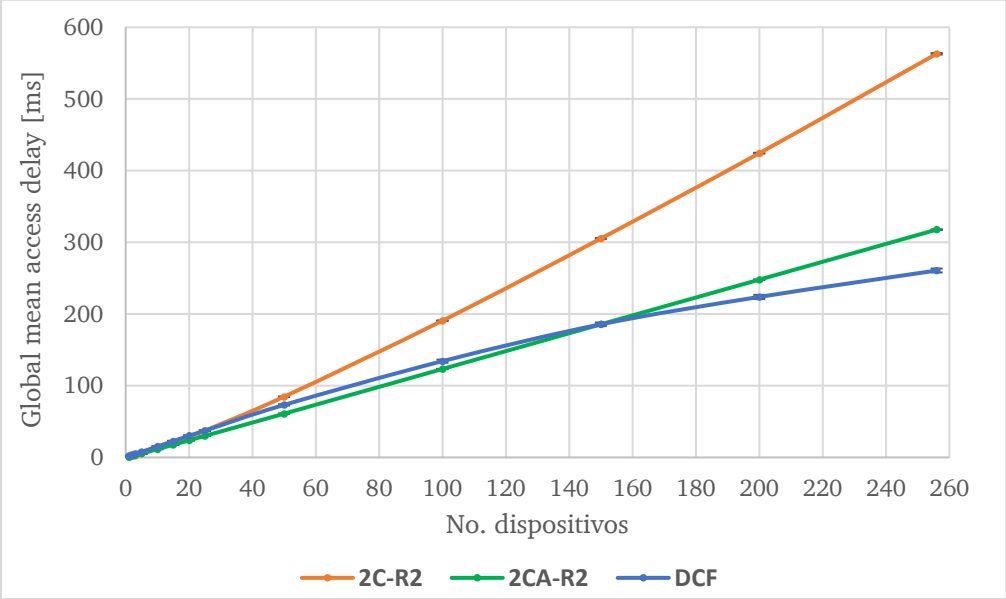

Figura 4.6. *Global mean access delay* con canal con pérdida de paquetes de 1%

En la Figura 4.5, se puede observar una tendencia similar, en términos de *mean network throughput*, con respecto al escenario con un canal libre de errores mostrado en la Figura 4.2. De hecho, la diferencia existente entre los dos escenarios, que se debe a la presencia de PER en el canal de transmisión, es poco perceptible. Nuevamente se aprecia que cuando hay entre 1 y 20 dispositivos contendientes en el escenario, el desempeño de 2CA-R^2 disminuye a medida que aumenta el número

de dispositivos, pero sigue siendo mayor que el alcanzado por DCF y 2C-R^2. Además, otra vez se aprecia que, cuando hay más de 20 dispositivos contendientes, el desempeño de 2CA-R^2 se mantiene estable a medida que aumenta el número de dispositivos y es mejor comparado con el de DCF y 2C-R^2, que disminuyen su desempeño a medida que aumenta el número de dispositivos.

A su vez, la Figura 4.6 también muestra una tendencia similar, en términos de *global mean access delay*, con respecto al escenario con un canal libre de errores que se muestra en la Figura 4.3. Aquí, el desempeño de 2C-R^2 también es inferior, comparado con el de DCF y 2CA-R^2, cuando hay más de 30 dispositivos en contienda. Nuevamente, cuando el número de dispositivos en contienda es mayor a 160, ocurre un cambio de tendencia en dónde el *global mean access delay* es mejor para DCF que para 2CA-R^2. Una vez más esto se debe a la cantidad de paquetes que son desechados por DCF, tras exceder el límite de reintentos de transmisión, mientras que para 2CA-R^2 y 2C-R^2 no existe desecho paquetes. La Figura 4.7 muestra el porcentaje promedio de paquetes desechados por DCF en este escenario, con sus respectivos intervalos de confianza con un nivel de confianza del 95%. Aquí también se observa una tendencia similar, con respecto al escenario con un canal libre de errores, en donde en el peor caso del escenario, alrededor del 16% de los paquetes que se intentan transmitir son desechados. Estos paquetes no contribuyen en el cálculo del *global mean access delay* ya que no son entregados exitosamente.

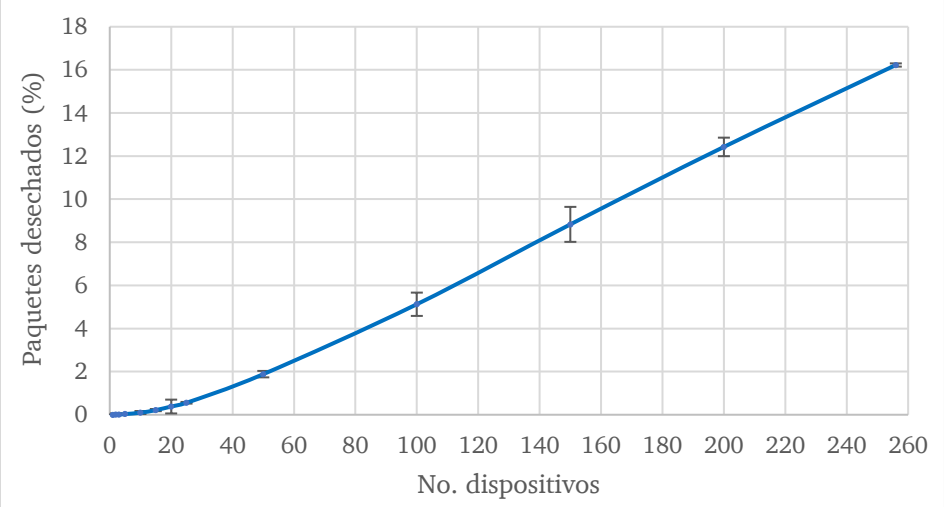

Figura 4.7. Porcentaje promedio de paquetes desechados por *Retry Limit* en DCF con canal con pérdidas del 1%

4.4.3. Un único paquete por dispositivo

Un tercer conjunto de simulaciones se llevó a cabo bajo un escenario en donde cada uno de los dispositivos contendientes generó un único paquete de datos que intentó transmitir desde el comienzo de la simulación. Una vez que un paquete era transmitido exitosamente, o se desechaba en el caso de DCF, el dispositivo correspondiente entraba en estado de reposo (*idle*) hasta el final de la simulación. La finalidad de este conjunto de simulaciones fue analizar un escenario recurrente en el contexto de las comunicaciones M2M, dado que en algunas aplicaciones de IoT es común que se produzcan ráfagas de acceso simultáneo, desde distintos dispositivos hacia un punto de acceso en común, seguidos de intervalos de inactividad [11]. En este escenario el canal de transmisión se modeló con un BER $= 1.9327 \times 10^{-5}$ para generar un PER ≈ 0.01, de acuerdo con la ecuación (4.11).

Estas condiciones sirven para evaluar el tiempo que el sistema tarda en transferir todo un conjunto de n paquetes al punto de acceso, cada uno de ellos proveniente de un distinto dispositivo. Es decir, n corresponde al número de dispositivos contendientes en el escenario. De esta manera, el tiempo de transferencia del conjunto completo de la *i*-ésima simulación ($T_S{}^{(i)}$) corresponde al tiempo en que el n-ésimo dispositivo recibe el acuse de envío exitoso de su paquete de datos, o al tiempo en que el paquete es desechado en el caso de DCF. La operación de cada protocolo fue simulada 30 veces para cada valor de n mencionado en la Tabla 4.1, con diferente semilla por cada simulación, tanto para 2CA-R^2 como para DCF. Así, el tiempo promedio de transferencia del conjunto completo (T_S) se calculó de acuerdo con la ecuación (4.12).

$$T_S = \frac{1}{30}\sum_{i=1}^{30} T_S{}^{(i)}. \tag{4.12}$$

Los resultados obtenidos se muestran en la Figura 4.8, con sus respectivos intervalos de confianza con un nivel de confianza del 95%. Como se puede observar, cuando hay más de 10 dispositivos contendientes en el escenario, el tiempo requerido por 2CA-R^2 para enviar un conjunto completo de n paquetes al punto de acceso es menor comparado con el que requiere DCF. También es notable que, a medida que aumenta el número de dispositivos, el desempeño de 2CA-R^2 es cada vez mejor que el de DCF. De hecho, en el peor caso de este escenario, que es cuando $n = 256$, existe una diferencia de más de 100 ms a favor de 2CA-R^2, con respecto a DCF. La importancia de este resultado recae en la diferencia en la cantidad de dispositivos que pueden transmitir su paquete de datos, al usar 2CA-R^2 o DCF, en una ventana de tiempo dada. Por ejemplo, si acotamos el tiempo a una ventana de 250 ms, lo

cual se señala en la Figura 4.8 con una línea horizontal roja punteada, alrededor de 145 dispositivos transmitirán su paquete de datos usando DCF, mientras que, para la misma ventana de tiempo, cerca de 200 dispositivos lograrán transmitir al usar 2CA-R^2. Esto es, el protocolo 2CA-R^2 logra una ganancia de aproximadamente 40% con respecto a DCF

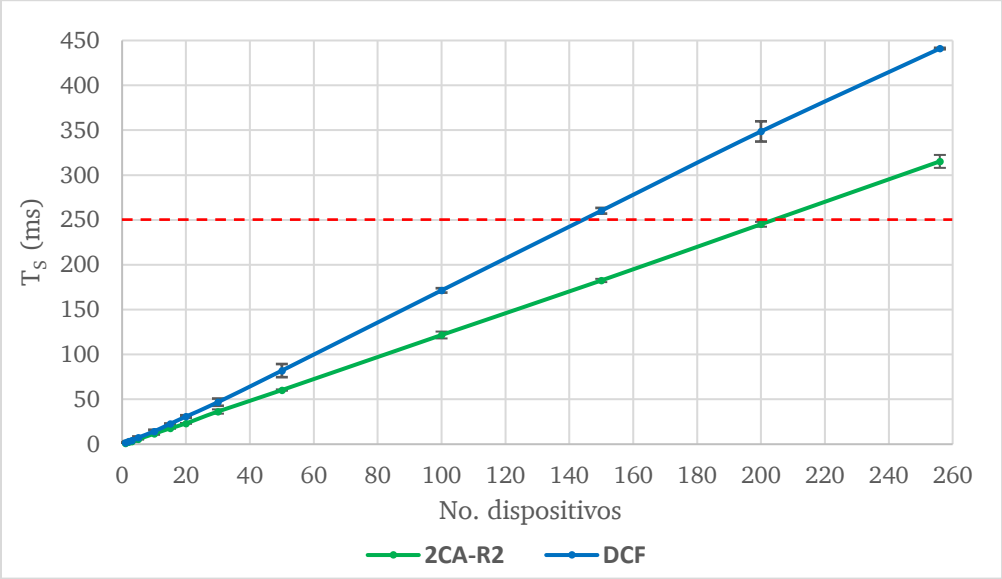

Figura 4.8. Tiempo de transferencia de un conjunto completo, con canal con pérdida de paquetes de 1%

Por otra parte, la Figura 4.9 muestra los resultados obtenidos, en términos de *global mean access delay*, para el mismo conjunto de simulaciones. De aquí se puede apreciar que el desempeño de ambos protocolos es similar pero ligeramente mejor al utilizar DCF, cuando hay más de 20 dispositivos contendientes en el escenario. La razón de este comportamiento se puede comprender al observar la desviación estándar del *access delay* de los paquetes entregados en cada i-ésima simulación ($s_{D_A{}^{(i)}}$). Esta medida permite apreciar el grado de dispersión que hay en los valores de *access delay,* de cada paquete, dentro de cada simulación. Después de obtener el promedio de la desviación estándar del *access delay* de las 30 simulaciones en cada caso del escenario (s_{D_A}), se compararon los resultados obtenidos para cada protocolo, mismos que se muestran en la Tabla 4.2. De aquí se puede observar que los valores resultantes para DCF son considerablemente más grandes que los de 2CA-R^2.

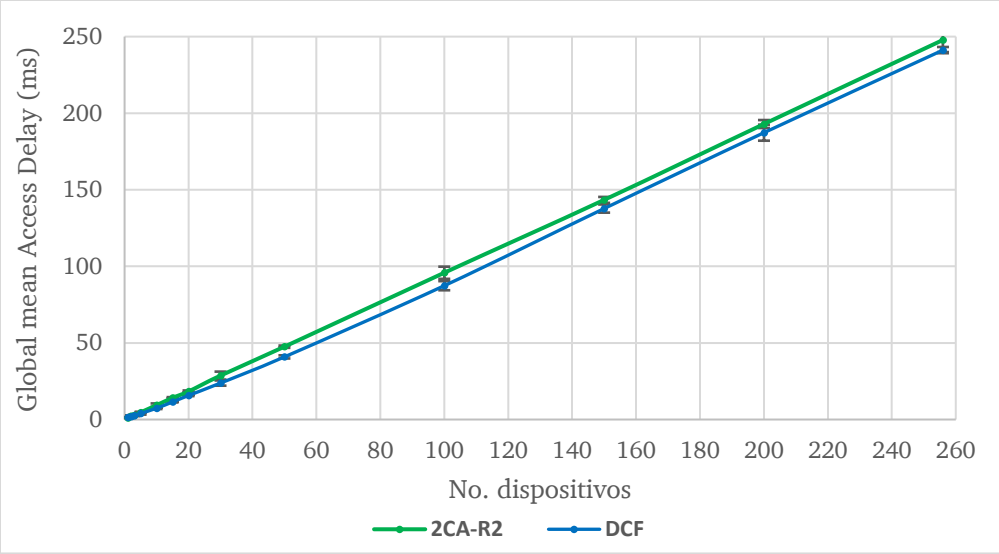

Figura 4.9. *Global mean access delay* con un único paquete de datos por dispositivo

Tabla 4.2. Promedio de desviación estándar de *access delay* con un único paquete por dispositivo

No. de dispositivos	s_{D_A} 2CA-R^2 (ms)	s_{D_A} DCF (ms)
1	0.00	0.00
2	0.37	1.04
3	0.53	1.48
5	0.84	2.33
10	1.60	4.29
15	2.37	6.61
20	3.11	8.84
25	4.65	13.34
50	7.69	22.37
100	15.38	45.47
150	22.96	68.43
200	30.58	90.99
256	39.26	115.10

A su vez, la explicación para la obtención de estos resultados se encuentra en el principio de operación de cada uno de los protocolos. Para ilustrar este punto, el diagrama de la Figura 4.10 muestra un ejemplo hipotético de los tiempos de recepción, o transmisión exitosa, según el funcionamiento de los mecanismos DCF y 2CA-R^2. En este escenario, 5 dispositivos han transmitido, cada uno de ellos, un paquete de datos. De manera similar al conjunto de las simulaciones de esta sección, en este ejemplo se asume que cada dispositivo generó su respectivo paquete de datos en $t = 0$.

