



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

**EVALUACIÓN DE TÉCNICAS PARA
LA RESOLUCIÓN DE COLISIONES
DINÁMICAS EN REDES WiMAX
IEEE 802.16**

T E S I S

QUE PARA OBTENER EL GRADO DE:

**MAESTRO EN INGENIERÍA
(COMPUTACIÓN)**

P R E S E N T A:

MIGUEL ÁNGEL PÉREZ VELÁZQUEZ

DIRECTOR DE LA TESIS: DR. VÍCTOR RANGEL LICEA



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Agradecimientos

A la Universidad Nacional Autónoma de México, por permitirme cursar los estudios de Maestría y formar parte de la comunidad universitaria, misma que fortaleció mis deseos de superación y amplió mi visión como ser humano.

Al Dr. Víctor Rancel Licea por su apoyo y paciencia durante el trabajo realizado bajo su asesoría. Su correcta guía permitió la culminación exitosa de este trabajo, por lo que le reitero mi admiración y gratitud por sus enseñanzas.

A la DGAPA-UNAM por el apoyo otorgado al proyecto de investigación PAPIIT IN104907 "Diseño de técnicas de mejoramiento de capacidad en redes inalámbricas de banda ancha tipo Mesh".

Al Consejo Nacional de Ciencia y Tecnología por el apoyo económico en mis estudios de posgrado.

Al los ingenieros Jaime Martínez Martínez, Carlos Pineda Figueroa, Fernando Sánchez Rodríguez, Efraín Torres Ortiz por su incondicional apoyo en la División de Ciencias Básicas de la Facultad de Ingeniería.

A todos los profesores del Posgrado de Ingeniería por su dedicación y enseñanza en las materias propias del área de redes de comunicaciones.

Así también agradezco la paciencia de la administración del posgrado que siempre se preocuparon por la situación de mi estudio.

Dedicatorias

A mis padres por todo su amor y comprensión en cada momento de mi vida. Los quiero mucho.

A mis hermanos que tanto quiero. Siempre me han apoyado y alentado para culminar mis metas.

A mi novia Victoria Martínez Enríquez por su amor y comprensión durante mucho tiempo. Te amo.

A todos mis amigos a quienes quiero mucho y de quienes he aprendido muchas cosas, la más importante el disfrutar con ellos de la vida. Gracias por su amistad.

CONTENIDO

CONTENIDO.....	v
LISTA DE FIGURAS.....	ix
LISTA DE TABLAS.....	xi
RESUMEN.....	xiii
CAPÍTULO 1.....	1
Introducción.....	1
1.1 Antecedentes.....	1
1.2 Definición del problema.....	3
1.2.1 Métodos de evaluación y medidas de desempeño de la eficiencia de una red.....	5
1.3 Objetivos.....	6
1.4 Método de investigación.....	7
1.5 Contribución y relevancia.....	7
1.6 Estructura de la tesis.....	8
CAPÍTULO 2.....	9
Descripción del estándar 802.16.....	9
2.1 Introducción.....	9
2.2 Capa Física.....	10
2.2.1. Estructura de <i>frame</i> y <i>slot</i>	11
2.2.2. Codificación y modulación adaptativa.....	13
2.2.3. Tasas de transmisión de capa PHY.....	13
2.3 Capa MAC.....	14
2.3.1 Construcción y transmisión de MAC PDU.....	16
2.3.2. Calidad de servicio.....	19
<i>Piggyback Request</i>	19
Bandwidth stealing.....	20
<i>Polling</i>	20
Servicio <i>Unsolicited Grant Service</i> (UGS).....	21
Servicios en tiempo real (rtPS).....	21
Servicios no en tiempo real (nrtPS).....	21
Servicio del mejor esfuerzo (BE).....	21
2.3.3. Solicitudes y asignación de ancho de banda.....	22
Solicitudes de ancho de banda.....	22
Modo de Asignación por Conexión (GPC).....	23
Modo de Asignación por Estación Subscriptora (GPSS).....	23

Consultas.....	23
2.4. Conclusiones.....	24
CAPÍTULO 3.....	25
Algoritmos de resolución de colisiones	25
3.1 Introducción.....	25
3.2 Antecedentes.....	25
3.3 Resolución de colisiones en IEEE 802.16	26
3.3.1 Operación de resolución de colisiones	27
3.4 Estudios sobre Binary Exponential Backoff	30
3.5 Estudios del funcionamiento del IEEE802.16 relacionados en la resolución de colisiones.....	31
3.6 Algoritmos analizados y propuestos en este trabajo de tesis	32
3.6.1. Simple-CSA.....	32
3.6.2. Forced-CSA.....	34
3.6.3 Variable-CSA	36
3.6.4. Multicast-CSA.....	37
3.6.5 IEEE 802.14-CSA	38
3.6.6. Collision Free-CSA	39
3.6.7 CDMA-CSA.....	41
3.7 Conclusiones.....	41
CAPÍTULO 4.....	43
Modelo de simulación y adaptación de las técnicas	43
4.1 Introducción.....	43
4.2 Modelado de <i>OPNET</i>	44
Dominio de Red (Network Domain).....	44
Nodo Dominio (<i>Node domain</i>).....	45
Proceso dominio (<i>Process domain</i>).....	46
4.3 Implementación de las técnicas en el modelo de simulación.....	46
4.3.1 Implementación 802.14-CSA	49
4.3.2 Implementación Collision Free-CSA	50
4.3.3 Implementación de Variable-CSA.....	52
4.3.4 Implementación CDMA-CSA	53
4.4 Conclusiones.....	54
CAPÍTULO 5.....	55
Simulación y resultados.....	55
5.1 Introducción.....	55
5.2 Parámetros de simulación.....	55

5.2.1 Retardo promedio de acceso (<i>Mean access delay</i>)	57
5.2.2 <i>Throughput</i> del Sistema.....	58
5.3 Diseño de la Red.....	59
5.4 Resultados.....	60
5.4.1 SIMPLE-CSA.....	61
5.4.2 Forced-CSA.....	63
5.4.3 Variable-CSA	65
5.4.4 Multicast-CSA.....	66
5.4.5 802.14-CSA.....	68
5.4.6 Collision Free-CSA	70
5.4.7 CDMA-CSA.....	72
5.6 Conclusiones.....	74
CAPÍTULO 6.....	75
Conclusiones generales.....	75
APÉNDICE A	77
Modelo de Simulación de Redes WiMAX	77
A.1 Escenario.....	77
A.2. Diseño de la Estación Base (<i>Base Station</i>)	78
Rx_point	78
From_network_cloud_rx.....	79
To_network_cloud_tx	79
To_stats_collection_tx	79
From_link_rx	79
Ant_rx.....	79
To_link_tx.....	79
Ant_tx	79
HE_MAC.....	80
A.2.1 Proceso <i>bwa_bs_mgr</i>	80
INIT.....	80
Ranging	81
Rng_Recv	81
Rng_complete	81
Idle	81
Upstream	81
Down_data	82
Mtn_MAP	82
MAP_Time.....	82

UCD_Time.....	83
Syn_Time.....	83
A.3 Estación Suscriptora (<i>Suscriber Station</i>).....	83
src_type_1.....	84
Downstream_sink.....	84
Ant_rx y ant_tx.....	84
From_link_rx y from_link_tx.....	84
MAC.....	85
A.3.1 Proceso <i>bwa_wm_mgr</i>	85
Init.....	85
Ranging.....	86
Syn_rsp.....	86
Idle.....	86
From_downstream.....	87
Down_data_traffic.....	87
MAP_Arrival.....	87
UCD_Arrival.....	87
Sync_Arrival.....	87
App_Traffic.....	87
A.3.2 Proceso <i>Bwa_wm_child</i>	87
Init.....	88
Q_Empty.....	88
No_Request_Outstanding.....	89
Tx_Opp_Pending.....	89
Collision_Resolution.....	90
Request_Outstanding.....	90
APÉNDICE B	91
Parámetros de capa MAC y PHY	91
B.1 Capa PHY.....	91
B.2 Capa MAC.....	94
APÉNDICE C	101
Artículo para su publicación	101
REFERENCIAS	107
Sitios web.....	110

LISTA DE FIGURAS

Figura 2.1. Representación de la estructura de un <i>frame</i> en modo TDD (a) y FDD (b).	11
Figura 2.2. Representación de la estructura de un <i>subframe</i> DL.	12
Figura 2.3. Representación de la estructura de un <i>subframe</i> de <i>uplink</i> .	12
Figura 2.4. Estructura de la capa MAC del protocolo 802.16.	15
Figura 2.5. Fragmentación y concatenación de SDUs en MAC PDUs.	16
Figura 2.6. Estructura de MAC PDU.	17
Figura 2.7. Encabezados de PDU: (a) genérico; (b) petición de ancho de banda.	17
Figura 3.1. Procedimiento de resolución de colisiones para IEEE802.16.	28
Figura 3.2. Procedimiento de Simple-CSA con MCs=0.	33
Figura 3.3. Procedimiento de Simple-CSA con MCs=2.	34
Figura 3.4. Procedimiento de Forced-CSA, con MCs=2.	35
Figura 3.5. Procedimiento de Multicast-CSA.	38
Figura 3.6. Procedimiento de Collision Free-CSA.	40
Figura 4.1 Diagrama de Estado de OPNET para los modelos de simulación.	45
Figura 4.2. Localización del estado SEND_NEX_MAP de la BS.	47
Figura 4.3 Ubicación del módulo <i>bwa_wm_child</i> de las estaciones suscriptoras.	49
Figura 5.1 Distribución de paquetes de Ethernet.	55
Figura 5.2. Diagrama de tiempo para la contabilización del retardo.	57
Figura 5.3 Escenario de simulación.	59
Figura 5.4 Parámetros de simulación de la estación suscriptora SS.	60
Figura 5.9. Throughput for all sources, para Simple-CSA.	62
Figura 5.14. <i>Throughput</i> para todas las SSs: Simple-CSA y Forced-CSA.	64
Figura 5.19. <i>Throughput</i> para todas las fuentes: Simple, Forced y Variable-CSA.	66
Figura 5.24. <i>Throughput</i> for all sources para Simple, Variable, Forced, Multicast-CSA.	68
Figura 5.29. <i>Throughput</i> for all sources para Simple, Forced, Variable, Multicast y 802.14-CSA.	70
Figura 5.34. <i>Throughput</i> for all sources para Simple, Forced, Variable, Multicast, 802.14 y Collision Free-CSA.	71
Figura 5.39. Gráfica de comparación de <i>Throughput</i> for All Sources para los esquemas analizados.	73
Figura 5.40. Utilización del canal de subida de las técnicas analizadas.	74
Figura A.1 Escenario de simulación.	77
Figura A.2 Configuración interna de la estación base (<i>Base Station</i>).	78
Figura A.3 Proceso <i>Bwa_bs_mgr</i> .	80
Figura A.4. Estación suscriptora (<i>Suscriber Station</i>).	83
Figura A.5. Atributos del canal de bajada.	84
Figura A.6. Proceso <i>bwa_wm_mgr</i> .	86
Figura A.7. Proceso <i>bwa_wm_child</i> .	88

LISTA DE TABLAS

Tabla 1.1. Datos básicos de la familia de estándares de IEEE 802.16.....	3
Tabla 2.1. Términos usados par la descripción del protocolo 802.16.....	10
Tabla 2.2. Esquemas de modulación y codificación de DL y UL.	13
Tabla 2.3. Tasas de transmisión de capa PHY.....	14
Tabla 2.4. Campos de la cabecera MAC genérica.	18
Tabla 2.5. Campos de la cabecera MAC para peticiones de ancho de banda.....	18
Tabla 2.5. Tipos de servicio para transmisiones en el UL.....	20
Tabla 5.1 Slots requeridos para cada paquete.....	56
Tabla 5.2 Parámetros generales del modelo de simulación.	58

RESUMEN

Debido a la gran demanda de las necesidades actuales, en el ámbito de las telecomunicaciones, ha traído consigo el desarrollo de las siguientes generaciones de servicios de comunicaciones que proporcionen altas velocidades de acceso a Internet, voz, video sobre IP y en general servicios multimedia que requieren grandes anchos de banda y además que sean capaces de soportar QoS. Tal es el caso del estándar IEEE 802.16.

Este trabajo se enfoca en optimizar el canal de subida. El canal de subida es un medio compartido, en donde se utiliza contención aleatoria, reservación y técnicas establecidas de reservación de acceso. El intervalo de distribución respecto al tiempo inicial de estos modos de acceso es dinámico y controlado por la estación base (Base Station). Los límites de acceso de contención y reservación son enviados periódicamente en forma de *broadcast* en el inicio de cada conexión. Cuando ocurre una colisión, un algoritmo de resolución de colisiones (CRA) es utilizado para resolverla, los sistemas de banda ancha basados el estándar IEEE 802.16 utilizan como CRA el algoritmo *Exponential Backoff*.

Esta tesis aborda e investiga el análisis de siete algoritmos dinámicos de resolución de colisiones para el estándar IEEE 802.16 —Simple-CSA, Forced-CSA, Multicast-CSA, Variable-CSA, IEEE 802.14-CSA, Collision Free-CSA y CDMA-CSA—, en un ambiente de simulación que de alguna u otra forma optimizan la resolución de colisiones que son la clave para servicios en donde el tiempo es un parámetro crítico. Generalmente para servicios interactivos en tiempo real, tales como VoIP, video, Streaming, etc.

Estas técnicas ajustan dinámicamente el número de *slots* de contención necesarios para resolver colisiones, según la carga de tráfico, el número de usuarios que colisionan y las probabilidades de que ocurran éstas, mejorando considerablemente el funcionamiento total del sistema. Los resultados obtenidos se analizan en relación al retardo promedio de acceso (*mean access delay*) y el funcionamiento del sistema (*throughput*), para tráfico IP.

A lo largo de este trabajo se presenta un panorama de las técnicas evaluadas y de las investigaciones en el área de resolución de colisiones para el estándar 802.16. Por otro lado, se muestra una descripción del estándar IEEE 802.16 haciendo énfasis en el control del acceso al medio (MAC).

También es importante mencionar que en este trabajo se presenta el modelo de simulación que se utilizó para simular, analizar y evaluar el sistema de banda ancha. Este modelo de esta basado en OPNET MODELER Ver 11.0, un lenguaje de simulación utilizado como una herramienta para la caracterización del desempeño y mejora de las redes de banda ancha. El principal motivo de su utilización, es porque posee un conjunto extenso de características diseñadas para soportar el modelado de redes que proveen un incremento en la flexibilidad del desarrollo de los modelos diseñados con este lenguaje de simulación.

CAPÍTULO 1

Introducción

1.1 Antecedentes

La gran demanda de nuevos servicios de información como aplicaciones sobre Internet y telefonía sobre el protocolo IP (*Internet Protocol*), tanto de audio y de video, ha motivado no solamente a la industria a extender y mejorar sus redes de comunicaciones, sino también a las instituciones de investigación a desarrollar nuevos protocolos de comunicaciones que soporten eficientemente tales aplicaciones.

Estos servicios de telecomunicaciones pueden ser ofrecidos a través de una variedad de tecnologías como: cable UTP/STP (*Unshielded Twisted Pair/Shielded Twisted Pair*), Cable-TV, redes inalámbricas fijas y móviles, fibra óptica, y redes satelitales. De estos sistemas la arquitectura más prometedora por su accesibilidad a corto plazo son las redes de Cable-TV y xDSL (*Digital Subscriber Loop*). Sin embargo, además de las exigencias de ancho de banda para el desarrollo de las diversas aplicaciones de comunicación, en los últimos años se ha incrementado la necesidad de contar con sistemas de acceso inalámbrico, tanto fijos como móviles, que ofrezcan conectividad de banda ancha en todo momento y lugar.

El término banda ancha inalámbrica (*wireless broadband*) se refiere generalmente a las altas velocidades de transmisión de datos (como mínimo, varios cientos de *kilobits* por segundo) que ocurren entre dos o más puntos fijos, dentro de una red. Ésta promete entregar una amplia gama de servicios informáticos, rápidos y rentables, a los negocios y clientes residenciales. Desafortunadamente esa promesa se brinda con resultados no muy eficaces, debido a que la tecnología existente no se ha desarrollado al grado requerido para ofrecer dichos servicios; además de los altos costos e implementación de los equipos en una red inalámbrica.

Es por eso que las tecnologías de acceso inalámbrico de banda ancha (*Broadband Wireless Access- BWA*) suenan como las más viables para, en un futuro no muy lejano, resolver los problemas que se presentan por transmitir datos a alta velocidad, además de reducir las probabilidades de error por el envío de paquetes y sincronización.

Existen varias propuestas para este tipo de tecnología y entre los más sobresalientes están: el protocolo europeo ETSI BRAN (*European Telecommunications Standards Institute Broadband Radio Access Network*), el cual pretende estandarizar tecnologías que alcancen velocidades de transmisión de 25 a 54 Mbps y el protocolo propuesto por la organización IEEE (*Institute of Electrical and Electronics Engineers*) cuyo nombre es IEEE 802.16 (*Worldwide interoperability for Microwave Access-WiMAX*). Sin embargo, éste último cuenta con un gran soporte de compañías líderes en el mercado a través del *Forum WiMAX*.

El *Forum WiMAX*, en la actualidad, está formado por más de 100 empresas dedicadas a diseñar los parámetros y estándares de esta tecnología. También se dedica a estudiar, analizar y probar los desarrollos implementados, lo cual le da una considerable ventaja con respecto a su homólogo europeo. El estándar 802.16, además de ser compatible con otros estándares como ATM (*Asynchronous Transfer Mode*) e IEEE 802.11 (*Wireless Fidelity-WiFi*), provee un gran ancho de banda con el cual se pueden alcanzar hasta 134 Mbps de transferencia de información.

En 1998, el IEEE formó un grupo llamado 802.16 para desarrollar el estándar *Wireless Metropolitan Area Network* o *wireless MAN*. Originalmente, este grupo se centró en el desarrollo de soluciones en la banda de 10 GHz a 66 GHz, con el principal objetivo de proporcionar conexiones de alta velocidad a las empresas que no podrían obtener una conexión de fibra. El grupo produjo un estándar aprobado en diciembre 2001. Este estándar, *Wireless MAN-SC* [11], especificó una capa física (PHY) que utiliza técnicas de modulación *single-carrier* y una capa MAC (*Media Access Control*) con una estructura TDM (*Time Division Multiplexing*), que soporta FDD (*Frequency Division Duplexing*) y TDD (*Time Division Duplexing*), para facilitar su interoperabilidad con otros sistemas inalámbricos y celulares.

Debido a que es muy compleja la fabricación y operación de los sistemas que actúan en el rango de frecuencias superior a 11 GHz, el grupo comenzó a trabajar para extenderlo hacia las frecuencias con y sin licencia en el rango de 2 GHz a 11 GHz, que permite un funcionamiento de NLOS (*Near Line of Sight*). Este trabajo, IEEE 802.16a [12], fue terminado en 2003 con los esquemas agregados de OFDM (*Orthogonal Frequency Division Multiplexing*) como parte de la capa física, para un funcionamiento en ambientes multidireccionales. Por ese tiempo, OFDM se había establecido como una opción para la multitrayectoria en banda ancha y era ya parte del estándar IEEE 802.11. Además de la capa física OFDM, el IEEE 802.16 también especifica opciones adicionales en la capa MAC incluyendo soporte para OFDMA (*Orthogonal Frequency Division Multiple Access*).

En 2004, se hicieron otras revisiones al estándar 802.16a y surgió el estándar revisado, IEEE 802.16-2004 [13], que sustituye al 802.16, 802.16a, y 802.16c [14] y que también se adoptó como base para HIPERMAN (*High-Performance Metropolitan Area Network*) por el ETSI. En 2003, el grupo 802.16 comenzó el trabajo sobre revisiones a las especificaciones para permitir aplicaciones de alta movilidad en vehículos y la revisión, 802.16e [15], fue terminada en diciembre 2005 y publicada formalmente como IEEE 802.16e-2005. Entre sus principales novedades, especifica un esquema OFDM escalable para la capa física y hace otras modificaciones a la capa del MAC para soportar movilidad de alta velocidad.

Las especificaciones del IEEE 802.16 son una colección de estándares con un alcance muy grande. Para acomodar las diversas necesidades de la industria, el estándar incorporó una amplia variedad de opciones. La interoperabilidad es el principal objetivo de usar la familia de estándares 802.16 y el *Forum WiMAX* es el encargado de asegurarla y de promover las soluciones basadas en el estándar IEEE 802.16.

La tabla 1.1 muestra en detalle las principales características de las diferentes versiones del protocolo 802.16, en la cual se resalta que, entre sus muchas bondades, puede ofrecer un

servicio de conexión a Internet de alta velocidad (de 1 a 134 Mbps) y una capacidad de cobertura de hasta 50 km para usuarios fijos o de 1 y hasta 8 km para usuarios móviles, permitiendo transmitir imágenes, video, teleconferencias, servicios multimedia, datos IP, datos ATM, voz, televisión celular, televisión interactiva y muchos otros servicios, con diferentes tipos de calidad.

	802.16	802.16-2004	802.16e-2005
Banda de frecuencia	2-11 GHz	2-11 GHz	2-11 GHz, 11-15 GHz, 17-22 GHz
Aplicación	Fijo, LOS	Fijo, NLOS	Fijo/Móvil, NLOS
Arquitectura MAC	Point-to-multipoint	Point-to-multipoint	Point-to-multipoint
Esquema de Transmisión	Portadora única	Single carrier, 256 QFDM o 2,048 QFDM	Single carrier, 256 OFDM o 5 escalable OFDM con 128 o 512, 1,024, o 2,048 subcarriers
Modulación	QPSK, 16QAM, 64QAM	QPSK, 16QAM, 64QAM	QPSK, 16QAM, 64QAM
Tasa de datos	32Mbps-134Mbps	1Mbps-75Mbps	1Mbps-75Mbps
Multicanalización	TDD	TDD, FDD	TDD, FDD, OFDMA
Duplexación	TDD and FDD	TDD and FDD	TDD and FDD
Ancho de banda del canal	5MHz-20MHz	5MHz-20MHz	5MHz-20MHz, 3.125MHz, 1.25MHz, 0.75MHz
Denominación de la interfaz aérea	WirelessMAN-SC	WirelessMAN-SCa WirelessMAN-OFDMA	WirelessMAN-SCa WirelessMAN-OFDMA

Tabla 1.1. Datos básicos de la familia de estándares de IEEE 802.16.

1.2 Definición del problema

La tecnología de los sistemas WiMAX está basada en el estándar IEEE 802.16, que es un protocolo de reservación de acceso múltiple y la eficacia del protocolo MAC depende más del ancho de banda asignado a la región de contención y los modos de reservación de acceso, que de los detalles de los algoritmos de resolución de colisiones.

Una red que utiliza un medio de acceso múltiple, requiere de un mecanismo eficiente para compartir el medio físico. Este es uno de los problemas que surgen para la tecnología WiMAX. Es decir, los usuarios que están dentro de la cobertura de una Estación Base (BS)

deben compartir el canal de transmisión, y esto se logra por medio de los protocolos de comunicación. Es en esta parte donde los algoritmos de resolución de colisiones tienen gran importancia.

El ancho de banda para el canal de subida en los sistemas WiMAX está ranurado en “frames” compuestos de “slots” fijos. El *frame* de “uplink” (canal de las SSs hacia la BS-canal de subida) está dividido en Región de Mantenimiento (Ranging inicial), Región de Contención (para solicitudes de ancho de banda) y Región de Reservación (para la transmisión de información de las estaciones).

Las Estaciones Suscriptoras (SSs) compiten en la Región de Contención para reservar ancho de banda en el *uplink* y transmitir su información. En consecuencia, un alto número de *slots* de contención asignados en el *uplink*, reduce el ancho de banda para la transmisión de datos en el área de reservación. Por otro lado, un número pequeño de *slots* de contención da lugar a un número creciente de colisiones con altas cargas del tráfico. Esto último, debido a que múltiples SSs en la red BWA están compitiendo por transmitir sus peticiones de ancho de banda, lo que provoca una degradación en el funcionamiento del sistema.

En el presente trabajo se muestran algunas asignaciones adaptativas de *slots* de contención para sistemas basados en 802.16. Estas técnicas ajustan dinámicamente el número de *slots* de contención necesarios para resolver colisiones, según la carga de tráfico, el número de usuarios que colisionan y las probabilidades de que ocurran éstas, mejorando considerablemente el funcionamiento total del sistema. Los resultados obtenidos se analizan en relación al retardo promedio de acceso (*mean access delay*) y el funcionamiento del sistema (*throughput*), para tráfico IP.

El estándar soporta cuatro calidades de servicio: *Unsolicited Grant Service* (UGS), esta calidad fue diseñada para soportar servicios de tiempo real que generen paquetes de longitud fija en forma periódica, por ejemplo, TI/EI y VoIP sin supresión de silencios; *Real Time Polling Service* (rtPS), diseñada para servicios de tiempo real que generan paquetes de longitud variable en forma periódica, por ejemplo, video MPEG; *Non Real Time Polling Service* (nrTPS), que soporta servicios que no son de tiempo real y que generan paquetes de longitud variable a intervalos regulares de tiempo, tales como FTP; y por último *Best Effort* (BE), este servicio es la calidad más baja que puede ofrecer el estándar, otorga el servicio lo mejor posible y puede utilizarse para tráfico de Internet y correo electrónico.

Es en este último tipo de calidad de servicio, BE, donde se enfoca el caso de estudio de la presente tesis. Las simulaciones que se presentan están basadas para una calidad de servicio BE, ya que es en este servicio donde ocurrirán las colisiones entre las SSs que compiten en la Región de Contención para reservar ancho de banda en el *uplink*.

Para la resolución de colisiones hay varios algoritmos en la literatura, a saber: “*exponential backoff*”, “*p-persistent*”, así como “*splitting tree algorithm*”. Pero el estándar adoptó el *Exponential Backoff Algorithm* (EBA) por su fácil implementación y además muchos sistemas, que utilizan un medio de acceso múltiple, hacen uso de él.

El principal argumento del EBA para resolver una colisión, es definir cuantos *slots* debe dejar pasar, “*backoff*” (retroceso), una SS para volver a realizar su petición de ancho de banda. El número de *slots* que debe dejar pasar es definido por $backoff = random[0, 2^r - 1]$, donde: $r = min\{i + c, t\}$, la variable i es un valor inicial establecido por la BS, c es el número de colisiones que ha sufrido la SS y t es el valor que trunca el crecimiento de la ventana. Para el estándar 802.16 este último parámetro tiene el valor de $t = 10$.

El valor *backoff* es inicializado desde 0 y se actualiza según el usuario haya caído en una colisión, es decir si lo hace una vez entonces el $backoff = random[0,1]$, si vuelve a caer cuando tiene la posibilidad de transmitir, entonces el $backoff = random[0,3]$ y así se incrementa hasta llegar a un valor máximo establecido. El usuario debe escoger un valor aleatorio de la ventana que tiene para transmitir. Entre más grande sea la ventana, significa que el usuario ha caído varias veces en colisión y su tiempo de espera, para transmitir su petición de ancho de banda, se vuelve cada vez más grande. Después de una transmisión exitosa, el *backoff* vuelve al valor inicial.

Una de las limitaciones del EBA es que castiga mucho a los usuarios y más a los que causan colisiones, haciéndolos esperar un tiempo muy grande para volver a intentar la transmisión. Esto provoca un incremento considerable en el retardo promedio de acceso a la red.

Con este trabajo retomamos la idea que presenta Mercedes López Bejarano en su tesis de maestría “Análisis de algoritmos de resolución de colisiones en redes inalámbricas de banda ancha” [22], al mostrar que el algoritmo adoptado por el estándar 802.16, presenta una baja eficiencia en condiciones de congestión de red. Es por ello que se implementarán nuevas técnicas de acceso dinámicas para mejorar el desempeño de una red BWA.

1.2.1 Métodos de evaluación y medidas de desempeño de la eficiencia de una red

Existen dos métodos principales para el modelado de redes. El primero es mediante modelos analíticos utilizando procesos estocásticos basados en probabilidad y teoría de colas [31]. Una desventaja que presenta este modelado es que arroja un análisis muy directo, al considerar o asumir ciertos parámetros que limitan la complejidad del sistema y no poder integrar el comportamiento dinámico de una red. El segundo método es mediante simulaciones por computadora, con esta opción se pueden simular ambientes más reales y seguir con más detalle el comportamiento de una red.

El diseño de simulación se basó especialmente para los sistemas de comunicaciones, los cuales reducen el desarrollo de modelos y el tiempo de análisis, teniendo una mejor precisión. Los errores pueden ser introducidos durante las etapas de diseño y también se pueden corregir con la adición de módulos para lograrlo. Los resultados de simulación se comparan contra los obtenidos por otros métodos de evaluación de desempeño. Claro está que la validación es una gran desventaja, ya que se requiere de un sistema existente para su correcta comparación y validación.

Los requerimientos para medir el análisis de eficiencia de las redes de comunicaciones incluyen:

- ✓ *MEAN ACCESS DELAY* (Retardo promedio de acceso)
El retardo promedio entre que el paquete está listo para ser transmitido hacia un nodo, hasta que es transmitido y recibido en su destino final (en este caso la estación base).
- ✓ *THROUGHPUT* (Máxima capacidad de transmisión de datos)
La tasa de datos que son transmitidos entre los nodos. El análisis es comúnmente hecho en términos del número total de bits transferidos en la capa WiMAX MAC (kbps, Mbps o como el porcentaje de la capacidad del canal).
- ✓ *UTILIZATION* (Utilización de la red)
La fracción de la capacidad total del canal que se usa, (incluyendo datos, protocolos de encabezado, peticiones de reservación, retransmisión y colisiones).

Estos son los requerimientos importantes en la evaluación de una red, de acuerdo a Stallings [29]. Los resultados, para este tipo de parámetros, son generalmente mostrados como una función de la carga ofrecida (la cual es la carga actual o el tráfico demandado en la red) o como una función del número de estaciones activas transmitiendo tráfico en la red. Adicionalmente, hay algunas otras propiedades que afectan el desempeño:

- ✓ Capacidad del canal.
- ✓ Retardo de propagación.
- ✓ Número de bits por *frame*.
- ✓ Protocolo local de la red.
- ✓ Número de estaciones.

Los primeros tres factores listados arriba pueden ser vistos como parámetros que caracterizan la red y generalmente son tratados como constantes. Los protocolos locales de las redes son el punto focal en el diseño de técnicas para estudiar el comportamiento de una red, y principalmente consisten en el acceso al medio y la capa física. Esta última no parece ser un factor relevante, generalmente los datos son transmitidos con un pequeño retardo. La capa de acceso al medio tiene un efecto significativo en el desempeño de la red y es discutido extensamente en esta tesis. Los dos últimos factores tienen que ver con el desempeño, como una función de la carga ofrecida y el número de estaciones activas.

1.3 Objetivos

Es de importancia remarcar que la presente investigación es la continuación de los estudios realizados por López Bejarano [22], que mostró que el EBA presenta una muy baja eficiencia en condiciones de congestión y que el *Splitting Tree Algorithm* (STA) es mucho más eficiente.

El estudio está basado en la evaluación de nuevas técnicas de control de acceso al medio (MAC), utilizando algoritmos de resolución de colisiones (CRA). Cuando dos o más estaciones transmiten sus mensajes de reservación al mismo tiempo, se genera una colisión y el algoritmo adoptado por el estándar IEEE 802.16 para la resolución de colisiones es el EBA. En esta investigación se diseñaron algoritmos de comportamiento dinámico que presentan un mejor desempeño de la red BWA y una mejor resolución dinámica de colisiones. Esto se ve reflejado en una disminución considerable del retardo promedio de acceso a la red.

Esta tesis aborda e investiga el análisis de siete algoritmos dinámicos de resolución de colisiones para el estándar IEEE 802.16 —Simple-CSA, Forced-CSA, Multicast-CSA, Variable-CSA, IEEE 802.14-CSA, Collision Free-CSA y CDMA-CSA—, en un ambiente de simulación de laboratorio con ayuda del software de simulación *OPNET Modeler*, cuyas licencias de operación se obtuvieron por medio del proyecto PAPIIT IN 104907.

Otro objetivo es generar investigación en el ámbito de las telecomunicaciones en nuestro país y que el conocimiento generado no se quede sólo en recintos de estudio. Se espera que los resultados obtenidos sean publicados en foros especializados nacional e internacionalmente.

1.4 Método de investigación

Se diseñó un modelo de simulación para una red BWA basada en el estándar 802.16 con ayuda del software *Optimun Network Performance Modeler (OPNET)*. Simulando tráfico de Internet y soportando una calidad de servicio *Best Effort*, en este modelo asumimos un canal de *uplink* de 6 MHz, modulación QPSK (*Quadrature Phase Shift Keying*) y una tasa de datos, en el *uplink*, de 9.6 Mbps donde las SSs transmiten tráfico de Internet a 32 kbps.

Los algoritmos diseñados fueron simulados con el modelo desarrollado en el laboratorio. Se evaluaron y analizaron cada una de las modificaciones realizadas a las especificaciones adoptadas en el estándar 802.16 para la resolución de las colisiones de los siete esquemas presentados como casos de estudio.

Los resultados obtenidos se compararon contra lo estipulado en el estándar 802.16 y los mostrados por López Bejarano [22].

1.5 Contribución y relevancia

La contribución primordial de esta tesis es el desarrollo de algoritmos dinámicos para la resolución de colisiones en el estándar IEEE 802.16, para que el acceso inalámbrico de banda ancha sea lo mejor posible y tenga un funcionamiento óptimo a una elevada carga de tráfico, además de que el retardo de acceso a la red sea mucho menor.

Por otro lado, contribuir con desarrollo tecnológico del país desarrollando nuevas estrategias que mejoren la capacidad de transmisión en redes WiMAX.

También desarrollar ambientes de tráfico simulados bajo la plataforma OPNET y colocar en la página de *OPNET Technologies* el modelo para que pueda ser utilizado por otros centros de investigación. En la actualidad el grupo de investigación en *OPNET* está formado por más de dos mil universidades a nivel mundial.

1.6 Estructura de la tesis

El capítulo 2 presenta los detalles de las capas MAC y PHY haciendo referencia al protocolo 802.16-2004 para una red punto-multipunto, operando en el modo TDD y FDD para el canal *uplink*. Este último modo es el que caracteriza el modelo de simulación.

El capítulo 3 detalla los algoritmos Simple-CSA, Forced-CSA, Multicast-CSA, Variable-CSA, IEEE 802.14-CSA, Collision Free-CSA y CDMA-CSA que son los casos de estudio de esta tesis. Los esquemas Simple-CSA, Forced-CSA y Multicast-CSA son estudiados por López Bejarano en [22].

En cuanto al capítulo 4 se presenta el modelo de simulación usado y la implementación de cada uno de los esquemas que se proponen en la tesis, descritos en el capítulo 3. Las implementaciones se detallan tanto en el módulo de la estación base (BS), como en el de las estaciones suscriptoras (SSs).

El capítulo 5 muestra los resultados de cada simulación y se discuten comparando los esquemas propuestos con los análisis en López Bejarano [22]. La comparación es modular, incorporando técnica por técnica, desde Simple-CSA hasta colocar todas en una sola figura.

El capítulo 6 discute las conclusiones generales de la tesis, abordando todos los objetivos que se lograron con el análisis de los esquemas analizados y propuestos.

CAPÍTULO 2

Descripción del estándar 802.16

2.1 Introducción

La definición de los servicios y características soportadas por la tecnología de acceso inalámbrico de banda ancha (BWA, *Broadband Wireless Access*) está contemplada en el protocolo IEEE 802.16 mediante las especificaciones correspondientes a las capas de control de acceso al medio (MAC) y física (PHY). Esta es una tecnología de banda ancha, cuya arquitectura punto-multipunto a través de un canal compartido, requiere de un mecanismo eficiente de acceso al medio capaz de soportar distintos niveles de servicio para el transporte de información con retrasos mínimos y un máximo desempeño en la distribución del ancho de banda para cada una de las estaciones transmisoras. Esto requiere de técnicas que permitan ofrecer distintas prioridades para cada una de las conexiones establecidas, pero que sean capaces de garantizar, a todas ellas, oportunidades de transmisión exitosas.

El protocolo IEEE 802.16 atiende estas necesidades mediante la descripción del funcionamiento de la capa MAC. El esquema presentado consiste en el envío periódico de mensajes de control desde la Estación Base (BS, *Base Station*) que determinan el tipo de las transmisiones que podrán realizarse durante un intervalo futuro. Este método se lleva a cabo mediante la combinación de TDMA (Acceso Múltiple por División de Tiempo) y DAMA (Acceso Múltiple por Asignación a la Demanda). La combinación de ambos mecanismos permite alcanzar tasas de transferencia máxima que fluctúan entre 32 y 130 Mbps dependiendo del ancho del canal y de la técnica de modulación. Para obtener la tasa de transmisión se utiliza la siguiente expresión:

$$R = \frac{m * BW}{1 + factor\ roll_off}$$

donde: m es el número de bits por símbolo, BW es el ancho de banda del canal en Hz y el factor *roll off* = 0.25 es un parámetro de exceso de ancho de banda para redes inalámbricas.

En este capítulo se abordan las especificaciones indicadas por el protocolo 802.16 para la capa MAC y la capa PHY. En cuanto a la capa PHY se aborda el modo de operación TDD y FDD; este último es el modo en el cual se estudia el comportamiento de los mecanismos propuestos para este trabajo; las tasas de transmisión y las tasas de codificación manejadas en capa PHY.

Para la parte de la capa MAC se muestra cómo se forman los paquetes de capa MAC y su fragmentación y concatenación para su envío por el medio físico. Una explicación de los

diferentes tipos de calidad de servicio que ofrece el estándar. Las diferentes formas en que una SS puede hacer sus peticiones de ancho de banda y los tipos de señalización. En este capítulo no se aborda el tema de cómo el estándar 802.16 resuelve el problema de contención, ya que el tema principal de esta tesis es el desarrollo de técnicas de resolución de colisiones dinámicas, en el capítulo 3 se presenta a detalle el funcionamiento del esquema adoptado por el estándar 802.16.

Para ayudar a comprender lo que se discute en este capítulo, se presenta la tabla 2.1, que contiene los términos técnicos, del estándar 802.16, más utilizados y una breve descripción de su significado.

Término	Descripción
<i>Frame</i>	Paquete de datos que se transmite en un canal de comunicación.
<i>Downlink</i>	Paquete de datos con dirección de la BS a las SS.
<i>Uplink</i>	Paquete de datos con dirección de las SS a la BS.
<i>Slot</i>	Recurso mínimo que puede asignarse a una SS en la capa PHY.
<i>Backoff</i>	Intervalo de tiempo que se espera para realizar una transmisión de un paquete de datos en un canal.
<i>Scheduler</i>	Algoritmo de la BS que asigna los recursos en un canal a una SS.
<i>Subheaders</i>	Paquetes de datos específicos que se transmiten en un canal.
<i>Feedback</i>	Término utilizado para indicar que existe una retroalimentación.
<i>Grant</i>	Permiso de recursos de comunicación que se otorga a una SS.
<i>Piggyback</i>	Se utiliza para determinar que una SS hace una petición de ancho de banda al final de un paquete de información en un área reservada.
<i>Overhead</i>	Paquetes de datos que se transmiten en un canal de comunicación, pero que no contienen información útil para las SS.

Tabla 2.1. Términos usados para la descripción del protocolo 802.16.

2.2 Capa Física

La capa física se basa en OFDM (*Orthogonal Frequency Division Multiplexing*), un esquema que permite la transmisión de datos de alta velocidad, vídeo y comunicaciones multimedia que además es utilizado por varios sistemas comerciales como ADSL (conexiones que siguen el estándar G.DMT, ITU G.992.1 [16]), Wi-Fi, *Digital Video Broadcast-Handheld* (DVB-H [32]) y MediaFLO [21]. OFDM es un esquema eficiente para tasas de transmisión altas en un ambiente de multitrayectoria o NLOS.

2.2.1. Estructura de *frame* y *slot*

El recurso mínimo de tiempo que se puede asignar a una conexión dada, por un sistema basado en el 802.16, se llama *slot* (ranura). La capa PHY es responsable de la asignación de *slots* y el envío de ellos. La figura 2.1 muestra un *frame* OFDM cuando opera en modo de TDD y FDD. En el modo TDD, el *frame* se divide en dos *subframes*: un *frame* de “*downlink*” (DL), después de un intervalo pequeño de guarda (TTG), y finaliza con un *frame* de “*uplink*” (UL), figura 2.1 (a). La relación de transformación del *frame* de *downlink* a *uplink* puede variar de 3:1 a 1:1 para soportar diversos perfiles del tráfico. El estándar también soporta FDD, en cuyo caso la estructura del *frame* es la misma sólo que el *downlink* y *uplink* se transmiten simultáneamente, utilizando un canal para *downlink* y otro canal diferente para *uplink*, como se muestra en la figura 2.1. (b).

