



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

“ Estudio de técnicas de slots de tiempo para mejorar el desempeño del MAC en el estándar 802.11 ”

T E S I S

QUE PARA OBTENER EL GRADO DE:

**MAESTRO EN INGENIERÍA
(COMPUTACIÓN)**

P R E S E N T A:

García Jiménez Luis Francisco

DIRECTOR DE TESIS: ” Javier Gómez Castellanos”

México, D.F.

2008.



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

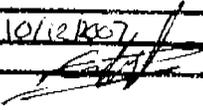
Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

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.

Autorizo a la Dirección General de Bibliotecas de la UNAM a difundir en formato electrónico a través el contenido de mi trabajo recepcional.

NOMBRE: García Jiménez Luis Francisco

FECHA: 10/12/2007

FIRMA: 

Agradecimientos

Agradezco el apoyo brindado a cada una de las personas que laboran en el Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas de la UNAM. Especialmente el apoyo por parte del Dr. Javier Gómez Castellanos, ya que sin su ayuda este proyecto no se hubiera realizado. Asimismo, las aportaciones y recomendaciones que surgieron durante la revisión de este trabajo para mejorarlo por parte de: Dr. Víctor Rangel Licea, Dr. Miguel López Guerrero, Dr. Demetrio Fablán García Nocetti, Dr. Sergio Rajsbaum Gorodesky. Por otro lado agradezco el apoyo brindado por parte del CONACYT.

Índice general

1. Introducción	11
1.1. Objetivo	13
1.2. Contribución	13
1.3. Descripción del contenido	13
2. El estándar 802.11	15
2.1. Topologías de Redes en el 802,11	15
2.1.1. BSS	15
2.1.2. IBSS	16
2.1.3. ESS	16
2.2. Pila de Protocolos	17
2.2.1. Problemas en los enlaces de radio frecuencia	17
2.2.2. Modos de acceso MAC	19
2.2.3. CSMA/CA	19
2.2.4. Exponential Backoff	21
2.2.5. Relación de tramas SIFS, PIFS, DIFS y slots del Algoritmo Backoff	22
3. Propuesta para mejorar el desempeño sobre la capa MAC del estándar 802,11	25
3.1. Deficiencias del protocolo CSMA/CA	25
3.2. Propuesta	27
3.2.1. Variante de la propuesta principal (Offset)	31
3.3. Justificación	32
3.4. Trabajos Relacionados	32
3.5. Modelo del sistema	34
3.6. Validación Matemática	35
4. Modelado del caso de estudio utilizando Network Simulator (NS-2)	39
4.1. Generalidades del simulador NS-2	39
4.1.1. Organizador de Eventos de NS-2	40
4.1.2. Modelo inalámbrico básico en NS	40
4.1.3. Nodos móviles en NS-2	41
4.2. Implementación de la propuesta de tesis	41
4.2.1. Implementación de la variante de la propuesta principal (Offset)	46

5. Resultados.	49
5.1. Esquema de simulación	49
5.2. Esquema de pruebas	51
5.3. Resultados	53
5.3.1. Resultados Offset	58
6. Conclusiones.	63
6.1. Trabajos Futuros	64
A. Script Otcl	67
A.1. Definición de variables	67
A.2. Organizador de eventos	67
A.3. Creación de la topología	68
A.4. Activación de la opción de trazado	68
A.5. Configuración y creación del nodo móvil	69
A.6. Movimiento del nodo	69
A.7. Generador de tráfico	70
B. Modelo Bidimensional de Markov	71
B.0.1. Modelo Matemático	71

Índice de figuras

2.1. Topología BSS	16
2.2. Topología IBSS	16
2.3. Nodo oculto	18
2.4. Intercambio RTS/CTS-DATOS-ACK	18
2.5. Nodo expuesto	19
2.6. Uso de NAV	20
2.7. Algoritmo Exponential Backoff	22
3.1. Disponibilidad del medio	26
3.2. Disponibilidad de Dominios en el medio	27
3.3. Superposición en el tiempo	28
3.4. Premisa 2	30
3.5. Offset	31
3.6. Canales ISM no superpuestos	32
3.7. Dominios mutuamente excluyentes	36
3.8. Throughput deseable	37
4.1. Esquema del nodo móvil	41
5.1. Throughput para 10 nodos	54
5.2. Throughput para 25 nodos	55
5.3. Throughput para 50 nodos	55
5.4. Colisiones para 10 nodos	56
5.5. Colisiones para 25 nodos	56
5.6. Colisiones para 50 nodos	57
5.7. Retardo promedio para 10 nodos	58
5.8. Retardo promedio para 25 nodos	59
5.9. Retardo promedio para 50 nodos	59
5.10. Throughput para 25 nodos con offset	60
5.11. Throughput para 50 nodos con offset	61
B.1. Cadena de Markov	72
B.2. T_s, T_c	76

Índice de tablas

5.1. Parámetros usados en DSSS	50
--	----

Capítulo 1

Introducción

El uso de sistemas inalámbricos ha dejado atrás la necesidad de estar fijos a una conexión física, ya que permiten la libertad de movimiento e incrementan la facilidad de comunicación en lugares de difícil acceso para los sistemas basados en cables. Las redes inalámbricas utilizan las ondas de radio como mecanismo de envío de datos, con el cual se facilita la comunicación entre sistemas que no deben permanecer estáticos; sin embargo, se genera un reto en el diseño eficiente para lograr la comunicación entre este tipo de sistemas. El medio electromagnético que es usado en los sistemas inalámbricos está dividido en bandas de frecuencias que ofrecen diferentes usos, donde la mayoría de estas bandas requieren una licencia para su utilización; no obstante, existen bandas de frecuencia libres que impulsan la creación de protocolos que permiten la comunicación entre sistemas sin la necesidad de pagar un derecho de uso. Uno de estos protocolos es el estándar 802,11, que asienta las reglas para que dos o más estaciones inalámbricas puedan comunicarse utilizando una banda de frecuencia libre llamada Industrial, Científica y Médica (ISM).

Uno de los problemas al usar una banda libre es el hecho de que coexisten grupos de redes inalámbricas dentro del mismo ancho de banda. Esto provoca que la existencia de interferencia y errores dentro del uso del canal aumenten, ocasionando que la cantidad de paquetes que la red es capaz de manejar disminuya considerablemente. Es por ello que la creación de protocolos que regulen el acceso entre las terminales que desean transmitir simultáneamente es de suma importancia. Es fácil observar que la banda de frecuencia del estándar 802,11 es un recurso limitado. Y por lo tanto el estándar 802,11 tiene varias extensiones que le han permitido mejorar sus características sobre la propuesta base. Este trabajo propone una mejora en el rendimiento del estándar a través de modificar la capa de Control de Acceso al Medio (MAC).

El estándar 802,11 hace uso del protocolo Acceso Múltiple con Detección de Portadora y Prevención de Colisión (CSMA/CA) para garantizar el uso equitativo y descentralizado entre las terminales que desean acceder al medio. Sin embargo, para escenarios donde el tráfico o el número de nodos solicitando transmitir es alto (más de 20 terminales), el desempeño de la red se reduce considerablemente. Se sabe que el número de terminales que intentan acceder al canal es uno de los problemas más predominantes que afectan el

rendimiento de las redes de área local, es por ello que trabajos relacionados en aumentar la eficiencia de la red basan su esfuerzo en la manipulación de los parámetros que maneja el estándar 802,11. Algunas de las propuestas son: el artículo [12] ajustar dinámicamente la ventana del algoritmo de retardo (exponential backoff) a través de estimar el número de terminales en contención. El artículo [9] maneja una probabilidad de colisión, al distribuir dinámicamente el backoff entre las terminales que desean acceder al medio. El artículo [11] administra la ventana del algoritmo backoff utilizando una probabilidad p después de una transmisión exitosa, etc. Este trabajo propone una manera simple de aumentar considerablemente la eficiencia de la red en estado de saturación al emplear el número de terminales en contención como la variable a modificar.

La propuesta de tesis introduce una técnica de slots de tiempo con el uso de CSMA/CA, para administrar la utilización del medio y asignar un número de usuarios a cada slot de tiempo, es decir, para un slot dado, sólo la mitad de nodos podrán acceder al canal, mientras los restantes deben esperar el siguiente slot de tiempo para acceder al medio. De esta forma se logra que solamente la mitad de las terminales contienda por el acceso al medio en un slot dado, esto trae como consecuencia que el número de colisiones se reduzca significativamente, y a su vez, la ventana del algoritmo *exponential backoff* no crezca tan rápido; tal como sucede en el estándar base. También produce que el valor de penalización promedio que las terminales seleccionan para acceder al medio se reduzca exponencialmente. Todo lo anterior refleja una mayor posibilidad de atender peticiones y administrar el medio de una manera más eficiente, además de que mejora considerablemente el consumo de baterías en los sistemas inalámbricos.

Esta propuesta de tesis hace uso de un modelo matemático (presentado en [4] y resumido en el apéndice B) basado en cadenas de Markov que permite calcular el desempeño de una red inalámbrica sobre el estándar 802,11. A través de este modelo matemático se realizaron los cambios pertinentes para obtener el modelo de técnicas de slots de tiempo y con ambos modelos se compara el beneficio de usar la técnica de slots de tiempo contra el estándar base. El modelo de técnicas de slots muestra que el uso de divisiones de tiempo obtiene el throughput para una red de $\frac{n}{2}$ nodos cuando la red tiene un tamaño de n nodos en contención. Que claramente es mayor, debido a que la eficiencia de la red está en función del número de nodos. El análisis del modelo supone la existencia tanto de colisiones como errores en el medio, pero deja a un lado el problema de la terminal expuesta. Finalmente, la propuesta de tesis se implementa bajo el simulador de redes NS-2 con el fin de comparar el desempeño práctico del modelo de técnicas de división de slots contra el desempeño del estándar base.

1.1. Objetivo

Modificar el estándar base 802,11 con el propósito de mejorar el rendimiento en términos de la capacidad de *carga-throughput* sobre la capa de acceso al medio en una red inalámbrica, a través del estudio basado en un modelo matemático que justifica la modificación del estándar base. Asimismo, implementar estas modificaciones sobre el simulador de redes NS-2 con el fin de cuantificar la ganancia de rendimiento de la propuesta con respecto al estándar base.

1.2. Contribución

Con la implementación de las modificaciones del acceso al medio del estándar base 802,11 sobre el simulador NS-2 y a través de los resultados obtenidos del porcentaje de ganancia en throughput de la red. Se pretende justificar la creación de una nueva extensión sobre el estándar base. Además el software obtenido se divulgará entre la comunidad científica interesada en el estudio de las redes inalámbricas bajo el estándar 802,11. A su vez, dado la ganancia obtenida y que además esta propuesta se puede utilizar en paralelo con muchas propuestas que mejoran el desempeño del 802,11, se tendrá la posibilidad que los fabricantes de hardware inalámbrico consideren la posibilidad de implementar la propuesta.