Vale la pena recordar que, de acuerdo con la ecuación (4.3), el *access delay* del j-ésimo paquete recibido exitosamente se calcula como $D_{A\,j} = t_{r_j} - t_{g_j}$. Sin embargo, el hecho de que todos los paquetes del ejemplo se generaran en $t = 0$, significa que $t_{g_j} = 0$. Por lo tanto, el tiempo de entrega de cada paquete es en realidad el *access delay* del mismo (es decir, $D_{A\,j} = t_{r_j}$). De la Figura 4.10 (a) se hace notar que, al emplear DCF como mecanismo MAC, un dispositivo puede transmitir su paquete de datos de manera exitosa en un intervalo de tiempo corto o, por el contrario, en un tiempo considerablemente largo, después de haberlo generado. Tal como se menciona en la sección 2.1.3, este tipo de aleatoriedad se debe a los tiempos de espera exponenciales que caracterizan al protocolo CSMA/CA y su implementación DCF.

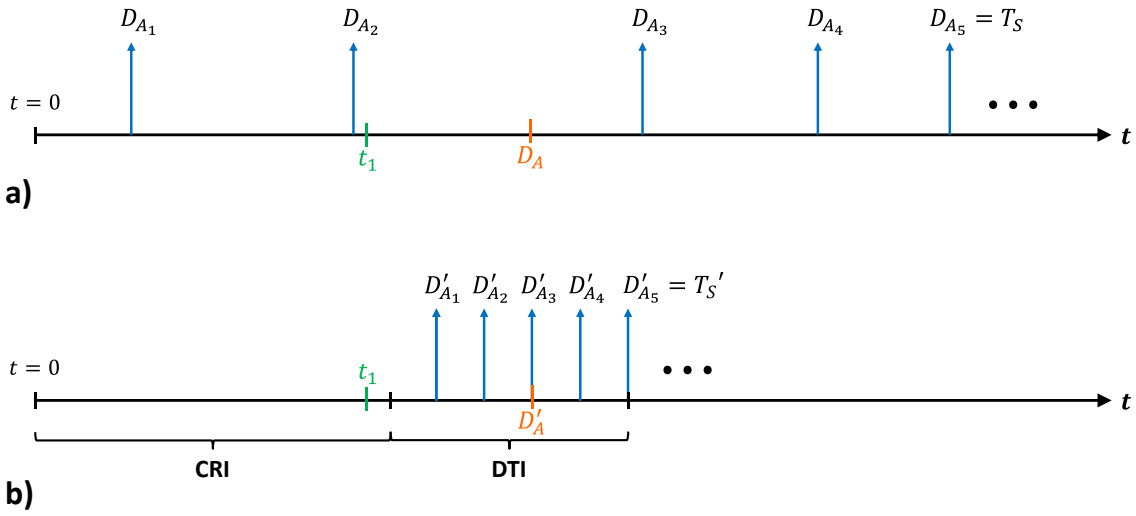

Figura 4.10. Ejemplos de tiempos de recepción en el funcionamiento de: **a)** DCF y **b)** 2CA-R^2

Por otra parte, de la Figura 4.10 (b), es destacable que, al emplear el protocolo 2CA-R^2, ningún dispositivo puede transmitir su paquete de datos en un intervalo de tiempo corto, después de haberlo generado. Esto se debe a que mientras exista un CRI en curso, solo habrá transmisiones de solicitudes RREQ. No obstante, una vez finalizado el CRI, en el DTI todos los dispositivos transmitirán su respectivo paquete en intervalos de tiempo regulares cuyo tamaño depende del tamaño de paquete, tamaño del acuse de recibo y la tasa de datos. Este valor corresponde a 528 μs para un tamaño de paquete de 65 bytes, con un acuse de 1 byte, a una tasa de 1 Mbps, valores idénticos a los parámetros utilizados en las simulaciones. Comparando los diagramas de tiempo de ambos mecanismos y tomando en cuenta

las características mencionadas, al emplear DCF se pueden haber transmitido uno o más paquetes de datos en un determinado tiempo $t_1 > 0$, mientras que al emplear 2CA-R^2 los dispositivos pueden encontrarse todavía en el CRI, es decir, no haber transmitido aun ningún paquete de datos en el mismo t_1. Sin embargo, utilizando DCF, el tiempo de transferencia del conjunto completo T_S será mayor que el tiempo T_S' obtenido al emplear el protocolo 2CA-R^2, aun cuando los tiempos de *mean access delay* sean similares en ambos casos, o lo que es lo mismo $D_A \approx D_A'$. Como es de esperarse, al utilizar DCF, la desviación estándar de los tiempos de *access delay* de cada paquete tendrá un valor más elevado comparado con la misma medida en caso de usar 2CA-R^2, ya que, al utilizar este protocolo híbrido, los tiempos de *access delay* de cada paquete se encuentran relativamente cercanos al promedio. Este último hecho también garantiza mayores condiciones de equidad en la entrega de los paquetes de cada dispositivo involucrado.

Dados los resultados mostrados en la Tabla 4.2, que muestran una mayor dispersión en los tiempos de recepción de cada paquete dentro de cada simulación en la que se utilizó DCF, en comparación con los tiempos de recepción dentro de cada simulación en la que se empleó el protocolo 2CA-R^2, se justifican los resultados encontrados para este escenario basándose en la explicación proporcionada a lo largo de esta sección.

4.4.4. Generación de tráfico mediante un proceso de Poisson

El cuarto y último conjunto de simulaciones se llevó a cabo bajo un escenario en donde la generación de paquetes de cada uno de los dispositivos contendientes se modeló mediante una distribución exponencial con media $1/\lambda = 0.4$ s. En otras palabras, los paquetes en cada dispositivo se generaron de acuerdo con un proceso de Poisson con una tasa $\lambda = 2.5$ paquetes cada segundo. Este valor de λ se seleccionó para mantener el tráfico total de la red por debajo de la velocidad de datos del canal, incluso con el mayor número de estaciones en el sistema. Al mismo tiempo, este valor permite observar la respuesta del sistema en una amplia gama de condiciones de utilización del canal al variar el número de estaciones en la red. Es importante señalar que, al utilizar el protocolo 2CA-R^2 bajo las condiciones de tráfico de este escenario, no todos los dispositivos participan al comienzo de cada CRI, por lo que el número de dispositivos activos entre CRI consecutivos puede variar debido a la naturaleza del tipo de fuente de generación de paquetes.

En este punto cabe hacer un par de aclaraciones. La primera de ellas es que, aunque el tráfico generado por las comunicaciones M2M tiene un comportamiento que no se puede modelar mediante el proceso

de Poisson, tal como se mencionó en la sección 4.3, se decidió realizar este conjunto de simulaciones por dos razones:

1. Los tres conjuntos de simulaciones anteriores contemplan escenarios en los cuales se tiene plenamente identificado que la implementación DCF de CSMA/CA presenta un desempeño limitado (ver [54] y [55]). Esta situación motivó generar un escenario en el cual se pudiera evaluar el desempeño del protocolo 2CA-R^2, en comparación con DCF, bajo condiciones en las que DCF muestra un desempeño eficiente, tal como ocurre con un tipo de tráfico que no se encuentre en condiciones de saturación.
2. A pesar de que una gran variedad de aplicaciones del IoT generan un alto porcentaje de sus paquetes debido a mensajes de actualización periódica, los cuales producen tráfico que se representa mejor mediante una fuente *greedy*, también generan paquetes esporádicos que se desencadenan por eventos o por el intercambio de información con su punto de acceso [11]. Estos dos últimos tipos de paquetes se producen aleatoriamente y pueden modelarse mediante el proceso de Poisson.

La segunda aclaración tiene que ver con la forma en que se mide el tiempo que tarda en entregarse un paquete desde el momento de su generación. Al utilizar una fuente *greedy* en las simulaciones, cada paquete se generó justo en el instante en que comenzaba una simulación o bien en el instante en que se recibía el acuse de una transmisión exitosa. Esta característica propicia que, en el momento en que se genera un paquete, inmediatamente después comienza el intento de envío por parte del dispositivo que lo genera, por lo que no se produce una cola en el transmisor. Sin embargo, con tráfico de Poisson, la generación y la transmisión de paquetes son procesos independientes. Es decir, uno o más paquetes de datos pueden generarse dentro de un dispositivo aun cuando exista un paquete que no se ha entregado, produciéndose así una cola de espera interna. Cuando esta situación ocurre, los paquetes dentro de un dispositivo, que están a la espera de su turno para intentar transmitirse, presentan retardo de encolamiento o *queuing delay*.

En este escenario, el desempeño de ambos protocolos, en términos de *access delay*, se midió utilizando la misma métrica que en los escenarios anteriores, indicada por la ecuación (4.3), es decir $D_{A_j} = t_{r_j} - t_{g_j}$. Sin embargo, se debe tener en cuenta que, para este conjunto de simulaciones, puede existir un retardo de encolamiento en cada paquete. Por esta razón, es importante resaltar que, en este escenario en específico, el *queuing delay* de cada paquete contribuye al *access delay* del mismo.

Por otra parte, este conjunto de simulaciones se llevó a cabo bajo un escenario con un canal de transmisión libre de errores, es decir, con un BER $= 0$, que diera como resultado un PER $= 0$, de acuerdo con la ecuación (4.11). A su vez, el tamaño máximo de la cola interna de cada dispositivo se estableció en un valor de 5. Vale la pena mencionar que en todas las simulaciones realizadas para estes escenario, no existió desbordamiento de cola de espera en ninguno de los dispositivos, tanto para 2CA-R^2, como para DCF.

Las figuras 4.11 y 4.12 muestran los resultados obtenidos, en términos de *mean network throughput* y de *global mean access delay*, respectivamente.

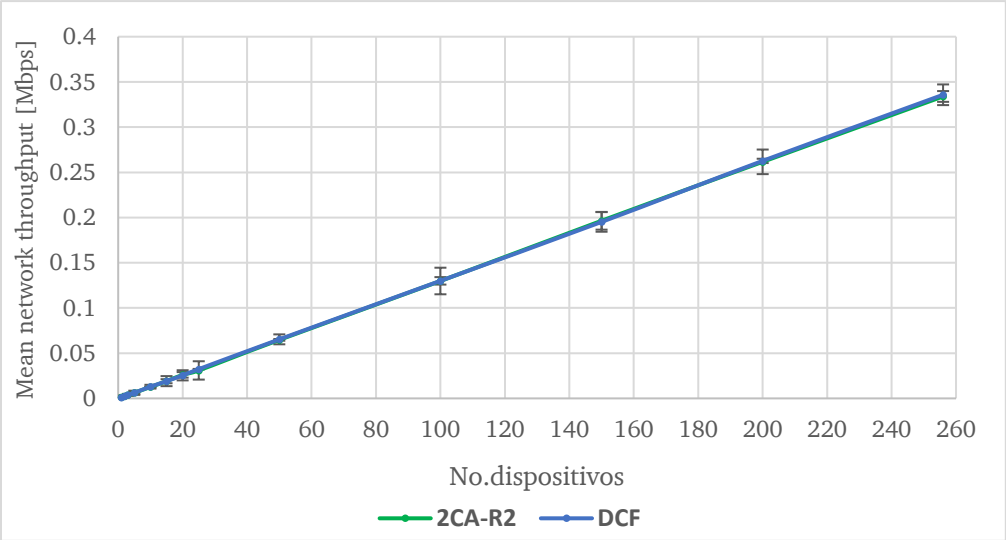

Figura 4.11. *Mean network throughput* con generación de paquetes mediante proceso de Poisson

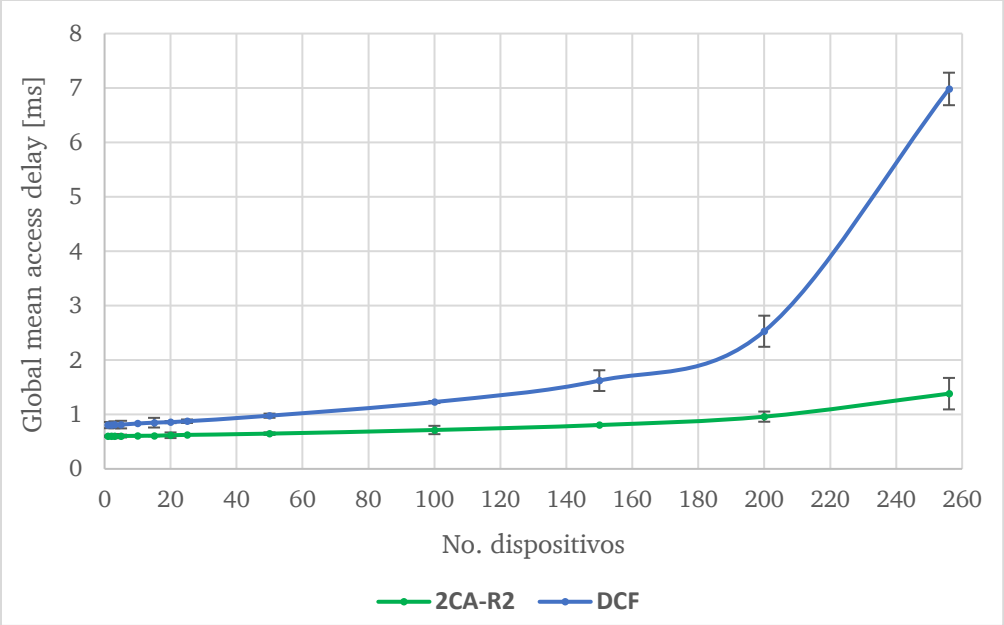

Figura 4.12. *Global mean access delay* con generación de paquetes mediante proceso de Poisson

En la Figura 4.11 se puede observar que, a diferencia de los escenarios con generación de tráfico *greedy*, el desempeño de DCF y 2CA-R^2, en términos de *mean network throughput*, aumenta a medida que incrementa el número de dispositivos. De hecho, se aprecia que la gráfica de ambos protocolos se encuentra sobrepuesta, lo que significa que el desempeño de 2CA-R^2 es similar al de DCF, bajo las condiciones de este escenario. Este comportamiento se puede explicar mediante la Tabla 4.3. En ella se puede observar la utilización y el uso del canal medidos en este escenario, tanto para 2CA-R^2 como para DCF. En este trabajo, la utilización del canal se definió como el porcentaje de tiempo que éste se mantiene ocupado debido exclusivamente a la transmisión de paquetes de datos de carga útil, mientras que el uso del canal se definió como el porcentaje de ocupación que se debe a cualquier transmisión, es decir, considerando también el envío de mensajes de señalización en ambos sentidos. De aquí se puede observar que incluso en el peor de los casos, es decir, cuando existen 256 dispositivos en el escenario, la utilización del canal que se alcanza con ambos protocolos es similar, de forma correspondiente con el *mean network throughput*. Esta situación ocurre debido a que, bajo estas condiciones, con ambos protocolos se transmiten el total de los paquetes de datos generados por los dispositivos. De hecho, cuando hay un número reducido de dispositivos en el escenario, la utilización del canal es muy baja, razón por la cual el *mean network throughput* aumenta a medida que aumenta el número de dispositivos. No obstante, también se puede apreciar que, al analizar el uso del canal, el

protocolo 2CA-R^2 alcanza un menor porcentaje comparado con DCF, lo cual indica que la transmisión de los mensajes de señalización producidos por DCF mantienen el canal ocupado durante una mayor cantidad de tiempo, comparado con 2CA-R^2.