El *frame downlink* (fig. 2.2) comienza con un preámbulo de *downlink* que es usado para los procedimientos de capa PHY, tales como sincronización de tiempo y frecuencia y valoración inicial del canal. El preámbulo de *downlink* es seguido por una cabecera de control de *frame* (FCH), que proporciona información sobre la configuración del *frame*, tal como la longitud de los mensajes de mapa de *uplink* y *downlink*, (DL-MAP y UL-MAP, mensajes que informan la estructura de cada *frame*), modulación, esquema de codificación, y los *subcarriers* utilizables. Después de la cabecera FCH, siguen los perfiles de las ráfagas para una o varias SSs transportando información solicitada por las SSs.

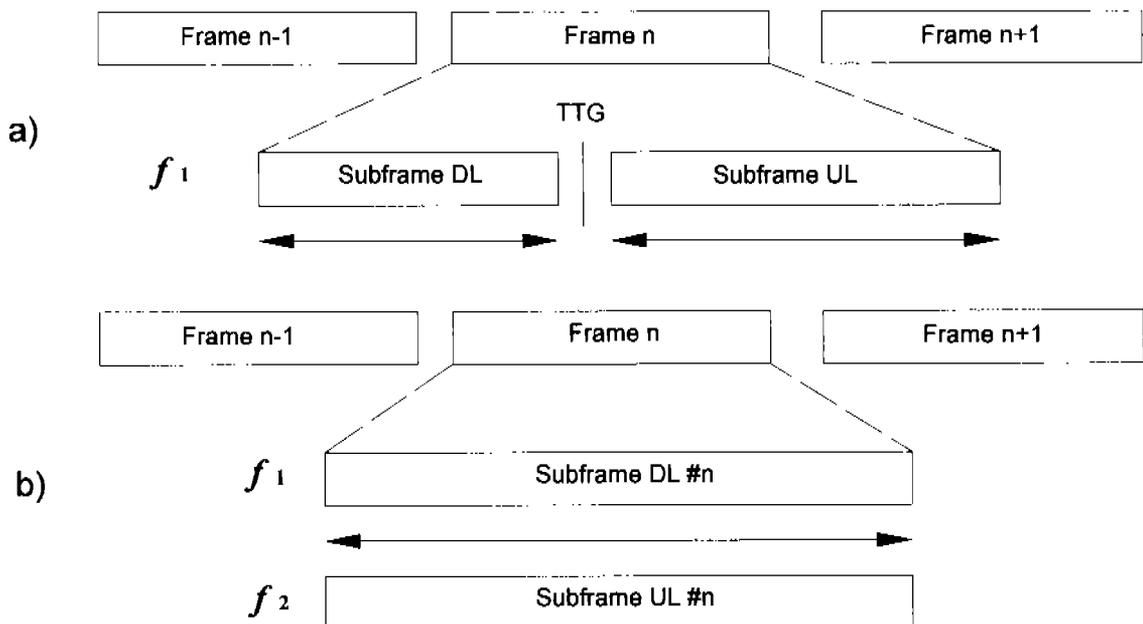


Figura 2.1. Representación de la estructura de un *frame* en modo TDD (a) y FDD (b).

Múltiples usuarios son asignados en regiones de datos dentro del *frame*, y estas asignaciones se especifican en los mensajes MAP, que contienen el perfil de la ráfaga para cada estación, definiendo el esquema de modulación y codificación usado en la conexión.

Puesto que los mensajes MAP contienen información crítica para todos los usuarios, se envía a menudo sobre una conexión muy confiable, tal como BPSK con la tasa codificación de 1/2, para que los usuarios más distantes de la BS puedan leer los mensajes de señalización, control y reservación incluidos en el MAP.

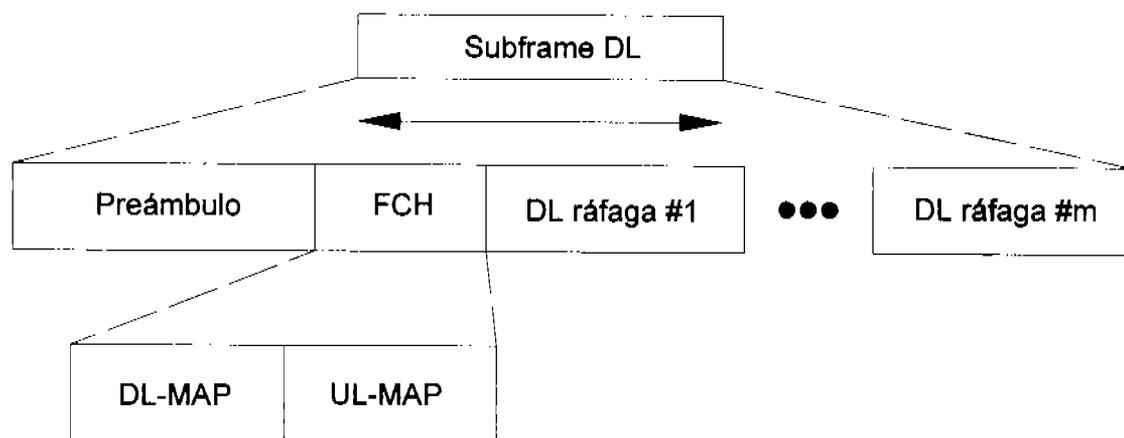


Figura 2.2. Representación de la estructura de un *subframe* DL.

El estándar 802.16 es absolutamente flexible para multicanalizar muchos usuarios y paquetes en un solo *frame*. Un solo *frame* de *downlink* puede contener múltiples ráfagas de datos de varios tipos y tamaños para diferentes estaciones. El tamaño del *frame* es también variable desde 2 ms hasta 20 ms, y cada ráfaga puede contener paquetes o fragmentos de tamaño fijo o variable concatenados.

El *frame* de *uplink* (fig. 2.3) está compuesto por tres regiones significativas. Una región de *slots* de contención para calibración y mantenimiento, utilizada para el proceso de registro e inicialización en la red. Después, le sigue una región de *slots* de contención, para que las SSs realicen sus peticiones de ancho de banda. Por último, una o varias regiones reservadas, donde las SSs envían su información de datos.

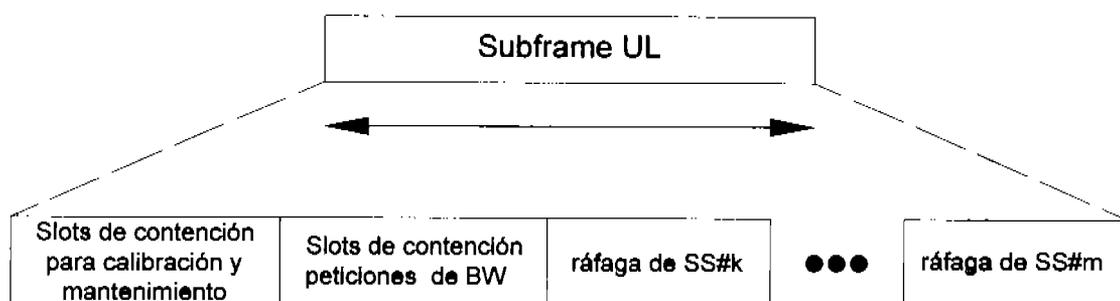


Figura 2.3. Representación de la estructura de un *subframe* de *uplink*.

2.2.2. Codificación y modulación adaptativa

El protocolo 802.16 utiliza una variedad de esquemas de modulación y codificación y permite que los esquemas cambien de conexión, dependiendo de las condiciones del canal. Para el *uplink*, la BS puede estimar la calidad del canal dependiendo la calidad de la señal recibida. El *scheduler* de la BS puede analizar la calidad del *uplink* y *downlink* de cada SS y asignar un esquema de modulación y codificación, que maximice el *throughput* para la relación señal a ruido disponible. La modulación y codificación adaptativa aumenta considerablemente la capacidad total del sistema, debido a que permite la compensación, en tiempo real, entre el *throughput* y la robustez de cada conexión.

La tabla 2.2 enlista varios esquemas de modulación y codificación para el protocolo 802.16. En el *downlink*, QPSK, 16 QAM, y 64 QAM son obligatorios para WiMAX fijo y móvil; 64 QAM son opcionales en el canal *uplink*. La codificación FEC (*Forward Error Correction*) que usa códigos convolucionales es obligatoria. Un total de 52 combinaciones de los esquemas de modulación y codificación se definen en el estándar 802.16.

	<i>Downlink</i>	<i>Uplink</i>
Modulación	BPSK, QPSK, 16QAM, 64QAM Obligatorio: códigos convolucionales con una relación de 1/2, 2/3, 3/4, 5/6	BPSK, QPSK, 16QAM, 64QAM Obligatorio: códigos convolucionales con una relación de 1/2, 2/3, 3/4, 5/6
Codificación	Turbo o Viterbi convolucionales con una relación de 1/2, 2/3, 3/4, 5/6 y códigos Reed-Solomon para OFDM	Turbo o Viterbi convolucionales con una relación de 1/2, 2/3, 3/4, 5/6 y códigos Reed-Solomon para OFDM

Tabla 2.2. Esquemas de modulación y codificación de DL y UL.

2.2.3. Tasas de transmisión de capa PHY

Ya que la capa PHY es absolutamente flexible, el funcionamiento de la tasa de datos varía conforme a los parámetros de operación. Los parámetros que tienen un impacto significativo en la tasa de datos de capa física son el ancho de banda del canal y el esquema de modulación y codificación usado, así como el número de *subcarriers* y el tiempo de guarda de OFDM.

La tabla 2.3 enlista la tasa de datos de capa PHY para varios anchos de banda del canal, así como la modulación y esquemas de codificación. Para el cálculo de las tasas binarias que se conseguirán con el uso de 256 *subcarriers* para SSs fijas y 128, 512 y 1024 *subcarriers* para SSs móviles, el estándar IEEE 802.16 proporciona la siguiente expresión:

$$R = \left(\frac{8 * BW}{7} * b_m * c_r * N_{utilFFT} \right) * \left(\frac{1}{N_{FFT} + \frac{N_{FFT}}{T_g}} \right)$$

Donde b_m es el número de bits por símbolo, c_r es la tasa de codificación, BW es el ancho de banda del canal en MHz, $N_{utilFFT}$ es el número de *subcarriers* utilizables, N_{FFT} es el número de *subcarriers* totales para el esquema usado. En OFDM, $N_{utilFFT} = 196$ y $N_{FFT} = 256$. Por otra parte en OFDMA, $N_{utilFFT} = 1536$ y $N_{FFT} = 2048$). T_g , es el tiempo de guarda para el esquema utilizado.

Ancho de banda del canal	3.5MHz		1.25MHz		5MHz		10MHz		5.75MHz	
Esquema de capa PHY	OFDM		OFDM		OFDMA		OFDMA		OFDMA	
Oversampling	8/7		28/25		28/25		28/25		28/25	
Modulación y Tasa de código	Tasa de transmisión de capa PHY (kbps)									
	DL	UL	DL	UL	DL	UL	DL	UL	DL	UL
BPSK, 1/2	946	375								
QPSK, 1/2	1,882	653	304	154	2,520	611	5,040	1,344	4,094	1,120
QPSK, 3/4			704	251		870	7,560	2,016	6,146	1,680
16 QAM, 1/2	3,763	1,306	1,008	307	5,040	1,306	10,080	2,688	8,288	2,240
16 QAM, 3/4			1,512	461	7,560	1,958	15,120	4,032	12,372	3,360
64 QAM, 1/2	5,645	1,958	1,512	461	7,560	1,958	15,120	4,032	12,372	3,360
64 QAM, 3/4	7,526	2,611						5,376	11,356	4,480
64 QAM, 5/6	8,467	2,938	2,268	691	11,340	2,938	22,680	6,048	20,712	5,040
64 QAM, 5/6	9,408	3,264					25,200	6,720	26,400	6,720

Tabla 2.3. Tasas de transmisión de capa PHY

2.3 Capa MAC

En una red, el propósito de la capa PHY es la entrega de los bits de información del transmisor al receptor, usando un medio físico como radiofrecuencia, ondas de luz o alambres de cobre. Generalmente, la capa PHY no tiene información de los requisitos de la calidad de servicio (QoS) y no está enterada de la aplicación, tal como VoIP (*Voice over Internet Protocol*), HTTP (*HyperText Transfer Protocol*), o FTP (*File Transfer Protocol*). La capa PHY se puede ver como un tubo, responsable del intercambio de información sobre una sola conexión entre un transmisor y un receptor.

La capa MAC es responsable de controlar y multiplexar varias conexiones sobre el mismo medio físico. Esta capa es orientada a conexión e identifica una conexión lógica unidireccional entre la BS y la SS por un CID (*Connection Identifier*). Los CIDs para las conexiones de *uplink* y *downlink* son diferentes. El CID se puede ver como un direccionamiento temporal y dinámico, de capa 2, asignado por la BS para identificar una conexión unidireccional entre las capas MAC y PHY, y se utiliza para el control del tráfico de los datos. Para asociar ese direccionamiento, la subcapa de convergencia (CS) no debe perder de vista la asociación entre la dirección destino y el respectivo CID.

La capa MAC, que reside sobre la capa PHY, como se puede observar en la figura 2.4, del protocolo 802.16 se divide en tres subcapas:

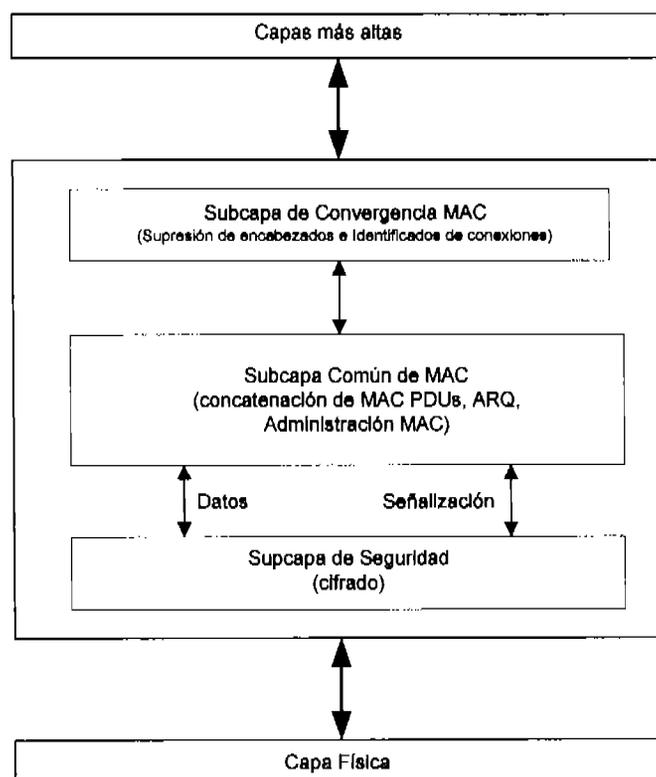


Figura 2.4. Estructura de la capa MAC del protocolo 802.16.

- ✦ **Subcapa de convergencia (CS).** Es la interfaz entre la capa MAC y la capa 3 (red). Recibe los paquetes de datos de las capas más altas. Estos paquetes también se conocen como *MAC Service Data Units (SDU)*. La CS es responsable de realizar todas las operaciones que sean dependientes de la naturaleza del protocolo de la capa más alta, tal como compresión de cabeceras y direccionamiento. La CS se puede ver como capa de adaptación que enmascara el protocolo de capas más altas.
- ✦ **Subcapa común.** Realiza todas las operaciones del paquete que son independientes de capas de más altas, tales como: fragmentación y concatenación de SDUs en MAC

PDU (*Protocol Data Units*), programar la transmisión de MAC PDUs sobre los recursos disponibles de PHY, control de la QoS y la retransmisión de los MAC PDUs que fueron recibidos erróneamente por el receptor cuando se utiliza ARQ (*Automated Repeat Request*, un mecanismo de control de errores para la transmisión de datos).

- ✦ **Subcapa de seguridad.** Responsable del cifrado, autorización y del intercambio apropiado de las claves de cifrado entre la BS y el SS.

2.3.1 Construcción y transmisión de MAC PDU

De acuerdo con el tamaño de la carga útil, múltiples SDUs pueden ser enviados en sólo un MAC PDU, o un SDU se puede fragmentar para ser transportado por múltiples MAC PDUs. Cuando un SDU se fragmenta, la posición de cada fragmento del SDU es etiquetada por un número de secuencia. El número de secuencia permite a la capa MAC, en el receptor, ensamblar los fragmentos del SDU en el orden correcto.

Para utilizar eficientemente los recursos de capa PHY, múltiples MAC PDUs son destinados al mismo receptor para ser concatenados y transportados en una sola oportunidad de transmisión o región de datos, según lo mostrado en la figura 2.5.

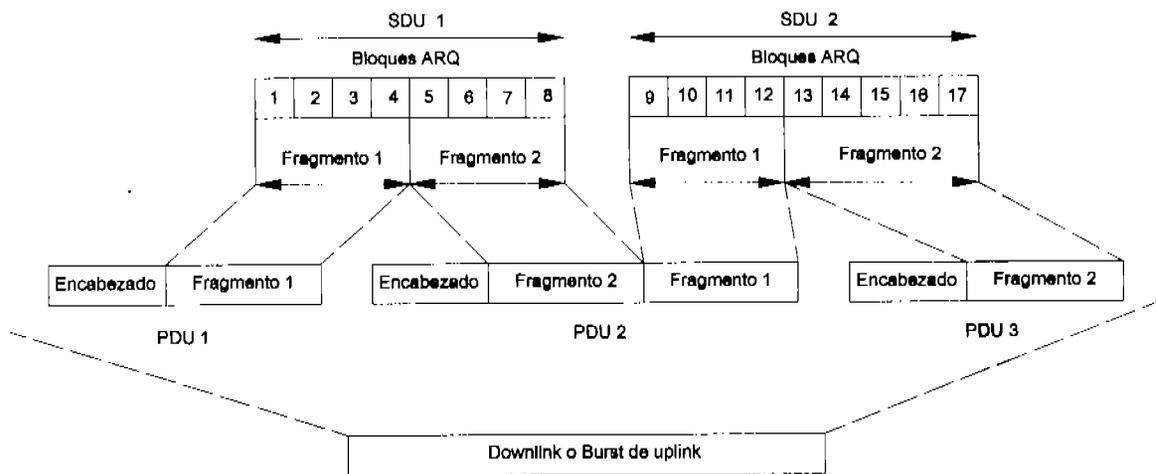


Figura 2.5. Fragmentación y concatenación de SDUs en MAC PDUs.

Las regiones de datos del *uplink* y *downlink* de una SS son un conjunto de *slots* adyacentes reservados para sus oportunidades de transmisión. Para las conexiones que no usan ARQ, cada fragmento del SDU se transmite en secuencia. Para las conexiones que usan ARQ, el SDU primero se fragmenta en bloques ARQ de longitud fija, y se les asigna un número secuencial de bloque (BSN) de ARQ. La longitud de los bloques de ARQ es especificada por la BS para cada CID, usando el parámetro ARQ-BLOCK-SIZE. Si la longitud del SDU no es un múltiplo de ARQ-BLOCK-SIZE, el último bloque ARQ se rellena. Después de que se fragmenta en bloques ARQ, el SDU se ensambla en MAC PDUs según lo mostrado en la figura 2.5.

Para las conexiones que tienen habilitado el ARQ, el encabezado de fragmentación y empaquetado contiene el BSN del primer bloque de ARQ y del siguiente. La retroalimentación ARQ del receptor viene en forma de ACK (*acknowledgment*), indicando la recepción apropiada de los bloques de ARQ. Esta retroalimentación se envía como un MAC PDU independiente o como *piggyback* en la carga útil de una MAC PDU normal. En el estándar 802.16, la retroalimentación de ARQ puede estar en la forma de ACK selectivo o de ACK acumulativo. Un ACK selectivo, para un BSN dado, indica que el bloque de ARQ se ha recibido sin errores. Por otra parte, un ACK acumulativo, para un BSN dado, indica que todos los bloques con números de serie menores o iguales al BSN se han recibido sin error.

El MAC PDU posee tres campos como se puede apreciar en la figura 2.6. El primer campo es de tamaño fijo y corresponde al *Generic MAC Header* (Encabezado Genérico), el segundo campo es opcional, de longitud variable, y corresponde a la carga útil (*Payload*) que puede tener subencabezados, mientras que el tercer campo correspondiente al CRC (*Cyclic Redundancy Check*), es también opcional y protege tanto a la cabecera como a la carga útil.

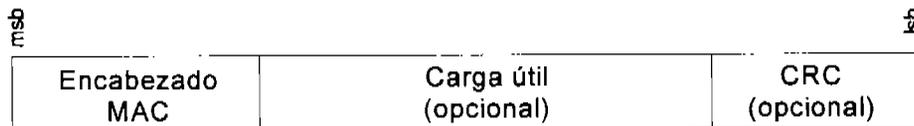


Figura 2.6. Estructura de MAC PDU.

El estándar 802.16 define dos tipos de encabezados de MAC PDUs, cada uno con una estructura diferente, según lo mostrado en la figura 2.7.

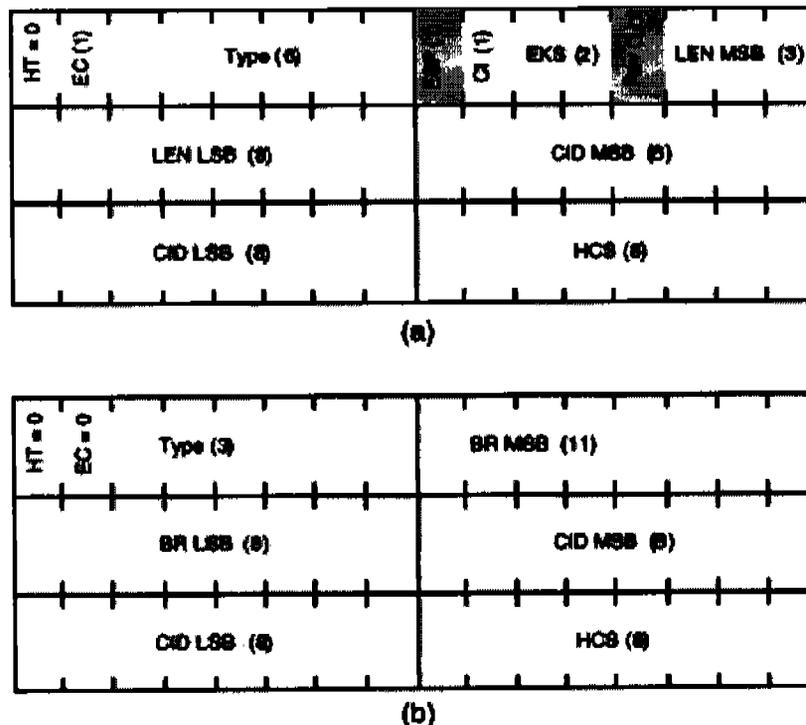


Figura 2.7. Encabezados de PDU: (a) genérico; (b) petición de ancho de banda

La MAC PDU genérica (Fig. 2.7a) se utiliza para transportar los datos y mensajes de señalización de capa MAC. Los elementos de información en la cabecera de una MAC PDU genérica se muestran en la tabla 2.4.

Campo	Longitud (bits)	Descripción
HI	1	Tipo de encabezado (0 para encabezado genérico)
FC	1	Control de cifrado (FC=0: no cifrado; 1: cifrado)
Type	2	Tipo de MAC PDU
ESF	1	Código de encabezado extendido (ESF). 1: está presente
CI	1	Indicador de CRC (CI). 1: presente; 0: no presente
EKS	2	Secuencia de la clave de cifrado (EKS)
Rsv	1	Reservado
LEN	16	Longitud en bytes del MAC PDU, incluyendo la cabecera
CID	8	Identificador de canal (CID)
HCS	8	Secuencia de verificación del encabezado

Tabla 2.4. Campos de la cabecera MAC genérica.

La MAC PDU de petición de ancho de banda (fig. 2.7b) es utilizada por las SSs para indicar a la BS cuánto ancho de banda requieren en el *uplink* para el envío de datos. Una MAC PDU de petición de ancho de banda consiste solamente en una cabecera de petición de ancho de banda, sin carga útil o el CRC. Los elementos de información de una cabecera de la petición de la anchura de banda se proporcionan en la tabla 2.5.

Campo	Longitud (bits)	Descripción
HI	1	Tipo de encabezado (0 para encabezado genérico)
FC	1	Control de cifrado (se fija en 0 para este encabezado)
Type	2	Tipo de MAC PDU
BR	16	Petición de ancho de banda (el número de bytes de ancho de banda del <i>uplink</i> solicitados por las SSs)
CID	8	Identificador de canal (CID)
HCS	8	Secuencia de verificación del encabezado

Tabla 2.5. Campos de la cabecera MAC para peticiones de ancho de banda.

Aparte de los encabezados genéricos y de petición de ancho de banda, el protocolo 802.16 también define cinco encabezados (*subheaders*) que se puedan utilizar en un MAC PDU genérico:

- ✦ *Subheader Mesh*. Cuando se utiliza una red de tipo Mesh.
- ✦ *Subheader* de fragmentación. Indica que el SDU está fragmentado en múltiples MAC PDUs.
- ✦ *Subheader Packing*. Este encabezado de empaquetamiento, indica que múltiples SDUs o fragmentos de SDU están empaquetados en una sola MAC PDU y están colocados al principio de cada SDU o fragmentos de SDU.
- ✦ *Fast-feedback allocation subheader*. El encabezado de asignación de retroalimentación rápida, indica que la PDU contiene retroalimentación del SS sobre la información del estado del canal DL. Proporciona la funcionalidad de retroalimentación de información del estado del canal para sistemas MIMO (*Multiple-Input and Multiple-Output*) y non-MIMO.
- ✦ *Grant-management subheader*. El encabezado de administración de concesión, es utilizado por la SS y transporta los mensajes relacionados con el manejo del ancho de banda, tal como peticiones por consulta y peticiones de ancho de banda adicional. Usar este encabezado es más eficiente que la petición de ancho de banda inicial, es más compacta y no requiere la transmisión de una PDU nueva.

Una vez que se construye un MAC PDU, se entrega al *scheduler*, que programa el MAC PDU sobre los recursos disponibles de capa PHY. El *scheduler* controla el ID del flujo de servicio y el CID de la MAC PDU, que permite que calibre sus requisitos de QoS. De acuerdo con los requisitos de QoS del MAC PDUs que pertenece a diferentes CIDs y IDs de flujo de servicio, el *scheduler* determina la asignación de recursos óptimos de capa PHY para todos los MAC PDUs, sobre los *frames*.

2.3.2. Calidad de servicio

Una de las funciones dominantes de la capa MAC es asegurarse de que los requisitos de QoS, para MAC PDUs, estén asignados dadas las condiciones de la carga del sistema. Esto implica que el tiempo de espera, la tasa de datos, la tasa de error del paquete y disponibilidad de sistema, deben ser óptimos para cada conexión. Se tienen tres mecanismos para la caracterización de QoS: solicitudes anidadas (*Piggyback*), robo de Ancho de Banda (*Bandwidth Stealing*) y consulta (*Polling*).

Piggyback Request

Se trata de solicitudes de asignación de ancho de banda incluidas en los encabezados de los paquetes enviados por las SSs durante una transmisión garantizada. Es decir, cuando la SS ha hecho una solicitud previa en el periodo de contención y la BS le ha proporcionado una cantidad específica de *slots* de transmisión, es posible que la SS aún tenga paquetes por enviar que no habían sido considerados cuando se realizó la asignación. Para evitar hacer una solicitud por contención nuevamente, la SS incluye en los encabezados de los paquetes que está enviando una solicitud adicional. Esta nueva solicitud tiene carácter preferencial, ya que su recepción en la BS está garantizada, aunque es posible que no se realice la asignación en su totalidad.

Bandwidth stealing

Se refiere al empleo de una porción del ancho de banda asignado a una SS para solicitudes adicionales. Este proceso tiene lugar cuando una SS se encuentra operando bajo un esquema de Asignación Garantizada por SS, esto es, que cuenta con oportunidades garantizadas para la transmisión de paquetes. En este caso, es posible que parte de estas oportunidades, destinadas para el envío de datos, se use para realizar solicitudes adicionales de ancho de banda. El término *stealing* se refiere a que parte de los *slots* destinados para enviar su información se usa para hacer solicitudes de ancho de banda.

Polling

Es el proceso mediante el cual la BS destina a una SS, o a una conexión, oportunidades específicas para el envío de solicitudes de ancho de banda. En este caso, la BS asigna regularmente algunos *slots* que la SS puede usar para hacer solicitudes de ancho de banda. Dichas oportunidades son independientes de aquellas que se asignan normalmente para las transmisiones de las SSs. Adicionalmente, la SS puede indicar que requiere de este servicio.

Las funciones anteriormente mencionadas permiten que el estándar 802.16 soporte la clasificación de cuatro tipos de QoS para las transmisiones en el *uplink*, como se describe en la tabla 2.5.

Calidad de Servicio (QoS)	Solicitud Piggyback	Robo de ancho de banda	Polling
UGS Servicio Garantizado no Solicitado (<i>Unsolicited Grant Service</i>)	Prohibida	Prohibida	Solicita transmisiones de UGS
rtPS Servicio de Consulta en Tiempo Real (<i>Real Time Polling Service</i>)	Permitida	Permitida para Asignaciones Garantizadas por Estación Suscriptor (GPSS, Grant per SS)	Solo se permite la consulta unicast
nrtPS Servicio de Consulta no en Tiempo Real (<i>Non Real Time Polling Service</i>)	Permitida	Permitida para RTGS	Reservado para consultas unicast de todos los tipos de servicios
BE Mejor Esfuerzo (<i>Best Effort</i>)	Permitida	Permitida para GPSS	Todas las formas de polling permitidas

Tabla 2.5. Tipos de servicio para transmisiones en el UL.

Servicio *Unsolicited Grant Service* (UGS)

Soporta servicios de tiempo real que generen paquetes de longitud fija en forma periódica, por ejemplo: TI/EI y VoIP sin supresión de silencios. La asignación de oportunidades de transmisión periódica y fija en tiempo real, permite reducir la latencia, sin dejar de cumplir con las demandas de la comunicación en tiempo real. La política de solicitud y transmisión impide a la SS utilizar las oportunidades de contención para la asignación de ancho de banda. Adicionalmente, la BS no podrá proporcionar a la conexión que usa este servicio la oportunidad *unicast* para realizar solicitudes. Por este motivo, la SS sólo podrá transmitir tráfico constante en oportunidades no solicitadas para esa conexión en particular.

Servicios en tiempo real (rtPS)

Soporta servicios de tiempo real que generan paquetes de longitud variable, en forma periódica, por ejemplo: Video MPEG. Mediante este servicio se ofrecen oportunidades periódicas *unicast* que permiten a la SS especificar la longitud deseada para cada una de las asignaciones. Puede mejorarse la eficiencia en la utilización del canal inalámbrico. La periodicidad de las asignaciones de oportunidades de transmisión debe cubrir los requerimientos del tráfico generado por la SS. Cabe mencionar que también es posible que la SS emplee el UGS, sin embargo, no está permitido que emplee la región de contención para el envío de solicitudes de asignación de ancho de banda para esa conexión en particular.

Servicios no en tiempo real (nrtPS)

Soporta servicios que no son de tiempo real y que generan paquetes de longitud variable a intervalos regulares de tiempo, tales como FTP. La BS deberá proporcionar periódicamente, durante intervalos menores a un segundo, oportunidades de transmisión para la SS, aún en el caso de congestión en la red. De acuerdo con la política de solicitud y transmisión, en este servicio la SS puede emplear los intervalos de contención para hacer solicitudes adicionales de ancho de banda, así como oportunidades *unicast* y del tipo UGS.

Servicio del mejor esfuerzo (BE)

No reserva ancho de banda, es decir, no asegura tasas de transmisión. Asigna tasas de transmisión disponibles y no da prioridad. Este tipo de servicio es característico del tráfico intermitente de IP. En donde, la SS puede emplear cualquier tipo de oportunidad para realizar solicitudes de asignación de ancho de banda, como son las regiones de contención y las oportunidades *unicast*. Es en este tipo de QoS en el cual se analizan los esquemas propuestos en la presente tesis.

Todos los tipos de servicio requerirán de aumentos o disminuciones de forma dinámica del ancho de banda asignado, exceptuando el UGS el cual tiene una tasa fija de transmisión asignada. Por esto, la BS debe permitir a las SSs que informen sus requerimientos de ancho de banda.

2.3.3. Solicitudes y asignación de ancho de banda

Con la finalidad de tener acceso al canal inalámbrico, cada una de las SSs participan en un mecanismo de solicitud de ancho de banda, a través del cual se le indica a la BS la necesidad de incrementar o disminuir los requerimientos de ancho de banda, de acuerdo con el tipo de tráfico generado.

Los mecanismos mediante los cuales las SS solicitan a la BS la asignación de ancho de banda de acuerdo con los requerimientos específicos de cada conexión se conoce como Acceso Múltiple por Asignación de Demanda (*Demand Assigned Multiple Access, DAMA*). Cada una de las SSs envía mensajes para solicitar la reservación de *slots* en el *uplink*. Las oportunidades de transmisión requeridas dependen del tipo de tráfico generado así como los parámetros de QoS especificados durante el establecimiento inicial de la conexión. Existen diversos métodos empleados por las SSs para realizar sus peticiones: Solicitudes, Reservaciones por Conexión, Reservaciones por Estación Suscriptora y Consulta.

Solicitudes de ancho de banda

Las solicitudes consisten en mensajes enviados por las SSs hacia la BS para solicitar la reservación de ancho de banda en el *uplink* para la transmisión de datos. Debido a las características dinámicas del canal inalámbrico, las solicitudes se expresan en términos de *bytes* sin tomar en cuenta el *overhead* de la capa física. Estas solicitudes pueden transmitirse durante el periodo de contención de solicitud de ancho de banda, o bien aprovechando algún periodo asignado por la BS para la conexión.

Las solicitudes realizadas durante una región reservada se realizan mediante los encabezados de solicitudes *piggyback*, que permiten optimizar el uso del canal inalámbrico, ya que reducen la posibilidad de que ocurran colisiones durante el envío múltiple de solicitudes de asignación de ancho de banda por varias estaciones.

Cabe señalar que las solicitudes pueden ser Incrementales o Totales, lo cual se indica en el campo *Type* del encabezado del mensaje de solicitud de ancho de banda. Dado que los encabezados para las peticiones *piggyback* no cuentan con el campo *Type*, se ha definido que siempre serán del tipo Incrementales. La BS cuenta con un registro de las peticiones realizadas por cada SS. En el caso de las solicitudes Incrementales, dicho registro del ancho de banda solicitado sólo se modifica de acuerdo con los mensajes enviados, esto es, cada SS indica que para la cantidad de *bytes* requeridos con anterioridad es necesario agregar otra cantidad, como lo indique la solicitud Incremental.

Por otro lado, en las Solicitudes Totales, la BS renueva su registro del ancho de banda solicitado por la SS, de acuerdo con la última petición enviada, sin importar que del ancho de banda requerido anteriormente sólo se haya reservado una porción. Debido a las características dinámicas del canal, la BS puede requerir que se generen solicitudes Totales bajo un esquema periódico, lo cual dependerá de los parámetros de QoS para un servicio en particular, así como de las características del canal. Es importante señalar que las solicitudes realizadas durante las regiones de contención siempre deberán ser Totales.

Modo de Asignación por Conexión (GPC)

En las Asignaciones por Conexión (GPC, *Grant per Connection*) el ancho de banda es asignado explícitamente para una conexión y las SSs usan el ancho de la asignación sólo para esa conexión.

Modo de Asignación por Estación Subscriptora (GPSS)

Con el modo de asignación por estación subscriptora, a las SSs se les otorga ancho de banda adicional dentro de la misma asignación. En este modo, las SS deberán ser más inteligentes en su manejo de calidad de servicio. Típicamente usará el ancho de banda para la conexión que la solicitó, pero no necesariamente. Además si la calidad de servicio de la SS ha cambiado desde la última petición, la SS tiene la opción de mandar los datos con mayor calidad de servicio.

Consultas

Mediante las consultas, la BS establece periodos para la transmisión en el *uplink* destinados específicamente para que las SSs envíen solicitudes de asignación de ancho de banda. Ya que la BS no envía mensajes explícitos para indicar tales oportunidades, dichos periodos se definen mediante los correspondientes elementos de información dentro de los UL-MAPs.

Las asignaciones de estas oportunidades pueden realizarse para una sola SS (*Unicast*) o para un grupo de SSs o de conexiones (*Multicast*). En este último caso, las estaciones o conexiones involucradas tendrán que contender para conseguir el envío de sus solicitudes. En base al tipo de consulta, la asignación podrá ser del tipo GPC o GPSS.

A continuación se mencionan las características para Consultas de tipo *Unicast* y de tipo *Multicast*.

- ✦ **Consulta *Unicast*:** Cuando se establece una Consulta *Unicast* para una SS en particular, la BS incluye en el UL-MAP una oportunidad exclusiva para la SS que podrá ser empleada sólo para solicitar ancho de banda.

- ✦ **Consulta Multicast:** Cuando la BS no dispone de ancho de banda suficiente para atender a cada una de las SS asociadas a ella, puede emplear consultas a un grupo de SSs. Tales grupos se identifican mediante CIDs exclusivos y las oportunidades que reciben se especifican a través del UL-MAP.

Las SSs involucradas sólo envían solicitudes si lo necesitan, participando en contenciones con las otras SSs de su grupo. Debido a que se encuentran sujetas a la ocurrencia de colisiones. En el caso de haberse presentado una colisión emplean el algoritmo *exponencial backoff* para determinar el *slot* en el que enviarán su solicitud para reprogramarla. Las solicitudes se considerarán exitosas sólo si en algún futuro UL-MAP se incluye una reservación para la CID en cuestión.

2.4. Conclusiones

En este capítulo se describieron las características de la capa PHY, las estructura de los *frames* de *uplink* y *downlink*. También se detallaron las diferentes formas de codificación y modulación adaptativa que soporta el estándar. También una tabla de cuáles son las tasas de transmisión que soporta el estándar en los diferentes modos de operación, ya sea OFDM, para usuarios fijos u OFDMA, para usuarios móviles, además de la expresión utilizada para obtener dichas tasas de transmisión.

También se detalló que la capa MAC proporciona una configuración flexible que puede utilizar eficientemente una variedad de requisitos de QoS. Se observaron las subcapas en que se divide la capa MAC y la funcionalidad de cada una de ellas. Se describió la forma en que la capa MAC concatena o fragmenta los paquetes de información PDU, dadas las características del medio físico.

Otra cosa importante es la forma en que las SSs pueden comunicarse con la BS para realizar sus peticiones de ancho de banda, además de los diferentes niveles de QoS que el estándar soporta.

CAPÍTULO 3

Algoritmos de resolución de colisiones

3.1 Introducción

Los algoritmos de resolución de colisiones o por sus siglas en inglés CRA (*Collision Resolution Algorithms*), definen las reglas usadas para resolver las colisiones. Estos juegan un papel muy importante en el desempeño de los protocolos de reservación de acceso múltiple. Esto es porque entre más rápido se resuelva la colisión, entonces el retardo será menor y por ende se obtendrá una mayor eficiencia para el sistema. Los algoritmos de resolución de contenciones toman mucho interés en los años 70's especialmente durante el desarrollo del proyecto ALOHANET [1]. Desde entonces, muchos trabajos de investigación se han dedicado a idear mecanismos eficientes para la resolución de contenciones de sistemas de acceso múltiple de redes LAN, MAN y satelitales.

En Jaquet [17] y Sala [25] y [26], los autores puntualizan que el desempeño de los protocolos de reservación de acceso múltiple depende en mayor medida de la estructura de los *frames*, la capacidad asignada para los *slots* de contención, así como de los modos de reservación de acceso, más que en los detalles de los CRA adoptados.

Se presentan los antecedentes relevantes a las técnicas investigadas de resolución de colisiones, desde la aparición del proyecto ALOHANET. Además, un estudio sobre la estabilidad del algoritmo *Exponential Backoff* (EB). También la forma en que el estándar 802.16 resuelve las colisiones para el acceso a la red usando el *Binary Exponential Backoff* (BEB).

Se muestra un panorama general de estudios relacionados con la misma línea de investigación presentada en este trabajo. De igual forma se detallan los esquemas analizados y propuestos para el caso de estudio presentado en esta tesis.

3.2 Antecedentes

Las estrategias que se han desarrollado para solucionar el problema de sistemas de acceso múltiple se pueden dividir en dos paradigmas básicos. Uno es el "*free-for-all*", similar a el que está usado para el ALOHANET, en el cual los nodos procuran retransmitir los mensajes colisionados esperando que no interfiera con otros nodos, descrito por Bertsekas en [3]. El otro paradigma consiste en dividir los nodos colisionados en una estructura tipo árbol. En estos mecanismos, basados en una estructura tipo árbol [6], todos los nodos implicados en una colisión se dividen en subconjuntos. El primer subconjunto es el que transmite

primero, seguido por el segundo subconjunto y así hasta los subconjuntos restantes. Las colisiones futuras son reducidas forzando a las estaciones, que colisionaron en el mismo *slot*, a retransmitir sus peticiones en diversos *slots*. Estos dos esquemas básicos se han desarrollado para adaptarse a varios ambientes y arquitecturas de red.