1.3. Descripción del contenido

- En el capítulo 2 se presenta la teoría relacionada a la propuesta de tesis. El capítulo está enfocado en mostrar los aspectos más importantes que abarca el estándar 802,11, principalmente la capa de control de acceso al medio (MAC) por su siglas en inglés *Media Access Control*.
- En el capítulo 3 se describe la propuesta del trabajo de tesis así como su justificación y su validación matemática.
- En el capítulo 4 se describen las características esenciales del simulador NS-2, el cual se utilizó para la implementación de la propuesta de este trabajo. Se definen los componentes necesarios para las redes móviles como son: sus mecanismos de enrutamiento, la creación de la pila de red que permite el acceso al canal, el soporte de trazado y la generación de escenarios de movimiento y de tráfico. Posteriormente se implementa la propuesta de tesis en las clases correspondientes del simulador.
- En el capítulo 5 se muestran los esquemas de simulación y pruebas realizadas sobre el simulador NS-2, con el propósito de evaluar el desempeño de la propuesta de tesis y comparar los resultados con respecto al comportamiento del estándar base 802,11.

- En el capítulo 6 se muestran las conclusiones del trabajo. Finalmente se proponen trabajos futuros.

Capítulo 2

El estándar 802.11

Esta sección presenta la teoría relacionada a la propuesta de tesis. El capítulo está enfocado en mostrar los aspectos más importantes que abarca el estándar 802,11, principalmente la capa de control de acceso al medio (MAC) por su siglas en inglés *Medium Access Control*.

2.1. Topologías de Redes en el 802,11

El estándar 802,11 define tres maneras diferentes de conectar sistemas [2] [1] [3]:

1. Área de servicio básico (Basic service sets-**BSS**).
2. Área Independiente de servicio básico (Independent basic service sets-**IBSS**).
3. Área extendida de servicio (Extended service sets-**ESS**).

Una área de servicio se define como el rango para el cual dos estaciones que desean comunicarse son visibles entre ellas.

2.1.1. BSS

El área de servicio básico utiliza una estación central que gestiona y controla las comunicaciones, esta estación es llamada punto de acceso (AP). A través del punto de acceso se establecen todas las comunicaciones dentro del área de servicio, esto es, si una estación *A* desea comunicarse con otra estación *B* dentro del área de servicio, la comunicación deberá ser gestionada por el punto de acceso de la siguiente forma: la estación *A* envía su paquete de datos al punto de acceso y el punto de acceso enviará el paquete al nodo *B*. Este tipo de tecnología normalmente está conectada a una red de cables como Ethernet y es por ello que también se conoce como red de infraestructura. Una BSS común se muestra en la Figura 2.1.

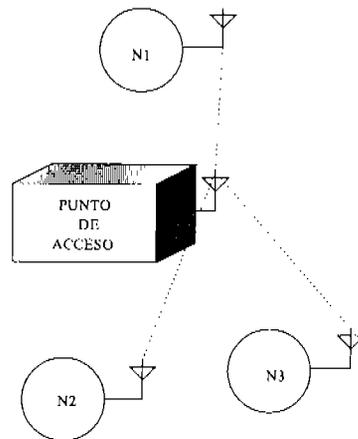


Figura 2.1: Topología BSS

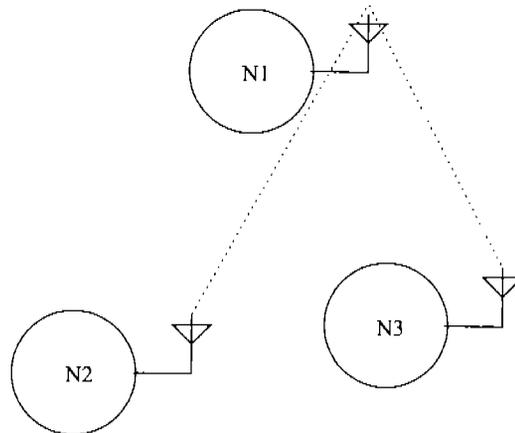


Figura 2.2: Topología IBSS

2.1.2. IBSS

Una IBSS está compuesta por dispositivos que se comunican directamente entre ellos sin la necesidad de un dispositivo central que regule las comunicaciones. El establecimiento de la comunicación necesita que tanto el transmisor como el receptor se vean directamente o conozcan previamente una ruta para establecer la comunicación. Una IBSS también es conocida como una red Ad-Hoc, porque es en esencia una red punto a punto como lo muestra la Figura 2.2.

2.1.3. ESS

Una ESS se utiliza para incrementar el área de servicio básico, con el fin de aumentar el rango donde los dispositivos pueden comunicarse. Para realizar esto se nombran todos los puntos de acceso de la misma manera y cada uno de ellos gestionará y controlará su propia BSS. Si una estación deseará comu-

nicarse con otra estación que no está en la BSS de la estación transmisora, los puntos de acceso se comunicarán entre ellos para enviar el paquete a su destinatario.

2.2. Pila de Protocolos

Al igual que muchos de los estándares de la IEEE que siguen como referencia el modelo OSI, el estándar 802,11 no es la excepción. La capa física del estándar 802,11 corresponde perfectamente a la establecida en el modelo OSI; sin embargo, la capa de enlace de datos se divide en dos subcapas [1]: la primera de ellas es *Medium Acces Control Layer* (MAC), que establece como se asigna el canal, es decir, la manera en que un dispositivo se apodera del medio para transmitir, la segunda es llamada Logical Link Control (LLC), cuyo propósito es ocultar las diferentes variantes de los estándares 802.X con el fin de que sean imperceptibles para la capa de red. Entre sus tareas están: controlar la sincronización de las tramas a transmitir (control de flujo) y el control de errores.

2.2.1. Problemas en los enlaces de radio frecuencia

El estándar 802,11 utiliza una banda libre para establecer sus comunicaciones, esto hace que la banda está sujeta a un sin número de problemas, como son: sistemas de microondas, fuentes de radio frecuencia, movimientos de las estaciones, desvanecimiento de la señal en múltiples rutas, etc. Esto hace que la comunicación dentro del estándar se torne más complejo que en un medio donde la transmisión de datos es por cables. Sin embargo, existen dos problemas de suma importancia en los sistemas móviles y son llamados: problema de la terminal oculta y la terminal expuesta.

2.2.1.1. El problema del nodo oculto

El área de servicio de un red inalámbrica suele ser de unos cientos de metros, pero esto depende de la potencia utilizada por las antenas de los dispositivos y de los obstáculos que se encuentren dentro de la comunicación de las terminales. En la Figura 2.3 el nodo $N2$ es capaz de ver tanto al nodo $N1$ como al nodo $N3$; sin embargo, el nodo $N1$ y $N3$ se encuentran lo suficientemente lejos para establecer una comunicación o existen obstáculos que no permiten que sus antenas puedan verse. Se dice que el nodo $N1$ es un nodo oculto para el nodo $N3$ y viceversa. Si los nodos $N1$ y $N2$ establecen una comunicación, el nodo $N3$ no sabe de dicha comunicación y podría enviar una trama de datos al nodo $N2$ interrumpiendo la comunicación.

El estándar 802,11 utiliza señales que garantizan que la comunicación entre dos terminales no sea interrumpida. Esto lo logra al enviar paquetes de control que reservan el medio. En la Figura 2.4 se ilustra el procedimiento. El nodo $N1$ reserva su BSS a través de enviar una trama llamada **Request To Send** (RTS),

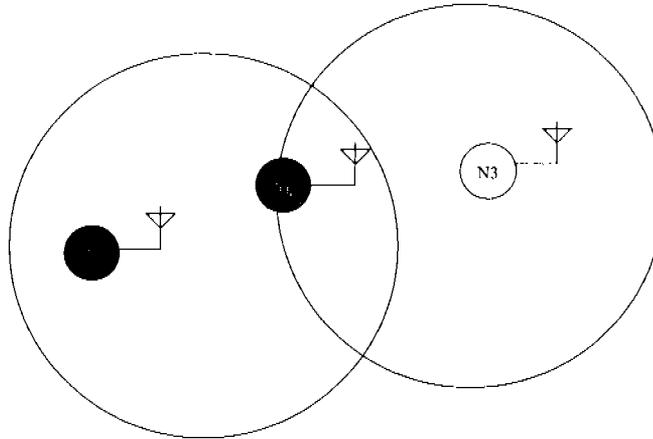


Figura 2.3: Nodo oculto

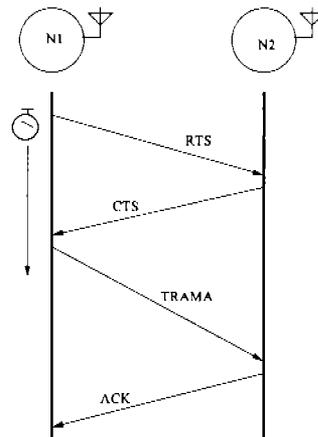


Figura 2.4: Intercambio RTS/CTS-DATOS-ACK

que asegura que ningún nodo dentro de la BSS del nodo $N1$ envíe paquetes. Si el nodo receptor recibe la solicitud RTS, éste contestará con un paquete de control llamado **C**lear **T**o **S**end (CTS), que de igual forma reserva la BSS del nodo receptor, que para el ejemplo es el nodo $N2$. De esta forma se evita que cualquier nodo dentro de las BSS's de los nodos transmisor y receptor envíen paquetes que interrumpa la comunicación. Una vez reservado el canal, se envía el paquete de datos y se confirma la correcta recepción del paquete en el nodo destino a través del uso de una señal de confirmación positiva (ACK).

2.2.1.2. Problema de la terminal expuesta

El problema inverso al nodo oculto es llamado de terminal expuesta. En la Figura 2.5, el nodo $N2$ desea enviar datos al nodo $N1$, por lo que $N2$ envía una trama RTS indicándole a $N1$ que quiere transmitir datos. En ese momento el nodo $N3$ se entera de la transmisión y debe silenciarse. $N1$ devuelve a $N2$ su trama CTS y se establece la comunicación. Entretanto, si $N3$ deseara trans-

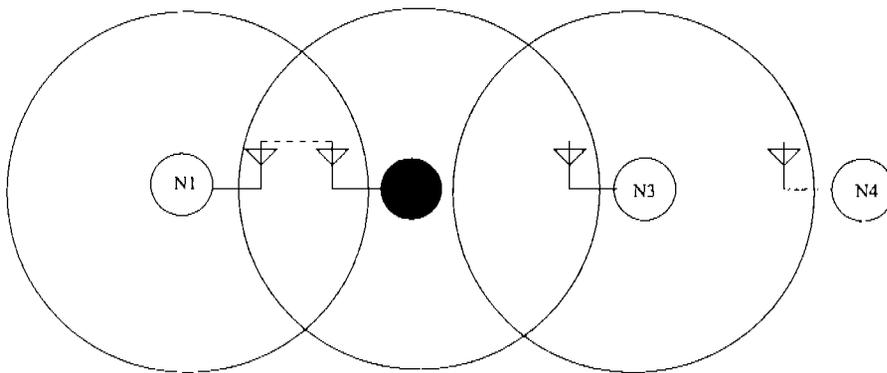


Figura 2.5: Nodo expuesto

mitir a $N4$, no puede hacerlo, debido a que fue silenciado por $N2$ aún cuando no interfiere con la transmisión de los nodos $N1 - N2$.