Tabla 4.3. Utilización y uso de canal con tráfico de Poisson

No. de dispositivos	Utilización del Canal [%]		Uso del Canal [%]	
	2CA-R^2	DCF	2CA-R^2	DCF
25	3.19	3.23	4.30	7.62
50	6.47	6.52	8.97	15.39
100	12.98	12.99	18.33	30.74
200	26.16	26.26	42.23	62.34
256	33.49	33.58	57.50	81.90

Al mismo tiempo, la Figura 4.12 muestra que, a medida que aumenta el número de dispositivos, el desempeño de 2CA-R^2 en términos de *global mean access delay*, disminuye. Sin embargo, se puede observar que el desempeño del protocolo 2CA-R^2 es superior al alcanzado por DCF en todo momento, situación que es más notoria cuando incrementa el número de dispositivos. De hecho, en el peor caso, es decir, cuando hay 256 dispositivos en el escenario, existe una diferencia de alrededor de 5 ms a favor de 2CA-R^2.

Con el fin de justificar la causa de este comportamiento se analizó el *global mean queuing delay* para cada uno de los protocolos, lo cual se muestra en la Figura 4.13, añadiendo sus respectivos intervalos de confianza con un nivel de confianza del 95%.

Se puede observar que el *global mean queuing delay*, al utilizar DCF bajo las condiciones de este escenario, aumenta a medida que incrementa el número de dispositivos. De hecho, en el peor caso, es decir, cuando hay 256 dispositivos en el escenario, existe un promedio de aproximadamente 3 ms de retardo de encolamiento. A su vez, el *global mean queuing delay*, al usar el protocolo 2CA-R^2 bajo las condiciones de este escenario, es cercano a 0, independientemente del aumento en el número de dispositivos. Así, es notable que el *global mean access delay*, para 2CA-R^2, se debe casi en su totalidad al tiempo de resolución de colisiones, a diferencia de DCF en el que este tipo de retardo tiene una contribución significativa al *global mean access delay*, situación que se exacerba al incrementar el número de dispositivos.

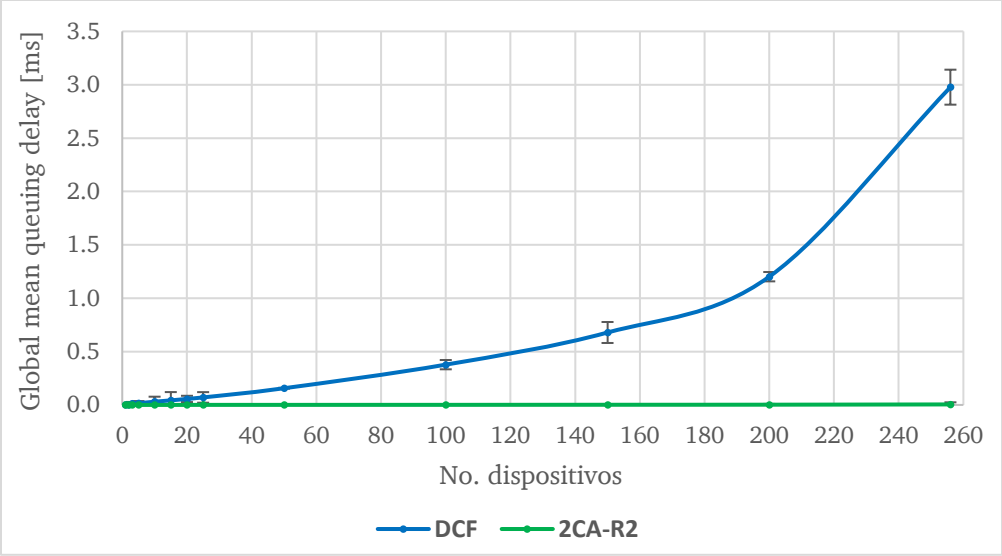

Figura 4.13. *Global mean queuing delay* con generación de paquetes mediante proceso de Poisson

Otro aspecto que se analizó bajo las condiciones de este escenario es el impacto de la variación en la multiplicidad de colisión que se presenta en los CRI al utilizar el protocolo 2CA-R^2. Este factor puede generar una estimación errónea de $\widehat{N}$ cuando el número de dispositivos contendientes en un CRI no es similar al número de transmisiones exitosas en el CRI anterior. En este sentido, la Tabla 4.4 resume algunos resultados de interés, en los casos de este escenario, donde la multiplicidad de colisión puede sobreestimarse o subestimarse. Estos resultados corresponden a la fracción promedio de CRI con 1, 2 y más de 2 dispositivos contendientes.

Tabla 4.4. Resultados relacionados con la variación en la multiplicidad de colisión

No. de dispositivos	Fracción de CRI´s con 1 dispositivo [%]	Fracción de CRI´s con 2 dispositivos [%]	Fracción de CRI´s con más de 2 dispositivos [%]
25	99.46	0.53	0.01
50	98.63	1.34	0.03
100	96.77	3.10	0.13
200	91.08	7.48	1.44
256	87.29	9.62	3.09

Como puede observarse, bajo las condiciones de este escenario, con generación de tráfico mediante el proceso de Poisson, un alto porcentaje de CRI del 2CA-R^2 comienzan con un solo dispositivo intentando transmitir (por lo tanto, se les concede acceso inmediato). Esta situación se mantiene

incluso al aumentar el número de dispositivos. Tomando como ejemplo el peor caso, que es cuando hay 256 dispositivos en el escenario, cerca del 87% de los CRI, en promedio, derivan en esa situación. Luego, la fracción promedio de CRI que comienzan con dos o más dispositivos es relativamente pequeña. Nuevamente tomando como referencia el caso cuando hay 256 dispositivos en el escenario, existen cerca del 9.6% de CRI en dicha circunstancia. Esto deriva en una variación mínima en la multiplicidad de colisión, y por tanto de su estimación $\widehat{N}$, la mayor parte del tiempo. Sin embargo, no se puede perder de vista que existen CRI que generan subestimación y sobreestimación en el cálculo de $\widehat{N}$.

Como se puede apreciar en la cuarta columna de la Tabla 4.4, en el peor caso de este escenario, es decir, cuando hay 256 dispositivos en él, existen más del 3% de CRI con una multiplicidad de colisión inicial mayor a 2. Una situación similar ocurre cuando hay 200 dispositivos en el escenario, en cuyo caso, ocurren más del 1.4% de CRI con una multiplicidad de colisión inicial mayor a 2. Estos eventos inevitablemente originan subestimación y sobreestimación en el cálculo de $\widehat{N}$, debido a la estrategia utilizada de estimación de orden 1. No obstante, es tan pequeña su ocurrencia durante el total del tiempo de simulación que su impacto en el *global mean access delay* es insignificante.

Aunado al mínimo impacto ocasionado por la baja ocurrencia de eventos generadores de subestimación y sobreestimación en el cálculo de $\widehat{N}$, en [48] se demostró que el algoritmo *Adaptive*-2C con estrategia de actualización por fase, el cual es la base del CRI del protocolo 2CA-R^2, es robusto en presencia de errores de estimación en el número de dispositivos en colisión. Como consecuencia, en caso de cambios abruptos en el número de dispositivos entre CRI consecutivos, no se espera una degradación significativa del rendimiento. Para corroborar esta afirmación, se realizó un conjunto de simulaciones adicionales en los cuales se calculó el tiempo promedio de transferencia del conjunto completo (T_S) al utilizar el protocolo 2CA-R^2, de forma similar a la realizada en la sección 4.4.3 (es decir, con un único paquete por dispositivo). Sin embargo, para cada número n de dispositivos en el escenario se proporcionó, de forma independiente, un valor $\widehat{N}$ con sobreestimación del 50%, con sobreestimación del 20% y con subestimación del 20%, con respecto del total. El objetivo fue comparar los valores obtenidos con el caso ideal, es decir, cuando el valor de $\widehat{N}$ corresponde al número n de dispositivos en el escenario. Para cada valor elegido de n se realizaron 15 simulaciones con un canal de transmisión ideal o libre de errores. La Tabla 4.5 muestra el porcentaje promedio adicional al tiempo de transferencia del conjunto completo para los distintos valores de n.

Tabla 4.5. Tiempo adicional a T_S con sobreestimación y subestimación de $\widehat{N}$

No. dispositivos	T_S con estimación ideal	% adicional (sobreestimación de 50%)	% adicional (sobreestimación de 20%)	% adicional (subestimación de 20%)
3	2.985	20.22	15.35	12.34
10	11.335	10.22	4.48	3.60
20	23.436	18.23	8.24	8.81
50	60.409	24.48	7.76	9.21
100	121.911	29.03	10.37	10.24
200	245.244	37.48	13.14	13.89
256	314.720	38.41	13.05	14.19

Se puede apreciar que, en un caso extremo, con $n = 256$ dispositivos en el escenario y una sobreestimación de $\widehat{N}$ del 50% , el sistema tarda, en promedio, poco más de un 38% adicional al tiempo que se requiere con una estimación precisa de la multiplicidad de colisión. Sin embargo, si esta sobreestimación es del 20% en condiciones similares del escenario, el tiempo promedio adicional es solamente de alrededor de 13%. Como se muestra en la quinta columna de la Tabla 4.5, un valor similar ocurre en caso de una subestimación del 20%, generándose un tiempo adicional de alrededor de 14%. Aunque algunos de estos valores son altos, principalmente cuando existe una sobreestimación de $\widehat{N}$ del 50%, el conjunto de simulaciones realizadas con tráfico de Poisson, muestran que la ocurrencia de estas variaciones es muy baja en un periodo de tiempo considerablemente largo, situación que minimiza su impacto en términos de *global mean access delay*.

Es importante enfatizar que la técnica de predicción empleada en este trabajo no forma parte del mecanismo 2CA-R^2 y que se pueden emplear muchas otras técnicas de predicción. Por lo tanto, en escenarios con requisitos más estrictos, es decir, con un tipo de tráfico de alta variabilidad, se puede emplear una técnica de estimación más precisa. Por ejemplo, un predictor de orden superior a 1, un predictor no lineal o un predictor basado en diferentes estrategias.

4.5. Validación del protocolo 2CA-R^2

Como se menciona al inicio de este capítulo, la evaluación del protocolo 2CA-R^2 se llevó a cabo exclusivamente a través de simulaciones con OMNeT++. Con el fin de validar estos resultados, se desarrolló un modelo matemático basado en una cadena de Markov. Los resultados obtenidos por ambos métodos fueron comparados entre sí como se reporta en esta sección.

El modelo que se presenta a continuación representa la evolución en el tiempo del CRI del protocolo 2CA-R^2. En otras palabras, representa el CRI del algoritmo *Adaptive*-2C con actualización por fase. En este modelo, cada estado de la cadena se identifica mediante el par ordenado (TC, WC), donde TC y WC indican el número de dispositivos en la celda de transmisión y en la celda de espera, respectivamente. La Figura 4.14 muestra el modelo donde los estados de la cadena se representan como círculos.

Las etiquetas de Inicio o «*Start*», en la Figura 4.14, indican los posibles estados iniciales de un CRI, dependiendo del número de dispositivos que colisionaron en el primer *minislot*. Es importante recordar que una colisión que involucra a todos los dispositivos activos marca el comienzo de una fase. Por lo tanto, la fase n comienza con una colisión de multiplicidad n, es decir, en el estado $(n, 0)$, y termina con una transmisión exitosa en el estado $(1, n - 1)$. Cuando la fase n termina, el sistema pasa a la fase $n - 1$, donde hay $n - 1$ dispositivos activos, es decir, en contienda.

Sean $A = (a, b)$ y $B = (c, d)$ dos estados de la cadena. Entonces, la probabilidad de transición del primero al segundo puede escribirse como $P_{A,B}$ o, más explícitamente, como $P_{(a,b),(c,d)}$. Sin embargo, con base en la siguiente observación y por razones de legibilidad, se utilizará una notación simplificada para las transiciones entre estados que pertenecen a la misma fase. Nótese que en cualquier fase n dada se cumple lo siguiente. Si hay i dispositivos en la celda de transmisión (con i en el intervalo $0 \leq i \leq n$), entonces debe haber $n - i$ dispositivos en la celda de espera. Por lo tanto, un estado en la fase n se puede identificar completamente utilizando solo un subíndice (es decir, i en este caso). Por lo tanto, denótese una transición del estado i al estado j de la fase n como $P_{i,j}^{(n)}$. La Figura 4.14 muestra las probabilidades de transición como etiquetas colocadas a lo largo de las flechas que conectan los estados.