Recientemente, los protocolos MAC tienden a ordenar el ancho de banda del *uplink* en una estructura de *frame* que se componga de dos *subframes*; reservaciones e información. La parte de reservación consiste en un número de *slots* de petición, que son utilizados por todos los usuarios que contienden para la reservación del canal. Un usuario que tenga éxito en el proceso de reservación puede ser asignado en los *slots* de datos dentro de la parte de información para su transmisión de información. Se define una longitud más corta de *slots* para la resolución de contención, que para los períodos de *slots* de información, dado que el ancho de banda del canal puede ser utilizado con mayor eficiencia.

El IEEE ha estandarizado el protocolo 802.16 para sistemas de acceso inalámbrico de banda ancha de redes metropolitanas, que señala que el esquema de acceso aleatorio está basado en el algoritmo BEB para *ranging* (calibración) y solicitudes de ancho de banda.

El algoritmo BEB se ha aplicado y su funcionamiento se ha investigado en varios sistemas de transmisión de datos. Por ejemplo, en el estándar tradicional IEEE 802.3, redes de área local o sistemas inalámbricos IEEE 802.11; sin embargo, la eficacia del algoritmo de BEB depende fuertemente del sistema en donde se aplica.

3.3 Resolución de colisiones en IEEE 802.16

Las colisiones son eventos que ocurren cuando dos o más SS's intentan transmitir información en el mismo *slot*. Esto provoca que se desperdicie parte del ancho de banda disponible en el *uplink*, ya que todas las peticiones que se hayan realizado durante una colisión se verán afectadas y se perderán. La BS, al escuchar que es una colisión, no se ocupará de lo que se encuentre en ese *slot*.

La BS especifica, a través del mensaje UL-MAP, las regiones de *slots* de contención (CSs), que están sujetas a la ocurrencia de colisiones. Las colisiones pueden ocurrir durante los intervalos de Mantenimiento Inicial y de Solicitud, cada uno definido mediante su respectivo IE's (*Information Elements*¹). Cuando ocurre una colisión en cualquiera de los períodos mencionados, las SS's involucradas son las responsables de aplicar los métodos de resolución.

El método para la resolución de colisiones que adopta el estándar 802.16, se basa en el *Binary Exponential Backoff Algorithm* (BEBA) cuya operación se desenvuelve por los parámetros de ventana de *backoff* inicial y ventana de *backoff* final, ambos controlados por la

¹ Un *Information Element* (IE) consiste en por lo menos tres campos: (1) Identificador de conexión (CID); unicast, multicast y broadcast. (2) Código usado para el intervalo de *uplink* (UIUC); define el tipo de acceso al *uplink*. Y (3) Desplazamiento (Offset).

BS en los mensajes Descriptores del Canal Ascendente (*Uplink Channel Descriptor*- UCD), mediante potencias binarias.

3.3.1 Operación de resolución de colisiones

Las SSs escuchan el mensaje UCD y se enteran de cuál es la región de contención, que es la ventana inicial de oportunidades de transmisión. En el instante en que las SSs tienen la necesidad de transmitir información, seleccionan un valor aleatorio de la ventana inicial, conformada por los *slots* de contención totales que tiene el *frame* de *uplink*, y envían una solicitud de ancho de banda. Si la SS no recibe una confirmación de que su petición fue exitosa, durante un periodo de espera de confirmación establecido por la BS, la SS sabrá que su petición generó una colisión con otra SS y activará el BEBA para resolver la colisión.

La estructura principal es la siguiente:

```

if( $c < 16$ ) then
     $k = \min\{c + i, t\}$  //elige el valor mínimo de los dos parámetros
     $b = \text{random}[0, 2^k - 1]$  // espera  $b$  slots de contención
    Retransmitir en el slot  $b + 1$ 
else
    Destruye la petición
  
```

donde:

c : número de colisiones de la SS

i : valor inicial del BEBA indicado por la BS

t : parámetro que trunca el crecimiento de la ventana de contienda,
para el estándar 802.16 $t = 10$

b : número de *slots* que debe dejar pasar para retransmitir su petición, (*backoff*)

De esta forma si $i = 0$, al ocurrir la primera colisión la ventana sería $[0,1]$ de la cual la SS elegiría un valor aleatorio, si vuelve a colisionar varias veces, la ventana se incrementa en potencias de dos hasta el valor truncado, $t = 10$, donde la ventana es $[0, 1023]$, si ocurren más colisiones la ventana se mantiene hasta que ocurre la colisión número 16, que es cuando la petición se debe destruir, ya que el periodo de contención es muy grande y el retardo de acceso a la red se eleva exponencialmente.

Hay muchos trabajos relacionados con el valor inicial de la ventana de *backoff*, ya que si la ventana inicial fuera $[0,1]$, es muy probable que las SSs involucradas en la colisión vuelvan a caer en colisión, para ilustrar el mecanismo, se fijará el valor de $i = 0$. También es de notar que para evitar periodos largos sin región de contención se toma en consideración el estudio de Stallings [30] que indica que en promedio se necesitan e *slots* de contención, 2.718, para transmitir una petición exitosa, entonces en cada *frame* de señalización habrá una región mínima de *slots* (*Minimum Contention Slots*, MCs), aparte de los *slots* de reservación que la BS no use en el *uplink*.

La figura 3.1 muestra un ejemplo detallado de la operación del BEBA para resolver las colisiones en el estándar 802.16. El *frame n* muestra cómo está estructurado el *uplink*. Se diferencian tres regiones, la primera región es de contención MCs, (que otorga la BS para todos los *frames* de *uplink*); la segunda región es de *slots* reservación (donde las SSs, que previamente realizaron sus peticiones de ancho de banda con éxito, realizan la transmisión de su información) y; la tercera región, CSs (que son los *slots* de reservación que la BS no utiliza y que convierte en *slots* de contención). Con esto, se observa que cualquier usuario tiene un total de oportunidades de transmisión $Tx_Opp = MCs + CS$. Para el caso del *frame n*, todas las SSs tienen en total $Tx_Opp = 8$.

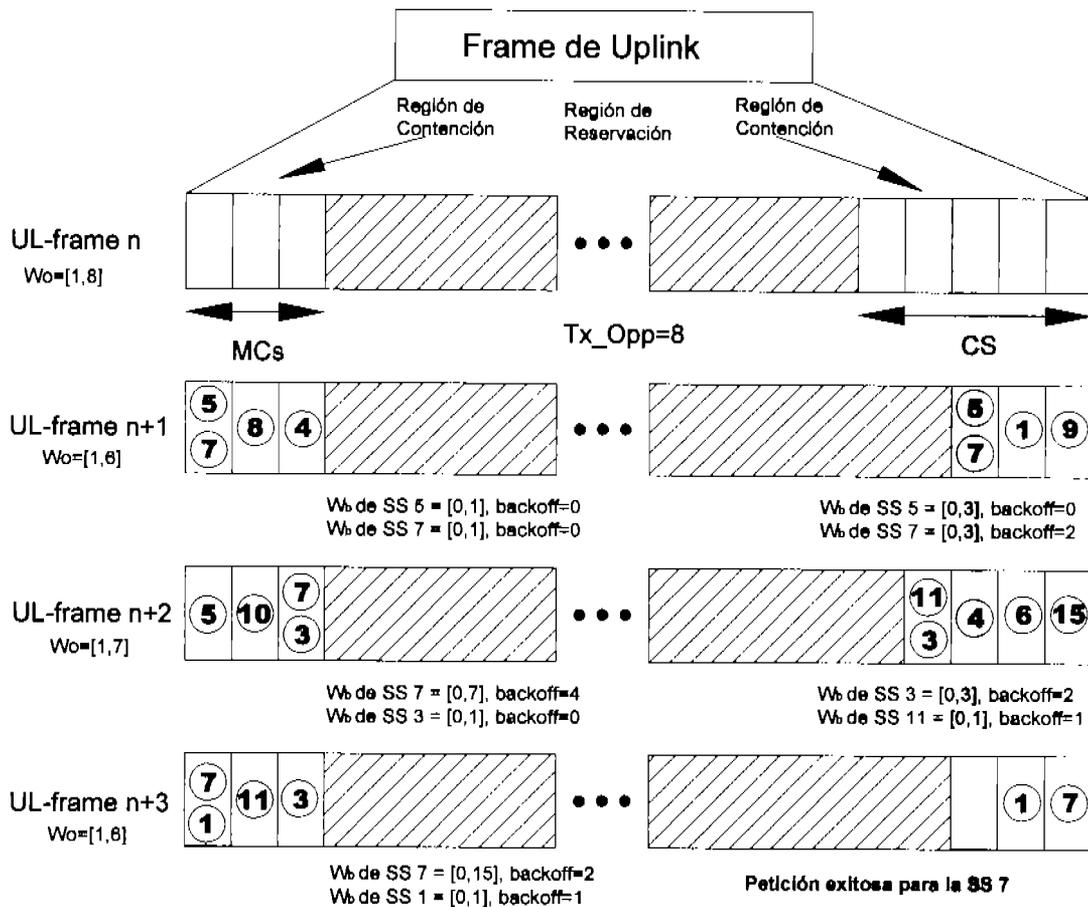


Figura 3.1. Procedimiento de resolución de colisiones para IEEE802.16.

Para el siguiente *frame*, $n + 1$, todos los usuarios tienen un total de $Tx_Opp = 6$. Entonces las estaciones 1, 4, 5, 7, 8 y 9 tienen una ventana inicial de transmisión igual a $W_0 = [1,6]$. Las estaciones eligen aleatoriamente una oportunidad de transmisión de su ventana inicial, esta ventana la pueden saber escuchando el mensaje UL-MAP.

La SS1 elige, aleatoriamente, transmitir su primera petición de ancho de banda en $Tx_Opp = 5$, la SS4 lo hace en $Tx_Opp = 3$, la SS5 en $Tx_Opp = 1$, la SS7 en $Tx_Opp = 1$, la SS8 en $Tx_Opp = 2$ y la SS9 en $Tx_Opp = 6$. Hasta este momento no se ha activado el mecanismo para resolver las colisiones del estándar 802.16.

En el canal de retroalimentación, que lo reciben las estaciones suscriptoras aproximadamente a la mitad de cada *frame*, las SSs se pueden dar cuenta si provocaron una colisión. Si no reciben una confirmación de éxito en su petición de ancho de banda o el tiempo máximo para recibir la confirmación se ha agotado, entonces las SSs se dan cuenta de que su petición provocó una colisión y activan el BEBA para resolver la colisión en que se involucraron.

Las estaciones 5 y 7 eligieron el mismo *slot* para transmitir, $Tx_Opp = 1$. Esperan la confirmación de su petición de ancho de banda pero no la reciben, entonces activan el BEBA. Como es la primera colisión de ambas estaciones, tienen una ventana de *backoff*, $W_b = [0,1]$, y las dos deben elegir un *backoff* aleatoriamente. Ambas estaciones eligen un *backoff* = 0, es decir que transmitirán su petición en el siguiente *slot* de contención que tengan disponible. Como ambas saben que al final del *frame* se encuentra otra región de contención, intentan su retransmisión pero vuelven a colisionar y su ventana se incrementa, $W_b = [0,3]$, y eligen aleatoriamente $backoff_{f_{SS5}} = 0$ y $backoff_{f_{SS7}} = 2$.

En el *frame* $n + 2$, la BS les indica a las estaciones activas que tienen $Tx_Opp = 7$ *slots* para realizar sus peticiones de ancho de banda. La SS5 transmite en el primer *slot* de contención y logra con éxito su petición, pero la SS7 deja pasar los primeros 2 *slots* de contención, como lo indica el BEBA, y hace su petición en el tercer *slot* de contención, espera su confirmación y no la recibe, lo cual indica que volvió a colisionar y ahora lo hace con la SS3. Como es la primera colisión para la SS3, su ventana de *backoff* es $W_{b_{SS3}} = [0,1]$ y elige aleatoriamente un $backoff_{f_{SS3}} = 0$, pero para la SS7 es su tercera colisión y su ventana es ahora de $W_{b_{SS7}} = [0,7]$ y elige un $backoff_{f_{SS7}} = 4$. En la parte final del *frame* $n + 2$, la SS3 realiza su retransmisión y vuelve a colisionar, pero ahora lo hace con la SS11. La ventana de la SS3 ahora es $w_{b_{SS3}} = [0,3]$ y elige un $backoff_{f_{SS3}} = 2$, por otra parte la ventana de SS11 es $W_{b_{SS11}} = [0,1]$ y elige un $backoff_{f_{SS11}} = 1$. Es de notar que la SS7 debe dejar pasar 4 *slots* de contención para volver a transmitir su petición de ancho de banda, como al final del *frame* $n + 2$ había 4 *slots* de contención entonces al inicio del siguiente *frame* realizará su retransmisión.

En el último *frame* de la figura 3.1, *frame* $n + 3$, la SS7 hace su retransmisión pero vuelve a colisionar, ahora con la SS1. La SS7 tiene ahora una ventana de $W_{b_{SS7}} = [0,15]$ y elige un $backoff_{f_{SS7}} = 2$ y la SS1 tiene una ventana de $W_{b_{SS1}} = [0,1]$ y elige un $backoff_{f_{SS1}} = 1$. Al final del *frame* $n + 3$, se observa cómo la SS1 retransmite su petición en el segundo *slot* de contención y logra con éxito su petición. La SS7 deja pasar dos *slots* de contención y vuelve a retransmitir su petición de ancho de banda, pero ahora realiza la operación con éxito.

Las SSs considerarán que la transmisión fue exitosa cuando la BS les indique que ha recibido su petición o cuando reciban un mapa que incluya la reservación, correspondiente al CID para el cual se hizo la solicitud, dentro de un periodo máximo definido mediante el mensaje Descriptor de Canal Ascendente. Entonces las SSs finalizarán la ejecución del BEBA —los parámetros del BEBA se reinician— e iniciará la transmisión de información correspondiente en los *slots* que la BS les asigne.

3.4 Estudios sobre Binary Exponential Backoff

Muchos estudios realizados sobre los algoritmos EB, incluyendo BEB, han sido en términos de su efecto sobre el funcionamiento de la red, mientras la carga de tráfico aumenta considerablemente. La mayoría de los trabajos disponibles en la literatura se enfocan en la estabilidad del EB y pocos trabajos están relacionados al análisis de su funcionamiento. Algunos de estos trabajos han arrojado resultados contradictorios sobre la estabilidad del EB, debido a que unos prueban su inestabilidad y otros su estabilidad bajo ciertas condiciones. Estos resultados son debido a las diferencias de los modelos analíticos, ya que simplifican o modifican el algoritmo y la definición de estabilidad en el análisis.

Los modelos simplificados y/o modificados del algoritmo *backoff* se utilizan a menudo para hacer un análisis más manejable, pero pueden conducir a resultados analíticos muy diferentes. Por ejemplo, Aldous [2] probó que el BEB es inestable para un modelo de usuarios infinito (un modelo simplificado) para cualquier tasa de arribo de paquetes diferente de cero, mientras que Goodman [8] mostró, con un modelo modificado de usuarios finitos, que BEB es estable para una tasa de arribo suficientemente pequeña.

Las múltiples definiciones de estabilidad, usadas en los estudios de los algoritmos *backoff*, se pueden clasificar en dos grupos. Un grupo utiliza un estudio basado en el rendimiento de procesamiento (*throughput*) y otro grupo, un estudio en relación al retardo (*delay*) de acceso para analizar su estabilidad. Bajo el análisis de *throughput*, Aldous [2] indica que el algoritmo es estable si el *throughput* no se colapsa cuando la carga ofrecida tiende al infinito. Y Shoch [27] establece que también es estable si el *throughput* no es una función del aumento de la carga ofrecida. Bajo el análisis de retardo, Håstad [9] mostró que el algoritmo es estable si se limita el tiempo de espera y esto se puede caracterizar por una reserva limitada de paquetes encolados, o mediante la utilización muy recurrente de cadenas de Markov.

La mayoría de los estudios analíticos y de simulación sobre EB, tratan al algoritmo *backoff* en el contexto de un protocolo específico de control de acceso al medio (MAC). Las características del protocolo en cuestión parecen tener mucho efecto en los resultados del funcionamiento de la red como del comportamiento del EB. Así, los resultados dependen considerablemente de qué protocolo MAC se utiliza en el estudio y no es posible entender el comportamiento del EB con los resultados. Algunos de los trabajos analíticos que se centran en EB se resumen a afirmar que el BEB es inestable para un modelo de memoria infinita, y para un modelo de memoria finita es estable si la tasa de llegada del sistema es bastante pequeña, pero inestable si la tasa de llegada es demasiado grande.

En la actualidad no existen modelos que analicen otros esquemas de resolución de colisiones para los sistemas 802.16. Es por esto que esta tesis se enfoca sobre nuevas técnicas para resolver el problema de los periodos largos de contención, reducir el retardo de acceso a la red y elevar el *throughput* del sistema.

3.5 Estudios del funcionamiento del IEEE802.16 relacionados en la resolución de colisiones

Kantawong y coautores en [18] analizan los diferentes algoritmos de resolución de contenciones para sistemas de comunicación inalámbrica. Los clasifican en dos grupos: acceso múltiple y acceso simple. El grupo de acceso simple son aquellos que, por cada señalización de *frame*, cada SS tiene sólo una oportunidad de contender por su reservación en el *uplink*. Si, en la región establecida para contención, colisiona o se le pasó, no puede contender en el siguiente *slot*, debe esperar hasta que llegue otro *frame* para contender en la región establecida. En el otro grupo, de acceso múltiple, las estaciones pueden contender en el siguiente *slot* a la derecha, no tiene que esperar a que llegue el siguiente *frame*. Estudian nueve algoritmos para la resolución de colisiones, en relación a ambos grupos, determinando que el BEB arroja mejores resultados.

Los autores de [34] establecen dos esquemas para analizar el funcionamiento del acceso aleatorio del protocolo 802.16, uno de simulación y el otro analítico. Observan que el modelo de simulación arroja resultados más apegados a los valores establecidos por el estándar. Pero no analizan ningún parámetro que permita un mejor funcionamiento de los sistemas basados en el protocolo IEEE 802.16

En [4] estudian el funcionamiento del *uplink*, del protocolo 802.16, en función del *throughput*, retardo promedio de acceso, probabilidad de colisión y la relación entre los *slots* de datos y de contención, para diferentes tamaños de carga útil. Muestran que el *uplink* tiene un mejor funcionamiento cuando el tamaño de la carga útil es grande, pero independientemente del tamaño de la carga útil, el *throughput* disminuye considerablemente debido a las colisiones ocasionadas por la cantidad de usuarios que contienden para reservar el canal.

Los autores de [5] analizan el funcionamiento del protocolo MAC del estándar IEEE 802.16, variado el *backoff* inicial para las peticiones de ancho de banda. Su estudio se basa en diferentes cargas de tráfico en función de los tamaños de paquetes a enviar por las SSs, para diferentes números de usuarios en la red. Realizan su investigación estableciendo un *backoff* inicial de 0, 16 y 32, pero en la simulación no consideran fragmentación y concatenación de paquetes MAC PDU ni tampoco el uso de peticiones *piggyback*.

Algo relacionado ocurre en [31], donde proponen una modificación del algoritmo adoptado para el IEEE 802.16, BEB, optimizando la ventana inicial de *backoff* según el número de usuarios activos en la red. En su escenario plantean un retardo de propagación despreciable y los paquetes se reciben sin errores. Implementan una función de costos normalizada en función del *throughput*, retardo de acceso y la probabilidad de colisión. Muestran que para el rango de usuarios [2, 17], la ventana inicial de *backoff* es $W_0=4$; para un rango de [17, 85], la ventana inicial de *backoff* es $W_0=8$ y para [85, 100], $W_0=16$.

Lidong y coautores en [20] proponen un modelo analítico para analizar el funcionamiento de peticiones en redes IEEE 802.16, ajustando la ventana de *backoff* inicial. Construyen el modelo basado en cadenas tridimensionales de Markov, con una distribución de probabilidad de *Poisson* en las peticiones y una distribución de probabilidad exponencial en la duración de la conexión. Los resultados muestran que el ajuste de la ventana de contención mejora lo propuesto en el estándar, además de que su implementación es más simple.

En los estudios mencionados, unos tratan de optimizar las técnicas propuestas por el estándar 802.16, otros encontrar parámetros que permitan un mejor funcionamiento del protocolo. Algunos proponen esquemas de simulación y otros procedimientos matemáticos, pero ninguno se enfoca en ofrecer nuevos mecanismos de resolución de colisiones que disminuyan el retardo promedio de acceso a una red que comparte el medio con múltiples usuarios. Que es lo que se estudia en este trabajo de tesis.

3.6 Algoritmos analizados y propuestos en este trabajo de tesis

En esta parte, se explican los algoritmos propuestos en [24] donde los autores ofrecen, entre sus alternativas, los algoritmos Simple-CSA, Forced-CSA y Variable-CSA. Estos mecanismos son analizados en [24] para redes CATV, basadas en el *European Cable Communications Protocol*. En López Bejarano [22], se ajustaron dos esquemas de [24], Simple-CSA y Forced-CSA, para su operación bajo el protocolo 802.16 y se estudia otro llamado Multicast-CSA. Para el presente trabajo continuamos la línea de investigación proponiendo cuatro nuevos mecanismos Variable-CSA, IEEE 802.16-CSA, Collision Free-CSA y CDMA-CSA bajo las especificaciones del protocolo 802.16.

3.6.1. Simple-CSA

Este mecanismo asigna todos los *slots* que no se utilizan para datos como *slots* de contención (CSs). Es decir, cuando la carga de tráfico es baja, muchos CSs serán asignados, más de los que se requieren. El exceso de CSs disminuye considerablemente el riesgo de colisión, lo cual también se traduce en un retardo de acceso muy bajo para las estaciones suscriptoras activas en la red. La figura 3.2, muestra un ejemplo sencillo del mecanismo Simple-CSA.

En la figura 3.2, el *frame n* no tiene ningún *slot* de contención, todos los *slots* son reservados para los usuarios que previamente realizaron sus peticiones de ancho de banda con éxito. El *frame n + 1* muestra tres *slots* de contención que son los que la BS no utiliza para las áreas de reservación. El *frame n + 2* muestra nueve *slots* de contención. Y por último, el *frame n + 3* no tiene ningún *slot* de contención para ese instante. Esas regiones son las que las SSs utilizarán para realizar sus peticiones de ancho de banda

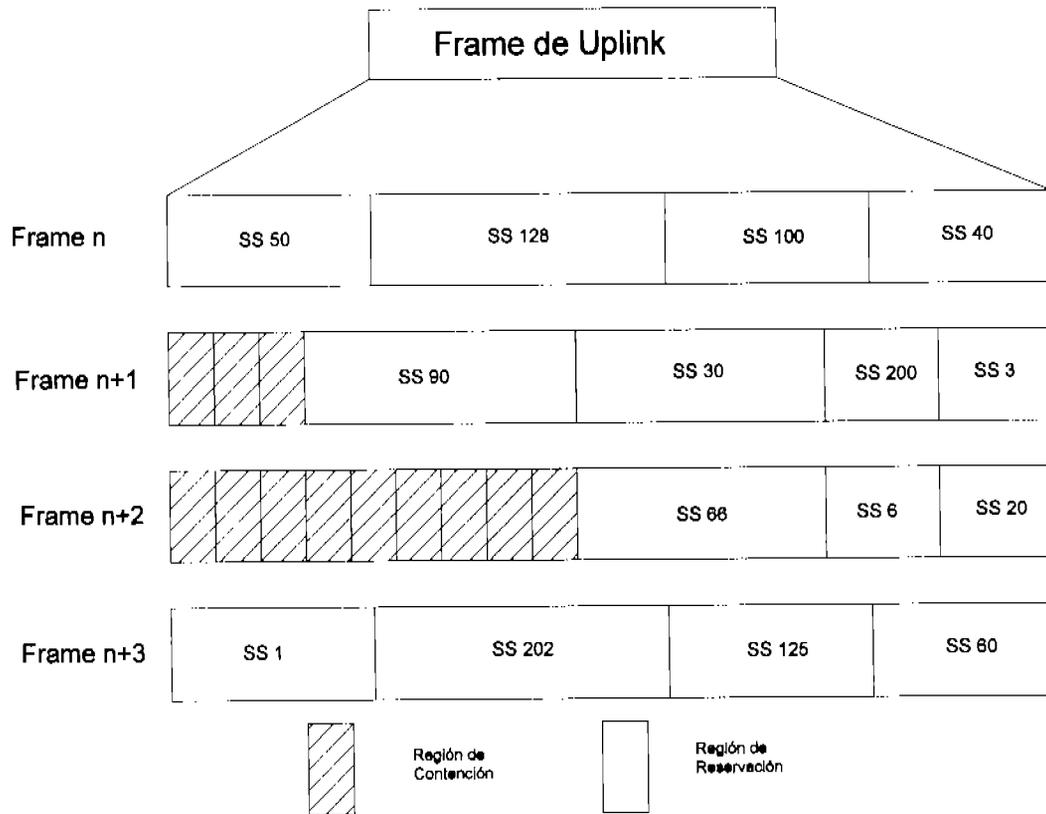


Figura 3.2. Procedimiento de Simple-CSA con $MCs=0$.

Este algoritmo es un mecanismo autorregulador, ya que si el número de CSs es demasiado bajo, las peticiones de los usuarios no llegarán a la BS y la región de reservación será muy pequeña, lo que genera que los slots sobrantes, del *frame* actual, se asignen como CSs automáticamente. Inversamente, si el número de CSs es demasiado alto, más peticiones alcanzarán a la BS y la BS volverá a incluir CSs hasta que estas peticiones hayan sido reservadas.

Para evitar que la BS deje pasar varios *frames* sin regiones de contención, se incluye un umbral mínimo "Minimum Contention Slots-MCs", que la BS debe asegurar en cada *frame*, para garantizar que por lo menos pocos slots deben ser reservados para el acceso de contención. Así, el funcionamiento de la red depende altamente del número mínimo de CSs asignado en cada señalización de *frame*. La figura 3.3 ilustra cómo actúa el mecanismo Simple-CSA, con el parámetro $MCs=2$.

En la figura 3.3 observamos que el *frame* n tiene dos slots de contención al inicio, correspondientes a MCs , y al final del *frame* cuatro slots que no utilizó para reservación y que los transforma en CSs. Para el *frame* $n + 1$, sólo fueron asignados los dos slots de MCs , lo mismo pasa para al *frame* $n + 2$, esto ocurre porque en el *frame* n muchos usuarios realizaron peticiones exitosas de ancho de banda y la BS no puede asignar CSs porque no le sobran slots para ello. En el *frame* $n + 3$, ahora la BS convierte los slots de reservación que no utilizó como CSs, además de los dos slots garantizados por el umbral mínimo, MCs .

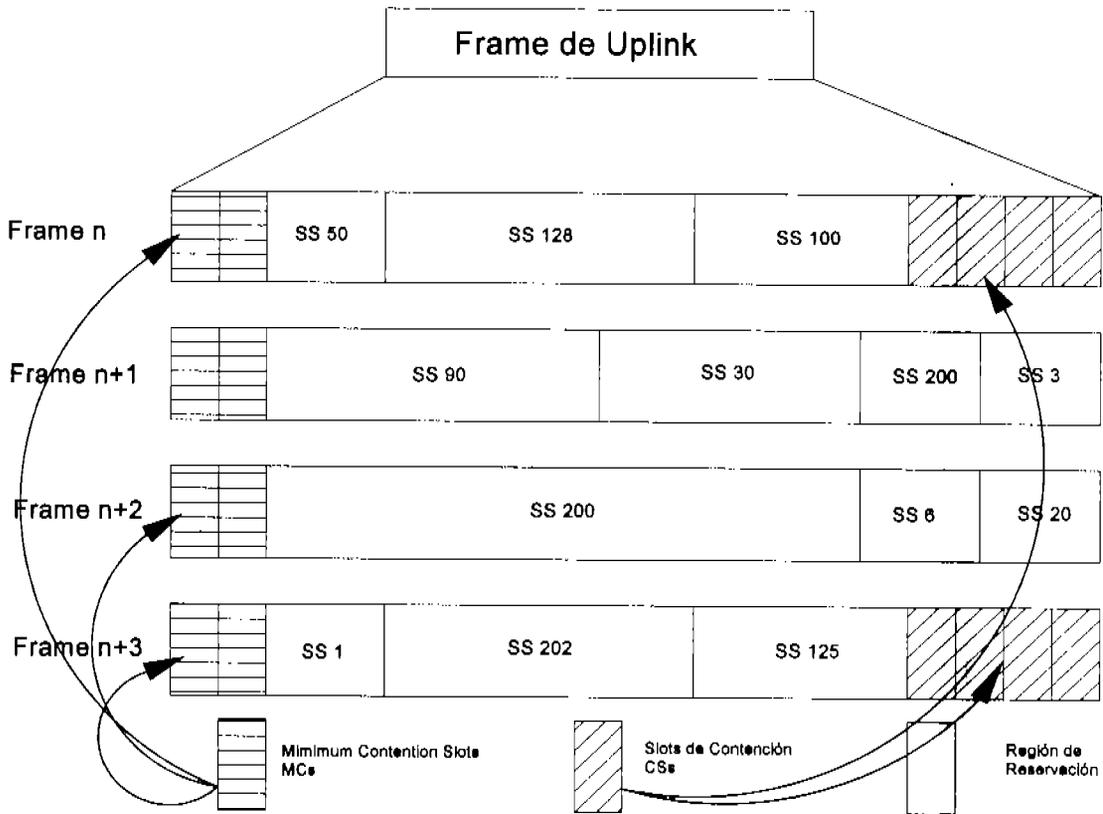


Figura 3.3. Procedimiento de Simple-CSA con MCs=2.

López Bejarano en [22] hace un estudio de este parámetro, MCs, encontrando y verificando lo que Stallings [30] muestra, que en promedio se necesitan $e=2.718$ slots de contención para transmitir una petición exitosa, 1 slot sin utilizar, 1 slot para transmisión exitosa y 0.718 slot para resolver la colisión.

3.6.2. Forced-CSA

Este mecanismo está basado en la dinámica del algoritmo *splitting tree*. Cuando una colisión ocurre, el algoritmo automáticamente asigna CSs en la siguiente señalización de *frame*, número especificado por el parámetro FCs (*Forced Contention Slots*) y pueden ser utilizados por las estaciones involucradas o no en la colisión. Es decir que el número total de CSs queda en función del número de colisiones en cada mapa, esto es:

$$CSs = Collision_per_MAP * FCs + MCs$$

De esta forma, el algoritmo Forced-CSA, aloja un número flexible de CSs. Con esta nueva funcionalidad, las SSs que compiten por el acceso en los slots de contención tienen más oportunidades de transmitir sus peticiones exitosamente. Debido a que más slots de contención son asignados cuando se necesitan, y no cuando sobran. Reduciendo

considerablemente el retardo promedio de acceso de los paquetes. Un ejemplo sencillo para este mecanismo es el mostrado en la figura 3.4.

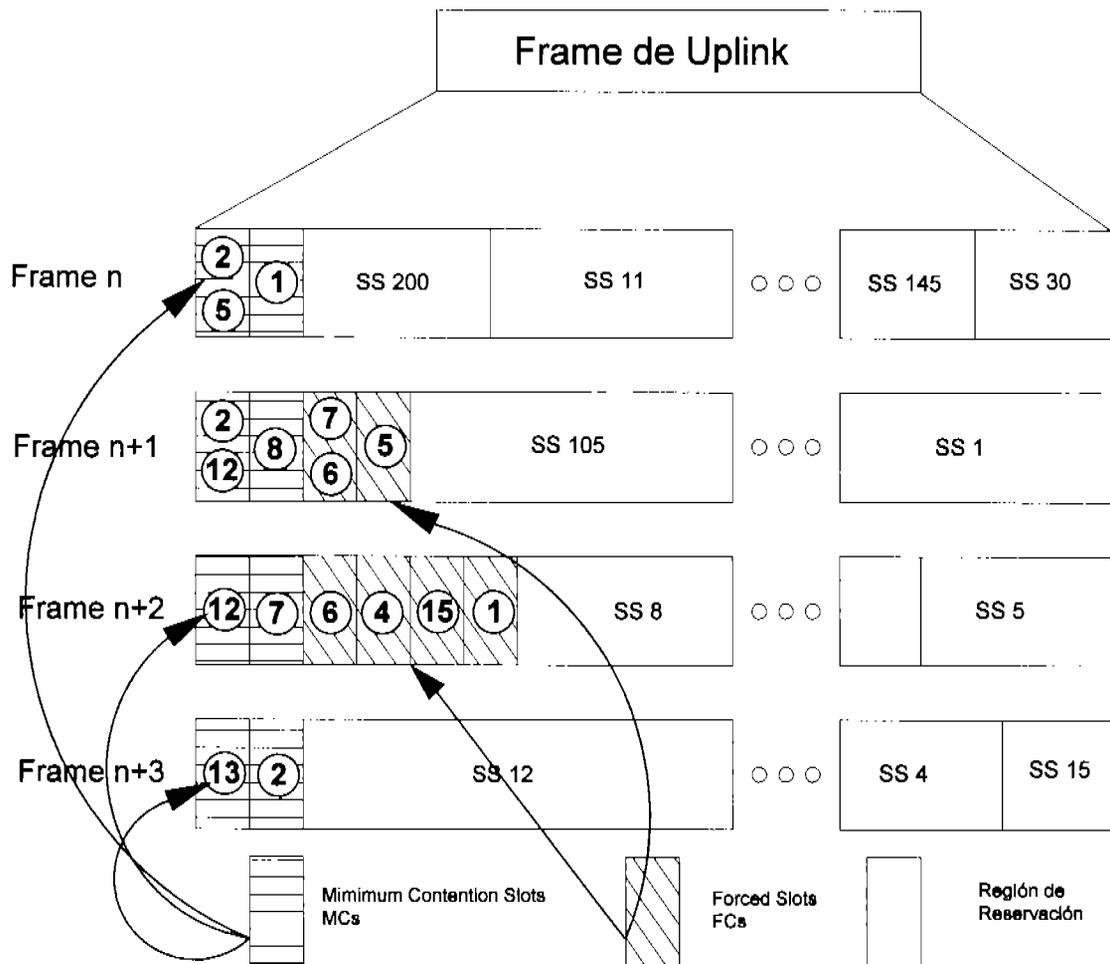


Figura 3.4. Procedimiento de Forced-CSA, con MCs=2

En la figura 3.4, el *frame n* contiene sólo los dos *slots* del umbral mínimo, $MCs=2$, y en el primero de ellos colisionan las estaciones 2 y 5. Este evento activa el mecanismo *Forced* y la BS deberá alojar un número de *slots* forzados (FCs, en el ejemplo $FCs=2$ *slots* por colisión) en el siguiente *frame* de señalización. Esta actividad se observa en el *frame n + 1*, donde además de tener dos *slots* mínimos, aparecen otros dos *slots* forzados. Entonces todas las SSs pueden hacer uso de no sólo dos *slots*, sino de cuatro. Y en este mismo *frame n + 1*, se registran dos colisiones, una causada por las estaciones 2 y 12 y la otra por las estaciones 7 y 6. Es de notar que la colisión se puede detectar, en el módulo de recepción de la BS, al escuchar una señal distorsionada, cuando se esperaba el envío de una petición de ancho de banda.

En el siguiente *frame n + 2*, se observa cómo las dos colisiones del *frame n + 1* provocan que la BS asigne ahora 4 *slots* más, a parte de los dos del umbral mínimo, es decir ahora todas las SSs tienen seis *slots* de contención para realizar sus peticiones de ancho de banda. Además

se observa que ninguna estación provoca colisión. En el *frame* $n + 3$ sólo aparecen los dos *slots* mínimo, ya que en el *frame* $n + 2$ no se registró ninguna colisión.

En resumen, la BS agregará *slots* de contención forzados cada que ocurra una colisión, por ejemplo si en el *frame* actual ocurren 4 colisiones, para la siguiente señalización de *frame*, la BS está forzada para alojar 8 *slots* ($2_{FCs} * 4_{colisiones} = 8_{slots}$), a parte de los *slots* propuestos por el umbral mínimo. En López Bejarano [22], se optimizó este mecanismo y el valor de FCs que presentó un mejor desempeño fue con $FCs = 3$, por lo que las simulaciones presentadas en el capítulo 5 utilizarán este valor óptimo.

3.6.3 Variable-CSA

Este nuevo mecanismo, propuesto en esta tesis, transforma algunos *slots* de reservación en *slots* de contención dependiendo de la carga de tráfico en la red, el número de colisiones que ocurren en cada mapa y haciendo una distinción de qué tipo de reservación se está recibiendo en la BS. El esquema es el siguiente:

```

if  $R_l > G_i$  &&  $C_i \leq 2$  // hay carga de tráfico y las colisiones son pocas
     $V_{CS} = 0$ 
else if  $R_l > G_i$  &&  $C > 2$  &&  $R_{C_i} > 2$  // hay tráfico y las colisiones son muchas
     $V_{CS} = \min\{e \cdot C, Max_{V_{CS}}\}$ 
else // no hay carga de tráfico
     $V_{CS} = Max_{V_{CS}}$ 

```

Donde R_l son las peticiones pendientes que la BS tiene almacenadas en la cola de concesiones (grants). La variable G_i son las concesiones que pueden ser alojadas en el siguiente *frame* de señalización de *uplink*. Esto es:

$$G_i = \frac{Slots_{pendientes}}{Max_{Data}}$$

donde $Max_{Data} = Slots_{totales\ por\ frame} - MCs$ y $Slots_{pendientes}$ son las concesiones pendientes en la BS, en *slots*. El parámetro C , es el número de colisiones que ocurrieron en el mapa anterior y R_{C_i} son las peticiones de ancho de banda que llegaron con éxito a la BS por medio de los *slots* de contención. V_{CS} es el valor de *slots* de contención que la BS deberá incluir en la siguiente señalización de *frame*. El parámetro $MAX_{V_{CS}}$ debe ser un valor adecuado para que el funcionamiento del sistema no se degrade. La determinación de este valor se realizó por medio de un mecanismo de prueba y error, arrojando un mejor comportamiento cuando $Max_{V_{CS}} = 7$.

Con el uso de este mecanismo el número total de CSs que se incluirá en la siguiente señalización de *frame* es representado por el MCs (según lo propuesto en el Simple-CSA), el valor de V_{CS} y los *slots* que la BS no utilice como *slots* de reservación. Así, cuando la red esté

congestionada, la BS no agregará más *slots* de contención en el *frame*, a menos que se presenten muchas colisiones, logrando que los periodos de contención de las SSs se reduzcan considerablemente, como se puede ver en la discusión de los resultados. En el capítulo 5, de esta tesis.

3.6.4. Multicast-CSA

Este mecanismo, propuesto por López Bejarano [22], asigna una pequeña región *multicast* para cada colisión, de tal forma que sólo los usuarios involucrados en la colisión hacen uso de la pequeña región reservada para realizar sus peticiones de ancho de banda eligiendo aleatoriamente un *slot* de la región. Los demás usuarios deben utilizar otros *slots* de contención diferentes. En el análisis realizado en [22] se puntualiza que la región *multicast* está conformada por tres *slots* de contención, para cada colisión ocurrida en el mapa anterior, además utiliza una región de *MCs* y la transformación de los *slots* de reservación no utilizados en *slots* de contención. Un ejemplo de este mecanismo se muestra en la figura 3.5, en la cual se considera un $MCs = 2$ y tres *slots* para las regiones *multicast*.

La figura 3.5 ilustra el funcionamiento del esquema Multicast-CSA. En el *frame* n , sólo se tienen los *slots* *MCs* y en el primer *slot* las estaciones 2 y 5 se involucran en la primera colisión, *C1*. Este evento es registrado por la BS, sabiendo que debería escuchar una petición y sólo captura interferencia. La BS deberá incluir una pequeña región *multicast* en el siguiente *frame*. Para el *frame* $n + 1$ se puede ver como la BS, además de alojar los *slots* mínimos, asigna una región para las estaciones que colisionaron en el *frame* anterior, *Región C1*, en la cual sólo las terminales 5 y 2 contienden. Los demás usuarios lo hacen en los *slots* de contención enviados por *broadcast*. También se observa que tres estaciones, 12, 18 y 7, colisionan (*C2*) en el primer *slot* de contención y hacia el final del *frame* $n + 1$ colisionan las terminales 11 y 20 (*C3*).

En el *frame* $n + 2$, la BS asigna una región *multicast*, *Región C2*, para las estaciones que colisionaron en *C2* y otra región *multicast*, *Región C3*, para las estaciones de la colisión *C3*. En la *Región C2* sólo contienden las estaciones 12, 18 y 7, las cuales tienen tres *slots* para volver a intentar sus peticiones, la terminal 7 logra con éxito su petición, pero las terminales 12 y 18 vuelven a colisionar, *C4*. Las estaciones 11 y 20 contienden en su región *multicast*, *Región C3*, y logran con éxito sus peticiones.

En el *frame* $n + 3$, la BS asigna una nueva región *multicast*, *Región C4*, para las estaciones 12 y 18 que provocaron la colisión *C4*, ambas estaciones contienden en esa región y logran con éxito sus peticiones.