2.2.2. Modos de acceso MAC

El acceso al medio inalámbrico se controla mediante funciones de coordinación. El estándar 802,11 establece tres tipos de coordinación de acceso al medio [1]:

1. **Distributes Coordination Fundation (DCF)**: es el mecanismo básico de acceso al medio en el estándar 802,11. Este mecanismo hace uso del protocolo **Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA)** para coordinar el acceso al medio. Funciona de la siguiente manera: si una estación desea transmitir escucha el canal y verifica si éste se encuentra libre, si es así, espera un tiempo aleatorio asignado por el algoritmo de demora *Exponential Backoff* y sólo si el medio sigue libre después de consumir el periodo de demora, enviará su trama. De esta forma el nodo que tenga el tiempo más pequeño de penalización ganará el medio. DCF puede o no utilizar el intercambio RTS/CTS con el fin de evitar colisiones, pero si lo utiliza, deberá verificar el correcto envío de datos a través del uso de una confirmación positiva (ACK).
2. **Point Coordination Funtion (PCF)**: Se utiliza en redes de infraestructura, lo que asegura que una estación central controle el envío de las tramas y de esta manera proporciona un medio sin contención.
3. **Hybrid Coordination Funtion (HCF)**: Esta función de coordinación es una combinación de las dos anteriores, con el propósito de equilibrar los servicios dentro de una red inalámbrica.

2.2.3. CSMA/CA

El estándar 802,11 coordina sus envíos de tramas por medio del protocolo CSMA/CA a través de dos modos de escucha de portadora, llamadas física y

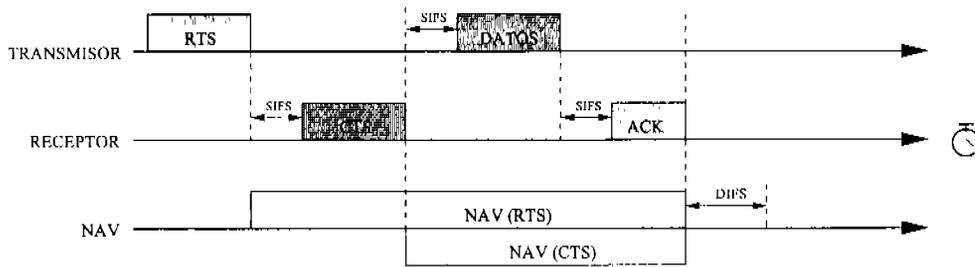


Figura 2.6: Uso de NAV

virtual.

2.2.3.1. Función de portadora física

La función de escucha física se realiza en la interfaz aérea por medio de la recepción de intensidad de las señales que otras fuentes están enviando. A través de los niveles de intensidad puede comprobar si el medio se encuentra libre o está ocupado por alguna estación.

2.2.3.2. Función de portadora virtual

La portadora virtual funciona al introducir campos de tiempo en los paquetes de control RTS/CTS, para establecer la duración que el canal va a permanecer ocupado por alguna estación que desea transmitir. Este campo dentro de los paquetes de control es llamado **Network Allocation Vector (NAV)**. Su funcionamiento es el siguiente (ver Figura 2.6): el nodo transmisor reserva el canal a través del uso del paquete de control RTS, dentro de este paquete se fija el tiempo que el nodo transmisor necesita para transferir sus datos. Al propagarse el paquete RTS dentro de su BSS, los nodos aledaños al nodo transmisor fijan su NAV y esperarán ese tiempo para tratar de acceder nuevamente al medio. El nodo receptor responde con su paquete de control CTS, en el cual se encuentra el tiempo que el nodo receptor necesita para enviar su trama. Al propagarse el paquete CTS, los nodos dentro de la BSS del nodo receptor fijarán su NAV y esperarán que el NAV tenga un valor de cero para intentar acceder al medio. De esta forma se asegura que ningún nodo interrumpa la comunicación entre dos nodos que previamente ya han establecido su comunicación. Cuando el NAV de todos los nodos tenga un valor de cero implica que el canal está libre y puede ser utilizado por cualquiera de los nodos que desean acceder al medio.

Dentro de la operación de intercambio de datos o también llamado operación atómica se definen espacios de tiempo que jerarquizan el acceso al medio o permiten crear niveles de prioridad, donde tráfico de alta prioridad debe ser atendido más rápido que el tráfico de baja prioridad, esto permite que si hablamos de una red de infraestructura y el punto de acceso necesita interrumpir la operación atómica, ésta pueda ser atendida, ya que tendría la mayor prioridad. La estructura de tiempos y prioridades se describen a continuación:

1. **Short Interframe Space (SIFS)**: Se utiliza para las transmisiones de prioridad más alta, como las tramas RTS, CTS, ACK y Trama de datos. Se define como el tiempo medido desde la última trama enviada en el canal hasta el comienzo de la siguiente trama. Es el periodo de tiempo más pequeño y es utilizado para coordinar el intercambio entre paquetes de control y tramas de datos.
2. **PCF Interframe Space (PIFS)**: Se utiliza únicamente en las redes de infraestructura, permitiéndole al punto de acceso tomar el canal sin competir contra las estaciones. Este periodo de tiempo le concede al punto de acceso que envíe tramas de datos (beacons) para verificar el canal y realizar alguna tarea.
3. **DCF Interframe Space (DIFS)**: Es un periodo de tiempo con el cual las estaciones aseguran que el canal se encuentra libre. Después de una trama de datos exitosa todas las estaciones deben esperar un tiempo DIFS para garantizar que el medio está libre.
4. **Extended Interframe Space (EIFS)**: Si una estación recibe una trama con errores deberá esperar un tiempo EIFS para reportar la trama.

2.2.4. Exponential Backoff

Después de transcurrir un periodo DIFS, los dispositivos escucharán el medio y si éste se encuentra libre cualquier dispositivo tienen el mismo derecho de tomar el medio, por lo que si al menos dos dispositivos tienen su valor de NAV en cero y ambos escuchan el medio y verifican que está libre, la probabilidad de una colisión es de uno. Para evitar esta situación en lo más posible la DCF utiliza un algoritmo llamado Exponential Backoff, éste consiste en: tras una colisión los dispositivos tienen que esperar un número aleatorio asignado mediante el algoritmo Exponential Backoff antes de volver a verificar el canal. Este número son periodos de tiempo denominados *slots*. Estos slots se asignan en el intervalo de 0 a $2^i - 1$, donde i es el número de colisiones que ha sufrido el paquete. Los acuses de recibo positivos (ACK) son la única indicación del éxito de una transmisión exitosa. Si se genera un problema dentro del intercambio de datos, la transmisión se considera un fracaso y el remitente considera la retransmisión. Por cada fallo se incrementa un contador de retransmisión i , esto es, por cada colisión o transmisión fallida, el dispositivo tomará un número aleatorio entre 0 y $2^i - 1$. Sin embargo, tras haber alcanzado 10 colisiones el contador se detiene. El estándar permite sólo seis intentos más con el mismo rango de números a escoger, si se diera este caso, se debe avisar a las capas superiores de un error.

Las estaciones toman un slot aleatoriamente entre 0 - $2^i - 1$ y esperan dicho tiempo antes de intentar acceder al medio nuevamente; todas las ranuras son selecciones igualmente probables. El número de penalizaciones o slots tomados mediante el algoritmo exponential backoff se reducen mientras el canal esté libre. En el momento que el dispositivo tenga un valor de slots

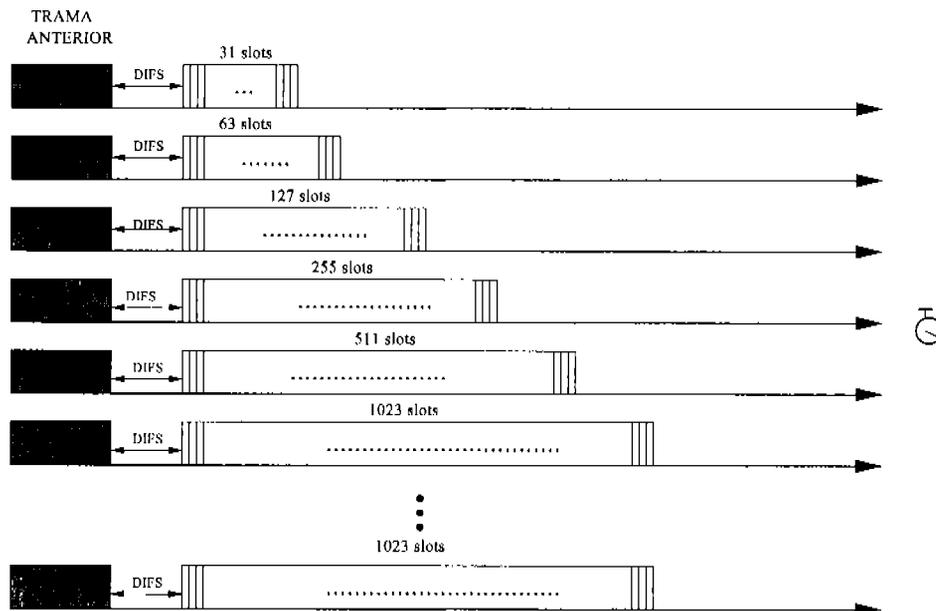


Figura 2.7: Algoritmo Exponential Backoff

igual a cero, éste verificará el canal y si está libre transmitirá; si se produjera una colisión debe llamar nuevamente al algoritmo y tomar un nuevo número aleatorio.

En la Figura 2.7 se muestra el flujo del algoritmo exponential backoff.

2.2.5. Relación de tramas SIFS, PIFS, DIFS y slots del Algoritmo Backoff

La relación entre tramas en un intercambio de datos depende únicamente del tipo de acceso usado en la capa física y no de la tasa de transferencia de datos. La capa física (PHY) del estándar está dividida en dos subcapas llamadas: Procedimiento de convergencia de capa física (PLCP) y la subcapa Dependiente del medio físico (PMD). La capa PLCP es la responsable de intercambiar datos entre la capa MAC y la capa PMD. Mientras la capa PMD se encarga de transmitir los datos provenientes de la capa PLCP utilizando una interfaz aérea. Además de estas capas, la capa física tiene una función llamada Función de valoración de canal limpio (CCA) que le avisa a la capa MAC cuando la interfaz aérea detecta una señal.

La siguiente numeración muestra los tiempos que las diferentes subcapas de la capa PHY necesitan para procesar los datos [1].

1. $RxRFDelay$: es el tiempo que tarda el último símbolo en salir de la interfaz aérea.
2. $RxPLCPDelay$: es el tiempo que la PLCP usa para enviar un bit de la PMD que fue recibido por la capa MAC.

3. `MACProcessingDelay` : es el tiempo que utiliza el MAC para procesar una trama.
4. `RxTxTurnarounTime`: es el tiempo máximo que la PHY requiere para cambiar de un estado de recepción a un estado de transmisión.
5. `CCATime`: es el tiempo mínimo que requiere el mecanismo CCA para verificar si el medio está libre o no.
6. `AirPropagationTime`: es el tiempo que la señal de radio utiliza para viajar desde la antena transmisora hasta la antena receptora.

Los tiempos entre tramas se definen de la siguiente forma [1]:

$$SIFS = RxRFDelay + RxPLCPDelay + MACProcessingDelay + RxTxTurnarounTime$$

$$Slot = CCATime + RxTxTurnarounTime + AirPropagationTime + MacProcessingDelay$$

$$PIFS = SIFS + Slot$$

$$DIFS = SIFS + 2*Slot$$

Capítulo 3

Propuesta para mejorar el desempeño sobre la capa MAC del estándar 802,11

En este capítulo se describe la propuesta del trabajo de tesis así como su justificación y su validación matemática.