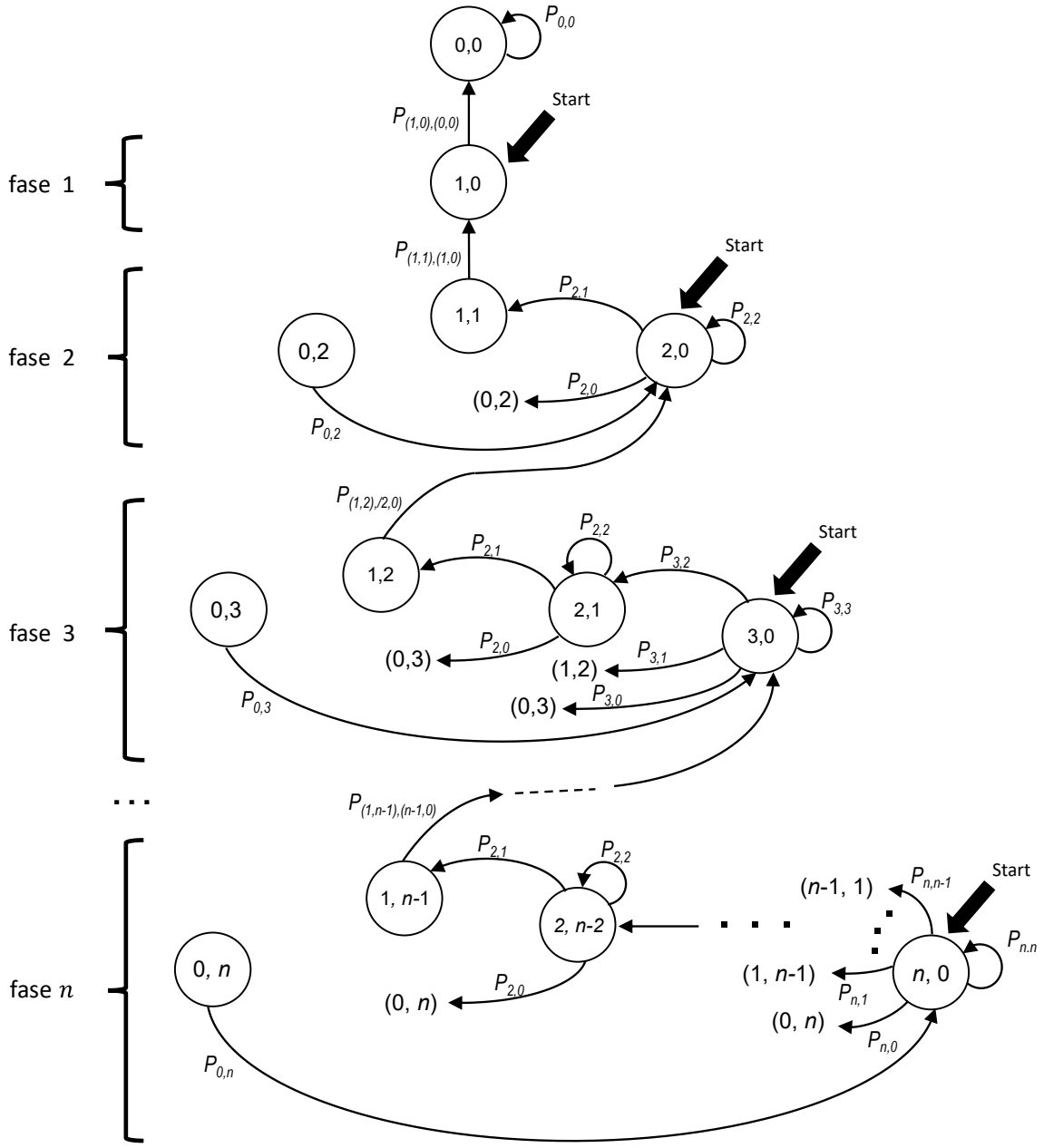

Figura 4.14. Modelo de cadena de Markov del algoritmo *Adaptive-2C* con actualización por fase

Con el propósito de ofrecer mayor claridad, la Figura 4.15 muestra los posibles eventos de un CRI al inicio de la fase $n = 3$, vinculándolos con la notación de la cadena de Markov que los representa. Como puede apreciarse, la Figura 4.15 (a) muestra el primer *minislot* de la fase, en la cual ocurre una colisión de multiplicidad 3. Es decir, en ese punto de la fase hay 3 dispositivos intentando transmitir ($TC = 3$) y ningún dispositivo en espera ($WC = 0$). Por esta razón, el estado que identifica el primer *minislot* de esta fase es $(3,0)$ y se representa en el modelo de la

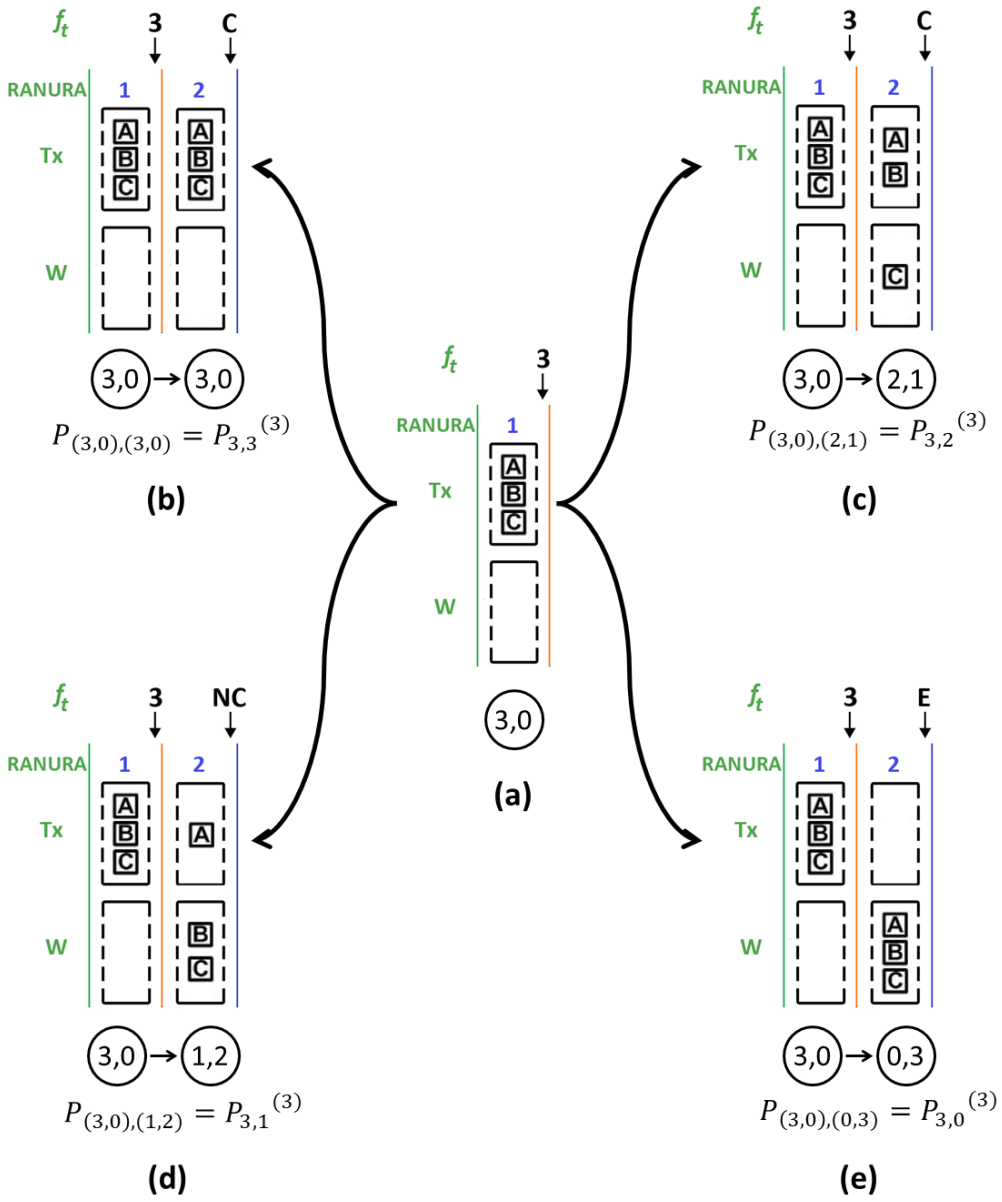

Figura 4.15. Evolución de CRI al inicio de la fase $n = 3$ con notación de cadena de Markov

cadena de Markov encerrado en un círculo. El segundo *minislot* de esta fase tiene cuatro diferentes posibilidades de ocurrencia, mismas que se ilustran en los casos (b), (c), (d) y (e) de la Figura 4.15. Tomando como ejemplo (c), caso en el cual dos de los tres dispositivos permanecen en la celda de transmisión ($TC = 2$) y el tercero de ellos pasa a la celda de espera ($WC = 1$), el estado de la cadena de Markov que identifica al segundo *minislot* es (2,1). De esta forma, el sistema pasa del estado (3,0) al estado (2,1) y se representa en el modelo de la cadena de Markov con ambos estados encerrados entre círculos y vinculados por medio de una flecha en el sentido de la ocurrencia de los eventos. La

probabilidad de que esta transición ocurra se representa en el modelo como $P_{(3,0),(2,1)}$ o bien en su versión simplificada como ${P_{(3,2)}}^{(3)}$. Nótese que en la versión simplificada se utilizan como subíndice únicamente los valores de la celda de transmisión TC, en los estados de origen y destino, y que el superíndice indica la fase en que se encuentra el sistema. Este mismo criterio se utiliza para explicar los casos (b), (d) y (e) de la Figura 4.15.

Por otra parte, en este análisis es conveniente distinguir entre tres tipos de transiciones. Una transición de tipo 0 ocurre cuando se parte desde el estado inicial de una fase. A su vez, una transición de tipo 1 se origina en un estado no inicial de una fase. Finalmente, una transición de tipo 2 es la que ocurre con probabilidad 1. Esta última situación ocurre en los dos casos siguientes: a) cuando se alcanza el estado $(0, n)$, que indica la ocurrencia de un *minislot* vacío, la única posibilidad es transitar al estado inicial de la misma fase, es decir, $(n, 0)$; y b) cuando se alcanza el estado $(1, n - 1)$, que indica la transmisión exitosa de una solicitud, la única posibilidad es transitar al estado inicial de la siguiente fase, es decir $(n - 1, 0)$. Como ejemplo de los tipos de transición, la Figura 4.16 (a) muestra todas las transiciones correspondientes a la fase $n = 3$. Adicionalmente, se pueden identificar los siguientes tres tipos de transiciones:

- Transiciones de tipo 0. La Figura 4.16 (b) muestra todas las transiciones posibles partiendo desde el estado inicial $(3, 0)$. Estas se identifican con letras en color azul.

- Transiciones de tipo 1. La Figura 4.16 (c) muestra todas las transiciones posibles a partir de $(2, 1)$, que no es un estado inicial. Estas se identifican con letras en color púrpura.

- Transiciones de tipo 2. La Figura 4.16 (d) muestra los casos con una única posibilidad de transición. Estas se identifican con letras en color rojo.

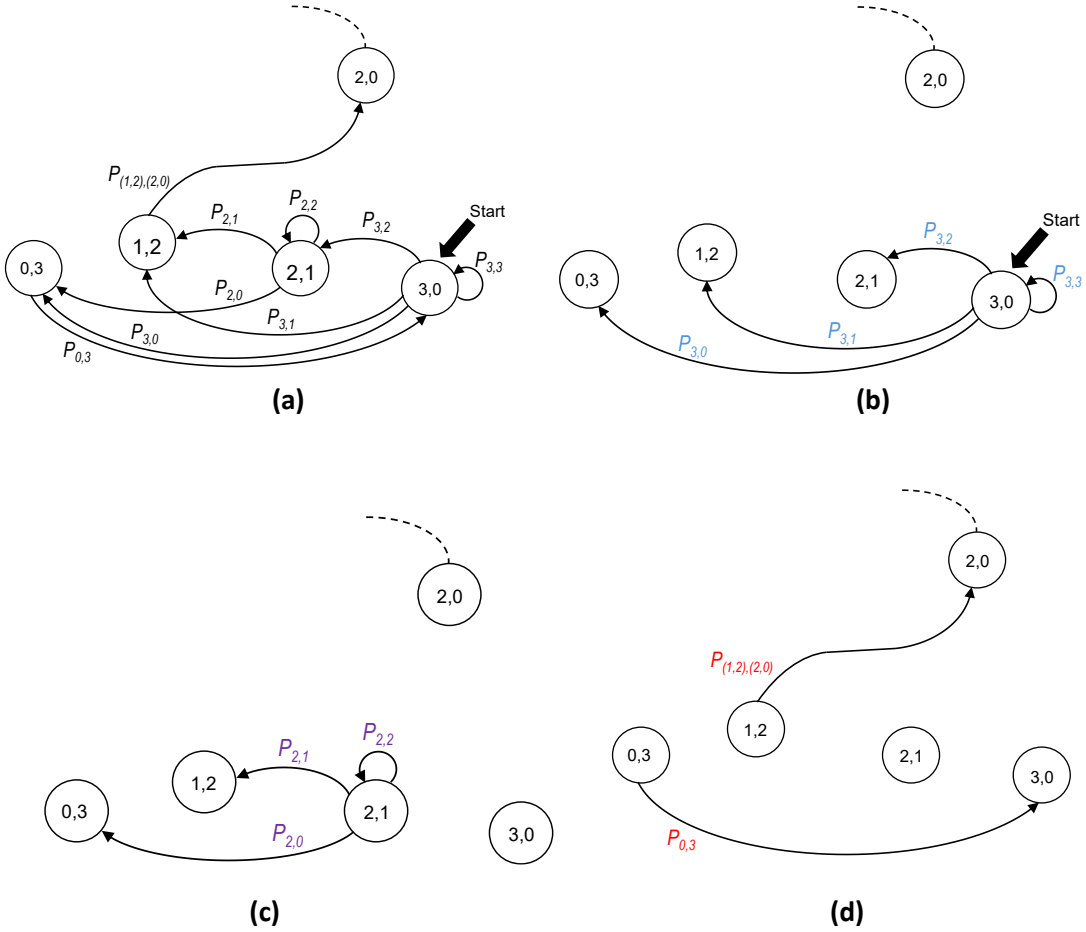

Figura 4.16. Posibles transiciones correspondientes a un CRI en la fase $n = 3$

En resumen, el tipo de transición depende del tipo de estado de origen. Por lo tanto, es conveniente observar que un estado que pertenece a la fase n puede clasificarse en una de las siguientes tres posibilidades: a) el estado inicial, es decir $(n, 0)$; b) estados no iniciales, que pertenecen al conjunto $\{(i, n - i) | 2 \leq i \leq n - 1\}$; y c) estados con un solo destino posible, es decir $(0, n)$ y $(1, 0)$. Tómese en consideración esta clasificación para calcular la probabilidad de las transiciones que parten de un estado en fase n (con $n \geq 2$), de la siguiente manera:

- Las transiciones de tipo 0, desde el estado inicial $(n, 0)$ a $(n - j, j)$, se calculan de acuerdo con la ecuación (4.13)

$$P_{n,n-j}^{(n)} = \binom{n}{n-j} p^{(n-j)}(1-p)^{j}, \tag{4.13}$$

donde

$$j = 0, \ldots, n; \;\; p = \begin{cases} \dfrac{1}{2} & si \;\; n = 2, \\ \dfrac{\sqrt{2n(n-1)} - 2}{(n-2)(n+1)} & si \; n > 2. \end{cases}$$

- Las transiciones de tipo 1, desde un estado no inicial $(i, n - i)$ a $(j, n - j)$, se calculan de acuerdo con la ecuación (4.14)

$$P_{i,j}^{(n)} = \binom{i}{j} p^j (1-p)^{(i-j)}, \tag{4.14}$$

donde

$$2 \leq i \leq n - 1; \;\; j = 0, \ldots, i; \; p = \frac{1}{2}.$$

- La transición de tipo 2, de $(0, n)$ a $(n, 0)$, ocurre después de un *minislot* vacío. En esta, todos los dispositivos en la celda de espera pasan a la celda de transmisión y se calcula con base en la ecuación (4.15)

$$P_{0,n}^{(n)} = 1. \tag{4.15}$$

- La transición tipo 2, de $(1, n - 1)$ a $(n - 1, 0)$, ocurre después de la transmisión exitosa de una solicitud. En esta, el CRI evoluciona desde la fase n a la fase $n - 1)$ y se calcula con base en la ecuación (4.16)

$$P_{(1,n-1),(n-1,0)} = 1. \tag{4.16}$$

Nótese que en este caso no se puede utilizar la notación simplificada ya que el origen y el destino pertenecen a fases diferentes.