Como se puede apreciar, con el uso de este mecanismo se reduce considerablemente el retardo de contención ya que las estaciones involucradas en una colisión, la resuelven mucho más rápido que utilizando el mecanismo, BEBA, propuesto por el estándar 802.16. Por ejemplo si colisionan 5 estaciones en un mismo *slot*, en el siguiente *frame* las 5 estaciones contendrán en su región *multicast*, la probabilidad de que ocurra una colisión es

muy alta pero el problema de la colisión se resuelve muy rápido, en el mejor de los casos, bastarían tres frames para resolver la colisión provocada por 5 estaciones, Pero aún así el tiempo promedio para resolver las colisiones es demasiado bajo, en comparación con el Simple-CSA, como se muestra más adelante en el capítulo 5.

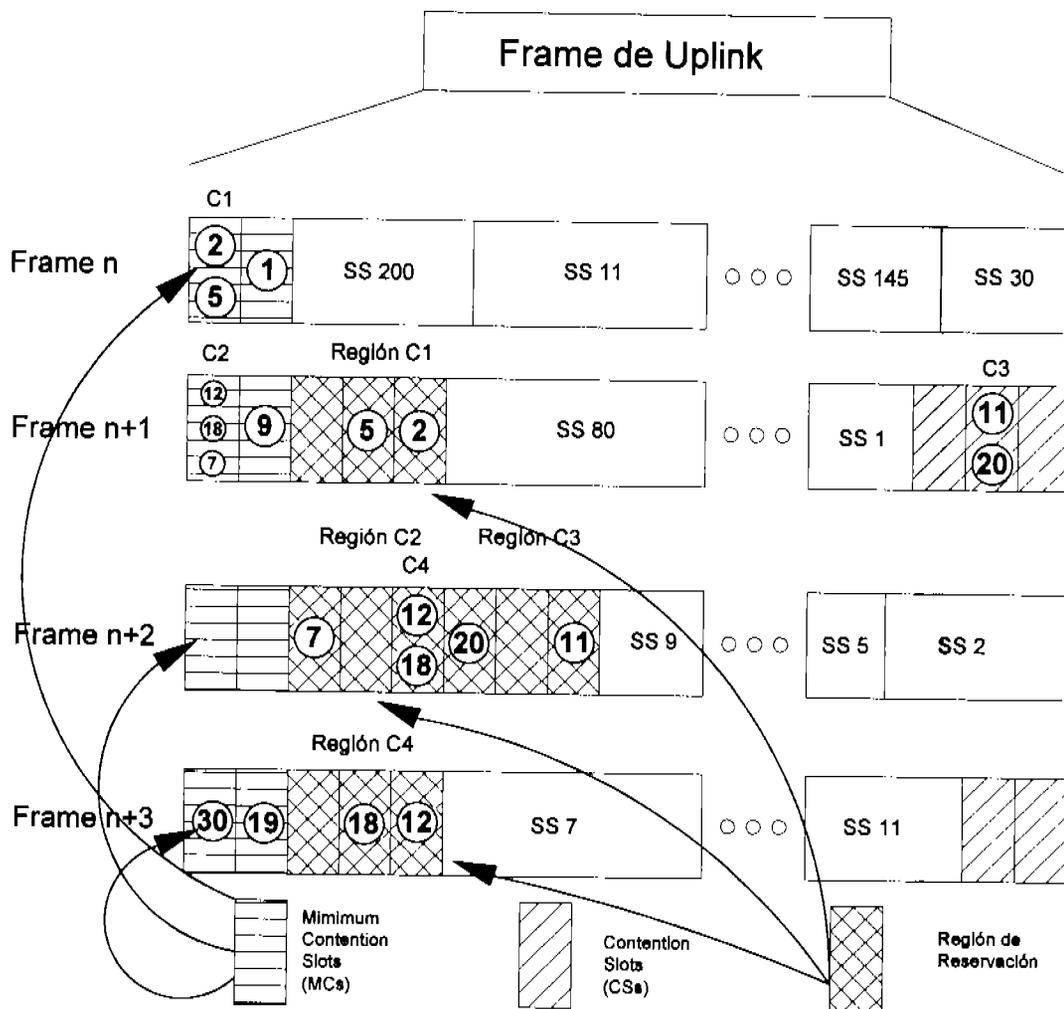


Figura 3.5. Procedimiento de Multicast-CSA.

3.6.5 IEEE 802.14-CSA

Este esquema, también propuesto en esta tesis, es una adaptación del mecanismo propuesto para el protocolo 802.14, un estándar para redes CATV (Community Antenna Television). Este esquema no fue muy comercializado ya que llegó tarde al mercado. Pero fue de los primeros sistemas de acceso múltiple para un medio compartido. Este esquema indica que los slots de contención son asignados, por la BS, de acuerdo a la carga de tráfico del sistema y el número de usuarios que se encuentran activos en la red.

El mecanismo presentado aquí es similar a la técnica presentada en Golmie [7] con algunas modificaciones para su implementación en el protocolo IEEE 802.16.

El número variable de CSs (VCSs) que se agregará en cada *frame* de señalización se ajusta dinámicamente por la BS según:

$$N_{RS} = \left\lfloor \frac{2 * MAX_{DATA}}{2 + Req_{size}} \right\rfloor$$

donde MAX_{DATA} es el número máximo de *slots* de datos que pueden alojarse en cada *frame* de señalización y Req_{size} es el número promedio de *slots* de reservación (RSs) que puede solicitar un usuario. Tomando en cuenta lo anterior, VCSs puede ser determinado como sigue:

$$VCS = \begin{cases} 0 & \text{if } MAX_{REQ} \geq \alpha * (MAX_{DATA} \cdot N_{RS}) \\ N_{RS} & \text{else} \end{cases}$$

donde MAX_{Req} es el número total de *slots* de datos solicitado que todavía la BS no ha concedido, α es un parámetro de diseño igual a 2.5 según lo sugerido en Golmie [7]. El número total de CSs que se incluirá en el siguiente *frame* de señalización es representado por el MCs (según lo propuesto en el Simple-CSA), la cantidad de *slots* de reservación convertidos en contención, VCS.

La utilización de este mecanismo en redes basadas en el estándar IEEE 802.14 presenta buenas mejoras en retardo y *throughput*, pero no fue el caso en redes basadas en el protocolo 802.16, como se verá en el capítulo 5, donde se discuten los resultados arrojados por todos los mecanismos estudiados. La principal causa es la duración de los *frames*.

3.6.6. Collision Free-CSA

Este nuevo esquema también se propone en esta tesis y el mecanismo asigna una oportunidad de transmisión *unicast* a las estaciones, mediante un sistema *Round Robin* (RR), mientras se tenga posibilidad de asignarlos el UL-MAP. Este método de RR sirve para seleccionar todos los elementos en un grupo de manera equitativa y en un orden racional. Normalmente comenzando por el primer elemento de la lista hasta llegar al último y empezando de nuevo desde el primer elemento. Cada estación usa esta oportunidad de transmisión *unicast* para enviar sus peticiones de ancho de banda, si es que tienen información que enviar.

Esta técnica elimina el acceso por contención, pero se registra un nuevo retardo, retardo por consulta, (*Delay Polling*). Esto es porque las estaciones están forzadas a esperar su oportunidad de transmisión *unicast*.

La figura 3.6 muestra el funcionamiento del mecanismo Collision Free-CSA, donde se puede observar que en el *frame* n , no existen *slots* para que los usuarios realicen sus peticiones de ancho de banda. En el *frame* $n + 1$, la BS detecta que le sobran 39 *slots*, esto

indica que la BS alojó todas las concesiones que tenía pendientes. Esos 39 slots sobrantes los asigna como slots unicast para el rango de estaciones de SS50 hasta la SS88, en estos slots, si los usuarios deben realizar peticiones de ancho de banda, pueden hacerlas en su respectivo slot unicast.

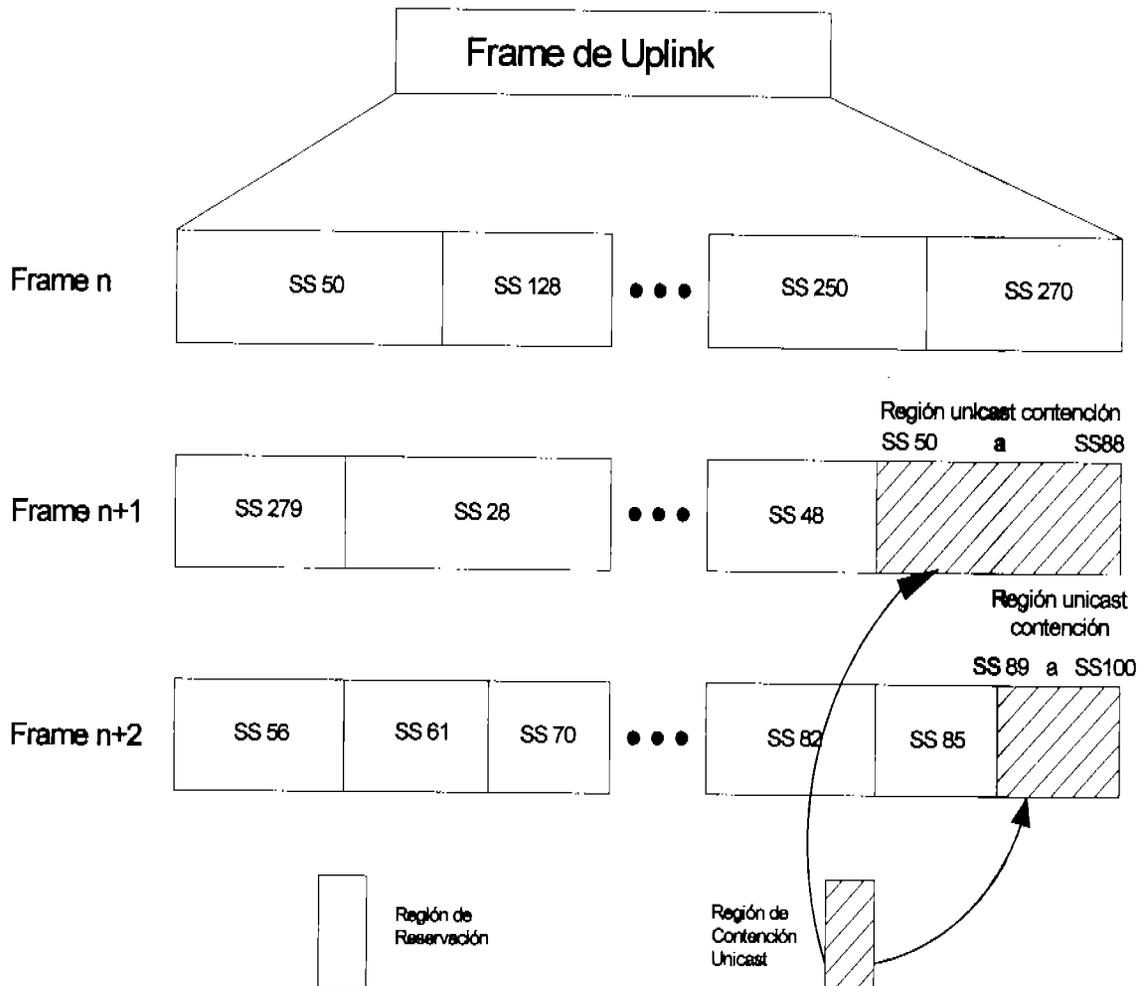


Figura 3.6. Procedimiento de Collision Free-CSA.

Para el *frame* $n + 2$, la BS les concede los recursos a los usuarios que realizaron sus peticiones en el *frame* $n + 1$ y detecta que le sobran 12 slots los cuales son asignados para los siguientes usuarios, del SS89 al SS100. Y así, hasta que cubre a todos los usuarios. La BS repite este proceso hasta llegar a la última SS de su lista de nodos activos y vuelve a asignar slots unicast desde el inicio de la lista de nodos activos, SS1.

Una desventaja de este mecanismo es que, en zona de congestión, si un usuario quiere hacer su petición de ancho de banda, y su slot unicast tiene muy poco que pasó, debe esperar por un tiempo muy largo para que la BS le asigne otro slot unicast. Pero este retardo de consulta, *Delay Polling*, se mantiene muy bajo, comparado con el retardo de contención de los demás mecanismos analizados en esta tesis.

3.6.7 CDMA-CSA

Este nuevo mecanismo propuesto, es adaptado para su funcionamiento en sistemas basados en el protocolo 802.16. La BS asigna sólo los *slots* de contención de umbral mínimo para todas las peticiones de ancho de banda de las estaciones activas en la red. Es decir que cada SSs tiene asignado un código de acceso para realizar su petición de ancho de banda en la región de contención. Ninguna estación provocará colisión ya que tiene su acceso al medio garantizado. Este mecanismo puede resolver todas las peticiones con sólo un *slot*, pero tomando en consideración el elevado nivel de interferencia de todas las estaciones enviando sus peticiones en un *slot*, se considera el umbral mínimo propuesto por Stallings [30], que es de tres *slots*. Los resultados de este mecanismo son muy alentadores para el retardo promedio de acceso al medio, pero en términos de *throughput* no son del todo satisfactorios, como se puede observar en el capítulo 5.

3.7 Conclusiones

Se presentaron tres esquemas que se analizan en López Bejarano [22], Simple-CSA, Forced-CSA y Multicast-CSA. Se detalla su funcionamiento y se ejemplifican cada uno de ellos. También se mostró el funcionamiento de cuatro nuevos mecanismos propuestos en este trabajo de tesis, Variable-CSA, IEEE 802.14-CSA, Collision Free-CSA y CDMA-CSA, todos ellos bajo su funcionamiento en redes 802.16, para mejorar el problema de resolver las colisiones en una red que utiliza un medio compartido y reducir considerablemente el retardo de acceso al medio.

Se presentó un panorama general de las líneas de investigación relacionadas con el caso de estudio que se presenta en esta tesis. Se hace énfasis en que el BEBA, adoptado por el estándar 802.16 para resolver el problema de las colisiones, presenta resultados muy diversos en otros sistemas y que su funcionamiento depende mucho del protocolo en el que se aplica.

CAPÍTULO 4

Modelo de simulación y adaptación de las técnicas

4.1 Introducción

Para el modelado de redes de comunicaciones, existen dos métodos principales. El primero de ellos se lleva a cabo mediante modelos analíticos y la segunda opción es a través de simulaciones. Cuando se habla de modelos analíticos, nos referimos al análisis matemático mediante el uso de ecuaciones. Para el caso de la opción de simulación, el uso de una herramienta que cumpla con dicho objetivo. Esta última opción nos ayuda a representar ambientes más reales y por lo tanto obtener varios detalles, respecto al comportamiento de dichos modelos.

El uso de los modelos de simulación permite examinar una variedad amplia de configuraciones necesarias para proporcionar conclusiones generales y nuevos mecanismos de mejoramiento, que permitan incrementar el desempeño de protocolos de redes inalámbricas de banda ancha relativamente en un tiempo corto.

El principal objetivo de este capítulo es presentar el modelo de simulación utilizado para la optimización de los algoritmos de colisiones CRA (*Collision Resolution Algorithm*) para el estándar IEEE 802.16. Se presenta una descripción de los módulos en donde se implementaron las técnicas de estudio. El apéndice A, incluye una descripción detallada del modelo de simulación utilizado para esta tesis, modelo que se logró gracias al proyecto PAPIIT IN110805 e IN104907.

En la actualidad existen dos principales lenguajes de simulación orientados al modelado de redes de comunicaciones. El primero es *OPNET (OPTimised Network Engineering Tool) Modeler*. Este software es un lenguaje de simulación de comunicaciones especialmente orientado al modelado de las redes de comunicaciones que utiliza Redes (*Networks*), Nodos (*Nodes*) y el Editor de Procesos (*Process Editors*) para la construcción del modelo de simulación.

La máquina del *OPNET Modeler* es un modelo de máquina de estado finito, en combinación con un modelo analítico. *Modeler* puede modelar protocolos, componentes y comportamientos de redes, con alrededor de 400 modelos de funciones de propósito especial. La Interfase Gráfica de Usuario (*Graphical User Interface GUI*) y el animador tienen una gran cantidad de documentación y casos de estudio disponibles. Hay disponibles tres editores durante el diseño de la simulación, proyecto, nodo y proceso, los cuales pueden modificar la configuración de la red, equipo, protocolos y aplicaciones a diferentes niveles.

También existe el software de simulación llamado *ns*, *Network Simulator*, (más conocido como *ns - 2* por su versión actual). Es un simulador de redes de eventos discretos, creado por la Universidad de Berkeley para modelar redes de tipo IP. En la simulación se toma en cuenta lo que es la estructura (topología) de la red y el tráfico de paquetes que posee la misma, con el fin de crear una especie de diagnóstico que muestre el comportamiento que se obtiene al tener una red con ciertas características.

ns - 2 es utilizado principalmente en ambientes académicos debido, a que está escrito en código abierto y, a la abundancia de documentación en línea. Se pueden simular tanto protocolos *unicast*, como *multicast* y se utiliza intensamente en la investigación de redes móviles *ad-hoc*. Puede simular una amplia gama de protocolos tanto para redes cableadas o redes *wireless*, así como mixtas.

De estos dos lenguajes de simulación presentados, *OPNET Modeler* (v 11.0 C) [23] ha sido utilizado como la herramienta de simulación para la caracterizaciones del desempeño y la optimización de las redes de banda ancha. Esta simulación de paquetes contiene un conjunto extenso de características diseñadas para soportar el modelado de redes y provee un incremento en la flexibilidad del desarrollo detallado de modelos diseñados. Además ofrece licencias para utilizar un modelo inalámbrico de simulación de redes y soporte para cualquier eventualidad. Esta característica fue una de las principales razones por las que se optó en utilizar *OPNET Modeler*.

4.3 Modelado de *OPNET*

Para tener una perspectiva de como son implementados los modelos en *OPNET*, en esta sección se dará una pequeña descripción del diseño jerárquico usado por este paquete de simulación. Los modelos de *OPNET* están basados en tres niveles de estructura jerárquicos. Esto se ilustra en la Figura 4.1.

OPNET tiene una extensa gama de herramientas que permiten desarrollar modelos detallados, identificar los elementos del modelo de interés, ejecutar simulaciones y analizar los datos generados. Estas herramientas se encuentran divididas de la siguiente forma: Constructor jerárquico de modelos, interfaz gráfica de simulación y Analizador de resultados.

El constructor jerárquico de modelos: Contiene cuatro editores para desarrollar los modelos de los sistemas que van a ser simulados: *Network editor*, *Node editor*, *Process editor* y *Parameter editor*. Los cuatro están organizados en forma jerárquica y pueden ser utilizados de tal forma, que un modelo desarrollado en un nivel, puede ser utilizado por otro modelo desarrollado en otro nivel más alto.

Dominio de Red (Network Domain)

El nivel más alto es *Network Domain* (Figura 4.1 a). La función de este dominio es definir la topología de la red de comunicación, las entidades de comunicaciones llamadas nodos y sus interconexiones (utilizando bus, punto a punto o enlaces de

radio). Basado en estos bloques básicos de construcción, más modelos complejos pueden ser desarrollados.

Nodo Dominio (Node domain)

El siguiente nivel es llamado *Node domain* (Figura 4.1b). Este nivel define la funcionalidad de cada dispositivo de comunicación que puede ser desarrollado e interconectado en la red (ejemplo enrutadores, puentes, terminales, switches, etc). Cada nodo está constituido de una fuente generadora de tráfico, procesos, colas y varios transmisores y receptores permitiendo a un nodo unirse con los enlaces de comunicación en la red.

Generador de tráfico (*Traffic generators*): Son objetos fabricados utilizados para generar fuentes de tráfico. La generación del tráfico del canal tiene longitudes de paquetes y distribución de paquetes inter-arriba, la cual es seleccionada de un conjunto que soporta la distribución de probabilidades.

Colas y procesos (*Queues y Processors*): Son objetos y son altamente detallados. Las colas son usadas para simular buffers de enlaces de comunicación y el comportamiento del protocolo. Algunos ejemplos de las estadísticas que las colas pueden recolectar son: su tamaño actual/promedio, el retardo y el sobre flujo de la cola. Cada objeto cola, puede generar múltiples subcolas en donde los paquetes pueden ser almacenados. Los procesos tienen la capacidad de manipular los paquetes pero no la habilidad de almacenarlos. Pueden ser utilizados como contenedores de paquetes o como generadores de tráfico. Esto último cuando el tráfico no puede ser modelado por la probabilidad de distribución predefinida.

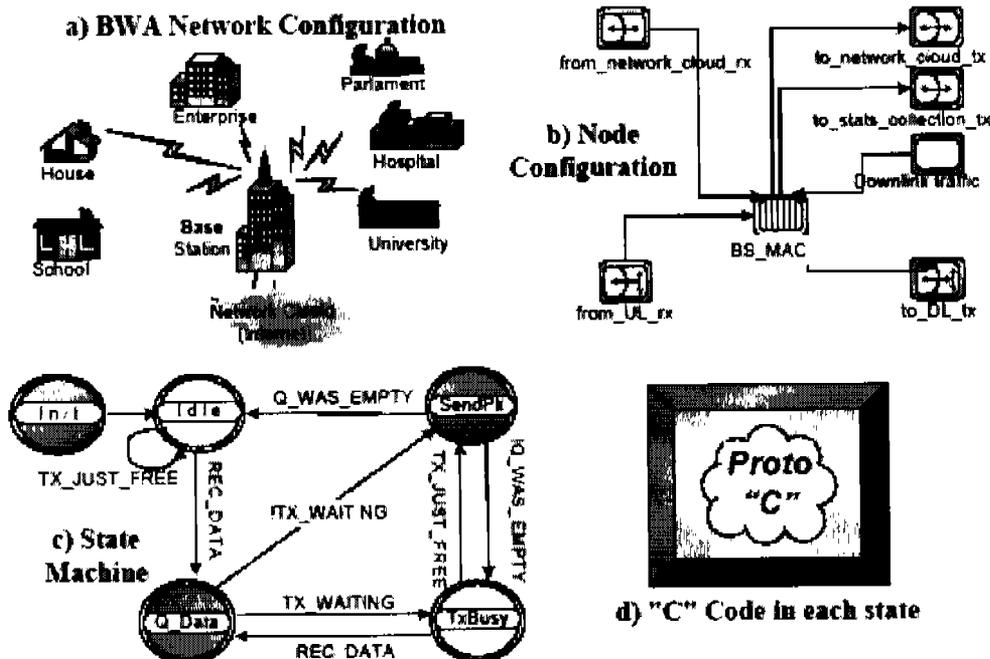


Figura 4.1 Diagrama de Estado de OPNET para los modelos de simulación.

Proceso dominio (*Process domain*)

El último nivel es llamado Proceso dominio (Figura 4.1.c). En este nivel la funcionalidad de cada objeto, cola o proceso, está definida usando la máquina de estado finito FSM (*Finite State Machine*). Estados y transiciones gráficamente definen el avance del proceso en respuesta a los eventos. En general, cada FSM puede definir estados privados variables y puede hacer llamadas al código y proveer al usuario de las librerías. La FSM es dinámica y puede ser activada espontáneamente por otras FSMs durante la simulación, en respuesta a eventos discretos específicos. El dinamismo de la FSM simplifica dramáticamente las especificaciones del protocolo que administra un número escalable de recursos o sesiones.

Cada estado, del modelo de un proceso, contiene código 'Proto-C' (Figura 4.1.d), soportado por una extensa librería de funciones diseñadas para la programación del protocolo.

Adicionalmente, OPNET posee dos herramientas: Interfaz gráfica de simulaciones y Analizador de resultados. La primera de ellas es para correr y depurar simulaciones con las que interactúa, es decir que cuando se corre una simulación es posible interactuar con la simulación para detectar posibles errores e identificar cuál objeto o proceso fue el que terminó la simulación.

La otra herramienta, Analizador de resultados, contiene los siguientes cuatro elementos: editor de pruebas (*Probe Editor*), que contiene la información que será recolectada; herramienta de análisis (*Analysis Tool*), se utiliza para el análisis estadístico de los resultados; herramienta de filtrado (*Filter Tool*) que procesa los datos y; el visor de animaciones (*Animation Viewer*).

4.4 Implementación de las técnicas en el modelo de simulación

La programación de las técnicas estudiadas en este trabajo es en los módulos de capa MAC del modelado de la BS y de las SSs. Para una descripción detallada de todo el sistema 802.16 referirse al apéndice A. Para la BS, todos los esquemas de Resolución de Colisiones se incluyen en el estado *SEND_NEX_MAP* del módulo *HE_MAC*. La figuras 4.2 (a) y (b) muestra en detalle la localización de los módulos para la BS.

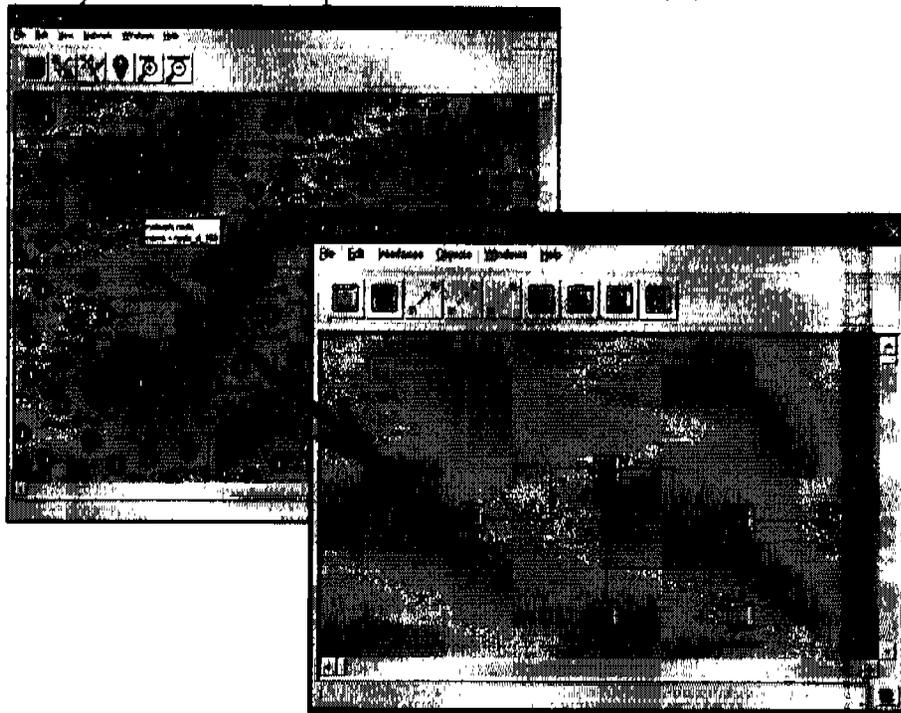
Para encontrar el módulo *SEND_NEXT_MAP*, del escenario de simulación, ingresamos en la BS, figura 4.2 (a). Se puede observar que el módulo *bwa_be* está conformado por diversos módulos como antenas de recepción y transmisión, módulos para preparar la recepción o transmisión de paquetes, *link_rx* y *link_tx*. Pero el más importante es el módulo *HE_MAC*, donde se encuentra todo lo relacionado con la capa MAC del protocolo 802.16. En la figura 4.2. (b), se observa que el módulo *HE_MAC*, está estructurado por una máquina de estado finito (*Finite State Machine-FSM*). Se observan los módulos : *INIT*, *Ranging*, *Rng_rcvd*, *Rng_comple*, *Upstream*, *Down_data*, *Mtn_MAP*, *MAP_Time*, *UCD_Time*, *Sync_Time*. De estos módulos, el importante es *MAP_Time*, que es en donde se realiza la planeación de cada *frame* de *uplink*

Por otro lado, los esquemas de Resolución de Colisiones de las SSs se localizan en el módulo *bwa_wm_child*, que es utilizado por cualquier SSs. Las figuras 4.3 (a) y (b) muestran en detalle la localización de los módulos para las SSs.

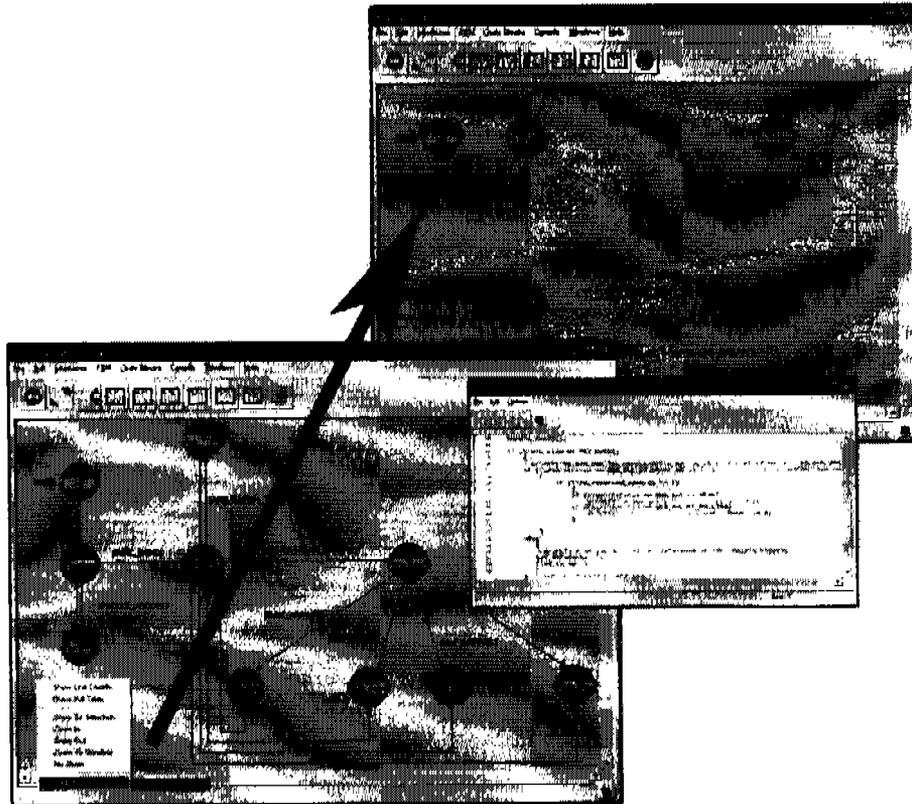
Para encontrar el módulo *bwa_wm_child*, del escenario de simulación, ingresamos en una SSs, figura 4.3 (a). Se puede observar que el módulo *bwa_data_station_IP* conformado por diversos módulos como antenas de recepción y transmisión, módulos para preparar la recepción o transmisión de paquetes, *link_rx* y *link_tx*, además de un módulo llamado *src_type_1*, donde se especifica el tráfico que la estación generará en la simulación. Pero el más importante es el módulo MAC, donde se encuentra todo lo relacionado con la capa MAC del protocolo 802.16, para las SSs.

En la figura 4.3 (b), se observa que el módulo MAC está estructurado por una FSM, conformada por los módulos: *INIT*, *Ranging*, *Syn_Rsp*, *App_Traffic*, *Idle*, *From_Downstream*, *Syn_Arvl*, *UCD_Arrival_MAP_Arrival* y *Down_Data_traffic*. De estos módulos, el módulo importante es *MAP_Arrival*, que es donde las SSs detectan si escuchan la llegada de un mapa. Al ocurrir la llegada de un mapa, las SSs invocan al módulo *bwa_wm_child* para realizar la transmisión de una petición de ancho de banda, si es que las SSs tienen información para transmitir.

El proceso *bwa_wm_child* se presenta también en la figura 4.3 (b), donde se observa que está conformado por una FSM con los módulos: *INIT*, *Q_Empty*, *Tx_Opp_Pending*, *No_Request_Outstanding*, *Request_Outstanding* y *Collision_Resolution*. En esta FSM las SSs resuelven todo lo relacionado con la resolución de colisiones, el envío de peticiones de ancho de banda y el cálculo de las oportunidades de transmisión.



(a) Diagrama de nodo y red para ingresar al modelado de las estaciones suscriptoras.



(b) Diagrama de FSM y código del módulo *bwa_wm_child*.

Figura 4.3 Ubicación del módulo *bwa_wm_child* de las estaciones suscriptoras.

4.4.1 Implementación 802.14-CSA

Para implementar este nuevo mecanismo, es necesario ingresar al modelo de la BS y modificar el estado *SEND_NXT_MAP* bajo las siguientes instrucciones:

```

if (multicast_resolution == 0) //no está activado el esquema multicast
{
  Nrs=(2*147)/(2+(1*26.6)); //número de slots de reservación
  carga=2.5*(147-Nrs); //carga de tráfico en la red
  if (slots_pending >= carga) //la red no tiene carga elevada asigna MCs
    lev_ie_request_prim (ld_ptr, CM_BROADCAST_SID, contention_slots);
  else //la red tiene carga elevada asigna MCs+Nrs
    lev_ie_request_prim (ld_ptr, CM_BROADCAST_SID, contention_slots+Nrs);
}
else //está activado el esquema multicast
{ realiza el esquema Multicast}

```

La BS esta diseñada de tal modo que lleva un registro minucioso de todas las tareas que realiza. Está estructurada por listas y la de más importancia, para en este esquema, es la de concesiones pendientes, *ie_lst_pending*. Con ayuda de un contador se puede determinar

cuántas concesiones, en términos de *slots*, tiene pendientes la BS. Ese contador se llama *slots_pending*, el cual se reinicializa si la lista *ie_lst_pending* se vacía.

La función *lev_ie_request_prim(ld_ptr, CM_BROADCAST_SID, contention_slots)* es la encargada de asignar los *slots* de contención para el siguiente *frame*. El primer parámetro, *ld_ptr*, es la estructura que compone a las regiones de contención. El segundo parámetro, *CM_BROADCAST_SID*, indica que la región que se incluye en el *frame* es del tipo *broadcast* y es para que las estaciones realicen sus peticiones de ancho de banda. El tercer parámetro, *contention_slots*, indicada la cantidad de *slots* de contención que se incluirán en el siguiente *frame* de señalización.

Este esquema, realizado para redes CATV, no tiene gran ventaja sobre algún otro que se muestra en este trabajo. Esto se debe a que en las redes CATV la duración del *frame* es de longitudes muy variables, lo cual no concuerda con el estándar 802.16 que establece una duración fija del *frame*. En este caso si la red tiene una carga de tráfico elevada, siempre asignará 10 *slots* de contención, a parte de los *slots* de contención mínimos, establecidos en el esquema Simple-CSA.

4.4.2 Implementación Collision Free-CSA

Para la implementación de este nuevo esquema propuesto, se hacen cambios considerables en el envío de los siguientes mapas en la BS. En el módulo *SEND-NEXT_MAP*. La estructura es la siguiente:

```
if (!lev_complete(ld_ptr, OPC_FALSE)) //verifica si el frame está completo
{
    int ff_slots, ds, counter_SS;
    ff_slots = lev_time_to_slots (Fixed_Frame_Duration); //convierte la duración del frame en slots
    ds = ff_slots - ld_ptr->num_slots; //número de slots restantes
    counter_SS = 0;
    if(ds > 0) //Si sobran slots
    {
        int a, listSize, u;
        qosMasterEntry *listEntry; //Importa la lista de usuarios activos
        listSize = op_prg_list_size(masterList);
        for(a = 0; (a < listSize) && (a < ds); a++) //coloca slots unicast hasta llenar el frame
            //o hasta llegar a la estación final
        {
            u = a + user_llmite;
            if(u < listSize) //No se ha llegado al llmite
            {
                listEntry = (qosMasterEntry *) op_prg_list_access(masterList, u);
                UNICAST_RESOLUTION_SID = -2000 - listEntry->SID;
                lev_ie_request_prim (ld_ptr, UNICAST_RESOLUTION_SID, 1); //slot unicast para
                // SS "u"
                counter_SS ++;
            }
            else
            {
                listEntry = (qosMasterEntry *) op_prg_list_access(masterList, u - listSize);
            }
        }
    }
}
```

```

        UNICAST_RESOLUTION_SID= -2000 - listEntry->SID;
        lev_ie_request_prim (Id_ptr, UNICAST_RESOLUTION_SID, 1);
        counter_SS ++;
    }
} //fin del for
if (u<listSize)
{
    user_limite=u+1; //guarda el usuario al que llegaste
}
else
{
    user_limite=u-listSize+1; //guarda el usuario al que llegaste
}
pad_slots=ds-counter_SS; /
if (pad_slots>0) //asegúrate que no sobró ningún slot en el frame
{
    lev_ie_request_prim (Id_ptr, -5000, pad_slots);
}
total_padding_slots = total_padding_slots + pad_slots;
} //fin del if(ds>0)
} //fin del frame completo

```

En éste mecanismo, la BS primero otorga las concesiones que pueden alojar en el siguiente mapa, una vez que se concedieron las regiones de reservación para las SSs, entra en el mecanismo Collision Free-CSA.

Se asegura de que la bandera *lev_completo* no esté activada, lo cual indica que el *frame* de *uplink* no se encuentra lleno. Se hacen los cálculos de cuántos *slots* le sobran al *frame*, esto se almacena en la variable *ds*. Después se importa la lista de SSs en la red y comienza a otorgar *slots* de contención *unicast* a cada una, por medio de un mecanismo *Round Robin* hasta que se llene el *frame* de *uplink*. Para que la BS sepa en cual SSs se quedó, la variable *user_limite* almacena el número de estaciones a las que se les ha asignado un *slot* de contención *unicast*, esta variable se reinicializa cuando la BS llega a la última SS de su lista de SSs activas.

Los *slots* de contención *unicast* son diferenciados de los de reservación por medio del CID, en este caso los CIDs *unicast*: -1001, -1002, -1003,... -1300, son para la SS1, SS2, SS3, ... , SS300, respectivamente. Para que las SSs transmitan en sus respectivos *slots* de contención *unicast*, sólo deben saber el CID que les corresponde. Esto se logra modificando, en el proceso *bwa_wm_child*, la función *find_tx_op*:

```

if (ie_ptr->sid < 0) /* Es un sid broadcast/multicast */
{
    else if (ie_ptr->sid == (-2000-station_SID)) //Es un sid para la estación
    {
        opportunity_size = 1; //es una oportunidad de transmisión
        cs_passed = 0;
        tx_opportunities = ((ie_ptr->ie_length/slot_time)-cs_passed) / opportunity_size;
        backoff -= (tx_opportunities+1);
        // envía la petición de ancho de banda
    }
}

```

Una vez que la estación suscriptora entra en la función *find_tx_op*, pregunta si el *sid* que está leyendo es para él. Si es así, entonces calcula el tiempo de su oportunidad de transmisión y envía su petición en el *slot* que se le asignó.

4.4.3 Implementación de Variable-CSA

Para este nuevo mecanismo propuesto, se debe hacer una distinción de que tipo de peticiones exitosas llegaron a la BS, si llegaron por contención o por reservación (*piggyback*). Para ello se modifica la estructura de peticiones de ancho de banda:

```
typedef struct
{
    int req_type; /* type of request 0 = minislots 1 = atm cells */
    int req_size; /* size of request in minislots or cells */
    int req_SID; /* SID of requester */
    double req_arrive_time; /* arrive time of a request */
    int bucketStatus; /* broadcom 7/1/98 */
    double req_CIR_time; /* UoS - Used for the implementation of the CIR */
    int flag_request; //Bandera para indicar el tipo de request
} REQUEST_STRUCT;
```

Con el parámetro *flag_request*, se puede determinar si la petición fue realizada por contención o *piggyback*.

La contabilización de estas peticiones se lleva a cabo en la lista de concesiones pendientes, de la siguiente forma:

```
/******
 * lev_ie_pending
 *
 * Issue a pending grant.
 *
 * Revision History
 * June 4, 1998 - MIL3 * Initial Development
 *****/
static void lev_ie_pending (lev_data *ld_ptr, REQUEST_STRUCT *rq_ptr)
{
    int tipo_req;
    FIN(lev_ie_pending());
    if (op_prg_odb_trace_active("func_track"))
        tipo_req=rq_ptr->flag_request;
        slots_pending=slots_pending + rq_ptr->req_size;
        cids_pending++;
        if (tipo_req==0) //la petición es de contención
            request_contencion=request_contencion + 1;
        else if(tipo_req==1) //la petición es de reservación
            request_data=request_data+1; //contabiliza la petición de reservación
        lev_ie_add (ld_ptr, IE_TYPE_ZERO_GRANT, rq_ptr->req_SID, 0, le_lst_pending);
        ld_ptr->num_pending_ies++;
        lev_ack_update (ld_ptr, rq_ptr->req_arrive_time);
        FOUT;
}
```

Ya que se determina la distinción de las peticiones, en el módulo `SEND-NEXT_MAP` de la BS se implementa lo siguiente.

```

if (multicast_resolution == 0)
{
    R=cids_pending; //peticiones pendientes
    MAX_DATA=147; //slots de datos máximos en el frame
    slots_cola=slots_pending; //slots pendientes
    G=slots_cola/MAX_DATA; //concesiones otorgables
    C=collisions_per_MAP; // colisiones en el mapa anterior
    Rc=request_contencion; //peticiones por contención

    if (R > G && C>=2) //Poca caga de tráfico y pocas colisiones
        Vcs=0;
    else if(R > G && C>=2 && Rc>2) //Poca carga de tráfico y muchas colisiones
        Vcs=(int)(min(2.7183*C,MAX_Vcs));
    else
        Vcs=MAX_Vcs;
    lev_ie_request_prim (ld_ptr, CM_BROADCAST_SID, contention_slots+Vcs);
        //Aloja los slots de contención necesarios
}
else
{Realiza el mecanismo multicast}

```

4.4.4 Implementación CDMA-CSA

Para este nuevo esquema propuesto, se incorporó una variable nueva, `CDMA_SCHEME`, en el archivo de especificaciones de capa MAC, `bwa_mac_default.ef`. Esto se logra en la función `cmts_get_sim_attributes (void)`. La variable `contetion_slots`, se fija en tres slots de contención según el esquema Simple-CSA. El apéndice B muestra el archivo de configuración `bwa_mac_default.ef`

Para implementar este mecanismo, es necesario ingresar al estado `SEND_NXT_MAP` de la BS y modificarlo bajo las siguientes instrucciones:

```

else if (use_cdma==1 && unicast==0)
{
    lev_ie_request_prim (ld_ptr, CM_BROADCAST_SID, contention_slots);
}

```

Lo anterior es concerniente sólo para la BS, en cambio, para la parte del modelo del usuario, es necesario establecer ciertas consideraciones.