3.1. Deficiencias del protocolo CSMA/CA

Diversos estudios [4] [24] [22] [6] muestran que el desempeño de una LAN inalámbrica basada en el estándar 802,11 depende de los siguientes parámetros: el número de nodos que intentan acceder al medio, la velocidad de transmisión de datos, el tamaño del paquete a enviar, la distancia entre nodos, etc. Es por ello, que un diseño eficiente sobre el protocolo que coordina el acceso al medio es de suma importancia para el rendimiento de una red inalámbrica.

El estándar 802,11 utiliza la función de coordinación distribuida (DCF) como mecanismo primario para acceder al medio. La DCF utiliza el protocolo (CSMA/CA) para garantizar el envío de información y mantener un buen desempeño. Para muchos escenarios el CSMA/CA provee una óptima utilización del canal; sin embargo, para escenarios con cargas muy altas, la utilización del medio tiende a reducirse considerablemente, en otras palabras, la estabilidad del sistema tiende a perderse conforme la carga aumenta.

Para optimizar el estándar 802,11 sobre la capa de acceso al medio, se pueden manipular los siguientes parámetros del protocolo: el algoritmo de retardo (*exponential backoff*), el número de terminales compitiendo por el canal, la probabilidad de colisión, la cantidad de paquetes entrantes sobre la red o carga, etc. De tal suerte que la modificación de uno o varios parámetros logre una ganancia significativa en la utilización del canal. Como todos estos parámetros se relacionan entre sí, la modificación de alguno de ellos repercutirá en los otros y determinará los valores que caracterizan o miden el estado de la red.

Si se toma el algoritmo de retardo como uno de los parámetros a modificar

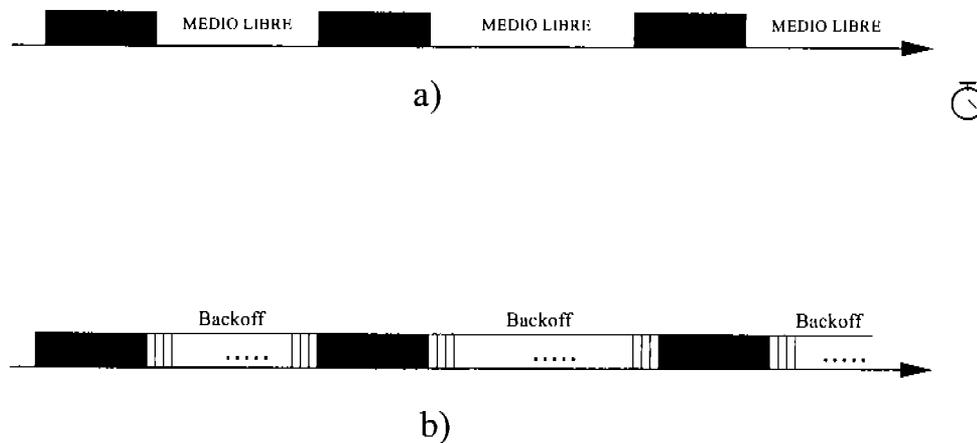


Figura 3.1: Disponibilidad del medio

se sabe que: el algoritmo de retardo aleatorio (*exponential backoff*) es un mecanismo que evita que dos o más estaciones transmitan al mismo tiempo, ya que utiliza un valor de penalización que es obtenido aleatoriamente y que las estaciones debe consumir para tratar de acceder al canal e intentar transmitir. Cuando una red permanece en estado lincal, la probabilidad de que una estación verifique el canal y pueda transmitir es alta, ya que en esta situación el canal permanece la mayor parte del tiempo libre y pocas estaciones desean transmitir (ver Figura 3.1 a)). Mientras en un estado de saturación, el número de colisiones se incrementa exponencialmente, ya que la probabilidad de que más de dos nodos transmitan al mismo tiempo es muy alta. Esto ocasiona que por cada colisión el algoritmo de retardo doble su ventana de penalización, y como consecuencia de ello, el tiempo promedio de slots de penalización aumenta. Todo lo anterior provoca una reducción en el número de paquetes que la red es capaz de transportar en un periodo dado (ver Figura 3.1 b)). Es por ello que otro punto de enfoque de este trabajo es lograr que el tiempo promedio de penalización sea lo más corto posible y como consecuencia la probabilidad de transmisión aumente; mejorando la eficiencia de la red.

Durante el intercambio de paquetes de control RTS/CTS en la DCF, una colisión tiene lugar durante la transmisión del encabezado RTS o durante la transmisión del paquete CTS, ya que una vez establecida la comunicación el canal queda reservado mediante la portadora virtual (NAV). Durante el estado de saturación de la red, el número promedio de colisiones aumenta conforme el número entrante de paquetes a la red rebasa el throughput máximo. Es en este escenario que la utilización del canal se ve afectada debido al gran número de colisiones producidas, ya que el tiempo de canal que utiliza la DCF para tratar una colisión es de un RTS + DIFS o RTS + DIFS + SIFS + CTS y son irre recuperables. A través de lo mencionado, esta tesis propone una solución simple que minimiza en gran porcentaje los efectos negativos sobre la eficiencia de una red inalámbrica basada en el estándar 802,11 en estado de saturación.

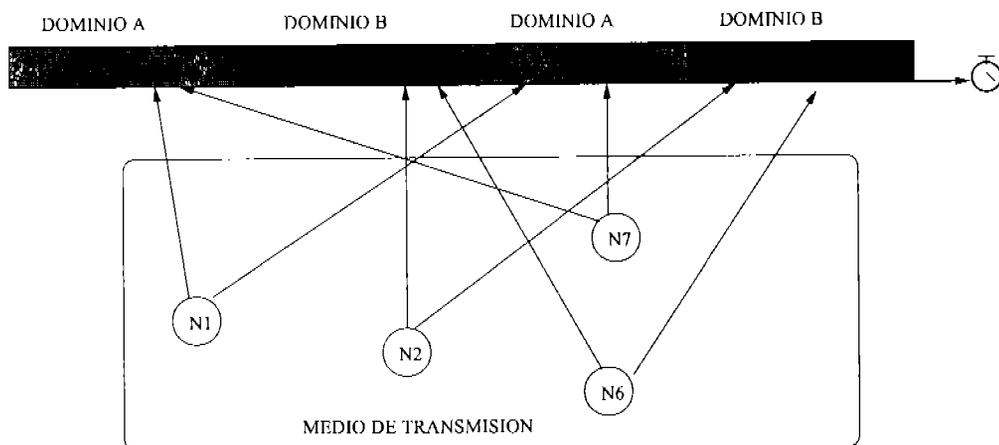


Figura 3.2: Disponibilidad de Dominios en el medio

3.2. Propuesta

Cualquier protocolo de acceso aleatorio tiene un comportamiento inestable con incrementos significativos de carga, especialmente cuando esta carga sobrepasa el throughput máximo. Se sabe [4] [22] [24] que el número de terminales compitiendo por el medio es uno de los factores predominantes que afectan el rendimiento de la red y hacen que ésta cambie de un estado lineal a uno de saturación. Para reducir los efectos negativos que producen el número de terminales compitiendo por el uso del canal, se introdujo una técnica de slots de tiempo, con el fin de administrar la utilización del canal al asignarle un número fijo de terminales a cada slot. Se define un slot como: un periodo de tiempo T para el cual la mitad de las terminales que están en el medio compiten por el canal.

Durante un slot de tiempo, las terminales definidas en un dominio compiten por el acceso al medio, esto es, la mitad de terminales asignadas como el grupo A o B . Por lo que una estación dentro del dominio A nunca competirá por el canal con una terminal del grupo B y viceversa. Durante el slot de tiempo, el dominio en posesión del canal se comporta bajo las reglas del protocolo base 802,11 tal cual están definidas. Por lo que utiliza una DCF y un protocolo de acceso al medio CSMA/CA que coordinan el uso del medio. Cada estación competirá con el resto de las estaciones en su dominio. La estación ganadora utilizará el envío de control RTS/CTS (siempre y cuando el tamaño del paquete rebasa el límite establecido en $RTSThreshold$) para establecer una comunicación y transmitir su paquete. La Figura 3.2 muestra el concepto de slots de tiempo, para el cual el número de terminales en contención está dividido en dos dominios A y B y a cada dominio le corresponde un slot de tiempo que es intercalando de uno en uno, esto es, para un tiempo dado, un dominio posee el canal y al término de su slot el otro dominio utilizará el canal.

Es importante notar que el estándar 802,11 define el establecimiento de una comunicación como una operación atómica (irrompible). Si una estación que en la línea del tiempo le pertenece el canal y un instante antes de que su slot

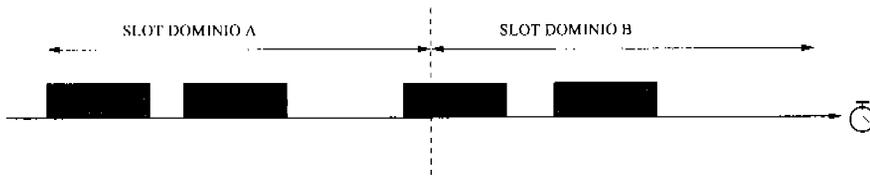


Figura 3.3: Superposición en el tiempo

termine ya ha enviado el protocolo de control RTS/CTS o la comunicación ya está establecida, este intercambio de datos no se puede romper. Entonces se traslapará en el tiempo con el dominio que ahora le corresponde tomar el control del canal, al término de la transmisión del paquete traslapado, las estaciones del dominio que tienen el uso del canal, intentarán transmitir. La Figura 3.3 muestra el caso en que una transmisión del dominio *A* se traslapa en el tiempo con el dominio *B*. Cualquier comunicación ya establecida no se puede romper, por lo que no importando en que momento ocurra el cambio de slot, esa comunicación se debe respetar hasta su finalización.

Mientras transcurre el tiempo de algún slot, las terminales de ambos dominios pueden generar paquetes a una tasa λ paquetes por segundo. Esto se debe al hecho de que no se le puede restringir al usuario que espere su slot de tiempo para generar su paquete, por ello el nodo debe ser capaz de almacenar el paquete en su buffer hasta que éste pueda ser enviado (respetando la capacidad máxima de almacenamiento disponible en el nodo). Sólo el dominio que tenga el permiso de acceder al medio podrá enviar sus paquetes, mientras los paquetes generados por el dominio que tiene el semáforo en rojo esperarán hasta que su slot de tiempo comience. De esta forma se garantiza que es una red de n nodos compitiendo por el canal, aunque en un slot de tiempo únicamente un dominio de $\frac{n}{2}$ nodos puedan acceder al canal. Esta acción es transparente para el usuario final, por lo que ningún usuario debe preocuparse por el dominio que le corresponde ni por el periodo de cambio de slot.

Durante la posesión del canal por alguno de los dominios (en un slot de tiempo), es importante no olvidar que exclusivamente las terminales que tienen el derecho de acceder al medio también tienen el derecho de reducir sus periodos de demora asignados por el algoritmo de *exponential backoff*, siempre y cuando el canal se encuentre libre. Las terminales del dominio que no poseen el canal no pueden reducir la cuenta asignada por el algoritmo de retardo debido a dos premisas importantes. Primero, se tiene un conjunto de n terminales que fue dividido en $\frac{n}{2}$ nodos, a los cuales se les denominó el dominio *A* y *B* respectivamente, cada dominio actúa independientemente del otro dominio. Por lo que en un slot de tiempo dado, únicamente uno de los dominios fungirá como el dueño del medio; sin embargo, el otro dominio tiene la capacidad de generar paquetes, por lo que ambos dominios trabajan en paralelo, es decir, ambos dominios se pueden ver como procesos independientes pero con el fin último de aumentar la eficiencia del canal. Segundo, si se permitiera la reducción del valor asignado por el algoritmo de retardo a las terminales

que no poseen el medio, la probabilidad de que más de dos estaciones reduzcan su cuenta regresiva a cero es muy alta, provocando que en el momento en que se les permitiera escuchar el canal (cambio de slot), estas estaciones colisionarían desperdiciando tiempo de canal.