La última fase de la cadena de Markov comienza cuando el sistema alcanza el estado $(1, 0)$ y el dispositivo restante puede transmitir sin contienda. En este caso, el sistema pasa al estado $(0, 0)$ y el CRI termina. Por este motivo, este estado se considera un estado absorbente en la cadena. Las probabilidades de transición descritas anteriormente se utilizan para crear la matriz de transición $\mathbf{P}$, que representa a un CRI completo. La Figura 4.17 ilustra la forma general de esta matriz.

	n,0	n-1,1	. . .	0,n	. . .	3,0	2,1	1,2	0,3	2,0	1,1	0,2	1,0	0,0
n,0	$P^{(n)}_{n,n}$	$P^{(n)}_{n,n-1}$	. . .	$P^{(n)}_{n,0}$	. . .	0	0	0	0	0	0	0	0	0
n-1,1	0	$P^{(n)}_{n-1,n-1}$	. . .	$P^{(n)}_{n-1,0}$	. . .	0	0	0	0	0	0	0	0	0
.	.	.	. . .	.	. . .	.	.	.	.	.	.	.	.	.
.	.	.	. . .	.	. . .	.	.	.	.	.	.	.	.	.
.	.	.	. . .	.	. . .	.	.	.	.	.	.	.	.	.
0,n	$P^{(n)}_{0,n}$	0	. . .	0	. . .	0	0	0	0	0	0	0	0	0
.	.	.	. . .	.	. . .	.	.	.	.	.	.	.	.	.
.	.	.	. . .	.	. . .	.	.	.	.	.	.	.	.	.
.	.	.	. . .	.	. . .	.	.	.	.	.	.	.	.	.
3,0	0	0	. . .	0	. . .	$P^{(3)}_{3,3}$	$P^{(3)}_{3,2}$	$P^{(3)}_{3,1}$	$P^{(3)}_{3,0}$	0	0	0	0	0
2,1	0	0	. . .	0	. . .	0	$P^{(3)}_{2,2}$	$P^{(3)}_{2,1}$	$P^{(3)}_{2,0}$	0	0	0	0	0
1,2	0	0	. . .	0	. . .	0	0	0	0	$P_{(1,2),(2,0)}$	0	0	0	0
0,3	0	0	. . .	0	. . .	$P^{(3)}_{0,3}$	0	0	0	0	0	0	0	0
2,0	0	0	. . .	0	. . .	0	0	0	0	$P^{(2)}_{2,2}$	$P^{(2)}_{2,1}$	$P^{(2)}_{2,0}$	0	0
1,1	0	0	. . .	0	. . .	0	0	0	0	0	0	0	$P_{(1,1),(1,0)}$	0
0,2	0	0	. . .	0	. . .	0	0	0	0	$P^{(2)}_{0,2}$	0	0	0	0
1,0	0	0	. . .	0	. . .	0	0	0	0	0	0	0	0	$P_{(1,0),(0,0)}$
0,0	0	0	. . .	0	. . .	0	0	0	0	0	0	0	0	$P^{(0)}_{0,0}$

Figura 4.17. Forma general de la matriz de transición $\mathbf{P}$

Para lograr una mayor claridad, en la Figura 4.17, las transiciones de tipo 0 se muestran en color azul, mientras que las transiciones de tipo 1 se muestran en color púrpura, siendo consistentes con la Figura 4.16. A su vez, las transiciones de tipo 2 se muestran en color rojo. Con el objetivo de alcanzar una mejor compresión conceptual, a continuación, se muestra la matriz de transición $\mathbf{P}$ para un CRI con una fase inicial $n = 4$, primero con la notación general en la Figura 4.18 y después con los valores numéricos correspondientes en la Figura 4.19.

	4,0	3,1	2,2	1,3	0,4	3,0	2,1	1,2	0,3	2,0	1,1	0,2	1,0	0,0
4,0	$P^{(4)}_{4,4}$	$P^{(4)}_{4,3}$	$P^{(4)}_{4,2}$	$P^{(4)}_{4,1}$	$P^{(4)}_{4,0}$	0	0	0	0	0	0	0	0	0
3,1	0	$P^{(4)}_{3,3}$	$P^{(4)}_{3,2}$	$P^{(4)}_{3,1}$	$P^{(4)}_{3,0}$	0	0	0	0	0	0	0	0	0
2,2	0	0	$P^{(4)}_{2,2}$	$P^{(4)}_{2,1}$	$P^{(4)}_{2,0}$	0	0	0	0	0	0	0	0	0
1,3	0	0	0	0	0	$P_{(1,3),(3,0)}$	0	0	0	0	0	0	0	0
0,4	$P^{(4)}_{0,4}$	0	0	0	0	0	0	0	0	0	0	0	0	0
3,0	0	0	0	0	0	$P^{(3)}_{3,3}$	$P^{(3)}_{3,2}$	$P^{(3)}_{3,1}$	$P^{(3)}_{3,0}$	0	0	0	0	0
2,1	0	0	0	0	0	0	$P^{(3)}_{2,2}$	$P^{(3)}_{2,1}$	$P^{(3)}_{2,0}$	0	0	0	0	0
1,2	0	0	0	0	0	0	0	0	0	$P_{(1,2),(2,0)}$	0	0	0	0
0,3	0	0	0	0	0	$P^{(3)}_{0,3}$	0	0	0	0	0	0	0	0
2,0	0	0	0	0	0	0	0	0	0	$P^{(2)}_{2,2}$	$P^{(2)}_{2,1}$	$P^{(2)}_{2,0}$	0	0
1,1	0	0	0	0	0	0	0	0	0	0	0	0	$P_{(1,1),(1,0)}$	0
0,2	0	0	0	0	0	0	0	0	0	$P^{(2)}_{0,2}$	0	0	0	0
1,0	0	0	0	0	0	0	0	0	0	0	0	0	0	$P_{(1,0),(0,0)}$
0,0	0	0	0	0	0	0	0	0	0	0	0	0	0	$P^{(0)}_{0,0}$

Figura 4.18. Ejemplo de la matriz de transición $\mathbf{P}$ para $n = 4$, con notación general

	4,0	3,1	2,2	1,3	0,4	3,0	2,1	1,2	0,3	2,0	1,1	0,2	1,0	0,0
4,0	0.007	0.069	0.254	0.415	0.254	0	0	0	0	0	0	0	0	0
3,1	0	0.125	0.375	0.375	0.125	0	0	0	0	0	0	0	0	0
2,2	0	0	0.25	0.5	0.25	0	0	0	0	0	0	0	0	0
1,3	0	0	0	0	0	1	0	0	0	0	0	0	0	0
0,4	1	0	0	0	0	0	0	0	0	0	0	0	0	0
3,0	0	0	0	0	0	0.05	0.25	0.44	0.25	0	0	0	0	0
2,1	0	0	0	0	0	0	0.25	0.5	0.25	0	0	0	0	0
1,2	0	0	0	0	0	0	0	0	0	1	0	0	0	0
0,3	0	0	0	0	0	1	0	0	0	0	0	0	0	0
2,0	0	0	0	0	0	0	0	0	0	0.25	0.5	0.25	0	0
1,1	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0,2	0	0	0	0	0	0	0	0	0	1	0	0	0	0
1,0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0,0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Figura 4.19. Ejemplo numérico de la matriz de transición $\mathbf{P}$ para $n = 4$

Con la información que proporciona la matriz de transición $\mathbf{P}$ se pueden calcular varias medidas de desempeño. Particularmente, es de interés calcular la duración promedio del CRI. Para ello, después

de aplicar algunos resultados conocidos de la teoría de cadenas de Markov, es posible obtener los tiempos de llegada o *hitting times* [56] resolviendo el sistema de ecuaciones lineales que se muestra en la ecuación (4.17)

$$H_{A,B} = \begin{cases} 0 & \text{para } A = B \\ 1 + \sum_{I \neq B} P_{A,I} H_{I,B} & \text{para } A \neq B, \end{cases} \tag{4.17}$$

donde A, B e I son tres estados de la cadena y $H_{A,B}$ es el número esperado de *minislots* necesarios para alcanzar el estado B partiendo del estado A. Del conjunto de resultados obtenidos al resolver el sistema de ecuaciones (4.17), es de interés conocer $H_{A,B}$ donde $A = (n, 0)$ y $B = (0, 0)$ para diferentes valores de n. Esto se debe al hecho de que un CRI comienza con una colisión de multiplicidad n, es decir, en el estado $(n, 0)$ en la fase n, y termina cuando el sistema alcanza el estado absorbente, es decir, el estado $(0, 0)$.

Con el objetivo de calcular el número promedio de *minislots* (ms_n), que se necesitan para resolver una colisión inicial de tamaño n, se utilizaron dos diferentes herramientas de software a fin de resolver el sistema de ecuaciones lineales implicado por la ecuación (4.17). Los resultados se compararon con los valores obtenidos mediante la simulación de eventos discretos con OMNeT++, mismos que se muestran en la Tabla 4.6 y que se explican a continuación.

Tabla 4.6. Promedio de *minislots* necesarios para resolver una colisión de tamaño n utilizando 2CA-R[2]

n	OMNeT++ (ms_n)	Matlab (ms_n)	Python (ms_n)
2	4.57	4.50	4.50
3	8.34	8.25	8.26
5	16.09	16.05	16.06
10	36.04	36.20	36.24
15	56.59	56.70	56.76
20	76.64	77.35	77.43
30	119.80	118.86	118.99
50	202.43	202.27	202.49
100	411.37	411.58	412.02
150	622.10	621.27	621.93
200	831.21	831.12	832.01
256	1068.79	ND*	1067.40

*ND - No Disponible

- La primera columna corresponde a los resultados obtenidos mediante simulaciones por computadora. Para ello se utilizó el simulador OMNeT++. Cada valor mostrado en esta columna proviene de un promedio de 1000 ejecuciones de simulación.
- La segunda columna muestra los resultados obtenidos a partir de un *script* realizado en Matlab que encuentra la solución de la norma mínima no negativa del sistema de ecuaciones lineales [57]. El caso indicado como ND, para $n = 256$, surgió por limitaciones del software con respecto al uso de matrices.
- La última columna muestra los resultados obtenidos con un *script* realizado en Python que aplica la teoría de cadenas de Markov para calcular el tiempo medio antes de la absorción [58].

Como se puede observar en la Tabla 4.6, el número de *minislots* (ms_n) necesarios para resolver una colisión de tamaño n en el protocolo 2CA-R^2, obtenidos mediante simulación en OMNeT++, es prácticamente el mismo que el obtenido mediante la cadena de Markov, lo que indica que el modelo matemático presentado aquí representa con precisión el modelo de simulación utilizado en condiciones ideales (es decir con un canal libre de errores).

En este punto es necesario hacer una observación. A fin de ilustrar la situación, considérese el ejemplo de la Figura 4.16, que muestra las transiciones en la fase $n = 3$. Nótese que, al estado $(3,0)$ se puede llegar por cuatro diferentes trayectorias.

a) El CRI pudo haber iniciado con una contienda de tres estaciones. En el modelo, el estado $(3,0)$ es considerado en este caso como estado inicial, tanto del CRI como de la fase 3.
b) Si la fase 4 concluye, el sistema transita desde el estado $(1,3)$ al estado $(3,0)$. En este caso el estado $(3,0)$ es considerado en el modelo como el estado inicial de la fase 3.
c) Después de una ranura vacía en la fase 3 todas las estaciones regresan al estado de transmisión, es decir del estado $(0,3)$ se transita al $(3,0)$. Dado este caso, el estado $(3,0)$ es también considerado en el modelo como el estado inicial de la fase 3.
d) Por último, se puede transitar al estado $(3,0)$ desde él mismo, si las tres estaciones en transmisión deciden permanecer en ese estado. Sin embargo, obsérvese que en este caso el estado $(3,0)$ será visitado por segunda vez consecutiva, el modelo seguirá considerándolo como el estado inicial de la fase 3.

Nótese que, en cualquiera de los cuatro casos anteriores, la siguiente transición será de tipo 0 (desde el estado inicial), lo cual está correctamente considerado en el modelo para los tres primeros casos, pero esto es incorrecto en el último caso (d), ya que el estado (3,0) no debería ser tomado como estado inicial de la fase si se visita dos o más veces consecutivas. Este hecho es una imprecisión del modelo.

Recordemos que en el algoritmo *Adaptive*-2C con estrategia de actualización por fase, aplicado en el protocolo 2CA-R[2], el punto de acceso solo es capaz de proporcionar la estimación de multiplicidad de colisión $\widehat{N}$ a los dispositivos después de la primera colisión del CRI en la fase n, de forma que, idealmente, $\widehat{N}$ sea igual a n. Este valor será utilizado por los dispositivos para calcular su probabilidad de transmisión $p = P(N)$ (donde $N \leftarrow \widehat{N}$), únicamente después de la primera colisión de cada fase o después de una indicación de ranura vacía. Además, N será actualizado, decrementando su valor en 1, al ocurrir una transmisión exitosa. En caso de que existan dos o más colisiones consecutivas dentro de cualquier fase n en el CRI, los dispositivos deben usar una probabilidad de transmisión $p = 0.5$, a partir de la segunda colisión, ya que es imposible para ellos saber cuántos dispositivos hay en la celda de transmisión y cuántos en la celda de espera.