El proceso `bwa_wm_child` de la SS es quien lleva toda la implementación. Cuando la simulación entra en este proceso, los primero que realizan las SSs es escuchar el mensaje descriptor del `uplink` y encontrar los tiempos en los que la región de contención inicia y termina. Esto se logra reconociendo los CID's denominados como `broadcast`, la SS detecta si

el CID que está analizando es CID=1001 (valor establecido en el modelo de simulación), si es el caso entonces el SS envía su petición de ancho de banda.

Para eliminar las colisiones, se le indica a la BS que haga caso omiso a los niveles de potencia recibidos en los *slots* de contención, esto se logró haciendo que las peticiones tuvieran una longitud de cero *bytes*. Entonces la BS puede recibir todas las peticiones que le hagan las SSs. Para este escenario, con un solo *slot* de contención, la BS podría soportar todas las peticiones, pero para reducir la interferencia se utilizarán los *slots* mínimos propuestos en el mecanismo Simple-CSA.

4.5 Conclusiones

En este capítulo se mostró cómo se implementaron las técnicas de estudio de los nuevos esquemas propuestos: 802.14-CSA, Variable-CSA, Collision Free-CSA y CDMA-CSA, en el modelo de simulación para sistemas basados en el estándar 802.16. Las técnicas Simple-CSA, Forced-CSA y Multicas-CSA son implementadas en [22], donde López Bejarano hace un análisis muy simple de su comportamiento. Se especificó la utilización de OPNET como herramienta de simulación para las técnicas analizadas y un panorama general de cómo está estructurada dicha herramienta.

CAPÍTULO 5

Simulación y resultados

5.1 Introducción

En esta sección se presentan los resultados que arrojan las simulaciones de los esquemas Simple-CSA, Variable-CSA, Forced-CSA, Multicast-CSA, 802.14-CSA, Collision Free-CSA y CDMA-CSA, estos son los mecanismos que se presentan en el capítulo 4. Se pondrá especial atención en el esquema Simple-CSA que es el adoptado por el estándar 802.16 para redes inalámbricas de banda ancha. Ese es el principal parámetro de comparación para el resto de los esquemas.

Se hace énfasis en los principales parámetros de simulación que son el “Throughput” y “Mean Access Delay” del sistema. Y se toma en consideración la longitud promedio de los paquetes generados por las SSs y por ende los *slots* necesarios para su transmisión. En términos de retardo, se hace una estimación utilizando el método presentado por Sriram [28] para calcular *end-to-end packet delay*, como el retardo promedio de acceso para el sistema basado en el estándar 802.16.

Por último se presenta la comparación de todos los esquemas que se analizan en este trabajo presentando importantes mejoras sobre el algoritmo adoptado en el estándar IEEE 802.16. Dichos resultados se presentan en forma modular, es decir al mecanismo Simple-CSA se le adjuntarán los demás mecanismos analizados y propuestos en esta tesis.

5.2 Parámetros de simulación

En el modelo de simulación observamos el desempeño de la red empleando tráfico de Internet. Este tráfico ya ha sido modelado con base en el tamaño de los paquetes Ethernet y su probabilidad de ocurrencia. Esta fuente emula el tráfico de acuerdo a la función de densidad de probabilidad PDF (*Probability Density Function*) recomendado por IEEE 802.14 *Working Group* [10]. La distribución del tamaño del mensaje de Ethernet se muestra en la figura 5.1.

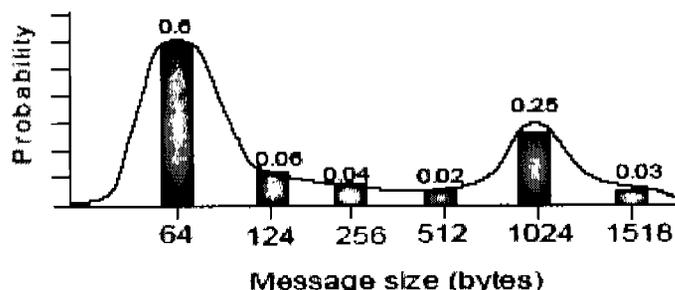


Figura 5.1 Distribución de paquetes de Ethernet.

El tamaño de los mensajes mostrado en la gráfica corresponde al *frame* en la capa MAC de Ethernet. En la capa de convergencia de IEEE 802.16 se agrega un encabezado de 6 bytes por lo que el tamaño del paquete a enviar por el medio físico es mayor y es el que se toma en cuenta para el envío de peticiones a la estación base.

La estación hace la conversión del tamaño del paquete de bytes a *slots*, ya que la solicitud de ancho de banda se hace en *slots*. En el estándar se manejan diferentes tamaños como 2, 4, 8, 16, 32, 64 y 128 para el tamaño de cada *slot*, pero el más utilizado es el de 16 bytes. A continuación se presenta la tabla 5.1 que muestra el número de *slots* correspondientes al número de bytes que cada paquete genera, utilizando el tamaño de 16 bytes para cada *slot*.

Con base en la tabla 5.1 podemos hacer una estimación del tamaño promedio, ℓ , de los paquetes a enviar por cada estación suscriptor.

Tamaño del paquete (bytes)	Slots requeridos	Probabilidad de ocurrencia
64	7	0.6
128	11	0.06
256	19	0.04
512	35	0.02
1024	69	0.25
1518	101	0.03

Tabla 5.1 Slots requeridos para cada paquete.

Este promedio es:

$$\ell_{\text{bytes}} = \left\{ \begin{aligned} &(64 * 0.6) + (128 * 0.06) + (256 * 0.04) + \\ &+(512 * 0.02) + (1024 * 0.25) + (1518 * 0.03) \end{aligned} \right\} = 368.1 \text{ bytes}$$

$$\ell_{\text{slots}} = \left\{ \begin{aligned} &(7 * 0.6) + (11 * 0.06) + (19 * 0.04) + \\ &+(35 * 0.02) + (69 * 0.25) + (101 * 0.03) \end{aligned} \right\} = 26.6 \text{ slots}$$

El tráfico generado para cada estación puede ser de 32 kbps o 64 kbps. En OPNET, esto se puede establecer mediante el tiempo de arribo entre cada paquete, $(1/\lambda)$, (*Packet interarrival time*). Si sabemos el tamaño promedio de cada paquete y el tráfico generado, tenemos que el tiempo de arribo entre cada paquete es:

$$\left(\frac{1}{\lambda}\right)_{32\text{kbps}} = \frac{368.1 * 8 \frac{\text{bits}}{\text{paquete}}}{32000 \frac{\text{bits}}{\text{segundo}}} = 0.092 \frac{\text{segundos}}{\text{paquete}} ; \lambda = 10.87 \frac{\text{paquetes}}{\text{segundo}}$$

$$\left(\frac{1}{\lambda}\right)_{64\text{kbps}} = \frac{368.1 * 8 \frac{\text{bits}}{\text{paquete}}}{64000 \frac{\text{bits}}{\text{segundo}}} = 0.046 \frac{\text{segundos}}{\text{paquete}} ; \lambda = 21.74 \frac{\text{paquetes}}{\text{segundo}}$$

5.2.1 Retardo promedio de acceso (*Mean access delay*)

Para hacer una estimación confiable del retardo de acceso promedio, se toma en cuenta: retardo promedio de encolamiento (D_q), que es el tiempo que el paquete permanece en la cola de la SS hasta que se realiza la primera petición de ancho de banda; retardo promedio de contención (D_c), que es el tiempo en que le toma a la SS transmitir una petición de reservación exitosa y; retardo de concesión (D_g), el retardo que le toma a la BS conceder el ancho de banda solicitado por la SS, hasta su total transmisión.

Un ejemplo del retardo promedio de acceso se aprecia en la figura 5.2, donde se observa cuándo empieza la contabilización de cada retardo y además el retardo de transmisión (D_{TX}), que para efectos de retardo, es despreciable en este análisis, Estableciendo un retardo promedio de acceso de la siguiente forma:

$$\overline{D_{ete}} = \overline{D_c} + \overline{D_q} + \overline{D_g}$$

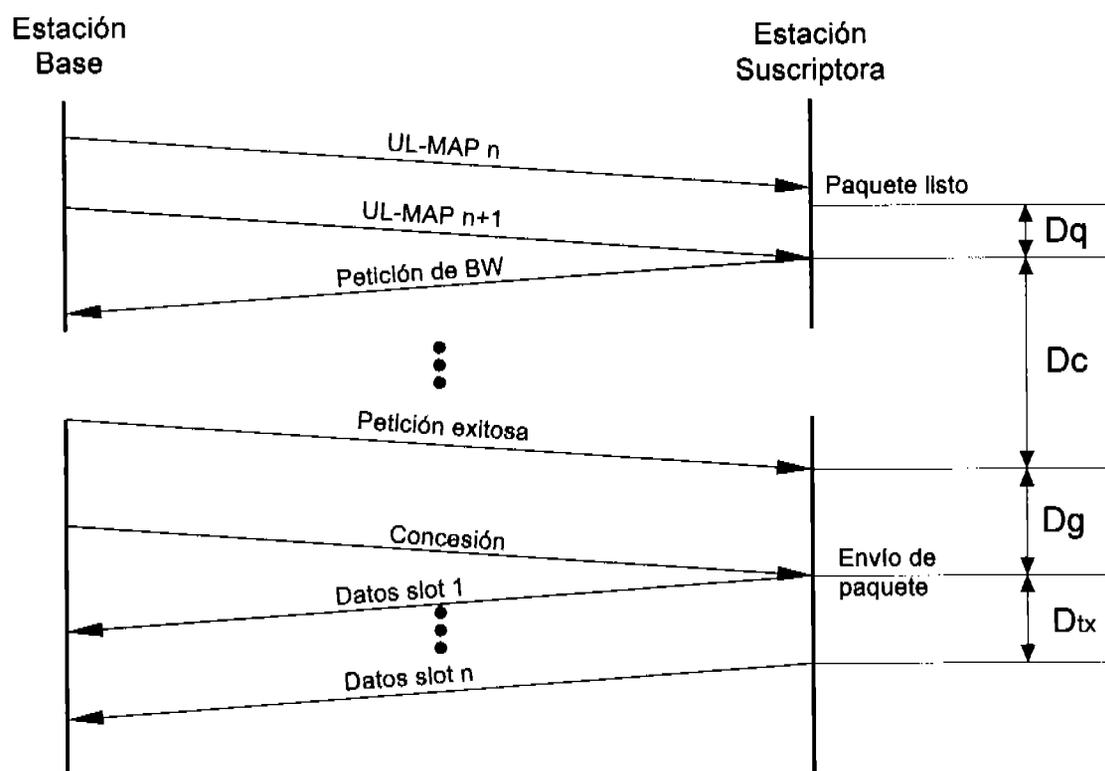


Figura 5.2. Diagrama de tiempo para la contabilización del retardo.

5.2.2 Throughput del Sistema

Utilicé la siguiente ecuación para estimar y validar el máximo *throughput* teórico en el *uplink* del sistema, se tiene:

$$S_{\max} = UL_{\text{bitrate}} \frac{Pk_{\text{MAC}}}{Pk_{\text{PHY}} + CS_{\text{REQ}}}$$

Donde las variables $Pk_{\text{MAC}} = 23$ slots y $Pk_{\text{PHY}} = 26.6$ slots, son el promedio, en *slots*, de las longitudes de los paquetes de tráfico de Internet en la capa MAC y PHY, respectivamente. El valor CS_{REQ} es el promedio de *slots* de contención por petición. Cuando el sistema opera en congestión, esto es cuando en la red hay más de 240 SS (indicado por el estándar 802.16), el valor $CS_{\text{REQ}} \rightarrow 0$, esto debido a que la red está congestionada y las SS hacen uso de las peticiones *piggyback* en el área de reservación. Tomando en consideración lo anterior, el *throughput* máximo teórico para el sistema analizado es:

$$S_{\max} = 9,600,000 \frac{23}{26.6 + 0}$$

$$S_{\max} = 8.3 \text{ Mbps}$$

La tabla 5.2 muestra en resumen los parámetros usados en el modelo de simulación.

Parametro	Valor
Esquema	OFDM
<i>Uplink</i> channel	3 MHz
<i>Uplink</i> bit rate	9.6 Mbps
Modulacion	QPSK
Tiempo del <i>frame</i> de señalizacion	2 ms
<i>Slots</i> por <i>frame</i>	150
Bytes por <i>slot</i>	64
Tráfico de Internet	33 Mbps
(1/λ)	2 s
Maximo numero de SS	300
Tiempo de simulacion	50 s

Tabla 5.2 Parámetros generales del modelo de simulación.

5.3 Diseño de la Red

Se generó una red con 300 estaciones suscriptoras (*Suscriber Stations*) fijas, cada una de ellas genera tráfico de Internet, atendidas por una estación base (*Base Station*). La distribución de los usuarios a lo largo de la red se realizó de manera aleatoria, para así tratar de simular una red apegada a la realidad. Esto se muestra en la figura 5.3.

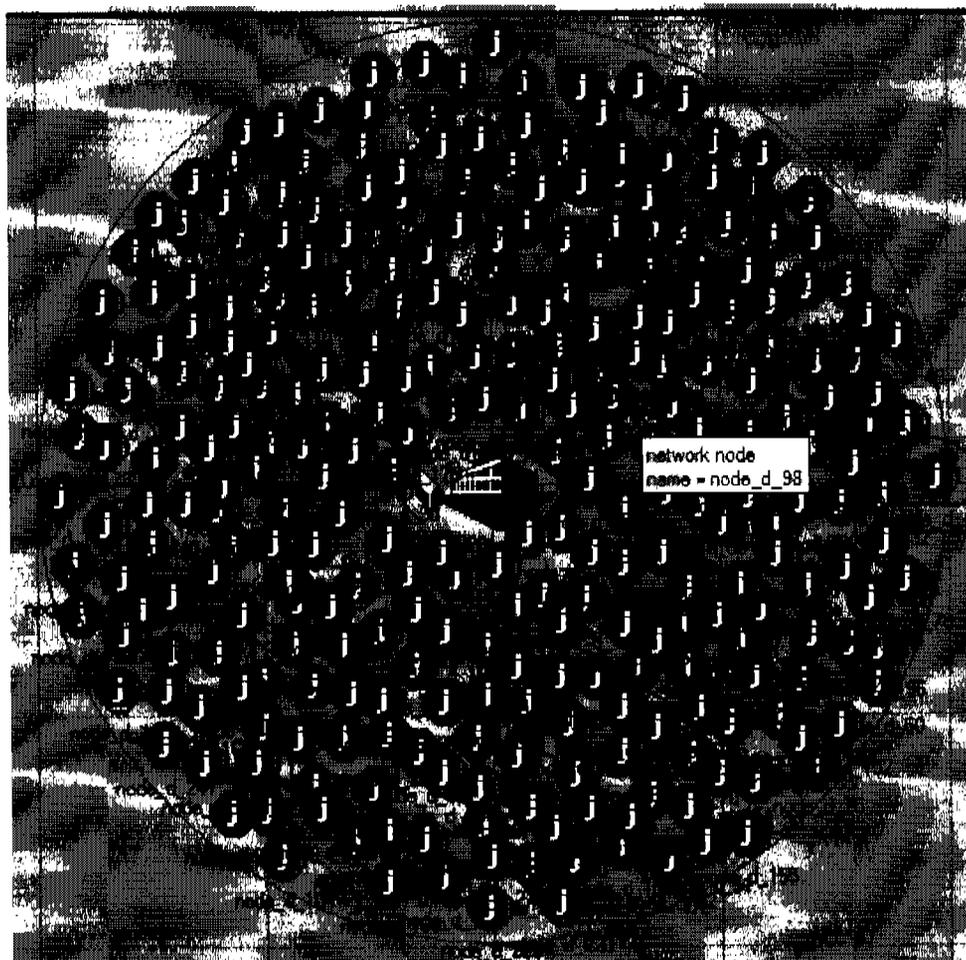


Figura 5.3 Escenario de simulación.

Cada simulación emula el comportamiento de una red por 60 segundos, los resultados que se generan se guardan en un archivo de tipo escalar, el cual contendrá todas las estadísticas promedio de los parámetros a analizar en las simulaciones.

Todos los parámetros relacionados con la capa física, PHY, se encuentran en el archivo `bwa_phy.ef`, y todos los relacionados con la capa BWA MAC se hallan en el archivo llamado `bwa_mac_default.ef`, como se detalla en el apéndice B.

Por otro lado, para que cada usuario pueda generar tráfico de Internet, es indispensable definir ciertos parámetros, los cuales indicarán la distribución de las llegadas de los

paquetes a transmitir y los periodos activos e inactivos de las SSs. Dichas variables se establecen como atributos del módulo generador de tráfico denominado “*src_type_1*”. En la figura 5.4 se presentan los parámetros de simulación para cada una de las SS's.

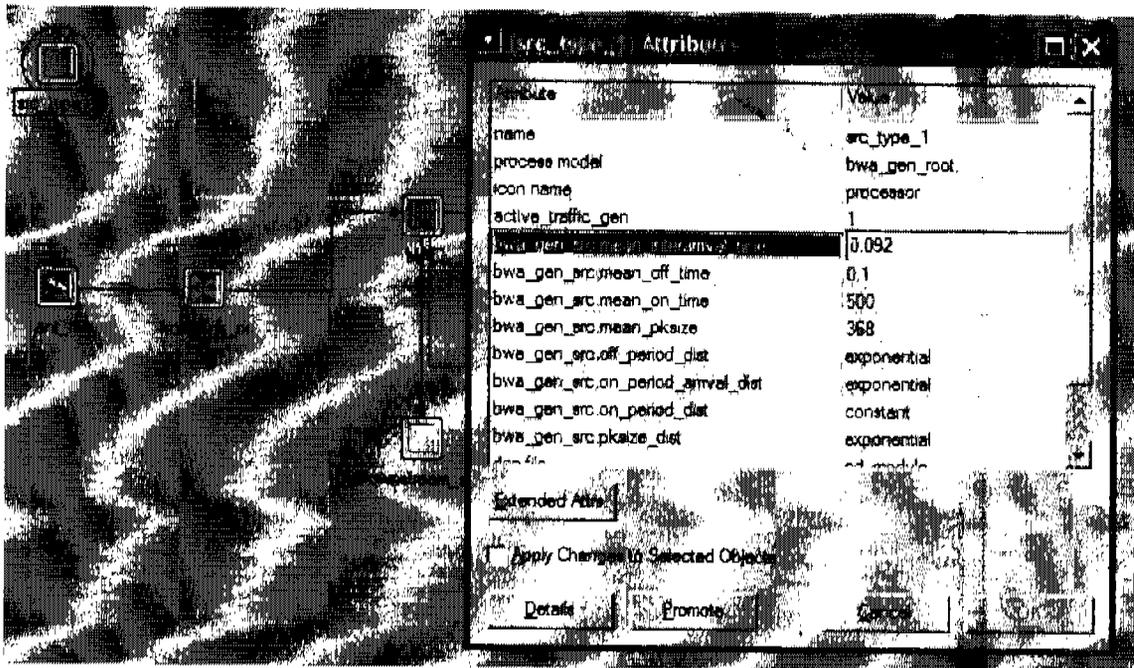


Figura 5.4 Parámetros de simulación de la estación suscriptoras SS.

Como se puede observar en la figura 5.4, la distribución que se emplea para generar tráfico de Internet es lo siguiente:

- ✦ Distribución de periodo de inactividad, (*off-period_distribution, mean_off_time*)
- ✦ Distribución de periodo de actividad, (*on-period_distribution, mean_on_time*)
- ✦ Distribución de las llegadas de los paquetes (*on_period_arrival_distribution, mean_interarrival time*) y la distribución del tamaño de los frames (*packet_distribution, packet_size*)

5.4 Resultados

Para la obtención de los resultados mostrados a continuación, es necesario indicar que la medición de tráfico se realizó con la colección de datos de a nivel nodo con la opción “Capture mode: All values” en OPNET. Esto permite ver el comportamiento a nivel-paquete. Todas las mediciones del retardo promedio de acceso se realizan a partir de todos los paquetes generados en la simulación.

Cada simulación de los esquemas analizados duró 60 segundos, considerando que durante ese tiempo se puede observar el comportamiento completo de los esquemas sobre una red basada en el estándar 802.16. OPNET permite realizar la simulación según la cantidad de estaciones suscriptoras que se tenga en la red. Para este caso, la simulación se realizó de 220 SSs hasta 300 SSs, realizando los promedios de retardo de acceso y *throughput*, cada 5

SSs. Por ejemplo, la SS40 genera un paquete y OPNET almacena en una lista el tiempo que permaneció el paquete en la cola del usuario, hace lo mismo para otro paquete generado por cualquier estación, de tal forma que al terminar la simulación, OPNET indica que cuando en la red había 240 SSs, en promedio, un paquete generado permaneció encolado en el usuario alrededor de 80 (msec). De la misma forma realiza la medición del retardo para contención y concesión.

Todas las estadísticas arrojadas por el modelo de simulación son realizadas para todos los paquetes generados, en promedio, cada simulación genera millones de paquetes, para cada uno de ellos y el número de estaciones suscriptoras en la red se hace la medición de las estadísticas de los paquetes.

5.4.1 SIMPLE-CSA

Este esquema es el que propone el estándar, asigna *slots* de contención que no se utilicen como reservados en cada *frame*. Mientras la carga de tráfico aumenta, pocos *slots* son asignados para contención, lo cual provoca un retardo elevado de encolamiento en las SSs a partir de 230 SSs, debido a que las SSs no pueden realizar sus peticiones de ancho de banda para enviar su información. Este detalle se puede observar en la figura 5.5, donde se muestra la gráfica de retardo de encolamiento.

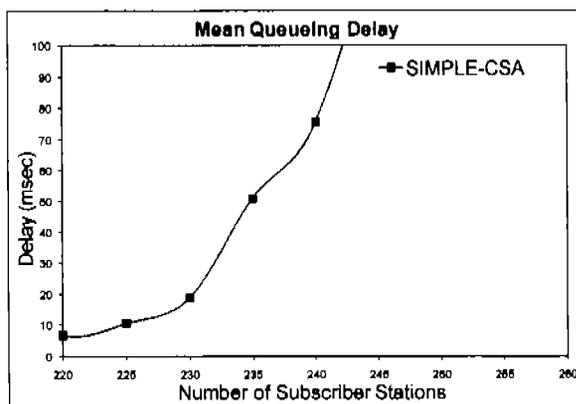


Figura 5.5. Queuing Delay.

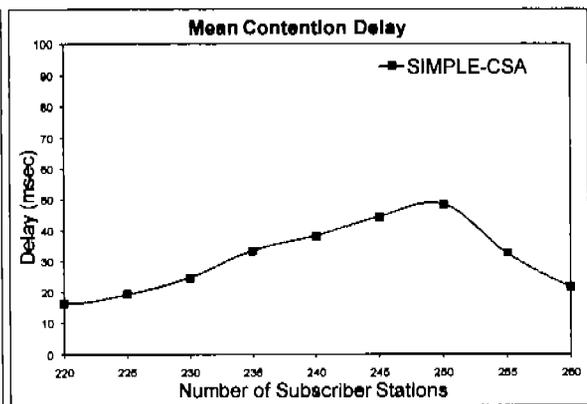


Figura 5.6. Contention Delay.

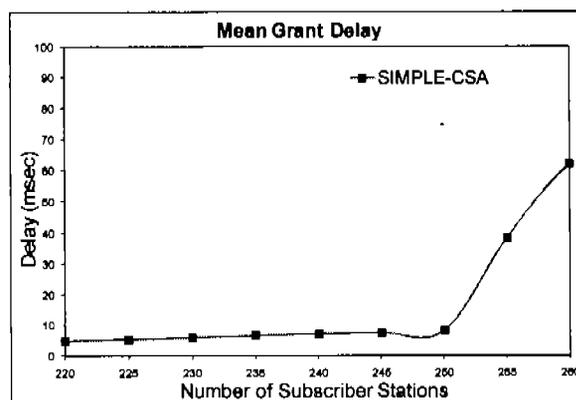


Figura 5.7. Grant Delay para Simple-CSA

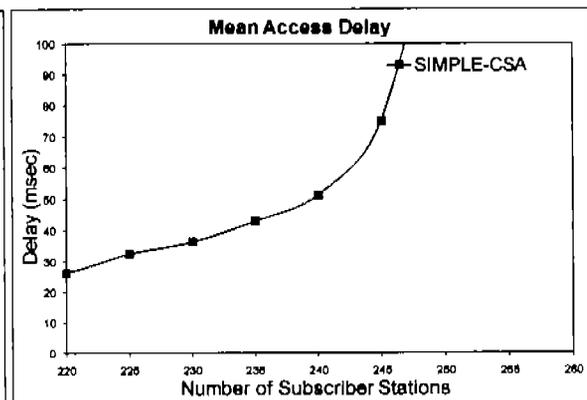


Figura 5.8. Mean Access Delay

En lo que se refiere a retardo de contención, D_c (fig. 5.6), el retardo es considerable, pero lo importante es que se observa con detalle el tiempo en el que actúa el esquema de peticiones *piggyback*, que se acentúa cuando en la red hay 250 SSs y el retardo de contención se reduce considerablemente porque las SSs dejan de mandar peticiones en contención, para mandarlas en las regiones de reservación de datos.

Para el retardo de concesión, D_g (fig. 5.7), se observa un aumento considerable a partir de 250 SSs, ya que los SSs realizan peticiones en *piggyback* y la BS tiene muchas concesiones pendientes, lo cual se traduce en tiempo de espera para conceder las regiones reservadas a todas las SSs que esperan su región para enviar su información.

En términos del *throughput* del sistema (fig. 5.9.), se puede observar que el *throughput* máximo para el sistema es de 8133.2928 kbps (8.1 Mbps), lo cual representa un porcentaje de error, contra el máximo teórico, de poco más del 2.5 %, tomando en consideración que el máximo teórico es de 8.3 Mbps. Este parámetro indica que el modelo de simulación incluye una implementación del esquema Simple-CSA correctamente.

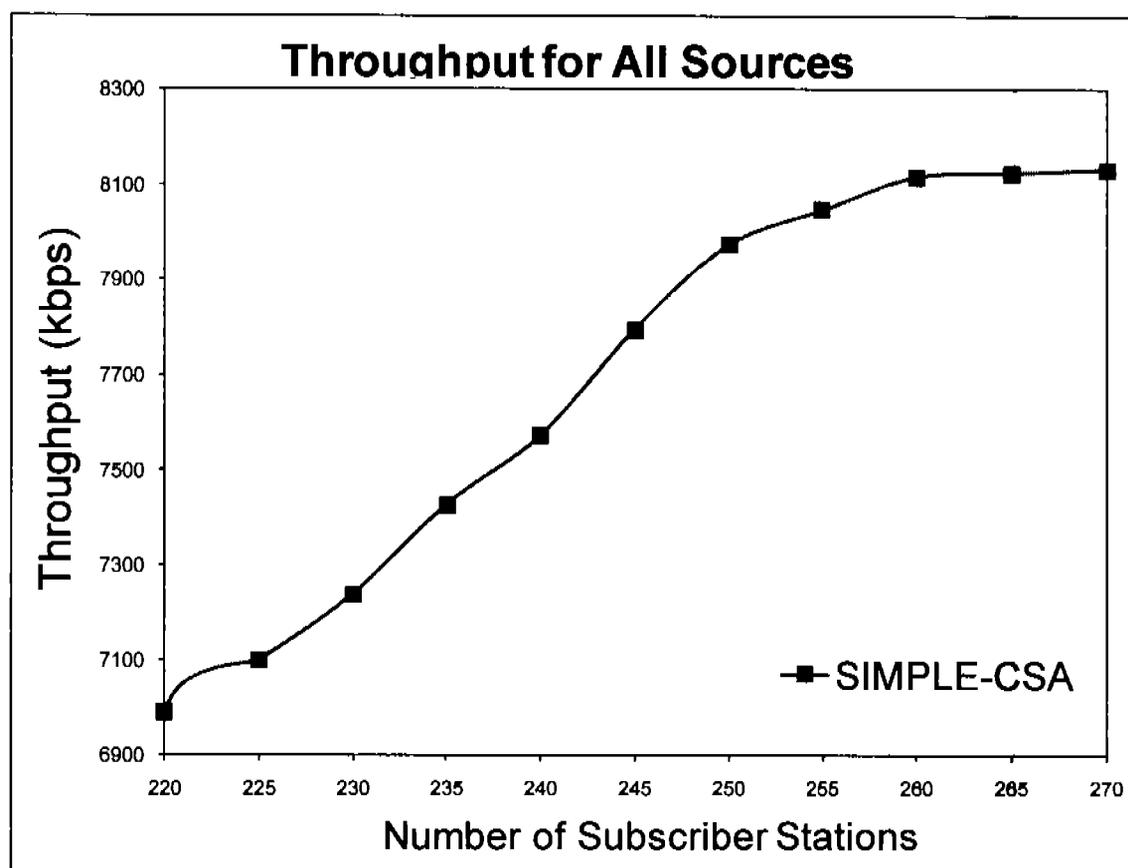


Figura 5.9. Throughput for all sources, para Simple-CSA.

5.4.2 Forced-CSA

El esquema Forced-CSA asigna slots de contención forzados “FCs” cada vez que ocurre una colisión. La BS asigna FCs por colisión, es decir que la BS aparta, de los slots fijos que componen a cada frame (150 slots), FCs slots por colisión. Se corrieron diferentes simulaciones con valores de FCs = 2,3,4 y 5, y los resultados que presentaron un comportamiento dinámico más eficiente fue con un valor de FCs = 3. Por lo tanto, los resultados que se presentan a continuación tienen como parámetro FCs = 3.

Como consecuencia de este esquema, el retardo de contención (fig. 5.11) se ve reducido ya que las SSs tienen más slots para poder realizar sus peticiones de ancho de banda y resolver las colisiones en las que se ven involucrados y las peticiones piggyback actúan a partir de que hay 245 SSs.

Para la gráfica de retardo de encolamiento (fig. 5.10), permanece en niveles bajos pero aumentan exponencialmente a partir de 235 SSs que es la principal causa del elevado retardo de acceso al sistema.

El mecanismo Forced-CSA hace que el retardo de concesión permanezca más tiempo por debajo, comparado con el esquema Simple-CSA.

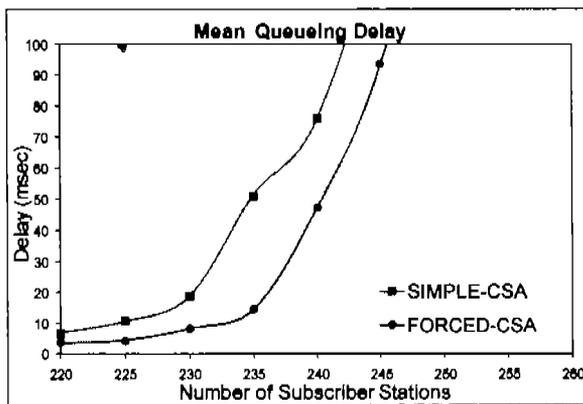


Figura 5.10. Queuing Delay.

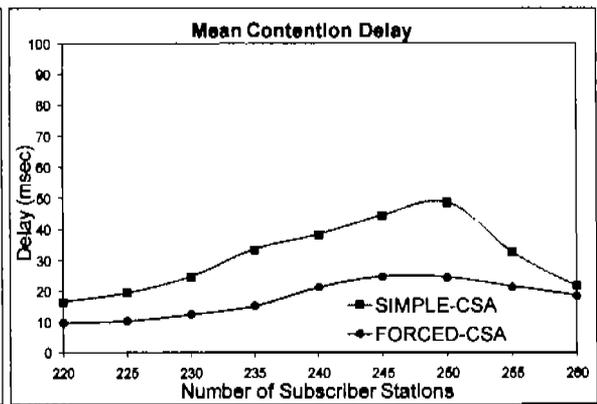


Figura 5.11. Contention Delay.

Tomando en consideración lo anterior y cuando el sistema se empieza a congestionar, se puede observar en la gráfica de retardo promedio de acceso, fig. 5.13, que el mecanismo Simple-CSA mantiene un retardo superior al 100% en la región izquierda, comparado con el mecanismo Forced-CSA. Pero en términos de retardo de contención supera al mecanismo Simple-CSA.

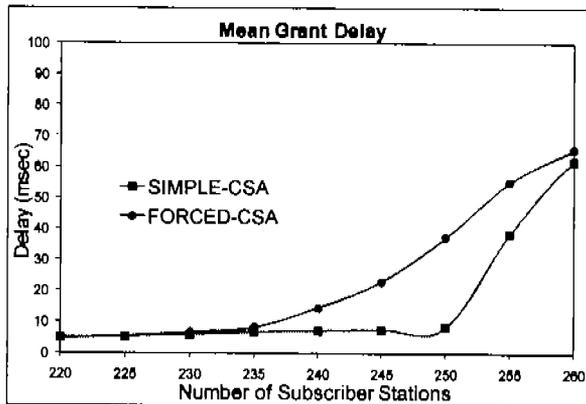


Figura 5.12. Grant Delay.

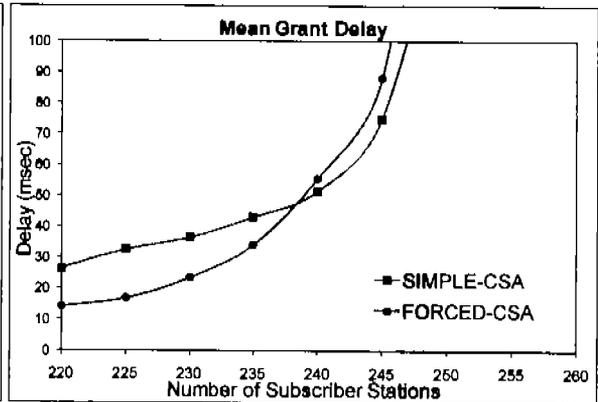


Figura 5.13. Mean Access Delay

Para lo relacionado al *throughput* del sistema (fig. 5.14.), se puede observar que el *throughput* máximo para el sistema, usando el mecanismo Forced-CSA, es de 8034.1456 kbps (8.0 Mbps), lo cual representa un porcentaje de disminución con el *throughput* máximo teórico de poco más del 3.6 %, tomando en consideración que el máximo teórico es de 8.3 Mbps y una disminución del 1.2% comparado con el mecanismo Simple-CSA.

Esta reducción es porque al hacer que las SSs tengan más *slots* de contención cuando se necesitan, y no cuando sobran, hace que la región de reservación disminuya en una muy pequeña fracción y el ancho de banda del *uplink* no se utilice de una forma óptima.

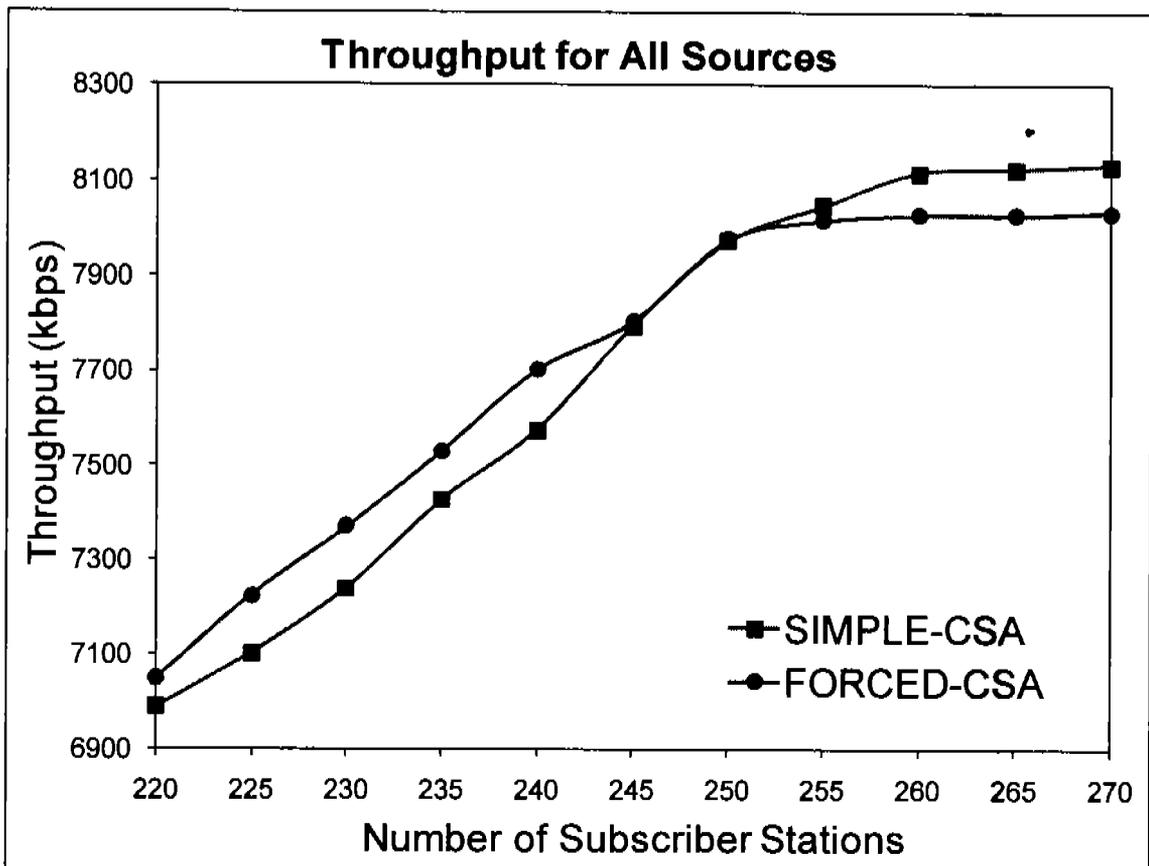


Figura 5.14. *Throughput* para todas las SSs: Simple-CSA y Forced-CSA.

5.4.3 Variable-CSA

En este esquema, se toma en consideración la carga de tráfico en la red, el número de colisiones en cada señalización de *frame*, así como el número de usuarios activos en la red. Prácticamente interactúa en la forma de asignar dinámicamente los *slots* de contención y los resultados de este esquema muestran que la BS concede las áreas de reservación a las SSs con mayor facilidad, esto es porque las peticiones de ancho de banda de las SSs tardan más tiempo en llegar a la BS. Esto se puede corroborar en los resultados que se muestran en la gráfica de retardo de concesión, figura 5.17.

El retardo de contención (fig. 5.16) aumenta desde 220 SSs y no se aprecia que las SSs hagan uso del mecanismo de peticiones de ancho de banda en regiones de reservación. Además el retardo de encolamiento, figura 5.15, aumenta exponencialmente a partir de 220 SSs. Esto indica que las peticiones exitosas de ancho de banda tardan en llegar a la BS y las que llegan son atendidas rápidamente.

Se puede apreciar que el mecanismo Variable-CSA tendría un comportamiento medio entre los mecanismos Simple-CSA y Forced-CSA, pero la gran ventaja del mecanismo Variable-CSA es que la BS concede más rápido las áreas de reservación de las SSs, porque las peticiones de ancho de banda que llegan exitosamente a la BS son pocas.

Esto indica que mientras más carga de tráfico exista en la red, la BS detecta más colisiones y debe alojar más *slots* de contención, robando espacio a las regiones de reservación.

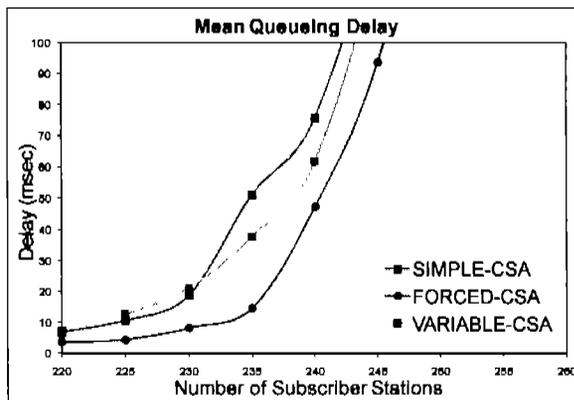


Figura 5.15. Queuing Delay.