Como ejemplo de la segunda premisa (ver Figura 3.4): las estaciones $N1$, $N2$, $N3$, $N4$, $N5$ y $N6$ tienen un valor de cuenta de demora: 2, 5, 3, 8, 6 y 4 respectivamente. Las terminales $N1$, $N3$ y $N5$ pertenecen al dominio A , por lo que las terminales $N2$, $N4$ y $N6$ pertenecen al dominio B . El dominio A tiene el derecho de transmitir. El nodo $N1$ transmitirá primero, ya que es la cuenta regresiva más pequeña y las terminales $N2$, $N3$, $N4$, $N5$ y $N6$ tendrán una cuenta regresiva de 3, 1, 6, 4 y 2 respectivamente. La siguiente terminal en transmitir será $N3$, por lo que ahora las terminales $N2$, $N4$, $N5$ y $N6$ tienen su cuenta regresiva en 2, 5, 3, 1 respectivamente. La última estación del dominio A en transmitir será la estación $N5$ que esperó otras tres cuentas regresivas. Para este momento la estación $N2$ tiene un valor de cero en su cuenta regresiva y está lista para transmitir en el momento que haya cambio de slot. El nodo $N6$ también está listo para transmitir inmediatamente cuando cambie el slot. Es importante notar que el nodo $N6$ y el nodo $N2$ tienen una cuenta regresiva con valor de cero; sin embargo, no se les permite transmitir debido a que su dominio no posee el canal. Si el slot es lo suficientemente grande para permitir la reducción de las últimas dos cuentas regresivas del nodo $N4$, los tres nodos del dominio B transmitirán en el momento del cambio de slot, lo que provocará una colisión entre las tres estaciones. Es por ello que no se permite que los nodos que no posean el canal reduzcan su *exponential backoff*. Se puede observar que si se permite la reducción de la cuenta regresiva al dominio que no posee el medio, la probabilidad de una colisión incluso cuando todos los valores del *exponential backoff* de cada estación son totalmente diferentes es muy alta, especialmente para slots de dimensión grande.

En esencia las terminales que poseen el canal podrán reducir sus slots ¹ o cuentas regresivas del algoritmo de demora durante la posesión del canal. Al final de su periodo de posesión del canal deberán congelar su *exponential backoff* hasta el próximo cambio de slot.

Es fácil observar que conforme el número de estaciones aumenta, la utilización del canal se verá disminuida debido al número de colisiones. Este problema se debe principalmente a la cantidad de números que el algoritmo de espera *exponential backoff* tiene para asignar. Como ejemplo, si se tienen 50 estaciones en contención, y el algoritmo comienza con un rango de valores del 1 al 16, la posibilidad de que al menos la mitad de nodos colisionen es muy alta, puesto que la cantidad de números en la primera etapa del *exponential backoff* no es suficiente para asignar un número distinto a cada estación. Por cada colisión que sufra un paquete se duplica el rango de números que el *exponential backoff* puede elegir. Este fenómeno se repite cada vez que un paquete colisiona, causando que el número promedio de slots o cuentas regre-

¹Se podrá nombrar como slot a: primero, el concepto de posesión del canal en un periodo de tiempo definido; segundo, el periodo de demora establecido en el estándar a través del algoritmo *exponential backoff*

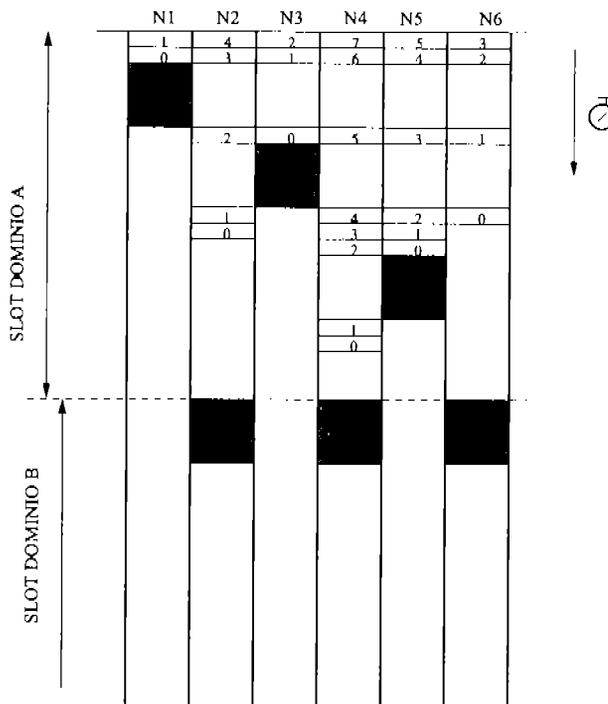


Figura 3.4: Premisa 2

sivas a escoger aumente, y como consecuencia existe un desaprovechamiento del canal. Por otro lado, si se comienza con una ventana o rango de números muy grande en el algoritmo de espera, se tiene la posibilidad de que el número de estaciones sea pequeño y sus tiempos de espera promedio se relacionan con el tamaño de la ventana, lo que causaría desperdicio de la utilización del canal. Se puede observar que existe una estrecha relación entre la ventana del algoritmo de espera, la utilización del canal y la cantidad de nodos que desean transmitir. Del ejemplo anterior se puede observar que al disminuir la cantidad de nodos que compiten por el medio a la mitad (de 50 a 25), la probabilidad de colisión disminuye y el rango de números de penalización que el algoritmo de demora tiene para asignar en su primer etapa se aproxima al número de terminales que se encuentran en el medio.

Se sabe [4] que la probabilidad de transmisión de una estación está en función del número de nodos que desean acceder al medio y el tiempo que le toma al canal manejar una colisión. De hecho, estos parámetros están relacionados inversamente con la probabilidad de transmisión, por lo que si la cantidad de nodos disminuye, la probabilidad de éxito de una transmisión aumenta. Por otro lado, si la cantidad de colisiones disminuye, la probabilidad de éxito aumenta. El tiempo que le toma al canal tratar una colisión es la suma de un RTS + DIFS o RTS + DIFS + SIFS + CTS, si ocurre en el paquete RTS o CTS respectivamente. Si se minimizan las colisiones, estos fragmentos de tiempo se pueden utilizar en el envío de carga de datos, que se traduce en un mejor aprovechamiento del canal. Si además se sabe que al reducir la cantidad de

nodos que compiten por el canal, el algoritmo de retardo no tendrá la necesidad de trabajar con ventanas muy grandes. Todo esto contribuye a la ganancia de tiempo de transmisión de datos.

3.2.1. Variante de la propuesta principal (Offset)

El uso de la técnica de slots de tiempo garantiza que el throughput del canal aumente; sin embargo, el uso de slots grandes permite que estaciones que generan paquetes y no les corresponde el canal, acumulen dichos paquetes con valores de penalización en el rango de la primera ventana del algoritmo backoff (1-31). Esto provoca que al cambio de slot o pertenencia del canal se produzcan colisiones, dado que la probabilidad de que varias estaciones escojan el mismo valor de penalización es alto. Es por ello que se decidió desfasar un slot de tiempo a los paquetes generados dentro de la no pertenencia del canal. Esto es: si una estación que no le corresponde el canal y genera un paquete, se recorrerá en la línea del tiempo exactamente la longitud de un slot con respecto al tiempo en el que se generó el paquete; con el propósito de distribuir los paquetes durante el slot de pertenencia del medio y de esta forma reducir la probabilidad de colisión.

En la Figura 3.5 el dominio en posesión del canal es A y los paquetes generados del dominio B se desfasan un slot de tiempo con el propósito de distribuir los paquetes y evitar las posibles colisiones como sucede en el escenario a). Se puede observar que tres paquetes tienen el mismo valor de penalización ($EB = 5$) y si no se desfasaran colisionarían, debido a que los tres paquetes tienen el mismo número de penalización. Esto sucede en el escenario b).

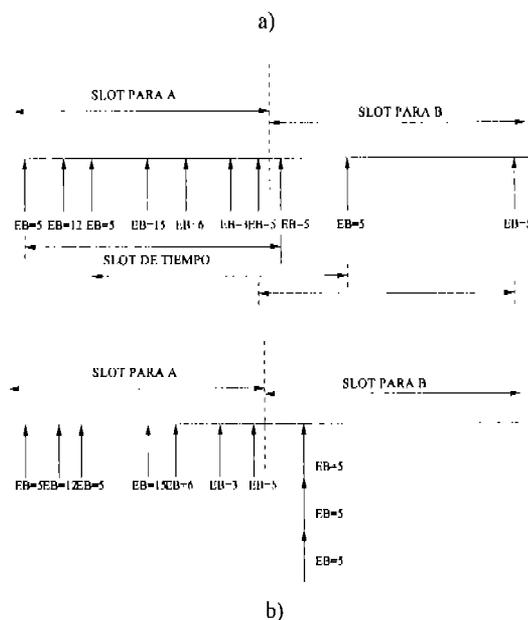


Figura 3.5: Offset

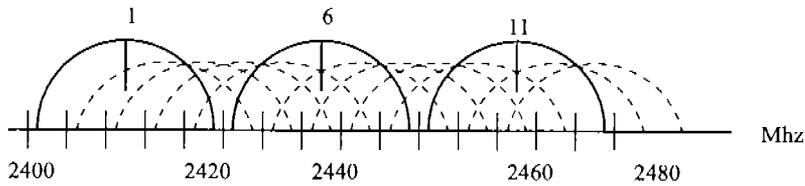


Figura 3.6: Canales ISM no superpuestos

3.3. Justificación

El estándar 802,11 permite configurar hasta catorce canales diferentes utilizando la técnica de espectro disperso DSSS, que es la técnica más usada actualmente. La banda ISM opera en la frecuencia de $2,4\text{ GHz}$ con una separación entre canales de 25 MHz entre portadoras. En la práctica solamente se configuran tres canales en la misma zona (ESS), debido a las interferencias entre canales adyacentes. El hecho de usar una banda de frecuencia libre conlleva a que se debe compartir los recursos entre redes vecinas. Esto es, si se utiliza el mismo canal para comunicarse, el ancho de banda debe ser repartido entre las redes. Por otro lado, si no se utiliza el mismo canal, la función de coordinación no verá las redes vecinas y la aparición de colisiones aumenta indiscutiblemente. La Figura 3.6 muestra la máscara espectral de las transmisiones en los canales denominados no superpuestos (1, 6 y 11). Estos canales proporcionan un medio libre de colisiones dentro del mismo ancho de banda. Este escenario muestra que la cantidad de nodos dentro de un ESS es limitado y por tanto es importante aumentar la capacidad de una red con el manejo del estándar. Es por ello que optimizar el desempeño del estándar 802,11 está justificado, ya que el número final de clientes conectados a una ESS se reduce a sólo tres puntos de acceso como máximo en una área sin interferencia. Debido a lo anterior esta propuesta se enfoca en el número de terminales que acceden al medio y desean ser atendidos a pesar de un nivel alto de carga en la red.