Para relacionar los eventos ocurridos al usar el algoritmo con el modelo de la cadena Markov, tómese como referencia el ejemplo de la Figura 4.20. En esta se observa la operación del algoritmo *Adaptive*-2C en la situación ya descrita. Se aprecia que, en el tercer *minislot*, es decir, en la primera colisión de la fase 3, están involucrados los 3 dispositivos, denotados por A, B y D. Representando este evento mediante la cadena de Markov, esta se encontraría en el estado $(3,0)$. Luego, al ser la primera colisión de la fase 3, continuarán en la celda de transmisión con una probabilidad $p = P(N)$ o pasarán a la celda de espera con una probabilidad $1 - p$. Desde el punto de vista de la cadena de Markov, las probabilidades de transición desde el estado inicial de la fase $(3,0)$, se calcularán a través de la ecuación (4.13) que involucra a $p = P(N)$.

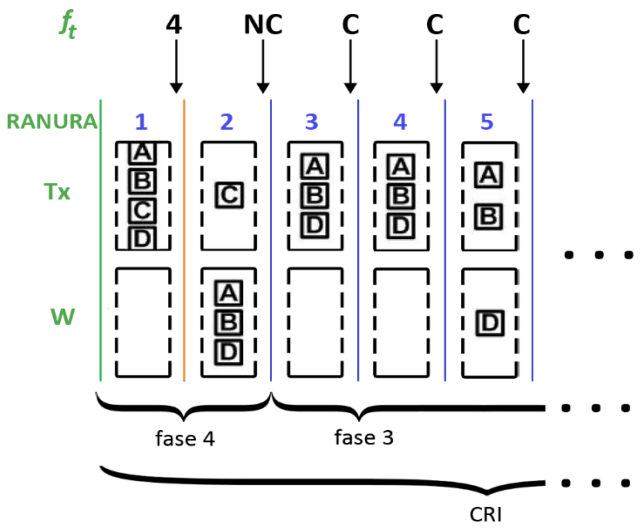

Figura 4.20. Ejemplo de CRI con transición del estado $(3,0)$ al estado $(3,0)$

En el ejemplo, los tres dispositivos A, B y D determinaron continuar en la celda de transmisión y sus paquetes vuelven a colisionar, lo cual se ilustra en el cuarto *minislot*. Nuevamente, representando este evento mediante la cadena de Markov, la transición de tipo 0 ocurrida en este punto fue aquella con probabilidad $P_{3,3}$, lo cual significa que se retornará al estado $(3,0)$. En ese instante, a los dispositivos solo se les proporciona un mensaje de retroalimentación C que indica una colisión, por lo cual no pueden saber cuántos dispositivos hay en la celda de transmisión y cuántos pasaron a la celda de espera. Por esa razón, el algoritmo indica que los dispositivos determinarán su permanencia en la celda de transmisión con una probabilidad $p = 0.5$ o su paso a la celda de espera con una probabilidad $1 - p$. No obstante, desde el punto de vista de la cadena de Markov, al haber regresado al estado $(3,0)$, es decir, al estado inicial de la fase 3, las probabilidades de transición volverán a calcularse a través de la ecuación (4.13) que involucra a $p = P(N)$, sin importar que el retorno estuviera precedido por una colisión. Esto es una discrepancia de la cadena de Markov con el algoritmo que, para evitarse, requeriría un sistema con memoria de más de un evento anterior, rompiendo así con la propiedad de Markov. Esta discrepancia se presenta en cada transición con probabilidad $P_{n,n}$ del estado $(n, 0)$ de cualquier fase $n \geq 2$ del CRI. Sin embargo, se ideó una solución que conservó la propiedad de Markov a costa de incrementar el número de estados en la cadena.

La solución implementada se ilustra en la Figura 4.21, en donde se muestran todas las transiciones correspondientes a la fase $n = 3$, de forma similar a la Figura 4.16 (a). Sin embargo, aquí se incorpora un estado adicional, el cual es un estado «fantasma» del estado inicial de la fase $(3,0)$, y se denota

como $(\widehat{3,0})$. Aquí, es posible notar que dicho estado «fantasma» no es un estado inicial de la fase y se llega a él desde el estado $(3,0)$ con una probabilidad de transición $P_{3,3}$. Obsérvese que esta transición es de tipo 0, la cual, al relacionarla con el ejemplo de la Figura 4.20, ocurriría tras la recepción del mensaje de colisión C al final del tercer *minislot*. Así, este estado «fantasma» $(\widehat{3,0})$ representa la colisión en el cuarto *minislot* del ejemplo de la Figura 4.20, ya que, de nueva cuenta hay 3 dispositivos en la celda de transmisión y ninguno en la celda de espera. No obstante, al no ser un estado inicial, su inclusión permite que las probabilidades de transición subsecuentes sean de tipo 1 y por tanto se calculen con un valor de $p = 0.5$, tal como debe de ocurrir. Esto se representa en la Figura 4.21 al colocar las posibles transiciones desde el estado «fantasma» $(\widehat{3,0})$ en color púrpura.

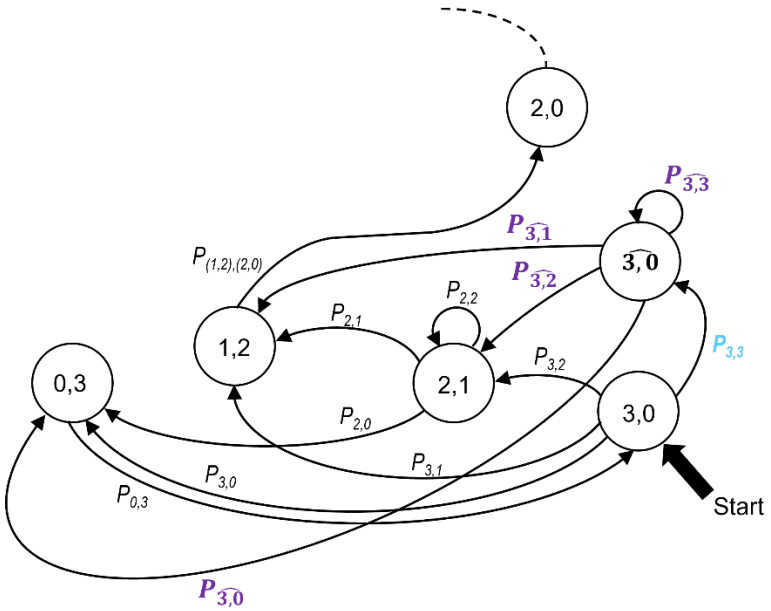

Figura 4.21. Posibles transiciones de la fase $n = 3$ con estado adicional

Como es de esperarse, este ajuste se realizó en cada fase del CRI en la cadena de Markov. Es decir, a cada estado inicial $(n, 0)$, para $n > 2$, se añadió un estado «fantasma» $(\widehat{n, 0})$. Es importante precisar que para la fase $n = 2$ no es necesario añadir un estado adicional ya que en este caso las probabilidades de transición siempre se basan en un valor de $p = 0.5$, mientras que para el caso de la fase $n = 1$, solo hay una probabilidad de transición.

Con esta nueva consideración, se desarrolló la versión extendida de la cadena de Markov, es decir, la versión que incluye los estados «fantasma» iniciando en una fase $n = 256$. Luego, se calcularon todas las probabilidades de transición basándose en las ecuaciones (4.13)(4.14)(4.15) y (4.16) y se construyó la matriz de transición $\mathbf{P}$ que representaba al CRI completo. Finalmente, se determinó el número

promedio de *minislots* (ms_n), que se necesitan para resolver una colisión inicial de tamaño n, a través del cálculo de los *hitting times*, mediante la resolución del sistema de ecuaciones lineales mostrado en (4.17). Este cálculo se realizó mediante un *script* realizado en Python.

La Tabla 4.7 muestra una comparación del número promedio de *minislots* (ms_n), que se necesitan para resolver una colisión inicial de tamaño n, obtenidos de la versión Simple de la Cadena de Markov (CMS) sin estados «fantasma» y de la versión Extendida de la Cadena de Markov (CME).

Tabla 4.7. Comparación de ms_n entre cadenas de Markov extendida y simple

n	ms_n CMS	ms_n CME
2	4.5049	4.5049
3	8.2567	8.2589
5	16.0645	16.0695
10	36.2417	36.2463
15	56.7634	56.7670
20	77.4325	77.4351
30	118.9883	118.9888
50	202.4878	202.4915
100	412.0167	412.0226
150	621.9263	621.9385
200	832.0123	832.0190
256	1067.3974	1067.4039

Como puede apreciarse, la diferencia entre las versiones simple y extendida de la cadena de Markov, en términos del número promedio de *minislots*, es despreciable. Esto indica que, a pesar de que la cadena de Markov simple (sin estados «fantasma») presenta una imprecisión en las probabilidades de transición en cada fase del CRI, representa de forma adecuada la evolución en el tiempo del CRI del protocolo 2CA-R^2. La razón de ello se encuentra en el valor que adquiere cada probabilidad de transición $P_{n,n}$ (para $n > 2$,) en el CRI. La Tabla 4.8 muestra los valores de las probabilidades de transición $P_{n,n}$ para las fases $n = 3,4,5,6,7$. En esta se puede observar que, en la fase n, la probabilidad de que un estado inicial $(n, 0)$ retorne a él mismo es pequeña. Además, se aprecia que esta probabilidad disminuye por un factor de 10 conforme aumenta el valor de n. Este comportamiento es consistente con los resultados presentados en [48], en dónde se comprobó que la ecuación (3.2), utilizada en el protocolo 2CA-R^2, optimiza el número de dispositivos contendientes la colisión inicial de una fase. Este valor óptimo, idealmente, permite que solo quede un dispositivo en la celda de

transmisión y el resto pase a la celda de espera. Por esta razón la probabilidad de transición más pequeña desde el estado inicial $(n, 0)$ en la fase n de la cadena de Markov, siempre será aquella que represente la permanencia en la celda de transmisión de todos los n dispositivos involucrados en una colisión inicial de la fase n, es decir, $P_{n,n}$.

Tabla 4.8. Valores de $P_{n,n}$ para las fases $n = 3$ hasta $n = 7$

Fase n	$P_{n,n}$
3	4.90×10^{-2}
4	7.10×10^{-3}
5	8.00×10^{-4}
6	7.47×10^{-5}
7	5.92×10^{-6}

Debido a la diferencia despreciable al usar la versión extendida de la cadena de Markov, con respecto a la versión simple sin estados «fantasma», se decidió presentar esta última como modelo matemático del CRI del protocolo 2CA-R^2, no sin antes precisar la discrepancia ya mencionada.

Por último, a partir de los valores mostrados en la Tabla 4.6, el *mean network throughput* en Mbps se puede calcular fácilmente si se conocen tanto el tamaño de los paquetes como la tasa de datos. Así, este cálculo se realiza de acuerdo con la ecuación (4.18)

$$Th_n = \left(\frac{(65)(8)(n)(1 \times 10^{-6})}{(ms_n)(1.68 \times 10^{-4}) + (n)(5.28 \times 10^{-4})} \right), \tag{4.18}$$

donde 1.68×10^{-4} es el tiempo (en segundos) necesario para enviar un RREQ de 20 bytes a 1 Mbps (es decir, la duración temporal de un $minislot$) y recibir un mensaje de retroalimentación de 1 byte desde el punto de acceso. A su vez, 5.28×10^{-4} es el tiempo (en segundos) necesario para enviar y reconocer un paquete de datos con una carga útil de 65 bytes a 1 Mbps. Este tiempo es la suma del tiempo de transmisión más el tiempo requerido para recibir el mensaje de retroalimentación correspondiente de 1 byte desde el punto de acceso.

La Figura 4.22 muestra una comparación entre los resultados obtenidos por medio de la relación (4.18) y los valores promedio obtenidos a partir de 1000 muestras de OMNeT++, bajo las mismas condiciones. Vale la pena mencionar que ambas curvas son indistinguibles entre sí porque se superponen. Este resultado verifica la validez del modelo matemático desarrollado. Dado que el modelo de Markov

supone un canal libre de errores, para fines de comparación también se incluyen resultados considerando tasas de error de paquete (PER) del 1% y 5%. Es decir, el escenario se configuró con un BER $= 1.9327 \times 10^{-5}$ para obtener un PER ≈ 0.01, y posteriormente con un BER $= 9.8636 \times 10^{-5}$ para obtener un PER ≈ 0.05, de acuerdo con la ecuación (4.11).

Dichos resultados muestran el efecto de los errores de canal en el desempeño pronosticado por el modelo. Todos los resultados experimentales se presentan utilizando intervalos de confianza con un nivel de confianza del 95%.

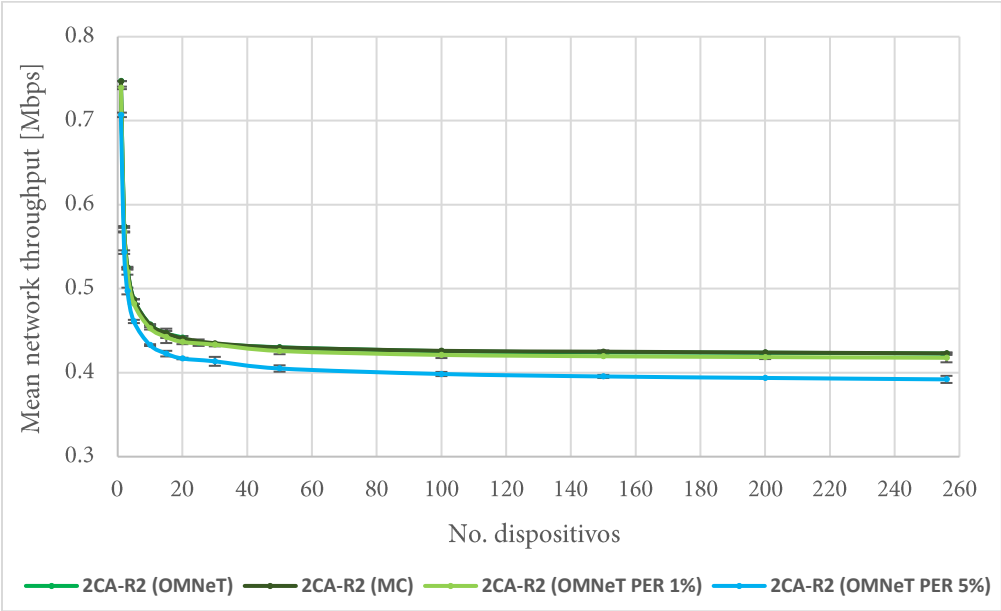

Figura 4.22. Comparación de *mean network throughput* entre los resultados de simulación y cadena de Markov

CAPÍTULO 5

Conclusiones y Perspectivas de investigación

5.1. Conclusiones

Se hizo una revisión documental de los protocolos MAC de contienda ALOHA, slotted ALOHA y CSMA/CA, incluyendo las variantes e implementación DCF de este último, así como de los protocolos de reserva TDMA, FDMA y CDMA. A su vez, también se realizó una revisión documental de los protocolos MAC para comunicaciones M2M, empleados en los estándares destinados a respaldar las aplicaciones del IoT. En esta revisión se identificó que todos estos protocolos tienen como base un protocolo, ya sea de contienda o de reserva, o bien en un protocolo híbrido conformado por ambos tipos. Posteriormente, se analizaron nueve diferentes protocolos MAC híbridos para comunicaciones M2M donde el ciclo de transmisión tuviera una estructura similar a la del protocolo 2CA-R^2, es decir, que tuviera al menos un intervalo de contienda y un intervalo de transmisión de datos. Como resultado de esta búsqueda bibliográfica se encontró que, al igual que con los protocolos utilizados en los estándares, estos mecanismos se basan en variantes de los mecanismos ALOHA o CSMA/CA durante su intervalo de contienda.