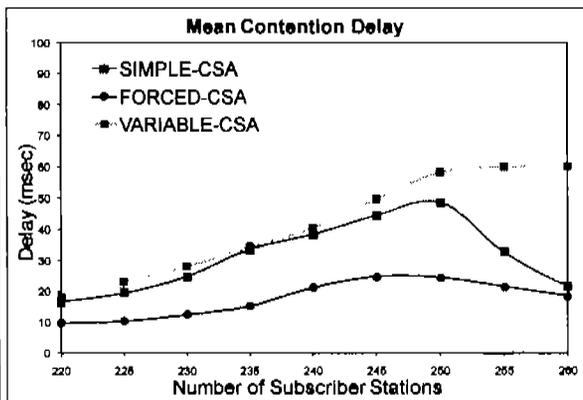


Figura 5.16. Contention Delay.

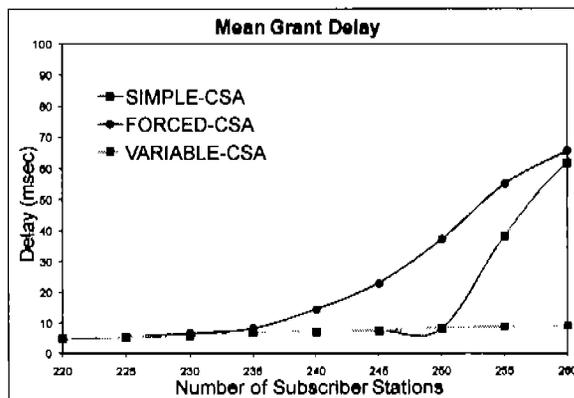


Figura 5.17. Grant Delay.

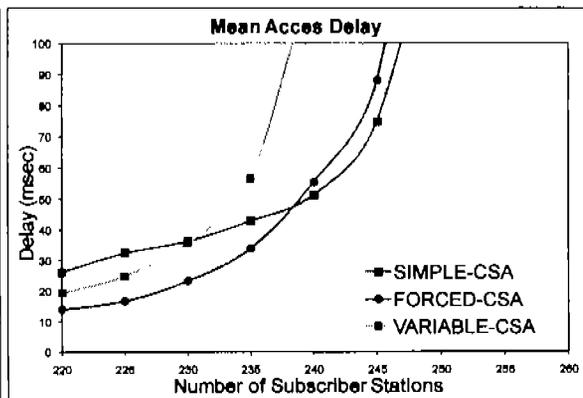


Figura 5.18. Mean Access Delay

En la figura 5.18 se puede observar que el mecanismo Forced-CSA, hasta ahora, es el que ofrece mejores resultados, en términos de retardo promedio de acceso.

En la gráfica de *throughput* del sistema (fig. 5.19), se puede observar que el *throughput* máximo para el sistema, usando el esquema Variable-CSA, es de 7926.592267 kbps (7.9 Mbps), lo cual representa un porcentaje de disminución con el *throughput* máximo teórico de poco más del 5.0 %, tomando en consideración que el máximo teórico es de 8.3 Mbps, y una reducción de 3 % en relación al esquema Simple-CSA.

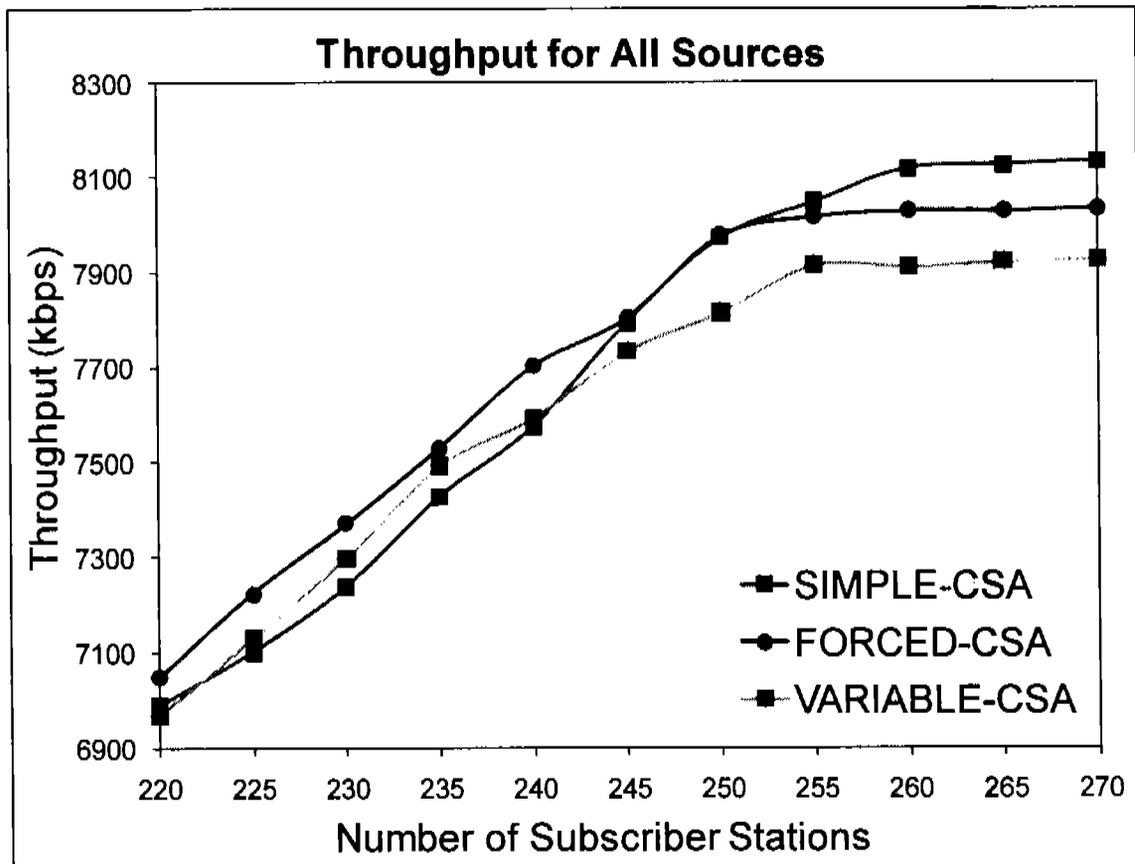


Figura 5.19. *Throughput* para todas las fuentes: Simple, Forced y Variable-CSA.

5.4.4 Multicast-CSA

El esquema *Multicast* es muy efectivo para el retardo de encolamiento (fig. 5.20.) y retardo de contención (fig. 5.21) ya que las colisiones se resuelven con mucha facilidad y las SSs no son castigadas como en el mecanismo Simple-CSA.

Se puede observar en la gráfica de retardo de contención, que los usuarios aún no han accionado el mecanismo de peticiones *piggyback* y el retardo se mantiene estable, por debajo de los mecanismos Simple-CSA, Forced-CSA, Variable-CSA.

El retardo de encolamiento se mantiene relativamente bajo hasta 235 SSs. A partir de éste punto, el retardo de encolamiento aumenta exponencialmente, pero por debajo de los demás esquemas.

Gran parte del retardo promedio de acceso, figura 5.23, lo provoca el hecho de que la BS tarda en conceder los recursos reservados para los usuarios que realizaron sus peticiones exitosas. Esto se debe a que la BS reduce la región de reservación para asignar slots de contención *multicast*, para que las colisiones se resuelvan lo más rápido posible.

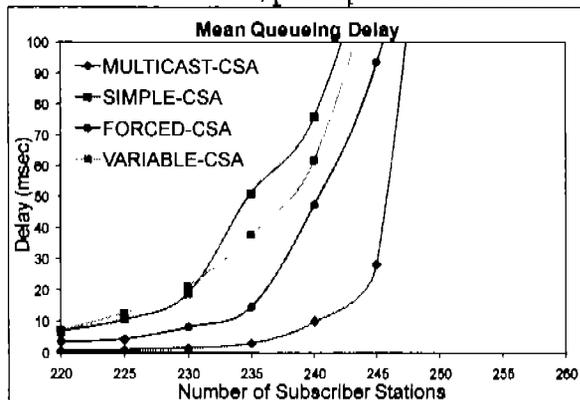


Figura 5.20. Queuing Delay.

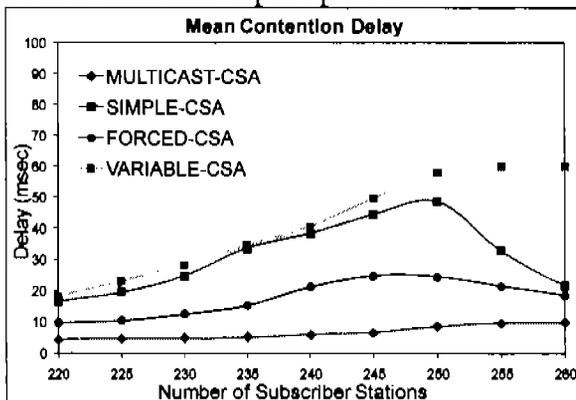


Figura 5.21. Contention Delay.

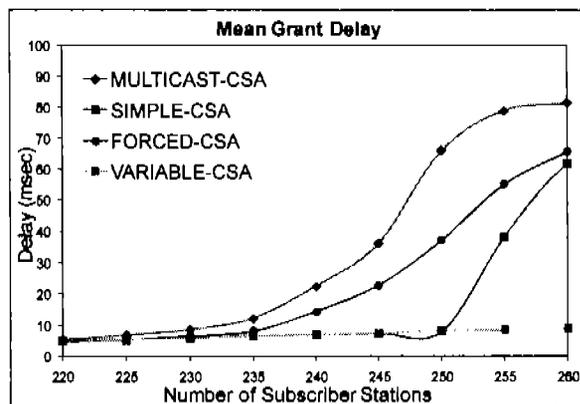


Figura 5.22. Grant Delay.

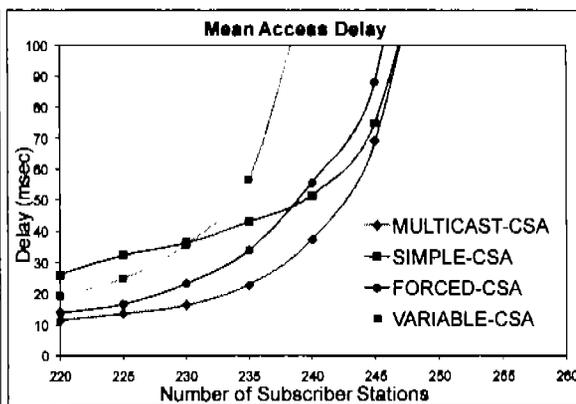


Figura 5.23. Mean Access Delay.

En la figura 5.22, retardo de concesión, se observa que la BS recibe muchas peticiones exitosas y para concederlas tarda mucho tiempo. Pero ante todo, se observa que el mecanismo Simple-CSA tiene un retardo promedio de acceso, cuando el sistema se empieza a congestionar, superior a 150 % comparado con el retardo que ofrece el mecanismo Multicast-CSA.

En este caso la gráfica de *throughput* del sistema (fig. 5.24), se puede observar que el *throughput* máximo para el sistema es de 8032.45376 kbps (8.0 Mbps), lo cual representa un porcentaje de disminución, con el *throughput* máximo teórico, de poco más del 3.6 %.

Comparando este *throughput* con el ofrecido por el mecanismo Forced-CSA, presentan los mismos resultados y con respecto al Simple-CSA, presenta una disminución de 1.2%.

Hasta ahora, el mecanismo que ofrece mejor *throughput* es el esquema Simple-CSA, pero el esquema que ofrece menor retardo de acceso, es el mecanismo Multicast-CSA.

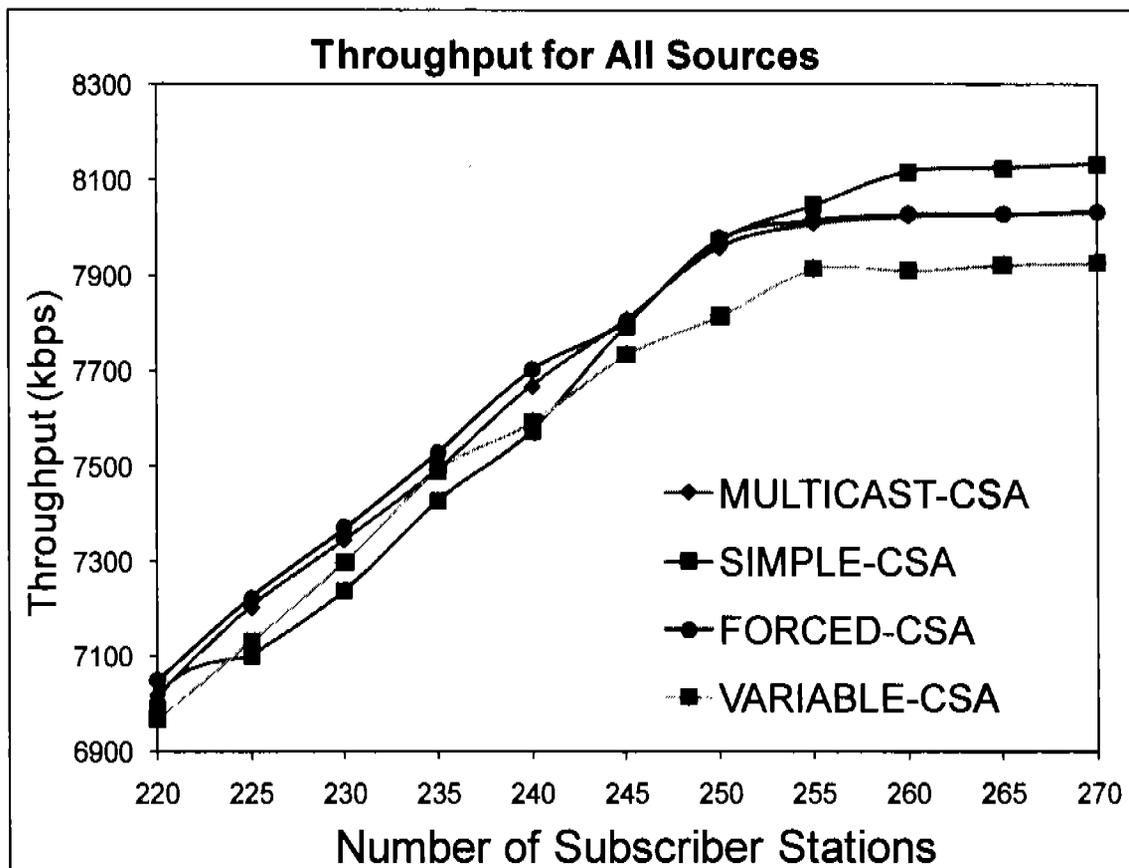


Figura 5.24. *Throughput* for all sources para Simple, Variable, Forced, Multicast-CSA.

5.4.5 802.14-CSA

Con este esquema se puede observar que el comportamiento de la red no es del todo óptimo ya que el retardo de encolamiento, figura 5.25, aumenta exponencialmente desde los 220 SSs, y eso degrada el funcionamiento del sistema. Tiene un comportamiento parecido al mecanismo Variable-CSA.

En términos de retardo de contención, figura 5.26, tiene un comportamiento muy similar al mecanismo Simple-CSA. También se observa que las SSs hacen uso de las peticiones de ancho de banda en régimen de reservación, *piggyback*, a partir de que hay 250 SSs.

En cuanto a retardo de concesión (fig. 5.27), la BS despacha rápidamente las peticiones de ancho de banda que le llegan con éxito en las regiones de contención, pero cuando las estaciones hacen uso de peticiones *piggyback*, cuando hay 250 SSs, se observa que la BS tarda en conceder los recursos a las SSs.

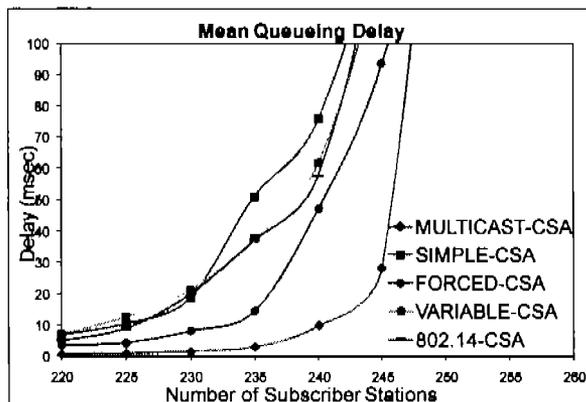


Figura 5.25. Queuing Delay.

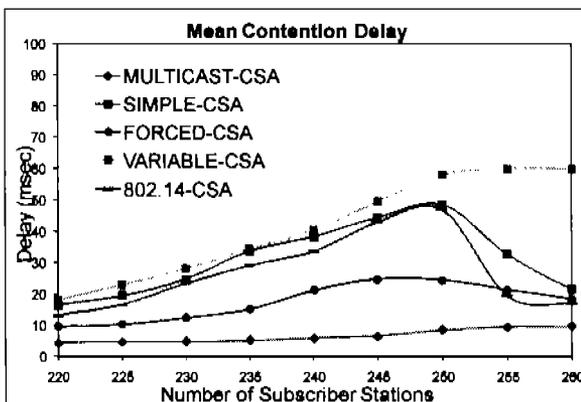


Figura 5.26. Contention Delay.

Este esquema 802.14-CSA, si bien fue un algoritmo que presentó una eficacia superior a Simple-CSA en el protocolo de televisión por cable IEEE 802.14 y DOCSIS, en los sistemas BWA no muestra una mejora, debido a que los *frames* son de tamaño fijo y las SSs hace uso de las peticiones *piggyback*, lo que hace que la región de contención no deba ser muy grande.

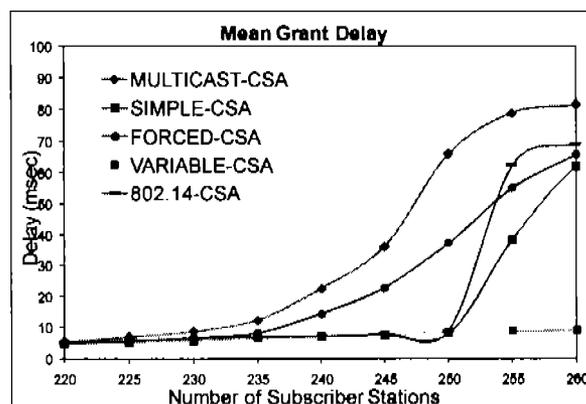


Figura 5.27. Grant Delay.

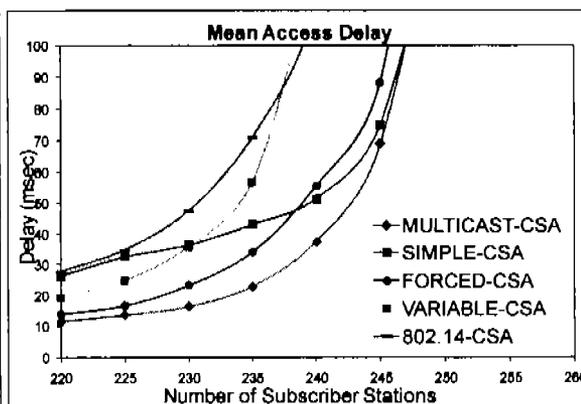


Figura 5.28. Mean Access Delay

El esquema 802.14-CSA arroja en la gráfica de *throughput* del sistema (fig. 5.29), un *throughput* máximo de 8019.390667 kbps (8.0 Mbps), lo cual representa un porcentaje de disminución con el *throughput* máximo teórico de poco más del 3.6 %, tomando en consideración que el máximo teórico es de 8.3 Mbps, y una disminución de 1.4 % comparado con el mecanismo Simple-CSA.

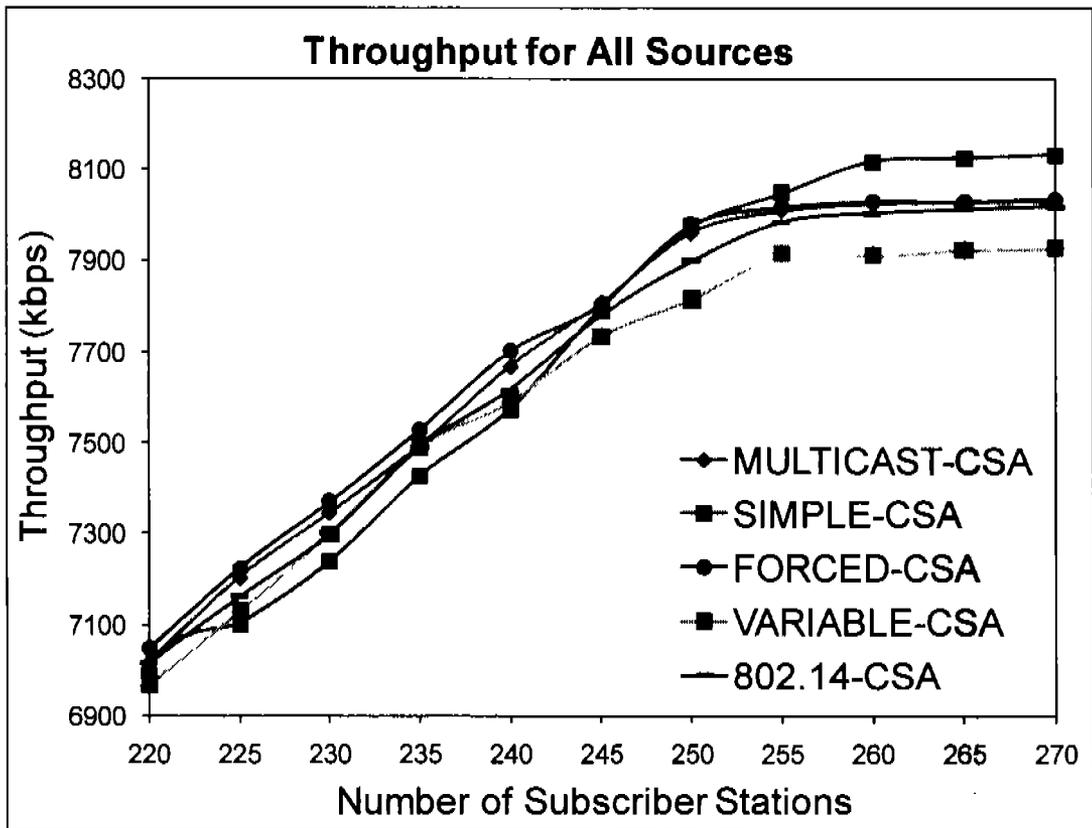


Figura 5.29. Throughput for all sources para Simple, Forced, Variable, Multicast y 802.14-CSA.

5.4.6 Collision Free-CSA

En los resultados obtenidos con la simulación del modelo Collision Free-CSA, podemos observar que el retardo de encolamiento, figura 5.30, es el más bajo, esto debido a que los paquetes son enviados más rápido. Pero el retardo de contención desaparece en este esquema, pero aparece un retardo de consulta (fig. 5.31), ya que las estaciones deben esperar hasta que se les asigne un *slot unicast* para enviar sus peticiones de ancho de banda. Se observa también que las SSs hacen peticiones *piggyback* a partir de que hay 255 SSs activas en la red.

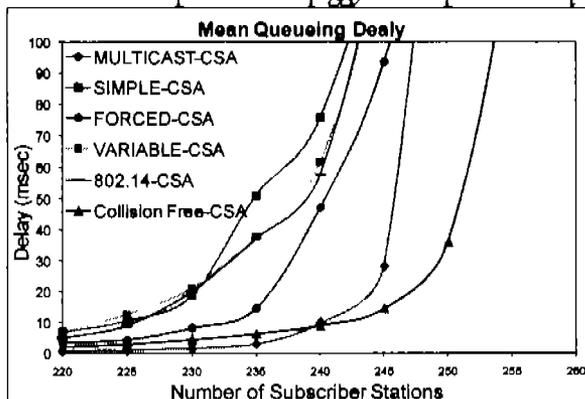


Figura 5.30. Queuing Delay.

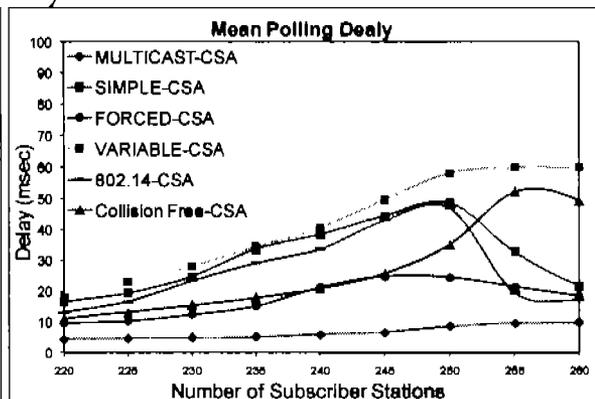


Figura 5.31. Polling Delay.

En cuanto al retardo de concesión (fig. 5.32), permanece por debajo del retardo del esquema Simple-CSA, debido a que la BS resuelve muy rápido las concesiones pendientes. Pero se observa que a pesar de todo, el retardo promedio de acceso, figura 5.33, permanece bajo hasta que la red está soportando 255 Ss.

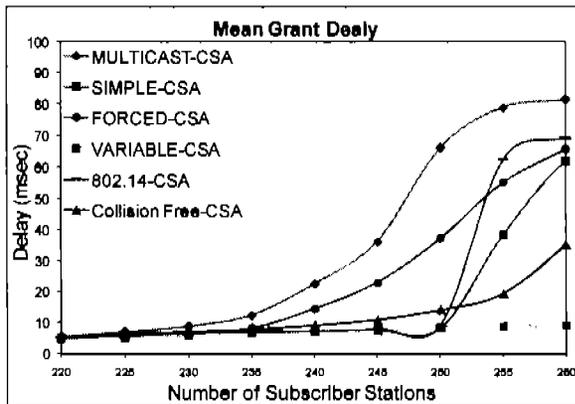


Figura 5.32. Grant Delay.

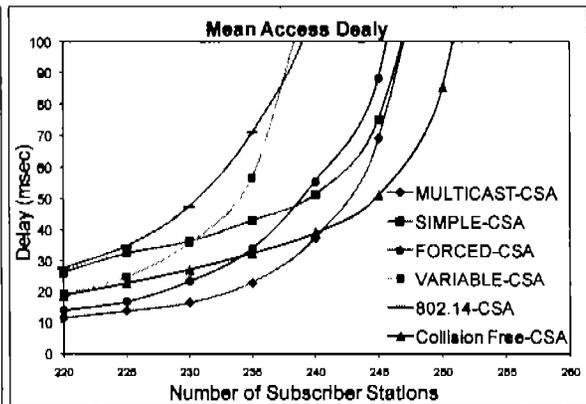


Figura 5.33. Mean Access Delay

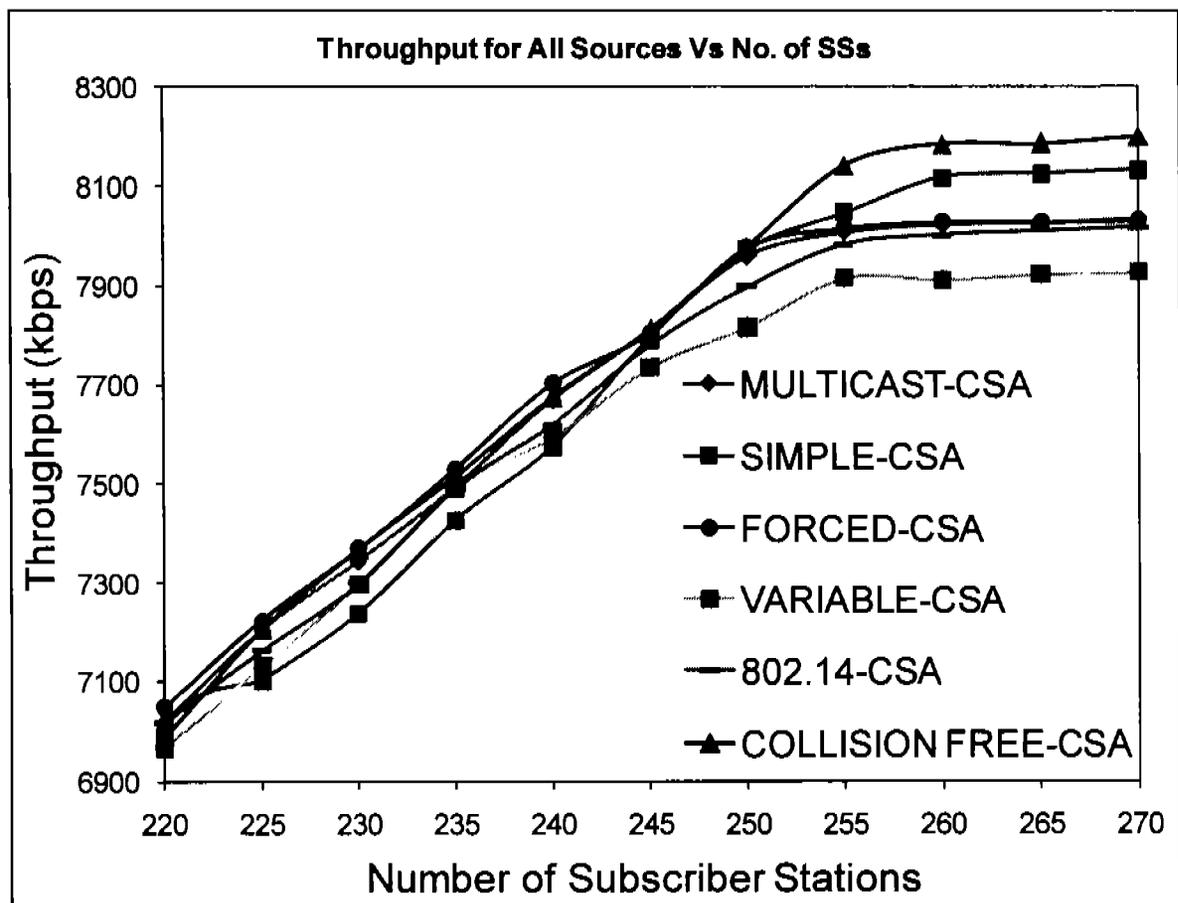


Figura 5.34. Throughput for all sources para Simple, Forced, Variable, Multicast, 802.14 y Collision Free-CSA.

Este esquema, Collision Free-CSA, arroja en la gráfica de *throughput* del sistema (fig. 5.34), un *throughput* máximo de 8201.4664 kbps (8.2 Mbps), lo cual representa un porcentaje de disminución con el *throughput* máximo teórico de poco más del 1.2 %. Además una mejora de 1% en comparación con el esquema Simple-CSA.

Esto indica que el esquema Collision Free-CSA presenta mejores resultados en *throughput* y retardo de acceso para sistemas basados en el estándar IEEE 802.16, que el adoptado por el propio estándar, Simple-CSA.

5.4.7 CDMA-CSA

Este nuevo mecanismo propuesto CDMA-CSA, es muy eficiente en cuanto a retardo de contención, como se observa en la figura 5.36, donde se observa que el retardo de contención utilizando la técnica CDMA está muy por debajo de los demás esquemas, además de que el retardo de encolamiento, figura 5.35, también permanece por debajo de los demás esquemas. Este comportamiento se debe a que en la BS recibe todas las peticiones de ancho de banda de las estaciones en la pequeña región de tres slots de contención, que la BS aloja en cada *frame*.

Las estaciones sólo deben esperar a que llegue dicha región, por lo que los paquetes permanecen muy poco tiempo encolados y por ende muy poco tiempo tratando de hacer peticiones de ancho de banda con éxito.

El retardo mayor, para esta técnica, es el generado por la concesión, figura 5.37. Se observa que el retardo de concesión aumenta, ya que la BS recibe todas las peticiones pero para alojar las regiones de reservación tarda bastante a partir de que en la red hay 235 SSs.

Sin embargo, el retardo promedio de acceso a la red es el más bajo de todas las técnicas analizadas en este trabajo de tesis. Lo cual se traduce en que el retardo promedio del mecanismo Simple-CSA es mayor en el orden de 200 % comparado con el esquema propuesto CDMA-CSA.

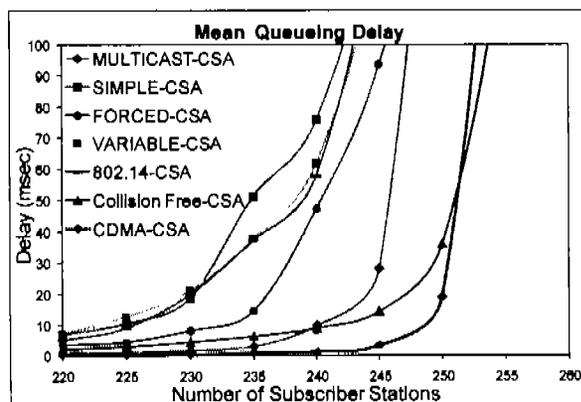


Figura 5.35. Queueing Delay.

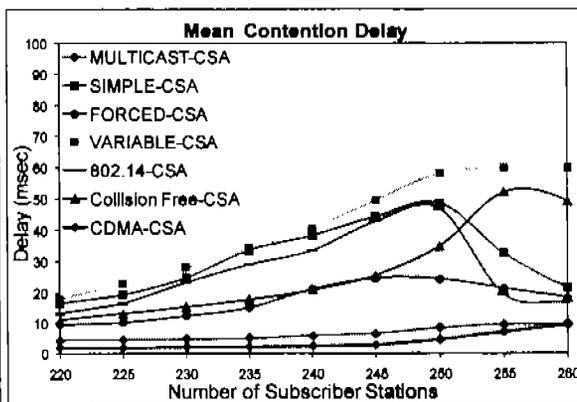


Figura 5.36. Contention Delay.

Sin embargo, pese a estos buenos resultados de retado, no se ve tan favorecido en cuando a *throughput*, ya que, como se observa en la figura 5.39, arroja un *throughput* máximo de 7935.508267 kbps (7.94 Mbps). Los que representa un error de 4 % con respecto al máximo teórico de 8.3 Mbps, y una disminución de 2.4% comparado con el mecanismo Simple-CSA.

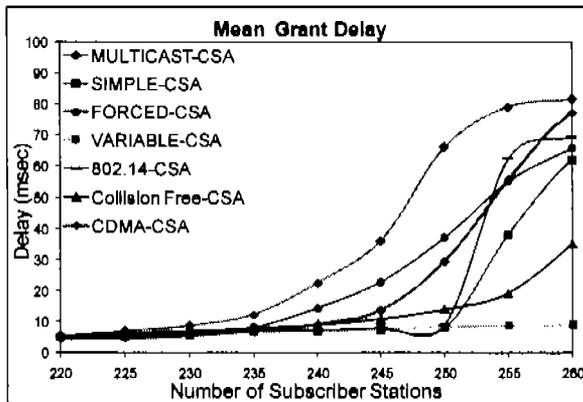


Figura 5.37. Queuing Delay para CDMA-CSA

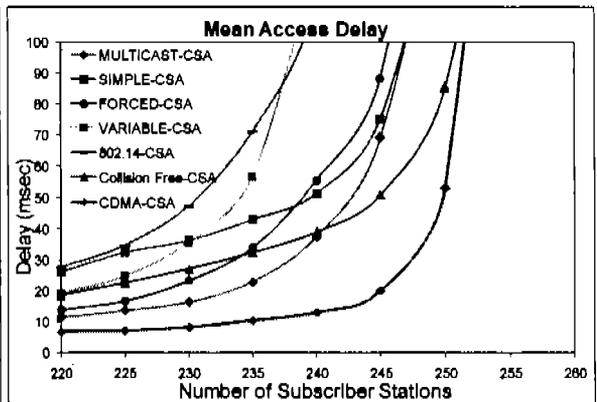


Figura 5.38. Contention Delay

Otro parámetro de importancia es la utilización del canal de subida, en la figura 5.40, podemos observar que la utilización del *uplink* para el esquema Collision Free-CSA es muy cercano al 100 %, en la región de congestión.

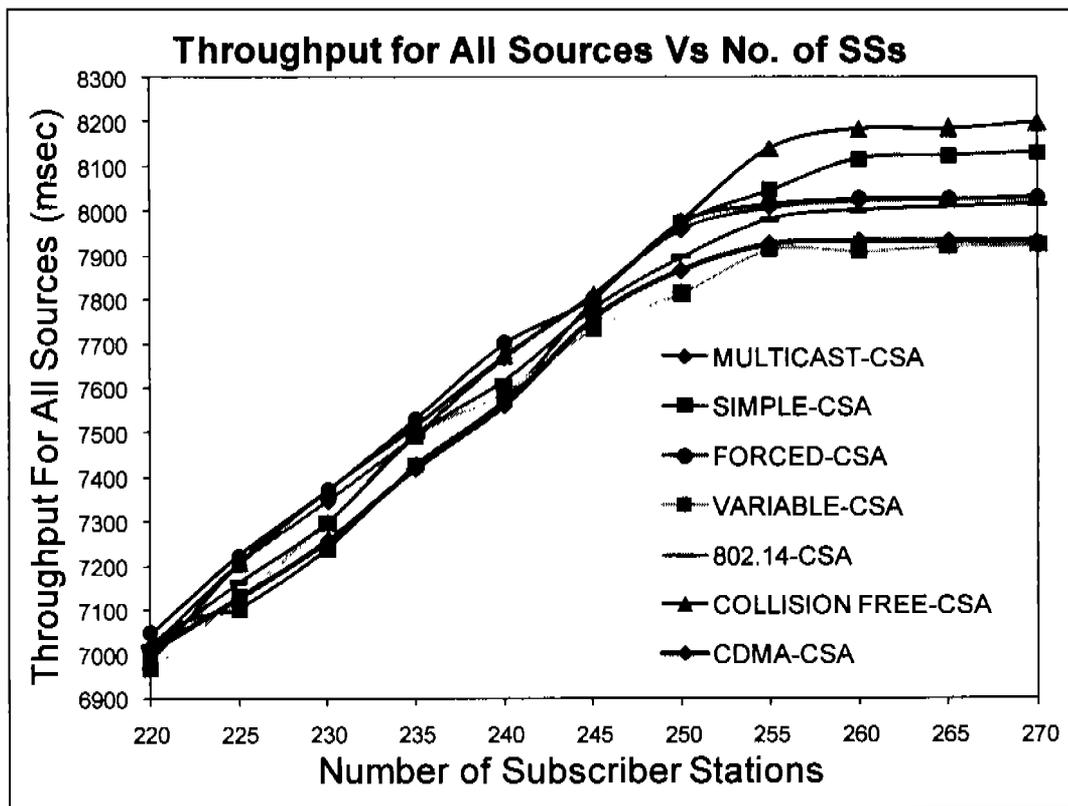


Figura 5.39. Gráfica de comparación de *Throughput for All Sources* para los esquemas analizados.

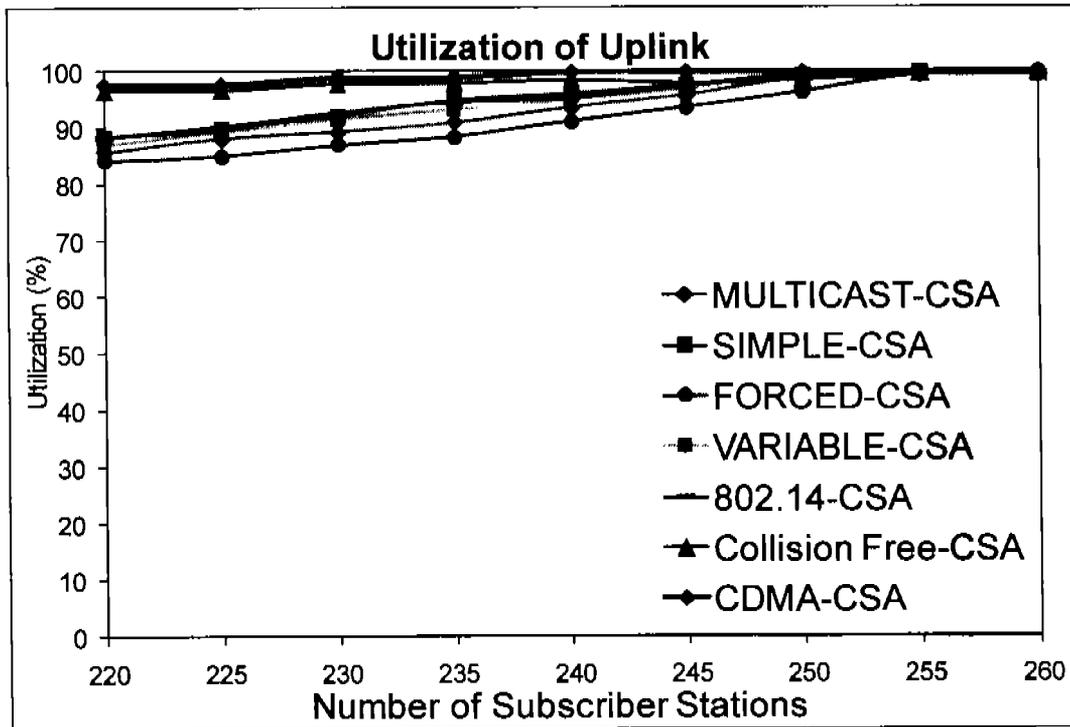


Figura 5.40. Utilización del canal de subida de las técnicas analizadas.