Uno de los problemas fuertes en el uso de las redes inalámbricas es el consumo de las baterías. Muchos de los esfuerzos por optimizar el estándar se encuentran en este punto, ya que la libertad de movimiento implica que el sistema debe ser autónomo. Es aquí donde esta propuesta toma un papel importante al reducir hasta un 50% el consumo de las baterías.

3.4. Trabajos Relacionados

Muchos de los trabajos propuestos en optimizar el estándar 802,11 sobre una DCF se reducen a adaptar los parámetros del estándar como son: balancear la carga de la red, manejar la ventana del algoritmo *exponential backoff*, asignar una probabilidad de transmisión, filtrar eficientemente entre canales adyacentes. Algunas de las propuestas mencionadas se presentan a continuación y se comparan con esta propuesta de tesis.

1. El artículo propuesto en [9] trata el problema de la probabilidad de colisión, al distribuir dinámicamente el backoff a través de todas las terminales intentando acceder al medio, esto lo logra al reducir el tiempo perdido en ventanas muy grandes del algoritmo backoff de la siguiente manera: dado un valor de penalización, el canal reducirá un valor de slots igual a $((CW_{min} + 1) * 2 - 1)$ y si aún el valor de penalización no tiene un valor de cero, entonces por cada slot que el canal reduzca, el número de penalizaciones se reduce a la mitad. También usa una probabilidad de colisión, tal que si el nodo detecta que es posible una colisión, aumentará su ventana de penalización y obtendrá un nuevo valor. Además, si la estación está disminuyendo los slots de penalización y observa que aparece un estado ocupado en la red, incrementará su ventana y tomará un nuevo valor de penalización. Sin embargo, la utilización de este algoritmo concede una mayor probabilidad de transmisión a terminales que recientemente han transmitido, destruyendo la premisa de equidad establecida en el estándar 802.11.
2. En [10] se presenta un protocolo que minimiza el número de colisiones a través de reducir a la mitad el tamaño de la ventana del algoritmo backoff después de c transmisiones exitosas. El algoritmo muestra que para un c que maximiza el sistema, éste logra una ganancia del 15% al 20% de throughput contra el estándar base.
3. En [11] se presenta un protocolo que reduce la ventana del algoritmo backoff al estado inicial a través de una probabilidad p después de una transmisión exitosa, mejorando los tiempos de uso del backoff. Los resultados muestran que hay ganancias de un 15% a un 20% en el throughput del sistema comparado con un estándar base.
4. En [12] se propone un ajuste dinámico a la ventana usada por el algoritmo backoff, dado que después de una transmisión exitosa, el algoritmo regresa a la primera ventana en el rango de (1-31). Esto trae como consecuencia que si la carga en la red es alta, la probabilidad de que más de dos estaciones escojan el mismo número de penalización es alta y conllevará a una colisión. El throughput del sistema se maximiza a través de:

$$W_{opt} = n \sqrt{2 * \frac{T_c}{\sigma}}$$

donde la ventana óptima se expresa en términos de la cantidad de nodos que intentan acceder a la red (n), el tiempo que el canal toma para tratar una colisión (T_c) y la longitud del tamaño del slot del algoritmo exponencial backoff (σ). Uno de los problemas es que el valor de n cambia conforme se añaden o reducen terminales en la línea del tiempo, por lo que se debe avisar a cada nodo de la alta o baja del número de terminales en cada instante, esto trae como consecuencia una sobrecarga innecesaria

de datos sobre la red. Una alternativa para la estimación del número de terminales es a través del cálculo de la carga actual en la red, un inconveniente es que la banda libre ISM es normalmente un medio con ruido y para varios escenarios pueden existir estimaciones no realistas. Además en el artículo se menciona que el uso de la ventana óptima tiene un mejor desempeño únicamente para paquetes de tamaño pequeño.

Este trabajo al hacer una división de dominios, resuelve de forma muy natural muchos de los efectos negativos generados por la utilización de una DCF con el protocolo CSMA/CA. Además, no existe la necesidad de generar tráfico redundante ni de control sobre la red. Esto da lugar a un mejor aprovechamiento del tiempo de uso del canal, que se traduce en una alza al throughput total. Por otro lado, uno de los problemas más fuertes en los sistemas inalámbricos es el consumo de batería, del cual, al permanecer la mitad del tiempo sin la necesidad de transmisión (debido al uso de dominios de tiempo), el dispositivo móvil puede mantenerse en estado de ahorro de energía durante la longitud del slot que no le pertenece el canal. También, cada una de las propuestas ya mencionadas se pueden utilizar en paralelo con esta propuesta de tesis, mejorando aún más el desempeño de una red inalámbrica.

3.5. Modelo del sistema

A continuación se especifican los términos y suposiciones del sistema utilizados en la mayor parte de la tesis, a menos que se especifique lo contrario:

- La arquitectura del sistema corresponde a un sistema inalámbrico que utiliza una DCF bajo la norma IEEE 802,11. Los nodos acceden al medio a través de un protocolo de acceso al medio aleatorio (CSMA/CA), y reservan el canal utilizando la portadora virtual (NAV) después de reducir su cuenta aleatoria provista por un mecanismo de demora llamado *exponential backoff*.
- El tamaño del paquete de datos o frame a transmitir es de Γ segundos y es igual para todos los paquetes generados en todos los nodos.
- Cada nodo es independiente del resto. En el sistema todos los nodos generan carga de tráfico de g paquetes por segundo bajo un proceso de Poisson.
- Los paquetes generados por cada nodo son almacenados en su buffer. El buffer almacena una cantidad finita de paquetes, por lo que una cantidad mayor de estos, serán paquetes perdidos.
- Todos los nodos puede comunicarse entre sí, ya que la línea de vista es directa y no se utiliza un protocolo de ruteo para alcanzar la línea de vista de algún nodo.

- Se considera un canal con errores y con colisiones.
- Colisión: Es el hecho en donde dos o más paquetes se traslapan en el tiempo ya sea parcial o totalmente. Si ocurre una colisión el mecanismo de espera alatorio (*exponential backoff*) será llamado con un tamaño de ventana $2^i - 1$, donde i corresponde a la i -ésima colisión del paquete.
- Throughput: Corresponde al número de transmisiones efectivas de datos en una fracción de tiempo, donde estas transmisiones pueden ser correctas o retransmitidas.
- Retardo de Acceso: Se refiere al tiempo que transcurre desde que un nodo genera un paquete hasta que dicho paquete se recibe correctamente por el nodo deseado.
- Se considera un canal en saturación, lo que garantiza que el buffer de cualquier nodo nunca esté vacío, por lo que siempre existe un paquete disponible para transmitirse.
- Se considera un canal en el cual no existe el problema de la terminal expuesta ni la terminal oculta.

3.6. Validación Matemática

Esta sección hace uso del análisis del throughput del estándar 802.11 propuesto por [4] y presentado en el Apéndice B. A través de este estudio se presentan los cambios realizados que permiten calcular el throughput de una red inalámbrica basada en la técnica de slots de tiempo y con ello comparar el beneficio que otorga el uso de la técnica de slots de tiempo contra el estándar base.

El análisis propuesto en [4] emplea un modelo bidimensional de cadenas de Markov, asumiendo un canal sin problemas de terminales ocultas y terminales expuestas, además que el número de terminales que desean acceder al medio es finito. El estudio detallado del artículo propuesto en [4] se presenta en el Apéndice B.

La Figura 3.7 muestra como en la línea del tiempo cada dominio es independiente del otro y por tanto se pueden ver como conjuntos mutuamente excluyentes. El throughput del modelo de técnicas de slots de puede calcular de la siguiente manera:

El throughput se define como [4] [1]:

$$S = \frac{\text{La cantidad de información transmitida exitosamente}}{\text{La longitud del periodo de tiempo}} \quad (3.1)$$

De la ecuación 3.1 se puede calcular la cantidad de paquetes que se transmitieron exitosamente durante la pertenencia del dominio A . Después se divide sobre el tiempo de pertenencia del canal:

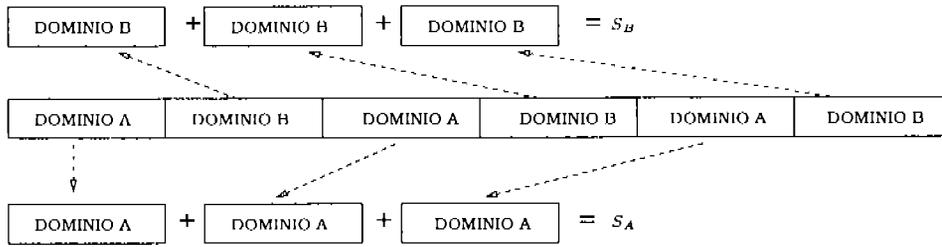


Figura 3.7: Dominios mutuamente excluyentes

$$S_A = \frac{\text{Cantidad de paquetes transmitidos exitosamente por } A}{0,5T} \tag{3.2}$$

donde T es la longitud del periodo de uso de la red.
 El throughput del dominio B se puede obtener como:

$$S_B = \frac{\text{Cantidad de paquetes transmitidos exitosamente por } B}{0,5T} \tag{3.3}$$

como la cantidad de paquetes recibidos por el dominio A es aproximadamente igual a la cantidad de paquetes recibidos en el dominio B , al sumar 3.2 y 3.4 el throughput total es:

$$S = \frac{\text{Cantidad de paquetes transmitidos exitosamente por } A \text{ y por } B}{T} \tag{3.4}$$

Dado que para cada dominio de tiempo la red se comporta como una red de $\frac{n}{2}$ nodos en lugar de una red de n nodos, el throughput total se aproxima al throughput de una red de $\frac{n}{2}$ nodos cuando la red debiese obtener el throughput de una red de n nodos. Se puede ver que el throughput obtenido es mayor, debido a que el desempeño de una red está en función de la cantidad de terminales que compiten por el medio, además este valor es el máximo alcanzable por el modelo para un T muy grande.

Al tomar el modelo propuesto en el Apéndice B.16 y variar la probabilidad de transmisión se puede comparar la eficiencia o throughput de una red de n nodos con respecto a una red de $\frac{n}{2}$ nodos. Por ejemplo, para una red de 50 nodos, la propuesta de tesis debería producir el valor de throughput de una red de 25 nodos. La Figura 3.8 muestra las gráficas de throughput normalizado con respecto a la probabilidad de transmisión para una red de 50 y 25 nodos en el medio. Se espera que la propuesta alcance los valores de una red de 25 nodos cuando tiene un tamaño 50 terminales.