La propuesta de protocolo MAC híbrido que se presenta en este trabajo expande las opciones disponibles para comunicaciones M2M y genera una ventana de oportunidades para la utilización de distintas estrategias de control de acceso al medio, ya que se utiliza un enfoque distinto al de aplicar variantes de los mecanismos ALOHA o CSMA/CA en el intervalo de contienda de un protocolo híbrido. A través de la implementación del algoritmo *Adaptive*-2C se demostró que una estrategia simple puede ser eficaz y eficiente en la solución de colisiones que se presentan en un medio compartido por una gran cantidad de dispositivos de transmisión.

Gracias a los resultados obtenidos en la evaluación de desempeño del protocolo 2CA-R^2 propuesto, se pudo observar que su aplicación en el contexto de las comunicaciones M2M propicia ventajas como:

- Incremento en el *mean throughput network* con fuentes *greedy*. En los dos escenarios en los que se consideró generación de tráfico *greedy* (es decir, con un canal libre de errores y con una tasa de perdida de paquetes cercana al 1%), se comprobó que, bajo las condiciones establecidas en las simulaciones, el *mean throughput network* al utilizar el protocolo 2CA-R^2, es mayor en comparación con DCF. Además, también se demostró que cuando hay más de 20 dispositivos contendientes, su desempeño se mantiene estable a medida que aumenta el número de dispositivos.

- Disminución del *global mean access delay* con fuentes *greedy* cuando no hay desecho de paquetes. En los dos escenarios en los que se consideró generación de tráfico *greedy* (es decir, con un canal libre de errores y con una tasa de perdida de paquetes cercana al 1%), se comprobó que, bajo las condiciones establecidas en las simulaciones, el *global mean access delay* del protocolo 2CA-R^2 es menor, en comparación con DCF, cuando no existe desecho de paquetes. Esto es, cuando el canal presenta un número reducido de colisiones y ningún paquete alcanza el límite de reintentos de transmisión en DCF, el *global mean access delay* al utilizar el protocolo 2CA-R^2, es menor comparado con DCF. Sin embargo, cuando el número de colisiones que se presentan en el canal es elevado, el parámetro *Retry Limit* de DCF favorece a un incremento controlado del *global mean access delay* en el sistema al desechar un porcentaje cada vez más elevado de paquetes, conforme se incrementa el número de dispositivos (de hasta 16% del total de paquetes transmitidos cuando hay 256 dispositivos contendientes). Por el contrario, el protocolo 2CA-R^2, presenta un incremento más acelerado del *global mean access delay* en el sistema (con una pérdida de hasta 60 ms, con respecto a DCF, cuando hay 256 dispositivos contendientes) al no restringir el límite de reintentos en la transmisión de los paquetes.

- Menor tiempo de transferencia de un conjunto completo de paquetes y equidad en la entrega. En el escenario cuando cada uno de los dispositivos tenía solamente un paquete por transmitir, en un canal con una tasa de perdida de paquetes cercana al 1%, se comprobó que, bajo las condiciones establecidas en las simulaciones, el tiempo requerido para transferir un conjunto completo de paquetes al utilizar el protocolo 2CA-R^2, es menor en comparación con DCF, cuando hay más de 20 dispositivos en el escenario. Este resultado significa que, al acotar el tiempo a una ventana mayor 10 ms y menor a 450 ms, el protocolo 2CA-R^2 es capaz de dar servicio a un número mayor de dispositivos que DCF, cuando cada uno de los dispositivos

cuenta solo con un paquete para su transmisión. La diferencia en el número de dispositivos que pueden ser atendidos, entre ambos protocolos, aumenta a medida que se incrementa el tamaño de la ventana, siendo de hasta 60 dispositivos cuando la ventana es de hasta 250 ms, lo cual significa que, para este caso, el protocolo 2CA-R^2 logra una ganancia de alrededor del 40% con respecto a DCF. A su vez, por medio del análisis de la desviación estándar del access delay de cada simulación, se comprobó que, los tiempos de *access delay* de cada paquete involucrado en un CRI se encuentran relativamente cercanos al promedio de *access delay* del ciclo completo, garantizando así equidad en los tiempos de entrega de cada paquete.

- Disminución del *global mean access delay* con tráfico de Poisson con una tasa $\lambda = 2.5$ paquetes cada segundo. En el escenario en que se consideró generación de tráfico mediante un proceso de Poisson con una tasa $\lambda = 2.5$ paquetes cada segundo y con un canal libre de errores, se comprobó que, bajo las condiciones establecidas en las simulaciones, el *global mean access delay*, es menor en comparación con DCF. Esta ventaja a favor del protocolo 2CA-R^2 se exacerba a medida que aumenta el número de dispositivos. Se demostró que, una contribución importante en la pérdida de tiempo al usar DCF, se debe al retardo de encolamiento que se presenta en los dispositivos. Este retardo representa casi el 40% del *global mean access delay* cuando hay 256 dispositivos en el escenario.

Por otra parte, para validar los resultados obtenidos por simulación, se desarrolló un modelo matemático basado en una cadena de Markov que representa el CRI del algoritmo *Adaptive*-2C con actualización por fase. Este algoritmo es el fundamento del intervalo de contienda del protocolo 2CA-R^2. Los resultados obtenidos mediante el modelo matemático coinciden de forma precisa con los valores producidos por el simulador, demostrando así que la cadena de Markov representa con precisión el modelo de simulación utilizado en condiciones ideales (es decir con un canal libre de errores). A pesar de la discrepancia existente entre el modelo matemático y el protocolo implementado en el simulador, se mostró que los resultados tienen una variación despreciable, comparados con la versión extendida que representa fielmente al CRI del protocolo. Sin embargo, tal versión extendida genera una mayor complejidad que se consideró innecesaria, presentando una versión más simple por motivos de practicidad.

Por último, se considera que los objetivos planteados en este proyecto fueron alcanzados satisfactoriamente a través de la metodología propuesta. Complementariamente, se identificaron algunas áreas que permiten extender este trabajo en el futuro.

5.2. Perspectivas de investigación

A lo largo del proyecto de investigación, se identificaron algunos tópicos relacionados con el protocolo 2CA-R^2 y su contexto, mismos que ya no pudieron ser implementados o evaluados. Dichos temas forman parte de las perspectivas de investigación debido a que se consideran nichos de oportunidad. Estos se describen a continuación.

- Escalamiento del protocolo. Se pretende evaluar el rendimiento del protocolo 2CA-R^2 con más de 256 dispositivos e implementar algunas estrategias de escalamiento. Como se puede constatar en la evaluación de desempeño, los resultados de este trabajo pueden aplicarse directamente a una situación donde el número de dispositivos contendientes está limitado a un máximo de 256. Sin embargo, es importante destacar que, en caso de que este número sea mayor, es posible dividirlo en grupos de hasta 256 dispositivos cada uno. Esta estrategia de escalado también se utiliza en el estándar IEEE 802.11ah [52].
- Implementación de modelos de tráfico más realistas. Para evaluar el desempeño de los mecanismos MAC diseñados para comunicaciones M2M, se requiere de modelos que representen con precisión el tráfico real. Los modelos utilizados típicamente en este tipo de evaluación se dividen en dos grupos: modelos de tráfico fuente y modelos de tráfico agregado. Los modelos de tráfico agregado representan el flujo de tráfico de un grupo de dispositivos dentro de un área determinada de forma simple y eficiente, a costa de perder algunas de las características individuales de cada dispositivo, derivando en la falta de precisión en el comportamiento de la red [59]. Este efecto se incrementa si los dispositivos tienen patrones de tráfico diferentes. A su vez, los modelos de tráfico fuente representan el comportamiento de tráfico de un dispositivo individual. Sin embargo, este tipo de modelos presentan un mayor costo computacional, lo cual genera un inconveniente en caso de aumentar de forma considerable el número de dispositivos involucrados [60]. Una opción que apareció en años recientes es el diseño de modelos de tráfico que ofrecen un equilibrio entre alta precisión y baja complejidad, los cuales en su mayoría hacen uso de cadenas de Markov para modelar el flujo entre estados de los dispositivos M2M de manera estocástica [59]. Uno de estos modelos

se conoce como CMMPP o *Coupled Markov Modulated Poisson Process* [61]. En el modelo CMMPP los eventos de generación de paquetes en un dispositivo se modelan mediante un proceso de Poisson modulado por Markov, es decir, un proceso de Poisson cuya tasa de llegadas depende del estado de una cadena de Markov que, a su vez, está asociada a un determinado estado del dispositivo que se está representando. Este modelo puede capturar la correlación temporal y espacial de las fuentes de tráfico M2M mediante el uso de un par de matrices de probabilidad de transición definidas globalmente para dispositivos perfectamente coordinados y no coordinados además de incorporar un proceso de fondo común que modela la correlación temporal entre las fuentes. A este modelo se le han realizado pruebas de desempeño donde se comprobó su capacidad de reproducir, de forma precisa, el comportamiento del tráfico generado por dispositivos M2M reales en aplicaciones y escenarios específicos [59]. Por esta razón, se desea evaluar el desempeño de los protocolos 2CA-R^2 y DCF en condiciones de tráfico real generado por aplicaciones específicas del IoT, utilizando para ello el modelo CMMPP.

- Estimación de multiplicidad de colisión. Se pretende investigar la incorporación de un predictor más complejo, en el protocolo 2CA-R^2, que el utilizado en los escenarios de evaluación de desempeño de esta tesis. Si bien la estrategia implementada demostró ser eficaz en los escenarios presentados, vale la pena destacar que el protocolo no está vinculado a este método de predicción y que se pueden emplear muchos otros en su lugar. En este contexto, existe evidencia que demuestra que el rendimiento puede mejorarse mediante la aplicación de técnicas de inteligencia artificial. Una línea de investigación particularmente prometedora consiste en el desarrollo de un mecanismo basado en *machine learning* para predecir la multiplicidad de colisiones, tal como se hace en [62] para redes LTE.
- Exploración de patrones de correlación temporal y espacial. Se pretende analizar con detalle los patrones de correlación temporal y espacial generado por las comunicaciones M2M, proveniente de algunas aplicaciones del IoT, de modo que puedan aprovecharse para disminuir el tráfico entrante hacia un punto de acceso. Específicamente, se desea generar grupos de dispositivos con propiedades espaciales o temporales similares, de forma que no sea necesaria la transmisión de los paquetes de cada dispositivo, sino solo de aquellos que posean la información representativa de todo el grupo al que pertenecen.
- Implementación de mecanismos de coordinación entre puntos de acceso. Dada la presencia de una gran cantidad de dispositivos que intentan transmitir de manera simultánea, puede

presentarse un escenario donde existan dos o más puntos de acceso, utilizando canales de frecuencias iguales o adyacentes, lo suficientemente cerca uno de otro como para interferirse mutuamente, afectando la comunicación con los dispositivos a los que prestan servicio. Por esta razón, se pretende investigar el efecto de implementar un mecanismo de coordinación entre los puntos de acceso, de una forma similar a la que se lleva a cabo en el estándar IEEE 802.11be, el cual se encuentra aún en desarrollo. Se busca que dicho mecanismo sea de baja complejidad tal como la coordinación en tiempo y frecuencia, o la coordinación de potencia adaptiva [63].

- Implementación de 2CA-R^2 en el estándar IEEE 802.11ah. Como ya se ha mencionado, uno de los protocolos MAC más utilizados para comunicaciones M2M en redes WLAN es el protocolo IEEE 802.11ah. El éxito de este protocolo se debe, en parte, a la división del número total de dispositivos contendientes en grupos de menor tamaño [27]. Posteriormente, a los dispositivos de cada grupo se les asigna una ventana RAW para contender, únicamente con los dispositivos de su grupo, por una oportunidad de transmisión. Debido a la estructura de las RAW, las cuales contienen fases de contienda, transmisión y retroalimentación de tamaño fijo, se pueden requerir varios ciclos hasta que todos los dispositivos que contienen un paquete de datos sean transmitidos, presentando un retardo considerable en caso de congestión en el medio. Por lo anterior, se pretende desarrollar una versión del protocolo 802.11ah sustituyendo el mecanismo DCF por 2CA-R^2 en las ventanas RAW. De esta manera, se busca mejorar el *network throughput* y minimizar el *access delay* para transmitir los datos de un dispositivo siempre que este lo requiera.

REFERENCIAS

[1] X. Yu y F. Sun, «A Study on Telecommunication Network, Internet, and Internet of Things,» de *10th International Conference on Information Systems and Computing Technology (ISCTech)*, Guilin, China, 2022.

[2] F. Malandra, S. Rochefort, P. Potvin y B. Sansò, «A case study for M2M traffic characterization in a smart city enviroment,» de *International Conference on Internet of Things and Machine Learning*, Liverpool, Inglaterra, 2017.

[3] A. Samuel y C. Sipes, «Making Internet of Things Real,» *IEEE Internet of Things Magazine,* vol. 2, nº 1, pp. 10-12, 2019.

[4] P. Verma, R. Verma, A. Prakash, A. Agrawal, K. Naik, R. Tripathi, T. Khalifa, M. Alsabaan, T. Abdelkader y A. Abogharaf, «Machine-to-Machine (M2M) Communications: A Survey,» *Journal of Network and Computer Applications,* vol. 66, pp. 83-105, 2016.