5.6 Conclusiones

En este capítulo se mostró que, en general, el desempeño de los sistemas de banda ancha utilizando el estándar IEEE 802.16 puede ser optimizado significativamente por el uso de algoritmos de contención.

Los algoritmos presentados en este trabajo fueron: Simple-CSA, Forced-CSA, Variable-CSA, Multicast-CSA, 802.14-CSA, Collision Free-CSA y CDMA-CSA, los cuales adaptan el número de slots de contención que deberán ser alojados en la siguiente señalización de *frame*, basados en la carga de tráfico. Los resultados presentados en este capítulo han hecho hincapié en una reducción considerable en el retardo promedio de acceso y el *throughput* del sistema, comparado con lo estipulado en el estándar 802.16.

En términos generales, los resultados obtenidos por los esquemas propuestos y analizados muestran un comportamiento superior, en términos de simulación de tráfico de Internet, para una calidad de servicio *Best Effort*. Los esquemas Multicast-CSA y Collision Free-CSA tienen una gran mejora sobre el comportamiento de Simple-CSA adoptado en el estándar 802.16.

CAPÍTULO 6

Conclusiones generales

El principal objetivo de esta tesis es la evaluación de técnicas para la resolución de contenciones dinámicas en redes WiMAX, que permitan disminuir los periodos de contención de las estaciones activas en la red. Esta reducción se ve reflejada en el retardo promedio de acceso a la red.

En el presente trabajo de tesis se propusieron cuatro nuevos esquemas: Variable-CSA, IEEE 802.14-CSA, Collision Free-CSA y CDMA-CSA, y el análisis de éstos se hizo junto con tres esquemas: Simple-CSA, Forced-CSA y Multicast-CSA, propuestos en López Bejarano [22]. Todas éstas técnicas se estudiaron para redes basadas en el protocolo 802.16, para redes inalámbricas de banda ancha.

Con el término de esta tesis se elaboró una herramienta que facilita el estudio de las redes inalámbricas de banda ancha, que permite realizar el análisis del comportamiento dinámico de redes BWA de una forma más sencilla. Esta herramienta tiene como características:

- ✦ Presentar una interfaz gráfica amigable para el usuario.
- ✦ Clasificar los parámetros de simulación en dos conjuntos, de acuerdo a las capas del modelo de referencia OSI:
 - Capa MAC
 - Capa PHY, y
- ✦ Crear modelos de red personalizados compatibles con *OPNET Modeler*.

Los parámetros de análisis en este trabajo fueron: retardo promedio de acceso (*Mean Access Delay*) y máxima capacidad de transmisión (*Throughput*), que indican el funcionamiento de la red. Observando la gráfica de retardo de contención de todas las técnicas (Fig. 5.36), los esquemas CDMA-CSA y Multicast-CSA son los que presentan un menor retardo de contención. El mecanismo CDMA-CSA recibe todas las peticiones de ancho de banda que lleguen en el *uplink*, y prácticamente desaparecen los periodos de contención. Mientras que la técnica Multicast-CSA presenta un funcionamiento dinámico para resolver rápidamente las colisiones. En cuanto a la repercusión de estas dos técnicas en el retardo promedio de acceso (Fig. 5.28), CDMA-CSA es la que presenta un menor retardo promedio de acceso, además se observa que mejora al mecanismo adoptado por el estándar, Simple-CSA, en poco más de 150 %.

En lo que se refiere a *throughput* de todas las técnicas (Fig. 5.39), el esquema Collision Free-CSA es el que presenta un mejor funcionamiento de la red, muy por encima de los esquemas Multicast-CSA y CDMA-CSA. Esto se debe a que la técnica Collision Free-CSA se enfoca a

que la BS aloje las regiones de reservación más rápido y de una manera eficiente para otorgar *slots* de contención *unicast* a las SSs.

Haciendo un balance de resultados, la técnica Collision Free-CSA —propuesta en esta tesis— presenta un menor retardo de acceso y un mayor *throughput* de sistema, para redes basadas en el estándar 802.16, de redes inalámbricas de banda ancha. Ésta disminuye un 50% el retardo promedio de acceso a la red y tiene un error de 1.2 % en relación al *throughput* máximo teórico de 8.3 Mbps

También es importante recalcar que la implementación, vía *software* en capa MAC, de las técnicas Forced-CSA, Multicast-CSA y Collision Free-CSA es muy sencilla ya que las estaciones sólo necesitan conocer los *slots* de contención en dónde pueden realizar sus peticiones de ancho de banda, la estación base es la que realiza todo el mecanismo.

Por otro lado el esquema CDMA-CSA sería muy costoso, porque habría que incluir su operación en *hardware*, pero debido a que el estándar 802.16-2004 utiliza CDMA para *ranging* (calibración), este funcionamiento se podría extender para el envío de solicitudes de ancho de banda.

Como contribución de la tesis, se produjo el artículo “*Performance Evaluation of Adaptive Contention Slot Allocators for IEEE 802.16 Based Systems*”, que se planea presentar en una conferencia internacional en 2008. El artículo se presenta en el Apéndice C, cabe recalcar que dicho artículo está en corrección para su envío.

APÉNDICE A

Modelo de Simulación de Redes WiMAX

A.1 Escenario

En este modelo se incluye una red con estaciones suscriptoras, estos nodos producen tráfico de Internet, como se muestra en la figura A.1.

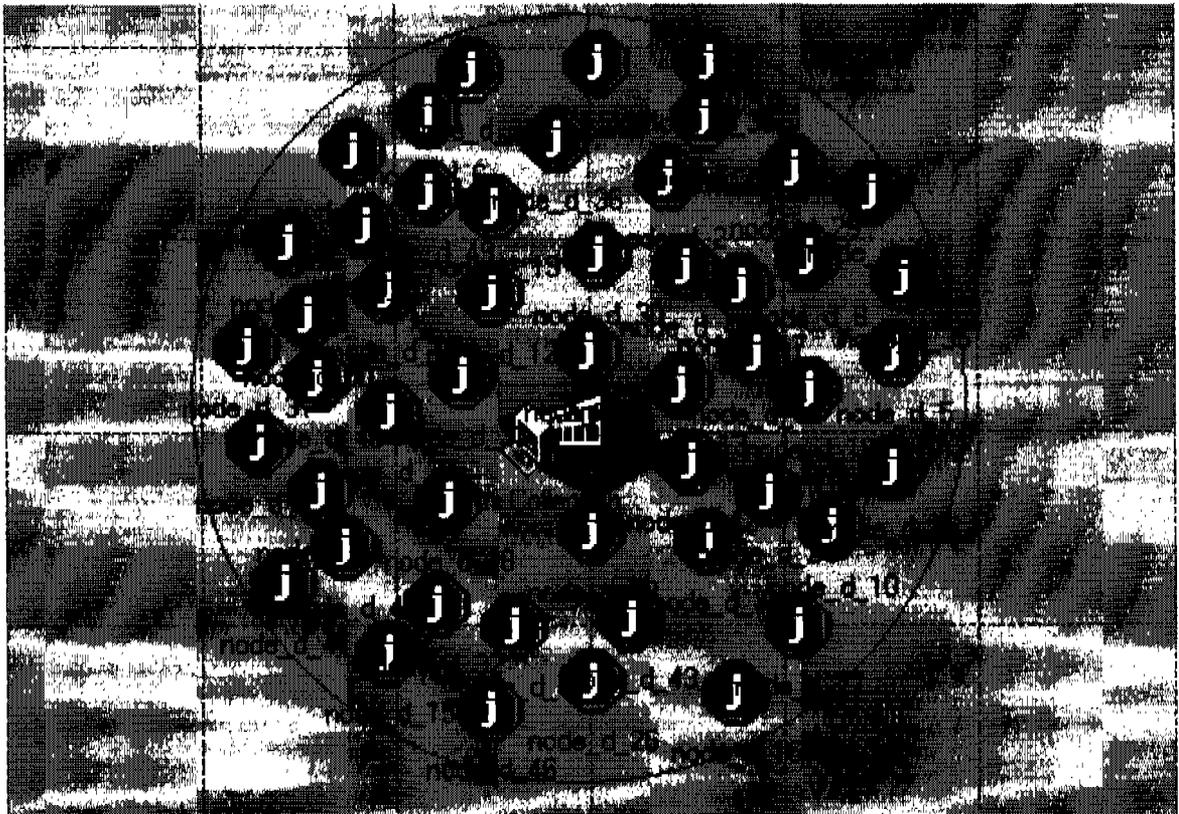


Figura A.1 Escenario de simulación.

El escenario está formado por una estación base (*Base Station-BS*) y estaciones suscriptoras (*Suscriber Stations-SSs*).

En los esquemas de simulación, la BS y las SSs soportan fragmentación *piggyback*. Un *slot* equivale a 16 bytes para todos los casos y todas las SSs transmiten con un esquema de modulación QPSK. La BS transmite con una modulación 16-QAM.

A.2. Diseño de la Estación Base (*Base Station*)

La BS permite el establecimiento de las conexiones a cada una de las SSs, también es la responsable de controlar el tráfico hacia éstas y administrar el canal de subida (*uplink*).

La representación de un nodo dentro de OPNET se muestra en la figura 4.1 (b). Aquí la MAC de la BS interactúa con el canal de bajada (*uplink*), definido con la dirección del flujo de datos de la BS hacia las SSs. Actualmente el generador de tráfico ha sido deshabilitado, por lo tanto los paquetes de datos no serán transmitidos hacia el canal de bajada, debido a que el principal objetivo de la investigación está centrado en el análisis de desempeño para el *uplink*, el cual es el único recurso crítico de la red debido a los mecanismos de reservación de acceso y a su marcada asimetría. Adicionalmente, la organización del canal de subida es parte del proceso (*Base Station Manager*).

La configuración interna de la BS lo representa la figura A.2. El nodo está conformado por los módulos: *from_network_cloud_rx*, *to_network_cloud_tx*, *rx_point*, *to_stats_collection_tx*, *HE_MAC*, *ant_rx*, *ant_tx*, *from_link_rx*, *to_link_tx*.

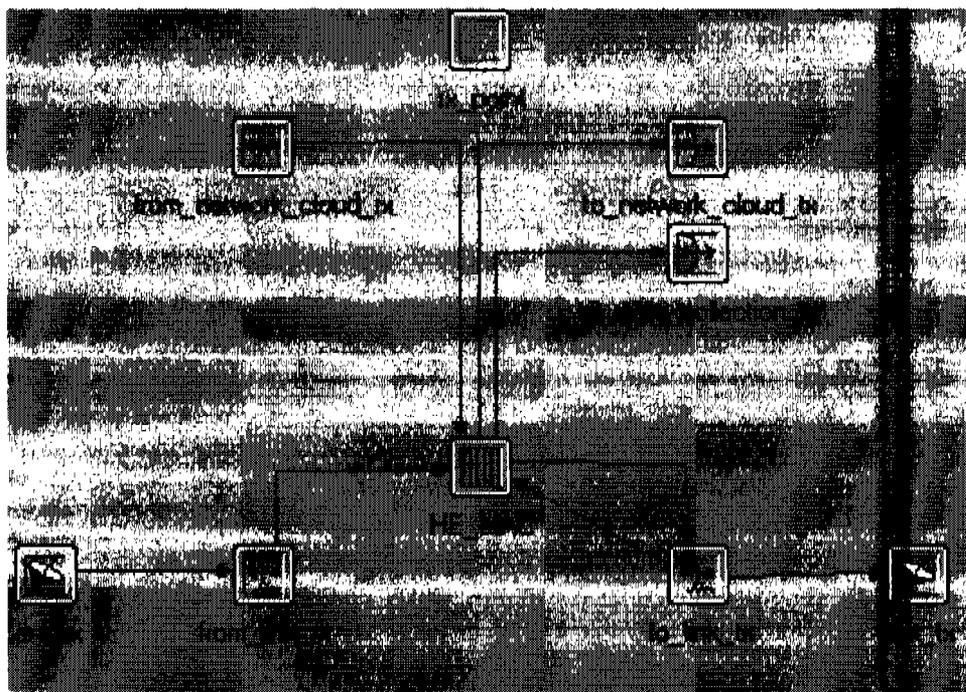


Figura A.2 Configuración interna de la estación base (*Base Station*).

Cada uno de estos módulos establece distintos tipos de conexiones con otros módulos para poder realizar las acciones de las cuales son responsables, esto es:

Rx_point

Este módulo es el encargado de la identificación de los nodos, la red a la que pertenecen y su ubicación en el espacio.

From_network_cloud_rx

Este módulo es usado para conexiones punto a punto y sirve como la puerta de entrada al nodo para los paquetes que son recibidos a través del canal, que puede ser uno sólo o varios, a una tasa de 1024 bps.

To_network_cloud_tx

Al igual que el módulo anterior se utiliza en la transmisión de paquetes para conexiones punto a punto y para este caso el módulo se convierte en la puerta de salida de los mismos. También puede contener varios canales especificándose su velocidad de transmisión, que en este caso es de 100 Mbps para la transmisión, mismo que tendrá que ser compatible con el del receptor.

To_stats_collection_tx

Este módulo es el encargado de recolectar estadísticas. Además cuenta con las mismas características que el *to_network_cloud_tx*.

From_link_rx

Se trata del canal de enlace con la BS que se encarga de recibir los paquetes provenientes de las SSs. Existen tres de ellos, debido a que en el modelo se implementó modulación adaptativa; de esta forma *from_link_rx* es el encargado de recibir las transmisiones provenientes de SSs que se encuentren trabajando con modulación QPSK.

Ant_rx

Es el dispositivo mediante el cual se recibe información proveniente de las SSs en forma de una onda electromagnética. Por comodidad se especificó una sola antena que está conectada a todos los canales de *uplink*. Es posible especificar el patrón de radiación de la misma, a través de valores angulares en coordenadas esféricas, para este caso se eligió una antena isotrópica.

To_link_tx

En este caso se trata del canal *uplink*, a través de él se envía información a las SSs con 16-QAM que es la única modulación disponible para la transmisión de la BS según el estándar IEEE 802.16. La SS cuenta con moduladores y demoduladores para poder recibir la información correctamente.

Ant_tx

Por comodidad y entendimiento del diagrama gráfico de nodo, se especificaron dos antenas, una de transmisión y otra de recepción, aunque prácticamente podría ser el mismo dispositivo el que realizara dichas funciones. En este caso, es la antena quien manda la información en forma de señal electromagnética a la SS y se encuentra por otro lado enlazada al módulo *to_link_tx* mediante un enlace de flujo de datos.

HE_MAC

Es el encargado de reunir los datos provenientes de los canales tanto de *uplink* como *downlink*. Además de estar conectado al módulo de estadísticas que recibe, a través

de este procesador, el flujo de datos de los módulos. Se comporta como una cola y se conecta con el *bwa_bs_mgr* que es el encargado de realizar, a través de varias funciones, los procesos necesarios para establecer regiones de contención en el mapa, canalizar el tipo de petición de acuerdo al tipo de servicio, etc. El diagrama de estados del proceso *bwa_bs_mgr* se muestra en figura A.3.

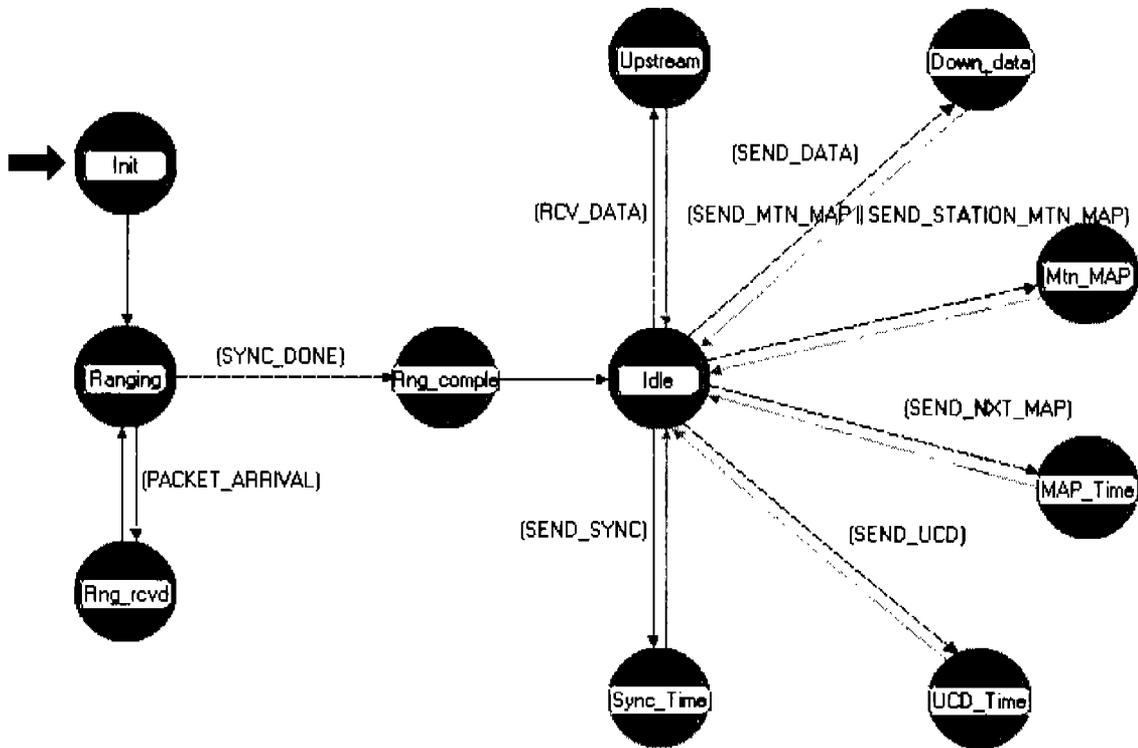


Figura A.3 Proceso *Bwa_bs_mgr*.

A.2.1 Proceso *bwa_bs_mgr*

INIT

Comienza con la ubicación de la BS en coordenadas del tipo “x” y “y”. Se encarga de aceptar las interrupciones que provienen del inicio de la simulación e invoca procesos para permitir la inicialización.

También es responsable de dar las interrupciones necesarias para poder recolectar estadísticas, crear las listas de IE’s y aquellas que pertenecen a peticiones de ancho de banda.

Finalmente se dan los valores iniciales de algunos parámetros como el retardo de propagación, contadores de estaciones, arreglos, etc., y se mandan a llamar los valores de otros como son: la duración de un *slot*, del mapa y la obtención de los parámetros de simulación.

Ranging

Terminadas las funciones del estado *Init*, este proceso tiene dos interrupciones que son activadas de acuerdo a dos parámetros. Si se ha terminado con el proceso de sincronía, la interrupción nos lleva al estado *Rng_complete*. Si se trata de la recepción de un paquete de información, el estado al que es canalizado ese paquete es *Rng_rcvd*.

Rng_Rcvd

Este estado se encarga de la recepción de los mensajes RNG_REQ de las SSs. Cuando recibe un paquete, determina su formato, la potencia con la que es recibido y la distancia entre el nodo receptor y el transmisor.

Lee el encabezado MAC, obtiene la información de QoS de la SS, el mensaje proveniente del tipo de paquete y verifica los valores de retardo de propagación máximos. Ya que obtuvo los parámetros regresa un mensaje RNG_RSP a la estación suscriptora que originó el mensaje.

Rng_complete

Es el encargado de enviar los primeros mensajes UCD y programar los siguientes. Emite el mapa inicial donde se encuentran solamente los IE's de los *slots* de contención y calcula los tiempos de transmisión para los mapas basándose en el retardo.

Se genera el encabezado MAC para la preparación del mapa y se envía un IE para que las SSs usen el espacio para petición de ancho de banda o transmisión de paquetes de información. El paquete es encapsulado y el envío del siguiente mapa se calcula de forma que sea posible que la estación más lejana lo reciba para transmitir a tiempo.

Calcula el número de IE's que se incluirán en el mapa de mantenimiento basándose en el número de SSs y el tamaño del búfer.

Idle

Este estado puede permanecer inactivo o canalizar el paso a otro estado dependiendo del tipo de información que reciba. En el caso de recibir un flujo de datos, proveniente de una SS, será enviado al estado *Upstream*, si ocurre el caso contrario, es decir, que la información se envíe por parte de la BS el estado siguiente será *Down_data*.

Las interrupciones también pueden ocurrir por la necesidad de generar de un mapa, pero éstas pueden tener distintos propósitos, si se retrata de un mapa que sólo incluya funciones de mantenimiento, la interrupción será canalizada a *Mnt_MAP*; en caso contrario la interrupción se dirige hacia el estado *MAP_Time* en donde se genera un mapa cada 2 ms de acuerdo al estándar IEEE 802.16.

Por último, si se requiere generar mensajes UCD o de sincronía, las interrupciones se programan para saltar a los estados *UCD_Time* o *Sync_Time* respectivamente.

Upstream

Recibe paquetes provenientes de la SSs, determina el nivel de potencia y la relación señal a ruido, incrementando el contador de paquetes recibidos.

Determina si el paquete entrante contiene una petición de ancho banda, si es así, se obtiene su CID, el número de *slots* solicitados y un historial de cuantas peticiones

han sido atendidas para esa SS. La petición se almacena en una cola y es atendida mediante el criterio de QoS establecido.

Analiza el encabezado del paquete para ver de qué tipo de información se trata, si está fragmentado o incluye peticiones *piggyback*. En el caso de que el paquete esté fragmentado puede ser identificado por su formato, obteniéndose así la longitud total y su carga útil.

El módulo encargado de las estadísticas recibirá una copia de cada paquete para procesar su información. En caso de que el paquete haya solicitado, en su encabezado, confirmación de envío, dicha petición también se pone en cola de espera para que una vez atendida se destruya el paquete.

Down_data

Cuando recibe un paquete, busca la dirección destino o en caso de no encontrar ese dato, la SS genera una aleatoriamente con base en su número dentro de la red. Se le asigna un apuntador al paquete, se añade su encabezado MAC y se encapsula dentro de una trama quien también contará con un apuntador de trama.

Mtn_MAP

Aquí se mandan los IEs por cada SS que se encuentre operando en la red, más un IE nulo para completar una serie que será enviada en un Mapa de mantenimiento utilizado exclusivamente para esta función.

MAP_Time

En este estado se crean mapas nuevos mapas y se establecen las condiciones en que dichos mapas están organizados. Los mapas están divididos en regiones que contemplan las regiones de contención (es esta región se establecieron las técnicas para la resolución de colisiones), peticiones y *padding*.

Existen regiones de contención, estas se encuentran alojadas al inicio del mapa, posteriormente se encuentra la región de las peticiones, que es la mayor de todas y finalmente la región de *padding* (utilizada también como región de contención).

Dependiendo de que clase de servicio se esté atendiendo, se ejecuta una rutina, en este caso es la que se refiere a *Best Effort*. Comienza indicando el inicio de un *slot* y si el mapa presente, en el que se requiere enviar la petición, está lleno, se autentica la petición y se borra de la cola una vez que ha sido atendida. Se verifica el espacio que se requiere para transmitir y se acomoda en uno o más mapas dependiendo de la longitud del mensaje emitido por la BS.

En casos especiales es posible que la longitud del mensaje genere una fragmentación ya que ésta puede ser mayor que el límite establecido. Para esos casos se fragmenta el mensaje y los paquetes resultantes de la fragmentación son atendidos revisando el espacio disponible en cada mapa.

En *MAP_Time* se implementaron los algoritmos de resolución de colisiones analizados en la presente tesis.

Fue necesario programar el modelo para conocer el número total de colisiones en un mapa y con ello establecer el número de *slots* de contención de forma adaptiva dando como resultado una mayor eficiencia al sistema.

UCD_Time

Aquí se prepara el formato de este tipo de mensaje y su contenido. Se incluye la tasa de transmisión del *uplink*, el tamaño del *slot* en *bytes*, parámetros de códigos de corrección de errores, localidades de contención, envío de información, mantenimiento inicial y mantenimiento final. Se asigna la longitud total al paquete y se crea su encabezado de control, finalmente se encapsula y envía a todas las estaciones suscriptoras de la red.

Syn_Time

Se da formato, encabezado, encapsulado y programación del envío del paquete de sincronía hacia todas las SSs de la red.

A.3 Estación Suscriptora (*Subscriber Station*)

Este nodo está implementado de tal forma que cumpla con todas las funciones de una SS común. Está conformado por módulos encargados de la generación y eliminación de paquetes de información, del envío y recepción al y desde el canal inalámbrico. El cumplimiento de las funciones propias de la subcapa MAC, así como de animaciones que permiten la obtención de resultados.

Este proceso se hace por medio de los módulos: *Animation*, *src_type_1*, *Downstream_sink*, *ant_rx*, *ant_tx*, *from_link_rx*, *from_link_tx* y *MAC*. Cada uno de ellos encargado de una función específica de la SS. La figura A.4 muestra un detalle de éstos módulos.

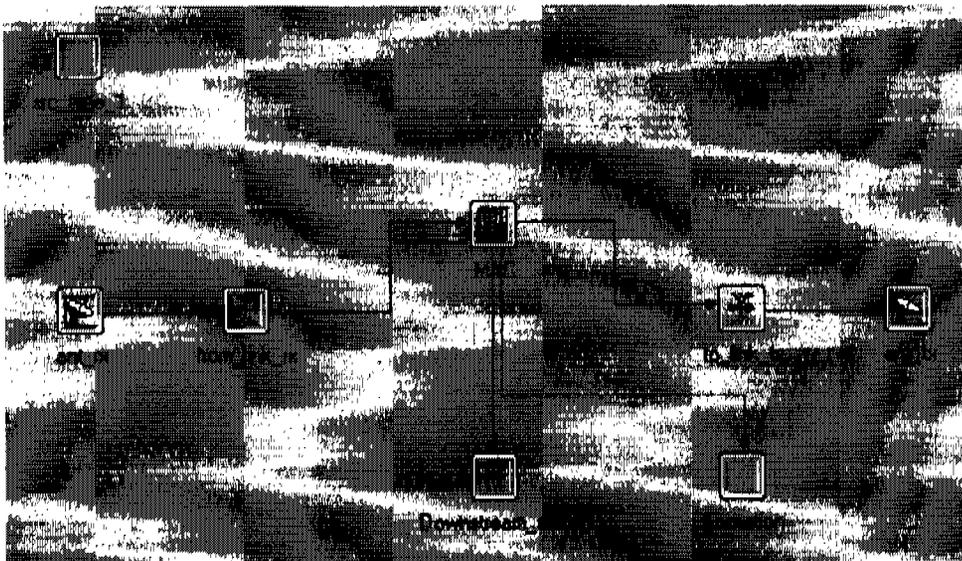


Figura A.4. Estación suscriptora (*Subscriber Station*).

Cada uno de estos módulos están conectados por medios de enlaces que simulan una conexión física o lógica para continuar con los procesos realizados por la estación. Simulando un flujo de paquetes entre los módulos o una conexión para el manejo de estadísticas independientes a las funciones de la estación suscriptora.

src_type_1

Este módulo se encarga de la generación de paquetes de información, está encargado de generar el tráfico en la red. Genera paquetes de longitud variable dependiendo del tipo de tráfico que se desee generar, puede generar tráfico de Internet, voz basado en *codecs* como: G711 y G723, además de tráfico de video.

Downstream_sink

Este módulo se encarga de la eliminación de los paquetes recibidos de la BS, ya que fueron procesados por el módulo MAC. Esto para que la cola no se llene y no se pierdan paquetes.

Además este módulo se encarga de la recolección de estadísticas para el tráfico de bajada, medición del retardo, tamaño de los paquetes y el tipo de mensajes que contenía cada uno de ellos.

Ant_rx y ant_tx

Estos módulos representan la antena de la SS, para facilitar la simulación se pone una antena para recepción (*ant_rx*) y otra para la transmisión (*ant_tx*).

From_link_rx y from_link_tx

Estos módulos establecen las características principales de los canales de *uplink* y *downlink*. El módulo *from_link_rx* recibe todos los paquetes percibidos por la antena y si sus parámetros coinciden con los del paquete recibido, los paquetes son aceptados por el simulador para así ser enviados hacia el módulo MAC, el cual procesa los paquetes. Los parámetros que puede soportar este paquete se establecen en la Figura A.5.

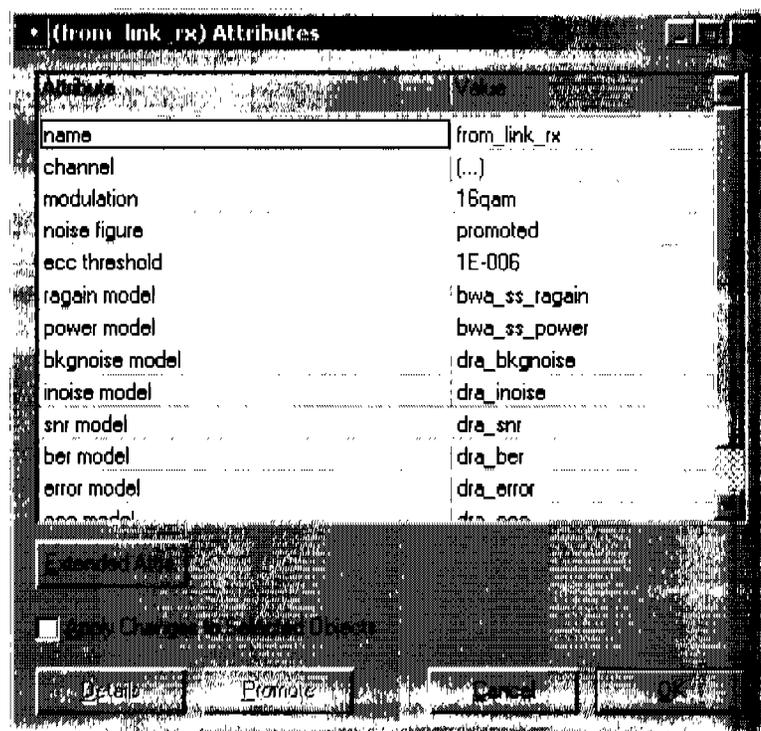


Figura A.5. Atributos del canal de bajada.

Entre los principales parámetros que se pueden modificar son: modulación, modelos de propagación a usar y las características de cada uno de canales de bajada utilizados. Para cada canal de bajada se pueden establecer parámetros, tales como: velocidad de transmisión, ancho de banda, frecuencia central, codificación y los formatos de paquetes aceptados.

En los modelos de propagación se establecen las condiciones para medir características de los paquetes recibidos, como por ejemplo: relación señal a ruido (SNR), tasa de bits erróneos (BER), ruido, potencia y ganancia.

El módulo *from_link_tx* recibe los paquetes enviados por el módulo MAC para su transmisión. Aquí se establecen las características y la forma en que se van a enviar los datos. Se establecen parámetros como: modulación, velocidad de transmisión, ancho de banda, frecuencia central y codificación utilizada. Después de pasar por este módulo, los paquetes son enviados a la antena (*ant_tx*) para su envío al medio inalámbrico con las características mencionadas anteriormente.

MAC

El módulo MAC es el más importante, éste es una cola que lleva a cabo el procesamiento de los paquetes de información recibidos desde capas superiores en la SS, así como de aquellos provenientes del canal de transmisión inalámbrico descendente. Este procesamiento incluye: agregar encabezados a los paquetes, programar oportunidades para el acceso al canal inalámbrico, envío de información durante oportunidades garantizadas por la BS, así como la resolución de colisiones durante periodos de contención.

A.3.1 Proceso *bwa_wm_mgr*

El módulo MAC se basa en el modelo de proceso *bwa_wm_mgr*, el cual realiza las funciones destinadas a la capa de convergencia MAC 802.16 tales como: calibración, recepción y procesamiento de tráfico de aplicación desde capas superiores, recepción y procesamiento de los paquetes de sincronización, UCD, MAP y tráfico en el canal *downlink* provenientes de la estación base. El proceso sigue los siguientes estados mostrados en la figura A.6.

Init

Se encarga de la obtención de los parámetros iniciales de la SS, simula el encendido de la estación. Aquí se obtienen las características que va a tener la BS y la simulación de la red en general. Se obtienen datos como: instante de Ranging, tamaño de los paquetes de solicitud de Ranging y tamaño de los paquetes de control MAC.

Se genera el CID de la BS, que es el identificador usado para mantener comunicación con la BS. Este número es generado por cada estación suscriptora tomando en cuenta un contador que inicia en 1000, y que cada estación suscriptora va incrementando en 1 al entrar a este estado.

Ya que se obtuvieron las características del canal de transmisión se hace el envío del mensaje RNG-REQ hacia la BS y se programa el tiempo de espera para que se reciba la respuesta del mensaje y así terminar con el *Ranging*.

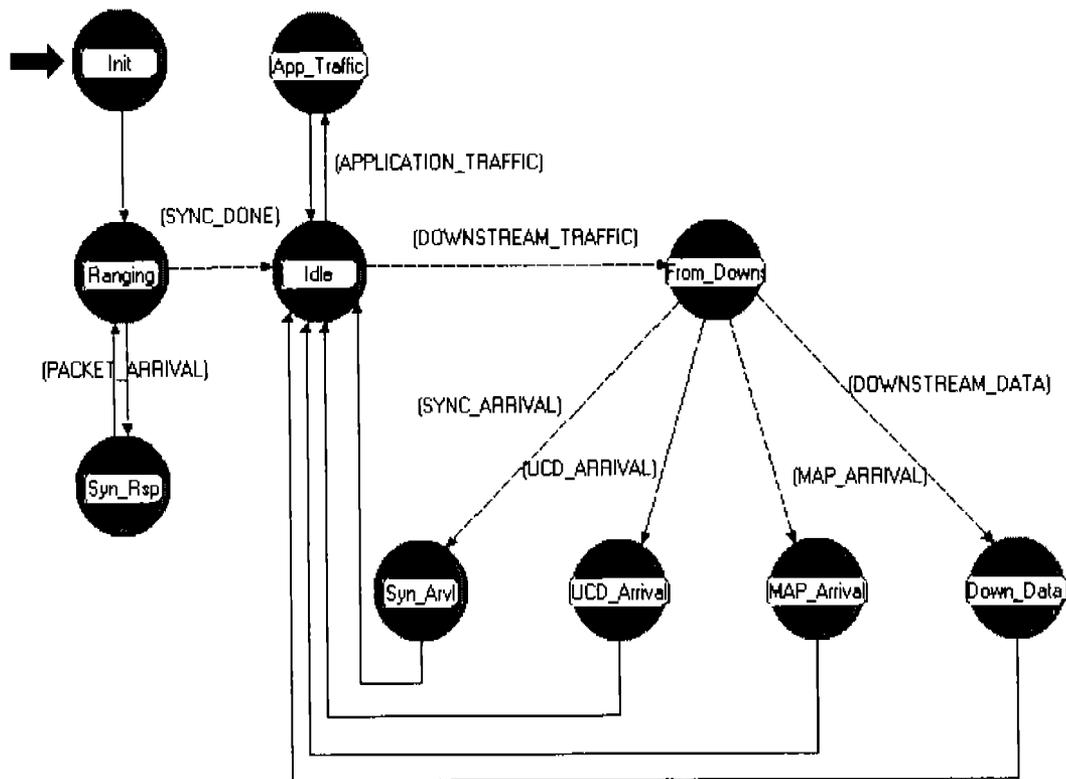


Figura A.6. Proceso bwa_wm_mgr.

Ranging

Ya que se envió el mensaje RNG-REQ, la estación espera una respuesta en éste estado. La SS se mantiene aquí hasta que recibe un paquete de la BS, si es así la estación cambia al estado *Syn_rsp*.

Syn_rsp

Si la SS entró a este estado, se debió a la recepción de un paquete. Aquí se lee el encabezado para saber si es para la estación en cuestión o es para otra. Si el paquete es para otra estación simplemente regresa al estado *Ranging* para seguir esperando, si es de tipo *broadcast* ó es dirigido a la estación, se lee el mensaje.

Si el mensaje recibido es RNG-RSP entonces se obtienen parámetros tales como el CID y dirección de la estación que son guardados para su uso en el envío de futuros mensajes. También se obtiene el retardo registrado en el paquete, para así sincronizarse con la BS ya que se tenga que enviar información en algún mapa. Ya que se leyó el mensaje se tira el paquete y la estación cambia nuevamente al estado *Syn_rsp*, sólo que ahora cambia al estado *Idle*, por lo que se entiende que el proceso de *Ranging* quedó finalizado y se realizó con éxito.

Idle

La SS se mantiene en este estado mientras no reciba o genere un paquete de información. Si la estación recibe un paquete proveniente de la BS en el canal

downlink, el proceso se dirige al estado *from_downstream*, en donde es procesado el paquete. En cambio, si se genera un paquete por la SS, el proceso cambia al estado *App_traffic* para su envío.

El proceso entra a este estado si ya terminó de hacer el *Ranging* o si ya procesó los paquetes para el canal de subida o bajada.

From_downstream

Realiza la lectura del encabezado de la trama recibida, para ver de qué tipo de paquete se trata. Si recibe un paquete de información, el proceso se va al estado *down_data_traffic*. Si se trata de un mapa, se va al estado *MAP_Arrival*. Si es un mensaje de sincronización, se dirige al estado *Syn_arvl* y si es un mensaje UCD se dirige a *UCD_Arrival*.

Down_data_traffic

Este estado procesa el arribo de paquetes de información del canal *downlink*. Si la dirección en el encabezado es correcta, la información es enviada a los niveles superiores. En el caso contrario, el paquete es destruido.

MAP_Arrival

Es el encargado del manejo de los mensajes UL-MAPs. Al recibir este mensaje, invoca al modelo de proceso *bwa_wm_child*, explicado posteriormente, el cual se encarga de su procesamiento.

UCD_Arrival

En este modelo se procesa el arribo de un mensaje UCD, proveniente de la BS, en donde se especifican las características del canal *uplink*.

Sync_Arrival

En este modelo se procesa el arribo de un mensaje de sincronización proveniente de la BS. En la simulación, la sincronización por medio de este mensaje no es necesaria por lo que el paquete recibido simplemente se destruye.

App_Traffic

Aquí se procesa el arribo de un paquete de información proveniente de niveles superiores, generado por el módulo *src_type_1*. Este modelo invoca el proceso *bwa_wm_child*, ya que este proceso se encarga del procesamiento de los paquetes para su envío.

A.3.2 Proceso Bwa_wm_child

El proceso hijo *bwa_wm_child* se encarga de procesar y redirigir los paquetes provenientes de la BS de tipo MAP y los paquetes generados por las fuentes de información de aplicación. Cada vez que el módulo MAC de una SS recibe alguno de estos paquetes, invoca este proceso desde los estados *App_Traffic* o *MAP_Arrival*. El proceso contempla una cola FIFO

(First In, First Out) mediante la cual se atienden los paquetes conforme se reciben. Se incluyen funciones para la resolución de colisiones, en específico, el algoritmo *Exponential Backoff*. La Figura A.7 muestra los estados que sigue el proceso.

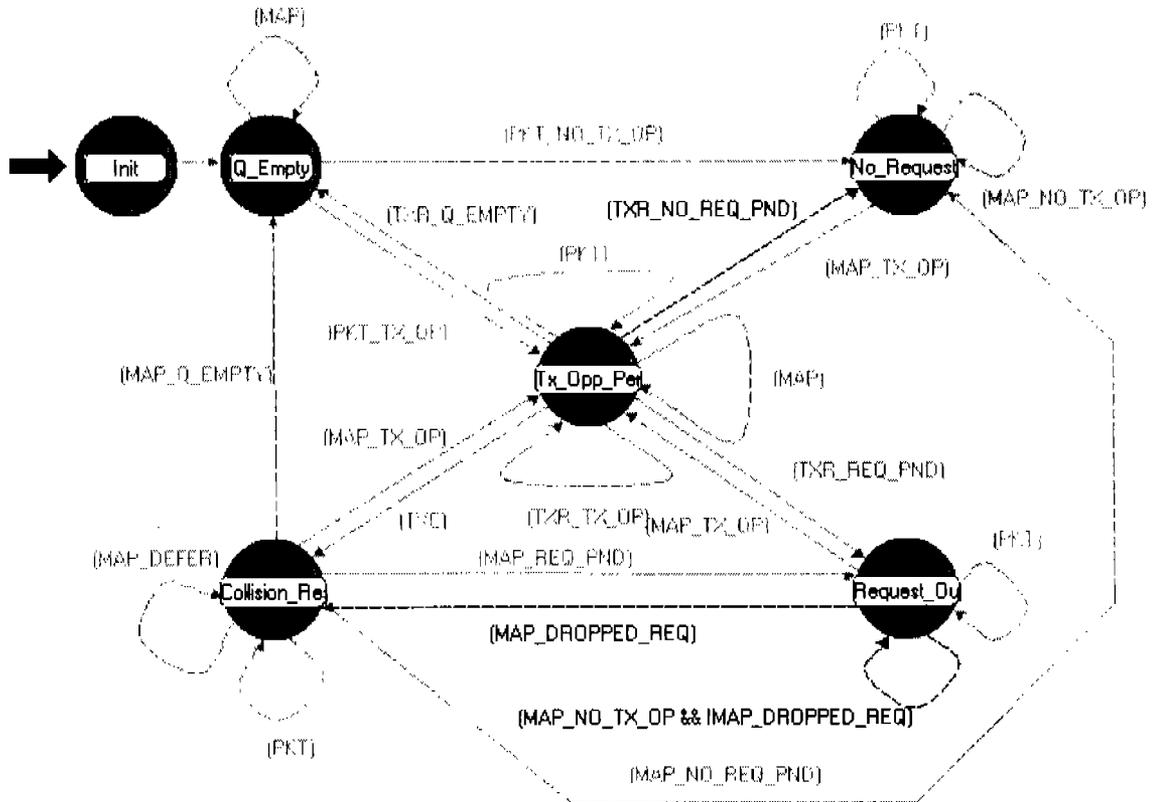


Figura A.7. Proceso *bwa_wm_child*.