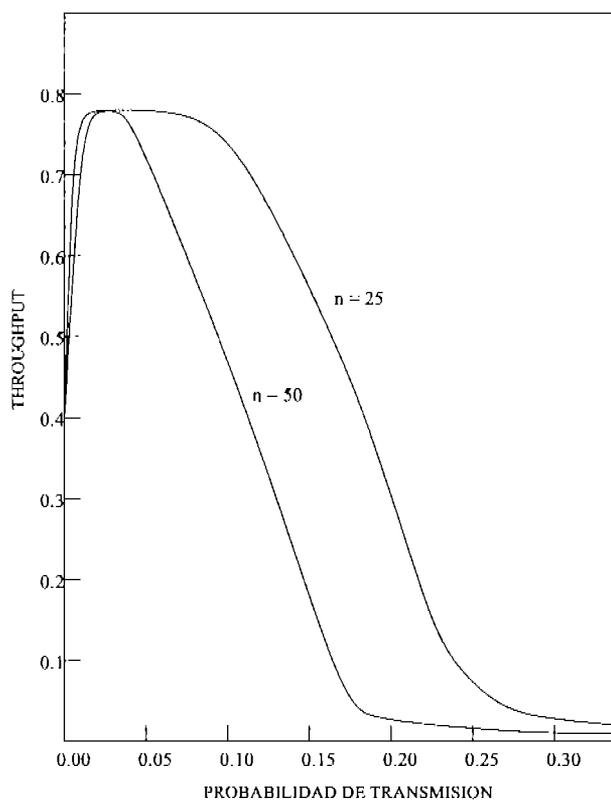


Figura 3.8: Throughput deseable

Capítulo 4

Modelado del caso de estudio utilizando Network Simulator (NS-2)

En este capítulo se describen las características esenciales del simulador NS-2, el cual se utilizó para la implementación de la propuesta de este trabajo. Se definen los componentes necesarios para las redes móviles como son: sus mecanismos de enrutamiento, la creación de la pila de red que permite el acceso al canal, el soporte de trazado y la generación de escenarios de movimiento y de tráfico. Posteriormente se implementa la propuesta de tesis en las clases correspondientes del simulador.

4.1. Generalidades del simulador NS-2

Network Simulator es un simulador de redes orientado a objetos y manejado por eventos discretos que simulan una gran variedad de redes IP; implementa protocolos de transporte tales como Protocolo de Control de Transmisión (TCP), Protocolo de Datagrama de Usuario (UDP); generadores de tráfico como Protocolo de Transferencia de Archivos (FTP), Telnet, web, cbr y real-audio; mecanismos de administración de filas de espera como Drop Tail, CBQ y variantes de Fair Queueing (FQ) tales como FQ, SFQ (Stochastic FQ) y DRR (Deficit Round-Robin); algoritmos de enrutamiento como Rutas de Fuentes Dinámicas (DSR), No ad-hoc Routing (NOAH), FLOODING; además, proporciona funciones de visualización y trazado de datos (NAM) [16] [17] [18].

El simulador NS utiliza dos lenguajes de programación (C++ y OTcl) para satisfacer las diferentes tareas que realiza el simulador. Por un lado requiere de un lenguaje que pueda manejar bytes y protocolos de forma eficiente y por el otro lado necesita crear escenarios y configurar parámetros para examinar la eficiencia de los diversos protocolos implementados. NS logra complementar estas dos tareas con el uso de los lenguajes C++ y OTcl, que embonan perfectamente para los dos propósitos principales del simulador.

Para usar NS, un usuario debe escribir un script en OTcl que inicia un

organizador de eventos (entrada al sistema), el cual debe contener la topología de red, los objetos de red y las funciones que le dicen a los generadores de tráfico cuando deben iniciar y cuando terminar de transmitir paquetes a través del organizador de eventos.

4.1.1. Organizador de Eventos de NS-2

Otro componente importante en el simulador NS es el organizador de eventos. Un evento en NS es un identificador de paquete único, el cual es apuntado a un objeto que maneja el evento. El organizador de eventos mantiene la información del tiempo de simulación de todos los eventos que deben ser atendidos mediante el llamado de componentes de red apropiados, estos componentes normalmente son aquellos que expidieron los eventos, y se les permite realizar una acción apropiada asociada con el paquete apuntado por el evento.

4.1.2. Modelo inalámbrico básico en NS

El modelo inalámbrico esencialmente consiste del nodo móvil como núcleo del modelo de movilidad, con características de soporte adicionales que permiten la simulación de redes adhoc de múltiples saltos, WLANs, redes de sensores, etc. Un nodo móvil es un objeto heredado de un nodo básico con funcionalidades agregadas, tales como: la capacidad de moverse dentro de una topología dada, capacidad de recibir y transmitir señales hacia y del canal inalámbrico, capacidad de encontrar rutas, las siguientes referencias son guías de aprendizaje para el simulador NS-2 [16] [17] [18] [19] [20].

Los archivos de configuración típicos de un nodo móvil contienen lo siguiente:

- Estructura genérica del Script en Otcl:
 - Definición de Variables Globales.
 - Creación del organizador de eventos.
 - Creación de la topología.
 - Activación de la opción de trazado.
 - Configuración del nodo móvil.
 - Creación del nodo móvil.
 - Movimiento del nodo móvil en caso de que se desee.
 - Generador de tráfico.
 - Definición del modelo de movimiento y de tráfico.
 - Finalización de la simulación.

Los detalles de la estructura genérica del Script Otcl como entrada al simulador se muestran en el Apéndice A

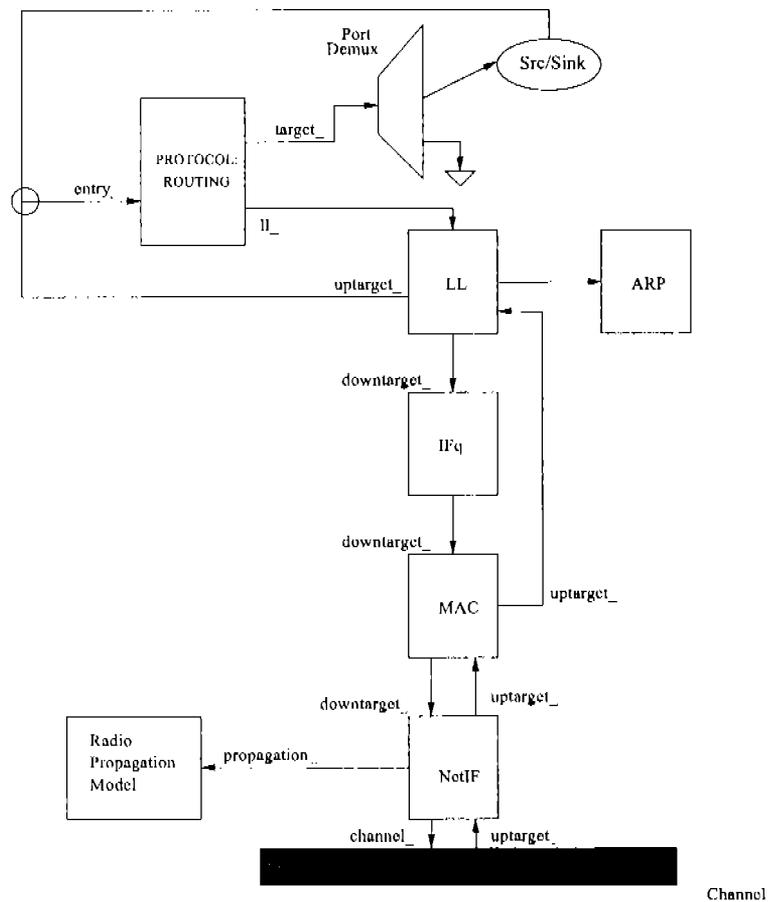


Figura 4.1: Esquema del nodo móvil

4.1.3. Nodos móviles en NS-2

Un nodo móvil es un objeto de NS con funcionalidades como: demultiplexores, agentes de enrutamiento, una pila de red que consiste de una capa de enlace (LL), un módulo ARP conectado a la capa LL, una interfase de fila de espera (IFq), una capa mac (MAC) e interfase de red (netIF), todos estos conectados al canal inalámbrico. El esquema de un nodo móvil se muestra en la Figura 4.1:

4.2. Implementación de la propuesta de tesis

El simulador NS-2 implementa el acceso al medio del estándar 802,11 tal cual fue definido en el capítulo 2 y en [1]. Por lo que una estación que desca transmitir debe esperar un tiempo DIFS y después transmitirá un paquete de control RTS sobre su BSS; esperará un SIFS + δ para recibir la contestación CTS del nodo receptor. Si esta respuesta no llega en un SIFS + δ de tiempo, se entenderá que hubo una colisión y el nodo deberá llamar al algoritmo de

retardo para seleccionar una penalización. En caso contrario, el canal queda reservado mediante la portadora virtual de ambos nodos. Después de un SIFS de tiempo el paquete de datos es transmitido del nodo transmisor al nodo receptor. Una vez terminada la transacción de datos, el nodo transmisor esperará un SIFS + δ con el fin de obtener una confirmación del nodo receptor, si es así, el nodo transmisor habrá confirmando que todo ha sido correcto, en caso contrario, se reintentara la transmisión de datos. Cada uno de estos eventos de tiempo son simulados a través del organizador de eventos de NS-2. Por ejemplo, el envío de un paquete RTS es construido en la capa MAC y enviado a la capa PHY cuando esta última le avisa que está lista para transmitir, es en este momento que un temporizador calcula el tiempo (a través de la cantidad de bits que contiene el paquete y la velocidad de transmisión del canal definida en la capa PHY) que el paquete de control va estar en el medio. Al transcurrir ese tiempo el "manejador"(Handler) llama a la función correspondiente para continuar con la transmisión de control o atender la colisión, si se dio el caso.

La primer función a definir en el acceso al medio es la división de los dominios *A* y *B*. Para ello se hizo uso de la dirección MAC del nodo. NS-2 asigna con números enteros las direcciones IP y MAC de los nodos. Se definió que los nodos con terminación par pertenecieran al dominio *A*, mientras que los nodos con terminación impar comprenden el dominio *B*. Se sabe que en la línea del tiempo la propuesta necesita de un periodo llamado slot, en el cual la mitad de las terminales que pertenecen a un dominio tratarán de acceder al medio; al término de este periodo, el dominio contiguo fungirá como dueño del medio durante el slot siguiente. Por lo que el tiempo se puede ver como pequeños periodos intercalados por el dominio *A* o *B*, dicho de otra forma, periodos par o impar. Si un nodo desea transmitir, primero debe verificar si le pertenece o no el medio, para ello debe saber si en la línea del tiempo se encuentra en un periodo par o impar. La manera en la que el nodo puede saber si se encuentra en un periodo par o impar sobre la línea del tiempo se describe en el siguiente algoritmo:

```
Mac802_11 :: is_allow(double slot)
{
double now = Scheduler::instance().clock();
if ((now % slot) == (now % 2 * slot) && (! (addr() % 2)))
    return 1;
if ((now % slot) != (now % 2 * slot) && (addr() % 2))
    return 1;
return 0;
}
```

Esta función miembro de la clase `Mac802_11()` llamada `is_allow()`, toma como argumento formal el tamaño del `slot`, y después llama al Handler Scheduler con el fin de que le proporcione el tiempo actual en que se encuentra la simulación, el valor regresado de Scheduler es asignado a la variable miembro `now`, la cual se operará con el argumento formal de la función (`slot`) utilizando la función módulo tanto para el tamaño del slot como para su doble.