[5] S. Watson, A. Piette, O. Sezgen y N. Motegi, «M2M Technology in Demand Responsive Commercial Buildings,» de *American Council for an Energy-Efficient Economy (ACEEE)*, Washington, Estados Unidos, 2004.

[6] Editado por C. Antón-Haro y M. Dohler, Machine-to-machine (M2M) Communications: Architecture, Performance and Applications, Woodhead Publishing - Elsevier, 2015.

[7] European Telecommunications Standards Institute (ETSI), *Machine-to-Machine Communications (M2M), Functional Architecture, tech. rep. ETSI TS 102 690 V2.1.1,* octubre 2013.

[8] oneM2M Partners (ARIB, ATIS, CCSA, ETSI, TIA, TSDSI, TTA, TTC) , *oneM2M Technical Specification, TS-0001-V1.13.1, Functional Architecture,* 2016.

[9] M. Chen, J. Wan y F. Li, «Machine-to-Machine Communications: Architectures, Standards and Applications,» *KSII Transactions on Internet and Information Systems,* vol. VI, nº 2, pp. 480-497, 2012.

[10] IoT Analytics, «State of IoT Summer 2024,» [En línea]. Available: https://iot-analytics.com/product/state-of-iot-summer-2024/. [Último acceso: 01 Octubre 2024].

[11] M. Sansoni, G. Ravagnani, D. Zucchetto, C. Pielli, A. Zanella y K. Mahmood, «Comparison of M2M Traffic Models Against Real World Data Sets,» de *IEEE 23rd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, Barcelona, España, 2018.

[12] S. Javier-Alvarez, P. Hernandez-Duran, M. Lopez-Guerrero y L. Orozco-Barbosa, «2CA-R2: A Hybrid MAC Protocol for Machine-Type Communications,» *Sensors,* vol. 25, nº 10:2994, 2025.

[13] W. Stallings, Comunicaciones y Redes de Computadores. 7a Edición., Madrid: Pearson Prentice Hall, 2004.

[14] A. Rajandekar y B. Sikdar, «A Survey of MAC Layer Issues and Protocols for Machine-to-Machine Communications,» *IEEE Internet of Things Journal,* vol. 2, nº 2, pp. 175-186, 2015.

[15] A. Gaurav y K. Pandey, «A Survey: Hybrid Medium Access Control Protocol for M2M Communication,» de *nternational Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, Chennai, India, 2018.

[16] A. Tanenbaum, Computer Networks 4a ed., Prentice Hall, 2003.

[17] J. Kurose y K. Ross, Redes de Computadoras. Un enfoque descendente. 7a Edición, Madrid: Pearson, 2017.

[18] H. Wu, S. Cheng, Y. Peng, K. Long y J. Ma, «IEEE 802.11 Distributed Coordination Function (DCF): analysis and enhancement,» de *IEEE International Conference on Communications. Conference Proceedings (ICC)*, New York, EU, 2002.

[19] J. Proakis y M. Salehi, Digital Communications, 5a Edición, McGraw-Hill, 2008.

[20] S. Haykin y M. Moher, Modern Wireless Communications, Pearson Prentice Hall, 2005.

[21] A. Amodu y M. Othman, «A Survey of Hybrid MAC Protocols for Machine-to-Machine Communications,» *Springer Telecommunication Systems,* vol. 69, nº 1, pp. 141-165, 2018.

[22] Y. Liu, C. Yuen, J. Chen and X. Cao, "A scalable Hybrid MAC protocol for massive M2M networks," in *IEEE Wireless Communications and Networking Conference (WCNC)*, Shanghai, China, 2013.

[23] A. Laya, F. Vázquez-Gallego y L. Alonso, «Goodbye, ALOHA!,» *IEEE Access,* vol. 4, pp. 2029-2044, 2016.

[24] L. Oliveira, J. Rodrigues, S. Kozlov, R. Rabêlo y V. Albuquerque, «MAC Layer Protocols for Internet of Things: A Survey,» *MDPI Future Internet,* vol. 11, nº 16, pp. 1-42, 2019.

[25] Bluetooth-SIG, «Bluetooth Specification v5.2,» 31 12 2019. [En línea]. Available: https://www.bluetooth.com/specifications/specs/core-specification/. [Último acceso: 07 05 2021].

[26] «IEEE Standard for Local and metropolitan area networks--Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs),» *IEEE Std 802.15.4-2011 (Revision of IEEE Std 802.15.4-2006),* pp. 1-314, Septiembre, 2011.

[27] T. Adame, A. Bel, B. Bellalta, J. Barcelo y M. Oliver, «IEEE 802.11AH: The WiFi approach for M2M Communications,» *IEEE Wireless Communications,* vol. 21, nº 6, pp. 144-152, 2014.

[28] 3GPP, «3rd Generation Partnership Project,» [En línea]. Available: https://www.3gpp.org. [Último acceso: 01 07 2025].

[29] R. Ratasuk, N. Mangalvedhe, D. Bhatoolaul y A. Ghosh, «LTE-M Evolution Towards 5G Massive MTC,» de *IEEE Globecom Workshops (GC Wkshps)*, Singapur, 2017.

[30] GSMA (Asociación GSM), «Mobile IoT in a 5G Future: NB-IoT and LTE-M in the context of 5G (Whitepaper),» Octubre 2024.

[31] P. Verma, R. Tripathi and K. Naik, "A robust hybrid-MAC protocol for M2M communications," in *International Conference on Computer and Communication Technology (ICCCT)*, 2014.

[32] E. Hegazy, W. Saad, M. Shokair y S. El-Halafawy, «Proposed MAC protocol for M2M networks,» *International Journal of Computing and Digital Systems (IJCDS),* vol. 5, nº 4, pp. 357-363, 2016.

[33] B. Yang, X. Cao y L. Qian, «A Scalable MAC Framework for Internet of Things Assisted by Machine Learning,» de *IEEE 88th Vehicular Technology Conference (VTC-Fall)*, 2018.

[34] X. Cao, B. Yang y Z. Song, «Performance analysis of hybrid MAC scheme with multi-slot reservation,» *Electronics Letters,* vol. 54, nº 4, pp. 250-252, 2018.

[35] X. Cao, Z. Song y B. Yang, «TR-MAC: A multi-step slot reservation-based hybrid MAC protocol for ad hoc networks (ICSPCC),» de *International Conference on Signal Processing, Communications and Computing*, 2015.

[36] W. Saad, S. El-Feshawy, M. Shokair y M. Dessouky, «Optimised approach based on hybrid MAC protocol for M2M networks,» *IET Networks,* vol. 7, 2018.

[37] A. Lachtar, A. Kachouri y M. Lachtar, «A Hybrid MAC Protocol for Heterogeneous M2M Networks,» de *Intelligent Systems Design and Applications - ISDA*, Auburn, WA, Estados Unidos , 2019.

[38] D. Olatinwo, A. Abu-Mahfouz and G. Hancke, "A Hybrid Multi-Class MAC Protocol for IoT-Enabled WBAN Systems," *IEEE Sensors Journal,* vol. 21, no. 5, pp. 6761-6774, 2021.

[39] D. Olatinwo, M. Abu-Mahfouz y P. Hancke, «Energy-Aware Hybrid MAC Protocol for IoT Enabled WBAN Systems,» *IEEE Sensors Journal,* vol. 22, nº 3, pp. 2685-2699, 2022.

[40] M. Salajegheh, H. Soroush y A. Kalis, «HYMAC: Hybrid TDMA/FDMA Medium Access Control Protocol for Wireless Sensor Networks,» de *IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications*, Athens, Greece, 2007.

[41] C. Lin, J. Lin y T. Chen, «Channel-Aware Polling-Based MAC Protocol for Body Area Networks: Design and Analysis,» *IEEE Sensors Journal,* vol. 17, nº 9, pp. 2936-2948, 2017.

[42] X. Fan, J. Hu y K. Yang, «Payload-Adaptive Hybrid MAC Protocol for Sustainable Internet of Things Networks,» de *IEEE 24th International Conference on Communication Technology*, 2024.

[43] J. Hu, G. Xu, L. Hu, S. Li y Y. Xing, «An adaptive energy efficient MAC protocol for RF energy harvesting WBANs,» *IEEE Transactions on Communications,* vol. 71, nº 1, pp. 473-484, 2023.

[44] J. Capetanakis, «Tree algorithms for packet broadcast channels,» *IEEE Transactions on Information Theory,* vol. 25, nº 5, pp. 505-515, 1979.

[45] W. Xu y G. Campbell, «A near perfect stable random access protocol for a broadcast channel,» de *[Conference Record] SUPERCOMM/ICC '92 Discovering a New World of Communications*, Chicago, Estados Unidos, 1992.

[46] M. Paterakis y P. Papantoni-Kazakos, «A simple window random access algorithm with advantageous properties,» *IEEE Transactions on Information Theory,* vol. 35, nº 5, pp. 1124-1130, 1989.

[47] Wi-Fi Alliance, « Discover Wi-Fi,» [En línea]. Available: https://www.wi-fi.org. [Último acceso: 27 01 2025].

[48] L. Alarcón, M. López y M. Makrakis, «Adaptive 2C: A Novel Access Control for Fair and Efficient Channel Sharing,» de *Canadian Conference on Electrical and Computer Engineering*, 2007.

[49] P. Hernández, «Propuesta y evaluación de un protocolo híbrido de control de acceso al medio (MAC) con reservación de recursos,» [Tésis de Maestría] Universidad Autónoma Metropolitana, 2010.

[50] OMNeT++, «Network Simulation Framework,» [En línea]. Available: https://omnetpp.org/. [Último acceso: Febrero 2025].

[51] INET Framework, «Open-source model for OMNeT++,» [En línea]. Available: https://inet.omnetpp.org/. [Último acceso: Febrero 2025].

[52] L. Tian, S. Santi, A. Seferagić, J. Lan y J. Famaey, «Wi-Fi HaLow for the Internet of Things: An up-to-date survey on IEEE 802.11ah research,» *Journal of Network and Computer Applications,* vol. 182, 2021.

[53] K. Smiljkovic, V. Atanasovski y L. Gavrilovska, «Machine-to-Machine Traffic Characterization: Models and Case Study on Integration in LTE,» de *International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems (VITAE)*, Aalborg, Dinamarca, octubre 2014.

[54] E. Ziouva y T. Antonakopoulos, «CSMA/CA performance under high traffic conditions: throughput and delay analysis,» *Elsevier Computer Communications,* vol. 25, nº 3, pp. 313-321, 2002.

[55] D. Prisca y T. Walingo, «Performance analysis of an adaptive OFDMA-based CSMA/CA scheme on a wireless network,» *Wiley IET Communications ,* vol. 14, nº 9, pp. 3480-3489, 2020.

[56] J. Norris, Markov Chains, Cambridge University Press, 1997.

[57] Mathworks, «Matlab: Compute markov chain hitting times,» [En línea]. Available: https://www.mathworks.com/help/econ/dtmc.hittime.html. [Último acceso: Febrero 2025].

[58] J. Stewart, Introduction to the Numerical Solution of Markov Chains, Princeton University Press, 1994.

[59] M. Sansoni, G. Ravagnani, D. Zucchetto, C. Pielli, A. Zanella y K. Mahmood, «Comparison of M2M Traffic Models Against Real World Data Sets,» de *IEEE 23rd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, Barcelona, España, 2018.

[60] K. Smiljkovic, V. Atanasovski y L. Gavrilovska, «Machine-to-Machine Traffic Characterization: Models and Case Study on Integration in LTE,» de *4th International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems (VITAE)*, Aalborg, Dinamarca, 2014.

[61] M. Laner, P. Svoboda, N. Nikaein y M. Rupp, «Traffic Models for Machine Type Communications,» de *The Tenth International Symposium on Wireless Communication Systems (ISWCS)*, Ilmenau, Alemania, 2013.

[62] D. Magrin, C. Pielli, Č. Stefanović y M. Zorzi, «Enabling LTE RACH Collision Multiplicity Detection via Machine Learning,» de *International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOPT)*, 2019.

[63] W. Ahn, «Novel Multi-AP Coordinated Transmission Scheme for 7th Generation WLAN 802.11be,» *MDPI Entropy,* vol. 22, nº 12, pp. 1-19, 2020.

Casa abierta al tiempo
UNIVERSIDAD AUTÓNOMA METROPOLITANA

ACTA DE DISERTACIÓN PÚBLICA

No. 00025
Matrícula: 2163802704

Desarrollo de un Protocolo de Control de Acceso al Medio en Comunicaciones M2M

En la Ciudad de México, se presentaron a las 11:00 horas del día 4 del mes de agosto del año 2025 en la Unidad Iztapalapa de la Universidad Autónoma Metropolitana, los suscritos miembros del jurado:

DR. ENRIQUE RODRIGUEZ DE LA COLINA
DR. LUIS ALBERTO VASQUEZ TOLEDO
DR. MIGUEL LOPEZ GUERRERO
DR. JAVIER GOMEZ CASTELLANOS
DR. RICARDO MARCELIN JIMENEZ

Bajo la Presidencia del primero y con carácter de Secretario el último, se reunieron a la presentación de la Disertación Pública cuya denominación aparece al margen, para la obtención del grado de:

DOCTOR EN CIENCIAS (CIENCIAS Y TECNOLOGÍAS DE LA INFORMACIÓN)

DE: SERGIO JAVIER ALVAREZ

y de acuerdo con el artículo 78 fracción IV del Reglamento de Estudios Superiores de la Universidad Autónoma Metropolitana, los miembros del jurado resolvieron:

APROBAR

Acto continuo, el presidente del jurado comunicó al interesado el resultado de la evaluación y, en caso aprobatorio, le fue tomada la protesta.

SERGIO JAVIER ALVAREZ
ALUMNO

REVISÓ

MTRA. ROSALIA SERRANO DE LA PAZ
DIRECTORA DE SISTEMAS ESCOLARES

DIRECTOR DE LA DIVISIÓN DE CBI

Román Linares Romero
DR. ROMAN LINARES ROMERO

PRESIDENTE

DR. ENRIQUE RODRIGUEZ DE LA COLINA

VOCAL

DR. LUIS ALBERTO VASQUEZ TOLEDO

VOCAL

DR. MIGUEL LOPEZ GUERRERO

VOCAL

DR. JAVIER GOMEZ CASTELLANOS

SECRETARIO

DR. RICARDO MARCELIN JIMENEZ