Init

Es el encargado de obtener los parámetros iniciales de la SS para la simulación. Aquí se establecen las variables iniciales tales como: servicio establecido para la estación, velocidad de transmisión, tiempo en segundos de cada slot, CID y tamaño de los encabezados a usarse. Las variables establecidas en el estado son útiles para el desarrollo de la simulación.

Q_Empty

El proceso se mantiene en este estado mientras no tiene un paquete esperando en la cola. Permanece aquí hasta que recibe un paquete, si el paquete es un mapa de la BS, simplemente lo lee, guarda su información y desecha el paquete, el proceso se mantiene en este estado.

Si se recibe un paquete de información proveniente de las capas superiores, cambia de estado. Si la SS esta generando tráfico de voz y tiene habilitado UGS como calidad de servicio entonces se dirige al estado *Tx_Opp_Pending*.

Si la estación no tiene habilitado UGS o genera un tipo de tráfico distinto, como por ejemplo tráfico de Internet, entonces se dirige al estado *No_Request_Oustanding*.

No_Request_Outstanding

Aquí la SS busca una oportunidad de transmisión en la región de contención en el mapa, para enviar la petición de ancho de banda requerida. Realiza el cómputo del tamaño de la carga y sus encabezados. Ya que tiene el tamaño del paquete total en *bytes* hace la conversión a *slots*, tomando en cuenta que cada *slot* es de 16 *bytes*, para hacer la solicitud de ancho de banda en *slots* a la BS.

En el modelo implementado, existen tres regiones de contención. La SS primero busca esta región en el mapa, si todavía está a tiempo de enviar una petición en esta región, realiza el algoritmo *Exponential Backoff* para encontrar el *slot* en el que transmitirá su petición.

Si la región de contención ya pasó o el algoritmo produjo un número fuera de esta región, entonces la estación busca en la región de *padding* otra oportunidad de transmisión. Esta región fue habilitada como contención para todas las modulaciones, por lo que todas las SSs podrán enviar solicitudes en esta región. De igual forma corre el algoritmo *Exponential Backoff* para ver si puede transmitir su petición.

Si se encontró una oportunidad de transmisión en el mapa entonces el proceso se dirige al estado *Tx_Opp_Pending*, si no encontró una oportunidad de transmisión se mantiene en este estado en la espera de nuevos mapas para intentar la transmisión.

Tx_Opp_Pending

En este estado la SS realiza el envío de los paquetes de información UGS y solicitudes de ancho de banda para datos de otro tipo. Aquí la SS pone los encabezados a los paquetes a enviarse y espera el tiempo indicado para su transmisión.

El proceso se mantiene en este estado hasta que los paquetes, que ya tienen asignado un *slot* para su transmisión, son enviados. Aquí se realizan procesos como concatenación, en los que varios paquetes son enviados a la vez con un único encabezado. También se realizan funciones de *piggyback*. En esta función, si se requieren enviar más paquetes de información a la BS, se hace la solicitud de ancho de banda en el último paquete de información enviado.

Cada vez que se envía un paquete, la SS inicia o actualiza un contador en el cual se establece el número de intentos que se ha realizado la SS para la transmisión del paquete.

Al terminar de enviar los paquetes, el proceso se dirige al estado *Collision_Resolution*. Si en la cola todavía existen paquetes, la estación ve su estado, si son paquetes que ya enviaron su solicitud de ancho de banda pero todavía no es atendida, el proceso entra al estado *Request_Outstanding*. Si en el proceso la SS generó un paquete de información, en las capas superiores, el proceso se va al estado *No_Request_Outstanding* para la generación del mensaje de solicitud de ancho de banda.

Collision_Resolution

Mantiene el seguimiento de los paquetes enviados hacia la BS. Es el estado encargado de la detección y resolución de colisiones en la SS, mediante la lectura de los mapas y mensajes enviados por la BS.

Si el paquete que se envió fue un mensaje de solicitud de ancho de banda, entonces la SS leerá todos los mapas en busca de la asignación dada por la BS. Si la red está saturada y la BS recibe una solicitud de ancho de banda, que no puede ser asignada, manda un mensaje de confirmación de recepción de la solicitud hacia la SS.

Cuando recibe la confirmación de recepción o ya se concedió el ancho de banda en algún mapa, entonces los paquetes enviados son desechados, ya que se considera que la transmisión fue exitosa. Si se hizo la asignación de ancho de banda y no existen paquetes en la cola, entonces el proceso se dirige al estado *Q_Empty*. Si se recibió un mensaje de confirmación indicando la recepción de la solicitud pero no la asignación de ancho de banda, entonces el proceso se dirige al estado *No_Request_Outstanding*.

Si no se recibió la asignación de ancho de banda y no existió un mensaje de confirmación de recepción enviado por la BS, la SS repetirá la búsqueda en cada mapa recibido hasta que encuentre la asignación o el mensaje, si esto no ocurre y se vence el tiempo de espera para su recepción, entonces la estación suscriptora determina que ha ocurrido una colisión, por lo que buscará una nueva oportunidad de transmisión del mensaje, mediante el uso del algoritmo *Exponential Backoff*. Si esto ocurre entonces el proceso se dirige al estado *Tx_Opp_Pending* para el nuevo envío del paquete.

Request_Outstanding

Al entrar a este estado, una solicitud de ancho de banda ha sido previamente hecha, pero aún no ha sido atendida. En este estado, la petición se mantiene en una cola y se espera una confirmación dentro de un mapa por parte de la estación base para su transmisión.

Cada mapa enviado por parte de la BS es analizado para obtener información de las peticiones, que ya están esperando en la cola. Si algún dato corresponde con el CID de la SS que envió el mensaje, se analiza la información almacenada y demás parámetros para calcular el tiempo de propagación entre la BS y la SS, y así programar el tiempo justo de transmisión.

En caso de que no se encontrara la información necesaria en alguno de los mapas para procesar la información y el tiempo máximo para la recepción de confirmación fuera excedido, por aquel que se registró en la petición de ancho de banda, se asume una colisión y se inicia el algoritmo *Exponential Backoff* para reenviar la solicitud.

APÉNDICE B

Parámetros de capa MAC y PHY

En esta sección se presentan los archivos que contienen todos los parámetros de capas PHY y MAC del estándar 802.16, para el modelo de simulación desarrollado.

B.1 Capa PHY

Revision History

Effective downstream data rate in bits/sec for 64 QAM
 # This includes MPEG TC overhead and FEC overhead
 #
 # All attributes below must have identical values

#"top.bwa.*.Downstream Data Rate": 44800000
 #downstream_data_rate: 44800000
 #"top.lb_1.data rate": 44800000

#"top.bwa.*.Downstream Data Rate": 9600000
 #downstream_data_rate: 9600000
 #"top.lb_1.data rate": 9600000

#"top.bwa.*.Downstream Data Rate": 2816000
 #downstream_data_rate: 2816000
 #"top.lb_1.data rate": 2816000

#Bw 7 MHz + 16-QAM
 "top.bwa.*.Downstream Data Rate": 22400000
 downstream_data_rate: 22400000
 "top.lb_1.data rate": 22400000

Upstream data rate in bits/sec
 # All attributes below must have identical values
 # was 2560000

#"top.bwa.*.Upstream Data Rate": 44800000
 #upstream_data_rate: 44800000
 #"top.lb_2.data rate": 44800000

#Bw 6 MHz + QPSK
 "top.bwa.*.Upstream Data Rate": 9600000
 upstream_data_rate: 9600000
 "top.lb_2.data rate": 9600000

```

#"top.bwa*.Upstream Data Rate":      2816000
#upstream_data_rate:                  2816000
#"top.lb_2.data rate":                2816000

# Downstream interleave delay in seconds
downstream_interleave_delay:          .0005

# Maximum frame size that can be transmitted downstream in bytes
max_frame_size:                        1518

# Size of an upstream minislot in bytes
# minislot_size:                       16
minislot_size:                          16

# These Values were taken from the IE802.16 standard
# Physical layer burst profiles for a request transmission
# Preamble and guardband in bits, FEC parity and codeword in bytes
Req_preamble:                           32
Req_FEC_parity:                           0
Req_FEC_codeword:                         20
Req_Guardband:                            48
Req_last_cdwrд_shorten:                   1

# Physical layer burst profiles for a request/data transmission
# Preamble and guardband in bits, FEC parity and codeword in bytes
Req_data_preamble:                        32
Req_data_FEC_parity:                       6
Req_data_FEC_codeword:                     36
Req_data_Guardband:                        48
Req_data_last_cdwrд_shorten:               1

# Physical layer burst profiles for an initial maintenance transmission
# Preamble and guardband in bits, FEC parity and codeword in bytes
Initial_mtn_preamble:                      32
Initial_mtn_FEC_parity:                    10
Initial_mtn_FEC_codeword:                  20
Initial_mtn_Guardband:                     48
Initial_mtn_last_cdwrд_shorten:            1

# Physical layer burst profiles for a station maintenance transmission
# Preamble and guardband in bits, FEC parity and codeword in bytes
Station_mtn_preamble:                      32
Station_mtn_FEC_parity:                     6
Station_mtn_FEC_codeword:                   36
Station_mtn_Guardband:                     48
Station_mtn_last_cdwrд_shorten:            1

# Physical layer burst profiles for a short data grant region
# Preamble and guardband in bits, FEC parity and codeword in bytes
Short_data_preamble:                       32

```

```

Short_data_FEC_parity:      6
Short_data_FEC_codeword:   36
Short_data_Guardband:      48
Short_data_last_cdwrld_shorten:  1

# Physical layer burst profiles for a long data grant region
# Preamble and guardband in bits, FEC parity and codeword in bytes
Long_data_preamble:        32
Long_data_FEC_parity:      10
Long_data_FEC_codeword:    255
Long_data_Guardband:       48
Long_data_last_cdwrld_shorten:  1

# Set the BS and SS Gains in [dBi]
SS_Antenna_Gain: 18
#SS_Antenna_Gain: 32
BS_Antenna_Gain: 15

#***** Parameters for path loss*****#
#Do: 100
#ht: 6.5
#hr: 50
Do: 100
ht: 1.5
hr: 50

#Terrain Field Category, 0 for CatA, 1 for CatB and 2 for CatC
Category: 2

#####For Cat A
#ABC_a: 4.6
#ABC_b: 0.0075
#ABC_c: 12.6
# Shadow Fading, values in dB.
#Sf_Media: 10.6
#Sf_Dstd: 2.3

#####For Cat B
#ABC_a: 4.0
#ABC_b: 0.0065
#ABC_c: 17.1
# Shadow Fading, values in dB.
#Sf_Media: 9.6
#Sf_Dstd: 3

#####For Cat C
ABC_a: 3.6
ABC_b: 0.005

```

```

ABC_c: 20
# Shadow Fading, values in dB.
Sf_Media: 8.2
Sf_Dstd: 1.6

```

B.2 Capa MAC

```

# Start and maximum backoff windows in the contention region
# as a power of 2

#####
### BAY Uses 3-10:
#####
# The following parameters were moves to the simulation screen
#data_backoff_start: 3
#data_backoff_end: 10

# 2 slots minimum fragment
BS_Min_Frag_Size: 64
# BE_Min_Frag_Size: 128
# Boolean that turns concatenation OFF/ON (0/1)
BS_Concatenation: 0
"top.bwa.*.MAC.bwa_wm_child.permit_concatenation": 0

# Maximum number of IE's a MAP can contain
#(Maximum value according to spec -- 240)
BS_MAX_MAP_IES: 300

# Maximum size of the MAP in minislots
# [From Chapter 6]
BS_MAX_MAP_SLOTS: 4096
#BS_MAX_MAP_SLOTS: 2048
#BS_MAX_MAP_SLOTS: 316

# Threshold that differentiates between a short and a long data grant in bytes
BS_Short_Grant_Limit: 245

# BE Administrative request limit in bytes
BS_Request_Admin_Limit: 3036

# Interval between generation of Maintenance MAPs downstream in seconds
# (Maximum value according to Appendix B -- 2 sec)
BS_MAINT_INTERVAL: 1.5

# MAC header base size in bytes
BS_MAC_Header_Base_sz: 6

```

```

# MAC extended header acknowledgement field size in bytes
BS_ACK_EH_sz:          2

# MAC extended header request field size in bytes
# in DOCSIS
#BS_Request_EH_sz:    3
# in 802.16 PBR = 2 bytes
BS_Request_EH_sz:     2

BS_REQ_DATA_Multicast_SID:    -2001
BS_MAC_Frag_Header_sz:        2

# Ranging request size
"Ranging Req Size":          0

# MAC Management packet size
"MAC MNG Pkt Size":         0

# Maintenance area in minislots
BS_MTN_Area:                4
#####
# Default Parameters          #
#####

verbose_sim:                TRUE
stats_start_time:           1

# Simulation time when the generators will begin generating traffic
gen_start_time_type1:       1.0
gen_start_time_type2:       1.0
gen_start_time_type3:       1.0
gen_start_time_mpeg:        1.0
gen_start_time_www:         1.0
gen_start_time_on_off:      1.0
gen_start_time_down:        1.0
gen_start_time_typev:       1.0

# number of priorities in simulation (maximum)
num_priorities: 8

# Number of list that are used for the SSSs with CIR requirements
CIR_Lists: 2

# These flags should be set to either 1 or 0. Turn on the flag by
# setting the value to 1 and turn off the flag by setting the value
# to 0. By turning on a flag, individual statistics will be collected
# for that particular type of traffic. Statistics will always be
# gathered for the conglomerate of all sources, and these flags should
# only be set if you are interested in statistics relating to an
# individual traffic type. If you are running long simulations and
# running out of disk space, you can turn these flags off and minimize

```

```

# the statistics that you are collecting.
collect_type1_stats: 1
collect_type2_stats: 1
collect_type3_stats: 1
collect_type4_stats: 1
collect_type5_stats: 1
collect_type6_stats: 1
collect_typeev_stats: 1

collect_dtype1_stats: 0
collect_dtype2_stats: 1
collect_dtype3_stats: 0
collect_dtype4_stats: 0
collect_dtype5_stats: 0
collect_dtype6_stats: 0
collect_dtypeev_stats: 0

"top.bwa.*.Call Ring Timer": 10.0
"top.bwa.*.Conversation Plateau Length": 30.0
#"top.bwa.*.Encoder Modulation Method": G.711
"top.bwa.*.Encoder Statistics Quantum": 2.5
"top.bwa.*.Packer Maximum Latency": 0.020
"top.bwa.*.Packer Maximum Size": 160
"top.bwa.*.Packetizer Destination Addr": 10001
"top.bwa.*.Packetizer Message Priority": 0.0
"top.bwa.*.Talk Ratio": two-way
"top.bwa.*.Voice Call Distribution": "exponential"
"top.bwa.*.Voice Stagger Start Period": 15.0
"top.bwa.*.Voice Mean Call Length": infinity
"top.bwa.*.Voice Call Rate": 10.0
"top.bwa.Stats Collection.voice_sink.sample_quanta": 2.5

"top.bwa.*.src_type_1.pksize_pdf": "docsis_srctype1_medium"
"top.bwa.*.Type 1 Source Arrival Rate": 17.21
# 17.21 frames/sec == 25 Kb/s with csf_phase4 distribution

# SS processing time in seconds
# The amount of time required by a SS station to process a MAP and
# start transmitting in the appropriate slots.
SS_processing_time: .00001

top.bwa.*.src_type_1.msg_priority: 0.0

#####
# BE && CIR Parameters #
#####

#Fixed Frame Mode (BE MAP interval)

#UNAM Jun 1, 2004: The MAP Frame in the IEEE 802.16 is fixed, we use MAP
Duration = 1 milisecond,

```

```
#put BS_Fragmentation = 1 to include fragmentation
#BS_Fixed_Frame_On:      0
BS_Fixed_Frame_On:      1
BS_Fixed_Frame_Duration: 0.002
#BS_Fragmentation: 0
BS_Fragmentation: 1
```

```
# Maximum grant time of the MAP in seconds
#BS_MAX_MAP_GRANT_TIME:      0.003
#BS_MAX_MAP_GRANT_TIME:      0.007
#BS_MAX_MAP_GRANT_TIME:      1.020
#UNAM Jun 1, 2004: Since the MAP frame in the IEEE 802.16 protocol is fixed, so
the grant time.
BS_MAX_MAP_GRANT_TIME:      0.002
```

```
# Number of contention based request slots allocated by the BE
# Parametro desabilitado, se define en la ventana de simulacion
```

```
#BS_Contention_Slots:      10
#BS_Contention_Slots:      16
#BS_Contention_Slots:      32
#BS_Contention_Slots:      100
#BS_Contention_Slots:      20
```

```
"top.bwa.*.Voice Request Policy Bitmask": 0
"top.bwa.*.Data Request Policy Bitmask": 0
"top.bwa.*.Voice Type of Service": 1
#"top.bwa.*.Data Type of Service": 1
"top.bwa.*.Data Type of Service": 4
"top.bwa.*.Voice Fragmentation Mode": 0
"top.bwa.*.Data Fragmentation Mode": 0
```

```
#####
# MAC Parameters #
#####
```

```
# UMAM Jun 2, 2004
# These variables define the maximum number of
# modems per priority and the first SID
# number for pri 0 modems
# Priorities are not implemented yet.
```

```
Priority_Spacing:          500
SID_start:                 1000
```

```
# UMAM Jun 2, 2004
# These variables define whether the individual statistics per
# priority are to be collected (0=Do not collect)
```

```

Col_pri0_stats:          1
Col_pri1_stats:          1
Col_pri2_stats:          1
Col_pri3_stats:          1
Col_pri4_stats:          1
Col_pri5_stats:          1
Col_pri6_stats:          1
Col_pri7_stats:          1

# UMAM Jun 2, 2004
# These variables define the number of
# modems per priority
cms_pri0:                0
cms_pri1:                0
cms_pri2:                0
cms_pri3:                0
cms_pri4:                0
cms_pri5:                0
cms_pri6:                0
cms_pri7:                0

"top.bwa.*.Max Burst Size":      1
"top.bwa.*.Max Packet Size":    146
"top.bwa.*.Voice Priority":      0
"top.bwa.*.Data Priority":      1

# End of the ranging period in seconds
# This is the initial time given to SS stations to figure out
# how far they are from the BE
End_of_Ranging:            1

# Interval between generation of UCD's downstream in seconds
# This is the time interval between subsequent UCD messages
# sent from the BE to the SS stations
UCD_interval:              5

# Interval between generation of SYNC messages downstream in seconds
# (Maximum value according to Appendix B -- 0.2 sec)
SYNC_interval:             0.2

#Set parameters for packet distribution
SS_use_packet_distribution: 1
  64b_pk_dist: 0.6
  128b_pk_dist: 0.66
  256b_pk_dist: 0.7
  512b_pk_dist: 0.72
  1024b_pk_dist: 0.97
  1518b_pk_dist: 1.0

#Indicate whether SS will use piggyback request
SS_PIGGYBACK_Enabled: 1

```

#Indicate if a new adaptive CRA will be used
SS_use_adaptive_CRA: 1

#Indicate the type of QoS
SS_Use_UGS_for_VoIP_traffic: 1
BS_Include_Round_Robing_For_UGS_Service: 1
BS_use_EDF_Scheduling: 1
SS_include_packet_loss: 0
Support_multicast_collision_resolution: 2
Collision_Free: 1
CDMA_SCHEME:1

APÉNDICE C

Artículo para su publicación

Performance Evaluation of Adaptive Contention Slot Allocators for IEEE 802.16 Based Systems

KEYWORD: Congestion/capacity management, Wimax Performance,

Corresponding Author

Victor Rangel Licea, (PhD)

Head of the Broadband Wireless Access Network Lab.
Department of Telecommunications,
National Autonomous University of Mexico

Address:

Edificio Valdes Vallejo 3er Piso, CU, Mexico D.F, CP 04510, MEXICO

Telephone:

+52 55 5622-3142

Fax:

+52 55 5622-3142

Email:

victor@fi-b.unam.mx

Abstract submitted to

2008 IEEE International Symposium
Broadband Multimedia Systems
Broadcasting

**Performance Evaluation of Adaptive Contention Slot Allocators for
IEEE 802.16 Based Systems**

Victor Rangel, Miguel Perez, Javier Gomez, Mercedes Lopez, Raul Aquino
Department of Telecommunications, National Autonomous University of Mexico

Abstract

A reservation based Medium Access Control (MAC) protocol has been adopted by the IEEE 802.16 standard as the basic protocol for data communication within the upstream channel. This standard adopts the binary exponential backoff algorithm with adjustable window size to resolve collisions of bandwidth request packets. The IEEE 802.16 MAC protocol establishes that bandwidth assigned for the uplink channel is conformed mainly by two regions: contention and reservation. The former is used by subscriber stations to transmit bandwidth requests. The later is used to transmit data information from subscriber stations in reserved slots. The efficiency of the MAC protocol highly depends on the bandwidth assigned to the contention access. A high number of contention slots (CSs) assigned to this region reduces the bandwidth for data transmission in the reservation region. On the other hand, a small number of contention slots give rise to an increased number of collisions during high traffic loads, resulting in degradation on system performance.

In this paper, two adaptive Contention Slots Allocators "CSA" are introduced for the IEEE 802.16 MAC protocol and compared with previous CSA schemes. The new techniques dynamically fit the number of contention slots needed to solve collisions according to the current traffic load, considerably improving the overall system performance. The CSA introduced in this paper indicate that the mean access delay could be reduced up to 50% compared with the simple CSA, which is adopted by the IEEE 802.16 standard. The results are also validated with the maximum theoretical system throughput and we achieved a deviation of less than 1.5% between theoretical and simulation.

1. Introduction

Broadband Wireless Access (BWA) Networks, based on the standard IEEE 802.16 has become the best way to meet residential and small business demand for high speed Internet, multimedia and voice services. As an emerging technology for broadband access, it provides the following advantages over its wired competitors: rapid deployment, easy to implement, high scalability, lower maintenance, low upgrade cost and higher data rates.

The IEEE 802.16 MAC protocol specification does not define any mechanism for bandwidth allocation, and this task has been left open to implementations and vendor differentiation. Therefore in this paper we propose and study several techniques in order to increase the system performance by adjusting the bandwidth to be allocated for contention access in each signaling frame, called "Uplink-MAP".

II. IEEE 802.16 MAC Protocol Overview

The IEEE 802.16 MAC protocol as defined in [1] and [2], regulates uplink (UL) channel access using Time Division Multiple Access (TDMA). Upon entering the IEEE 802.16 network, each Subscriber Station (SS) has to go throughout the Initialization and Registration setup process. Once this process is complete, a SS can create one or more connections over which its data is transmitted to the BS, using a defined Quality of Service (QoS) level. The standard support four different QoS agreements: Unsolicited Grant Service (UGS), Real-Time Polling Service (rtPS), Non Real-Time Polling Service (nrtPS), Best Effort (BE). The later is where collisions happen. In the other QoS levels, subscriber stations can transmit their data or bandwidth requests by using the grants or the polled request opportunities (Unicast slots), respectively, provided by the Base Station (BS).

Within the BE policy, upon receiving bandwidth request form SSs, the BS gathers these requests and determines the number of time slots (grant size) that each SS will be allowed to transmit in the next UL-Frame. This information is broadcasted in the DL channel by the BS using the UL-MAP message at the beginning of each DL-Frame.

III. Adaptive Contention Slot Allocators

As introduced in [3], [4] and [5], the authors have pointed out that system performance of a multi-access reservation protocol depend on the overall framing structure and the capacity assigned to the reservation channel than the details of the adopted collision resolution algorithm.

In this paper two adaptive CSA mechanisms are introduced, which dynamically adjust the number of contention slots (CSs) per UL-MAP, according to the current traffic load, mean packet size, mean requested slots and possible collisions. These mechanisms improve the maximum system performance for the Exponential Backoff Algorithm by assigning more CSs when they are needed, (and not when they are available) and by reducing the average number of CSs needed by requests to a value close to the optimum " $e = 2.718$ " as suggested by [6]. We have referred to the new mechanisms are *Collision Free-CSA* and *Multicast-CSA*. In order to demonstrate its superior performance, we also describe the main functionality of previous CSA, such as *Simple-CSA* adopted by the IEEE 802.16, *Forced-CSA* and *Variable-CSA*, as described in [3] for the European Cable Modem DVB/DAVIC protocol and the IEEE802.14-CSA as described in [5] for the IEEE 802.14 protocol.

Simple-CSA: This mechanism allocates all slots that are not being used for data as CSs. At low traffic loads, many CSs are allocated but are not required. The surplus of CSs significantly decreases the risk of collision (of bandwidth requests) to a very low level.

Forced-CSA: This mechanism is based on the dynamics of the splitting tree algorithm, as reported in [7]. When a collision occurs, the splitting tree algorithm automatically allocates three CSs in the next signaling frame, which are then used by all stations either involved in previous collisions or not. The Forced-CSA allocates a flexible number of CSs in the next UL-MAP. We refer to these additional slots as Forced contention slots (FSs). With this new functionality, stations competing for contention access have a better probability of transmitting successfully their bandwidth requests, since more CSs are allocated when they are needed.

Variable-CSA: This mechanism is also described in detail in [7] and uses a variable slot regime in which the ratio of CSs to reservations slots is varied from signaling frame to signaling frame, based on the current traffic load, mean packet size and mean grant sizes.

IEEE 802.14-CS: This mechanism is similar to variable-CSA, but adapted to the IEEE 802.14 protocol, as defined in [5].

Multicast-CSA: This new mechanism allocates a multicast contention region for each collision. Thus, the users that provoke a collision will have a short reserved contention area in the next UL-MAP in order to retransmit their bandwidth request. New requests are not allowed to be transmitted in these multicast contention opportunities.

Collision Free-CSA: This is also a new mechanism. This scheme assigns a unicast transmission opportunity to subscriber stations in a Round Robin discipline, as long as space in the current UL-MAP is available. A subscriber station uses its assigned transmission opportunity to send its bandwidth request. SSs are forced to wait for its unicast transmission opportunity avoiding contention access.

All mechanisms use the piggyback function of the IEEE 802.16 MAC protocol.

IV Performance Evaluation

A detailed simulation model of the IEEE 802.16 MAC protocol was implemented using OPNET MODELER v.14. For the description of this model the readers are referred to [8]. For the system performance, we assume a 6 MHz uplink channel, a roll off factor of 0.25, QPSK modulation, UL bit rate (UL_{bitrate}) of 9.6 Mbps. We also assume that SSs transmit Internet traffic at 32 kbps, where the mean packet size is of 23 slots and 26.6 slots for the MAC and PHY layer respectively, as described in [8].

In order to validate the results we estimate the maximum system throughput using equation 1.

$$S_{\max} = UL_{\text{bitrate}} \frac{\overline{Pk}_{MAC}}{\overline{Pk}_{PHY} + CS_{REQ}} [bps] \quad (1)$$

where $\overline{Pk}_{MAC} = 23$ slots and $\overline{Pk}_{PHY} = 26.6$ slots are the mean packet size of Internet traffic at the MAC and PHY layer, respectively. CS_{REQ} is the mean average of contention slots per request. When the network is congested (e.g. more than 260 stations) $CS_{REQ} \rightarrow 0$. This is because when the network is congested most subscriber stations avoid contention access by piggybacking requests in the reservation area. Thus the maxim theoretical system throughput $S_{\max} \rightarrow 9600000(23/26.6) = 8.3\text{Mbps}$.

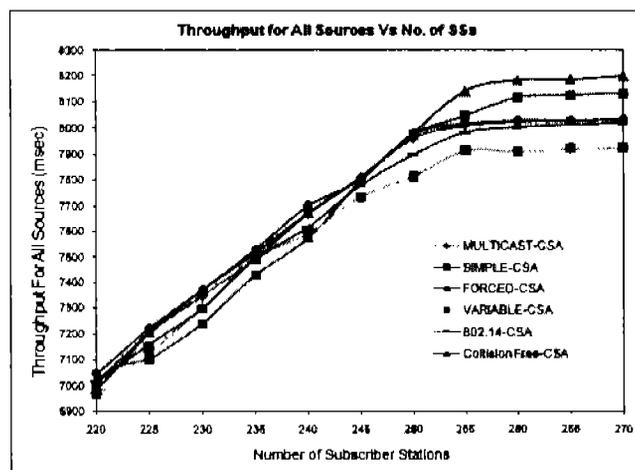
In Figure 1a, when the network is congested, we can observe that the Collision Free-SCA, is the only one that provides a similar system throughput of 8.2Mbps, achieving a deviation less than 1.5% between theoretical and simulations results. The simple-CSA adopted by the IEEE 802.16 protocol, achieves a lower system throughput of 8.1 Mbps, which results in a deviation of approximately 2.5% from theoretical results.

As we can see in Figure 1a, the throughput performance for the Simple-CSA is slightly better than the proposed Multicast-CSA, which achieves 8.0 Mbps (3.5% deviation from theoretical results). This reduction in system throughput is to be expected, since this scheme allocates short multicast contention regions for each collision. By allocating more contention slots when they are needed to resolve current collision, SSS have a better probability of retransmitting bandwidth requests successfully, which results in a reduced access delay. Thus, in figure 1b we can see that the mean access delay is considerably improved compared with Simple-CSA, this reduction is up to 50%.

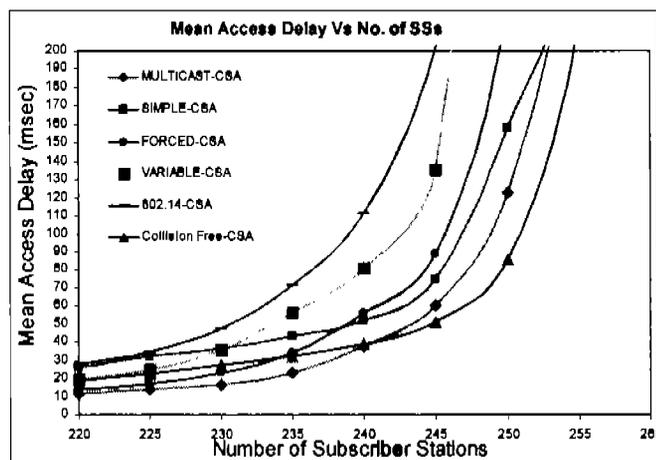
The other mechanisms Variable, Forced and the IEEE 802.14, provide a lower system performance compared with the two schemes proposed in this paper.

V. Conclusions

In this paper a performance evaluation and analysis of two new contention slot allocators have been presented for the IEEE 802.16 protocol. Simulations results revealed that the overall system performance could be significantly improved by adopting the proposed CSA (Collision Free and Multicast). Simulation system throughput agreed well with results from theoretical analysis with a deviation in the results not exceeding 1.5% for the Collision Free CSA.



(a) Throughput for all sources



(b) Mean Access Delay

Figure 1. Performance comparison of the proposed CSA with previous CSA.

References

- [1] IEEE 802.16-2001, "IEEE Standard for Local and Metropolitan Area Networks - Part 16: Air Interface for Fixed Broadband Wireless Access Systems," April 2002.
- [2] IEEE 802.16-2004, "IEEE Standard for Local and Metropolitan Area Networks - Part 16: Air Interface for Fixed Broadband Wireless Access Systems," October 2004.
- [3] V. Rangel and R. M. Edwards, "Performance Evaluation of Adaptive Contention Slot Allocators for CATV Networks based on the European Cable Communications Protocol DVB/ DAVIC", Journal of the SCTE, vol. 24, Sep. 2002, pp 24-30.
- [4] B. Doshi, S. Dravida, P. Magill, C. Siller, and K. Sriram, "A Broadband Multiple Access Protocol for STM, ATM, and Variable Length Data Services on Hybrid Fiber-Coax Networks", Bell Labs Technical J. (BLTJ), Vol. 1, No. 1, pp. 36-65, Summer 1996.
- [5] N. Golmie, Y. Saintillan, and D. Su, "A Review of Contention Resolution Algorithms for IEEE 802.14 Networks",
- [6] D. Bertsekas and R. Gallager., "Data Networks", Englewood Cliffs, NJ, Prentice-Hall, Second edition. pp 271-362, 1992.
- [7] V. Rangel, R. Edwards, and K. Schunke, "Contention Resolution Algorithms for CATV Networks Based on the DVB/DAVIC Cable Modem Protocol Specification (ETS EN 200 800), Journal of Cable Telecommunication Engineering (CTE), vol. 24, pp. 23-30, December 2002.
- [8] Víctor Rangel, Javier Gómez y Javier Ortiz, "Performance Analysis of QoS Scheduling in Broadband IEEE 802.16 Based Networks", OPNETWORK Technology conference 2007.

REFERENCIAS

- [1] N. Abramson, "Development of the ALOHANET," IEEE Trans. on Info. Theory, vol. IT-31, no. 2, March 1985.
- [2] D. J. Aldous, "Ultimate instability of exponential back-off protocol for acknowledgment-based transmission control of random access communication channels," IEEE Trans. Inform. Theory, vol. 33, no. 2, pp. 219–223, Mar. 1987.
- [3] D. Bertsekas and R. Gallager, Data Networks, 2nd Ed., Prentice Hall, 1992.
- [4] Bhandari, B.N.; Kumar, R.V.R.; Maskara, S.L.; "Uplink performance of the IEEE802.16 medium access control (MAC) layer protocol". Personal Wireless Communications, 2005. ICPWC 2005. 2005 IEEE International Conference on 23-25 Jan. 2005 Page(s):5 – 8.
- [5] Bhandari, B.N.; Kumar, Ratnam V. Raja; Maskara, S.L.; "Performance of the IEEE 802.16 MAC Protocol with Varying Request Backoff Start". TENCON 2006. 2006 IEEE Region 10 Conference. 14-17 Nov. 2006 Page(s):1 – 4.
- [6] J. Capetanakis, "Tree Algorithms for Packet Broadcast Channels," IEEE Trans. on Info. Theory, vol. IT-25, no. 5, Sept. 1979.
- [7] N. Golmie, Y. Saintillan, and D. Su, "A Review of Contention Resolution Algorithms for IEEE 802.14 Networks", IEEE Communication Surveys, Vol. 2, No. 1, 1999.
- [8] J. Goodman, A. G. Greenberg, N. Madras, and P. March, "Stability of binary exponential backoff," J. ACM, vol. 35, no. 3, pp. 579–602, 1988.
- [9] J. Håstad, T. Leighton, and B. Rogoff, "Analysis of backoff protocols for multiple access channels," SIAM J. Comput., vol. 25, no. 4, pp. 740–744, 1996.
- [10] "Evaluation Models for digital bi-directional IEEE 802.14 MAC protocol", IEEE 802.14 Subcommittee, 14/95-061R2, January 1996.
- [11] IEEE 802.16.2-2001, "IEEE Recommended Practice for Local and Metropolitan Area Networks – Coexistence of Fixed Broadband Wireless Access Systems". Sept. 2001.
- [12] IEEE 802.16a "IEEE Standard for Local and metropolitan area networks – Part 16: Air Interface for Fixed Broadband Wireless Access Systems— Amendment 2: Medium Access Control Modifications and Additional Physical Layer Specifications for 2–11 GHz".
- [13] IEEE 802.16-2004 "IEEE Recommended Practice for Local and metropolitan area networks Coexistence of Fixed Broadband Wireless Access Systems". Marzo 2004.

- [14] IEEE 802.16c “Standard for Local and metropolitan area networks Part 16: Air Interface for Fixed Broadband Wireless Access Systems— Amendment 1: Detailed System Profiles for 10–66 GHz”.
- [15] IEEE802.16 e “Draft Amendment to IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems — Amendment for Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands”.
- [16] ITU-T Rec. G.992.1, “Asymmetrical Digital Subscriber Line (ADSL) Transceivers,” 06/09.
- [17] P. Jaquet, P. Muhlethaler, and P. Robert, “Asymtotic average access delay analysis: adaptive p-persistence versus tree algorithm”, IEEE 802.14 Working Group, paper 96-248, 1996.
- [18] Kantawong, S.; Sivamok, N.; Wuttisittikulij, L.; “Investigation of contention resolution algorithms for wireless communication systems. Vehicular Technology Conference, 2001. VTC 2001 Fall. IEEE VTS 54th Volume 1, 2001 Page(s):222 - 226 vol.1
- [19] A. Law, M. McComas, “Simulation software for communications networks: the state of the art”, IEEE Communs. Mag., Vol. 32, No. 3, pp.44-50, Mar. 1994.
- [20] Lidong Lin; Weijia Jia; Bo Han; Lizhuo Zhang; “Performance Improvement using Dynamic Contention Window Adjustment for Initial Ranging in IEEE 802.16 P2MP Networks”. Wireless Communications and Networking Conference, 2007.WCNC 2007. IEEE. 11-15 March 2007 Page(s):1877 - 1882 .
- [21] F. Ling and K. Walker, “FLO: An Optimized Mobile Multimedia Multicasting Physical Layer,” 24th AIAA International Communications Satellite Systems.
- [22] M. López Bejarano. “Análisis de Algoritmos de resolución de colisiones en redes inalámbricas de banda ancha”. Tesis de Maestría, junio 2006.
- [23] OPNET Modeller v11.0 C, “OPNET Modeller Simulation Package”, OPNET Technologies (MIL 3), 1997, <http://www.opnet.com>
- [24] V. Rangel and R. M. Edwards, “Performance Evaluation of Adaptive Contention Slot Allocators for CATV Networks based on the European Cable Communications Protocol DVB/ DAVIC”, Journal of the SCTE, vol. 24, Sep. 2002, pp 24-30.
- [25] D. Sala, J.O. Limb, and S.Khaunte, “Adaptive MAC Protocol for a Cable Modem”, Georgia Tech Technical Report GIT-CC-97-014, May 1997.

- [26] D. Sala, J. O. LimbK, and S. U. Khaunte, "Adaptive Control Mechanism for Cable Modem MAC Protocols", Proceedings of INFOCOM'98, ISBN 0-7803-4383-2, San Francisco CA, Mar. 1998.
- [27] J. R. Shoch and J. A. Hupp, "Measured performance of an ethernet local network," Commun. ACM, vol. 23, no. 12, pp. 711-721, Dec. 1980.
- [28] K. Siriram, "Methodologies for Bandwidth Allocation Transmission Scheduling, and Congestion Avoidance in Broadband ATM Networks", Computer Network & ISDN Systems, Vol. 26, pp. 43-59, 1993.
- [29] W. Stallings, "Local and Metropolitan Area Networks", Prentice Hall International Editions, 5th edition, ISBN 0-13-253733-8, pp. 426-460, 1997.
- [30] W. Stallings, "Self similarity upsets data traffic assumptions", IEEE Spectrum, pp. 28-29, Jan. 1997.
- [31] Sung-Min Oh; Jae-Hyun Kim; "The Optimization of the Collision Resolution Algorithm for Broadband Wireless Access Network". Advanced Communication Technology, 2006. ICACT 2006. The 8th International Conference. Volume 3, 20-22 Feb. 2006 Page(s):1944 - 1948.
- [32] TR 102 377, Digital Video Broadcasting (DVB); DVB-H Implementation Guidelines, ETSI, November 2005.
- [33] P. Tzerefos, "On the Performance and Scalability of digital upstream DOCSIS 1.0 conformant CATV channels", Department of Computers Science, The University Of Schiffield, Ph. D. Thesis, Oct. 1999.
- [34] Vinel, A.; Ying Zhang; Lott, M.; Tiurlikov, A.; "Performance Analysis of the Random Access in IEEE 802.16". Personal, Indoor and Mobile Radio Communications, 2005. PIMRC 2005. IEEE 16th International Symposium on Volume 3, 11-14 Sept. 2005 Page(s):1596 - 1600.

Sitios web

<http://www.ieee.org/portal/site>

<http://www.opnet.com>

<http://www.intel.com/technology/itj/archive/2004.htm>

<http://www.intel.com/netcomms/bbw/>

<http://www.intel.com/technology/magazine/standards/st08031.pdf>

<http://www.intel.com/cd/ids/developer/asmo-na/eng/strategy/trends/95510.htm>

<http://www.intel.com/cd/ids/developer/asmo-na/eng/dc/pca/optimization/94553.htm?page=5802.16>

http://www.intel.com/standards/case/case_802_11.htm

<http://grouper.ieee.org/groups/802/16/index.html>

http://www.intel.com/standards/case/case_wimax.htm

<http://grouper.ieee.org/groups/802/>

<http://grouper.ieee.org/groups/802/11/>

<http://grouper.ieee.org/groups/802/16/pubs/P80216e.html>

<http://grouper.ieee.org/groups/802/20/>

<http://www.wimaxforum.org/home>

<http://www.intel.com/cd/ids/developer/asmo-na/eng/strategy/trends/95510.htm?page=3>