Posteriormente se igualarán estos valores con el fin de saber a cual dominio le pertenece el canal. Si ambos valores son iguales, se define que está en un periodo par, por el contrario, si son valores distintos, se define que está en un periodo non. Para ejemplificar esta función véase el siguiente caso:

Si un nodo desea transmitir a los 15,67 segundos, este valor es almacenado en la variable miembro `now`; el tamaño del slot definido por el usuario fue de 10 segundos. Esto quiere decir que cada 10 segundos habrá un cambio de dominio, si la simulación comienza con un periodo par; para 15,67 segundos el dominio a transmitir debe ser impar, debido a que está en el periodo que comprende de 10 a 20 segundos. Para que el nodo sepa en que periodo se encuentra es necesario comparar el valor devuelto de la operación `now % slot` y `now % 2 * slot`. Esto es: $15,67 \% 10$ es 5,67 y $15,67 \% 20$ es igual a 15,67, al igualar ambos resultados se observa que 5,67 es diferente de 15,67 exactamente por un periodo de slot, esto quiere decir que se encuentra en un periodo impar, si ambos resultados hubieran sido iguales se sabe que pertenecen a un periodo par. Después sólo resta verificar si el nodo es par o impar, con el fin de permitirle o no que escuche el canal.

La próxima función a definir es `checkBackoffTimer()`, esta función realiza lo siguiente: verifica dos casos. Primero, si el medio se encuentra libre y la cuenta regresiva del backoff está detenida, y además es permitido que el nodo escuche el canal, entonces restablece la cuenta regresiva de penalizaciones. Segundo, si el medio no está libre y la cuenta regresiva del backoff está disminuyendo, se detendrá la cuenta regresiva. Esto corresponde al hecho de que el MAC sólo permite disminuir slots de penalización cuando el medio esté libre. Tal cual está definido en el estándar 802,11; no puede haber decrementos de backoff mientras otro nodo ocupe el canal.

```
Mac802_11:: checkBackoffTimer()
{
if(is_idle() && mhBackoff_.paused() && is_allow()) {
    mhBackoff_.resume(phymib_.getDIFS());
}
if(! is_idle() && mhBackoff_.busy() && ! mhBackoff_.paused()) {
    mhBackoff_.pause();
}
}
```

El objeto `mhBackoff_` pertenece a la clase de temporizadores. Los cuales tienen la capacidad de detener el tiempo del contador, restablecerlo, actualizarlo o simplemente observarlo. Es a través de estos comportamientos que se puede manipular el estado del contador. Con el uso de la función `is_allow()`, se puede asegurar que ningún nodo que no le pertenezca el canal puede acceder a él. A su vez garantiza que ningún nodo que no le pertenece el medio puede disminuir su cuenta regresiva o penalización.

La siguiente función a definir es `navHandler()`, esta función es llamada por el temporizador `NavTimer` cuando la transmisión atómica ha terminado y el medio puede ser escuchado por las terminales. Este evento es tratado de la siguiente forma:

```

Mac802_11::navHandler()
{
if(is_idle() && mhBackoff_.paused() && is_allow())
    mhBackoff_.resume(phymib_.getDIFS());
}

```

La función miembro `navHandler()` verifica si el medio está libre, y que la cuenta regresiva `backoff` esté detenida, dado que el canal había sido silenciado a través de la portadora virtual NAV. Además, verifica si el nodo puede acceder al medio, si esto es cierto, restablece la cuenta regresiva del `backoff` sumándole un periodo DIFS, debido a que después de una transmisión exitosa, todos los nodos deben esperar un periodo DIFS para tratar de acceder nuevamente al medio. El uso de la función `is_allow()` garantiza que únicamente las terminales que les pertenece el medio puedan actualizar su temporizador de penalización después de que el NAV expiró.

La siguiente función a definir es la función miembro `navSHandler()` que tiene a su cargo las transiciones de slots de pertenencia del canal y como consecuencia debe manejar las siguientes situaciones:

- Si el medio está libre y la terminal está reduciendo su cuenta regresiva de penalización, y además el medio ya no le pertenece, debe detener su cuenta regresiva, ya que en este instante el tiempo de pertenencia del canal ha terminado.
- Si el medio está libre y su cuenta regresiva está suspendida, y el canal le pertenece, entonces debe restablecer su cuenta regresiva. Esto es debido a que su slot de pertenencia ha comenzado.
- Independientemente de la situación del nodo, éste debe actualizar la duración del siguiente periodo de slot, con el propósito de que al término de éste, el evento vuelva a ocurrir.
- Si un nodo está transmitiendo en el cambio de slot, no hay necesidad de verificar el `backoff` de las terminales, ya que el canal está reservado por la portadora virtual NAV y la función `navHandler()` se encarga de seguir el correcto funcionamiento del protocolo.

El código de la función se presenta a continuación:

```

Mac802_11::navSHandler()
{
if(is_idle() && mhBackoff_.busy() && ! mhBackoff_.paused() && ! is_allow()){
    mhBackoff_.paused();
}
if(is_idle() && mhBackoff_.busy() && mhBackoff_.paused() && is_allow()) {
    mhBackoff_.resume(0.0);
}
set_navs(slot);
}

```

La función `navSHandler()` hace uso de la función `set_navs(slot)`, con el fin de establecer el valor del slot en el temporizador, en el momento que este tiempo expire, el evento llamará de nuevo a la función `navSHandler()`. Se puede observar que la función `resume()` del objeto `mhBackoff_` toma como argumento formal el valor de 0,0, debido a que en esta ocasión no existe la necesidad de actualizar la cuenta regresiva con un DIFS. Solamente se necesita continuar con la cuenta regresiva que fue detenida en el slot anterior. La función `set_navs(slot)` se presenta a continuación.

```
inline void set_navs(u_int16_t us) {
double now = Scheduler::instance().clock();
double t = us * 1e-3;
if ((now + t) > navs_) {
    navs_ = now + t;
    if(mhNavs_.busy())
        mhNavs_.stop();
    mhNavs_.start(t);
}
}
```

La siguiente función a definir es la función miembro `send()` que tiene a cabo las siguientes tareas:

Primero, verifica el modelo de energía, y tira el paquete si el nodo se encuentra en modo de ahorro de energía. Después llama a las funciones `sendDATA()` y `sendRTS()`, las cuales construyen los encabezados para el paquete de datos y el paquete RTS. Estos paquetes son almacenados en `pktTx_` y `pktRTS_` respectivamente. Seguidamente el encabezado del paquete de datos es asignado con un número único con respecto al nodo, esto con el fin de darle seguimiento a los paquetes que van almacenándose en el buffer del nodo. Luego el MAC verifica su backoff, si el backoff no está activo, entonces el nodo verifica si el canal está libre y si además se le permite transmitir; si es así, el nodo toma un valor aleatorio en la ventana actual y llamará a un temporizador para asignarle un DIFS + número de slots seleccionados. Por otro lado si el medio se encuentra ocupado el nodo deberá comenzar un nuevo backoff. En este punto la función `send()` ha terminado y sólo espera a que alguno de los temporizadores expiren y llame a la función correspondiente que trata el evento (`deferHandler` o `backoffHandler`). Es importante ver que el modelo de energía sólo tira los paquetes que se envían hacia el nodo cuando éste está en estado de ahorro de energía; sin embargo, el nodo tienen la capacidad de generar nuevo tráfico independientemente del dominio que posea el medio. Con el uso de la función `is_allow()`, nuevamente se garantiza que únicamente los nodos que posean el medio pueden acceder a él, mientras los nodos que no les pertenezca el medio deben almacenar sus paquetes en su buffer. El código de la función `send()` se muestra a continuación:

```
void
```

```

Mac802_11::send(Packet *p, Handler *h)
{
double rTime;
struct hdr_mac802_11* dh = HDR_MAC802_11(p);

EnergyModel *em = netif_->node()->energy_model();
if (em && em->sleep()) {
    em->set_node_sleep(0);
    em->set_node_state(EnergyModel::INROUTE);
}

callback_ = h;
sendDATA(p);
sendRTS(ETHER_ADDR(dh->dh_ra));

/*
 * Assign the data packet a sequence number.
 */
dh->dh_scontrol = sta_seqno++;

/*
 * If the medium is IDLE, we must wait for a DIFS
 * Space before transmitting.
 */
if(mhBackoff_.busy() == 0) {
    if(is_idle() && is_allow()) {
        if (mhDefer_.busy() == 0) {
            /*
             * If we are already deferring, there is no
             * need to reset the Defer timer.
             */
            rTime = (Random::random() % cw_)
                * (phymib_.getSlotTime());
            mhDefer_.start(phymib_.getDIFS() + rTime);
        }
    } else {
        /*
         * If the medium is NOT IDLE, then we start
         * the backoff timer.
         */
        mhBackoff_.start(cw_, is_idle() && is_allow());
    }
}
}

```

4.2.1. Implementación de la variante de la propuesta principal (Offset)

La única función que se necesita modificar para lograr el desfase de los paquetes generados durante un slot que no les pertenece el medio es la fun-

ción miembro `send()`. En ella se debe utilizar un temporizador que desfase una longitud de tiempo de un slot, de tal manera que distribuya los paquetes generados durante el slot que sí les pertenece el medio. Al término del temporizador, el `DeferHandler` deberá tratar el evento y verificará si el medio está libre o está ocupado. La función miembro `send()` con offset se presenta a continuación:

```
void
Mac802_11::send(Packet *p, Handler *h)
{
    double rTime;
    struct hdr_mac802_11* dh = HDR_MAC802_11(p);

    EnergyModel *em = netif_->node()->energy_model();
    if (em && em->sleep()) {
        em->sct_node_sleep(0);
        em->set_node_state(EnergyModel::INROUTE);
    }

    callback_ = h;
    sendDATA(p);
    sendRTS(ETHER_ADDR(dh->dh_ra));

    /*
     * Assign the data packet a sequence number.
     */
    dh->dh_scontrol = sta_seqno_++;

    /*
     * If the medium is IDLE, we must wait for a DIFS
     * Space before transmitting.
     */
    if(mhBackoff_.busy() == 0) {
        if(is_idle() && is_allow()) {
            if (mhDefer_.busy() == 0) {
                /*
                 * If we are already deferring, there is no
                 * need to reset the Defer timer.
                 */
                rTime = (Random::random() % cw_)
                    * (phymib_.getSlotTime());
                mhDefer_.start(phymib_.getDIFS() + rTime);
            }
        } else if ( ! is_idle && is_allow()) {
            /*
             * If the medium is NOT IDLE but is allow, then we start
             * the backoff timer.
             */
            mhBackoff_.start(cw_, is_idle() && is_allow());
        }
    } else {
        /*
         * If NOT ALLOW, then we start de defer timer.
         */
    }
}
```

```
    rTime = (Random::random() % cw_)
            * (phymib_.getSlotTime());
    mhDefer_.start(slot + rTime)
    }
}
}
```

Este conjunto de funciones son las primordiales para el correcto funcionamiento de la propuesta de tesis. En el siguiente capítulo se evaluará la propuesta con el fin de encontrar un slot de tiempo que maximice el throughput del sistema.

Capítulo 5

Resultados.

En este capítulo se muestran los esquemas de simulación y pruebas realizadas sobre el simulador NS-2, con el propósito de evaluar el desempeño de la propuesta de tesis y comparar los resultados con respecto al comportamiento del estándar base 802.11.

5.1. Esquema de simulación

Para generar el escenario de simulación que evalúe el desempeño del acceso al medio, se requiere eliminar la sobrecarga de paquetes causada por el protocolo de ruteo, con el fin de conocer el máximo throughput que la red es capaz de transportar. Es por ello que se utilizó el protocolo llamado No Adhoc routing (NOAH) [26], este protocolo evita la búsqueda de rutas y únicamente hace una conexión directa entre dos nodos. Para su uso es necesario añadir la librería y volver a compilar el proyecto NS-2. Después, se añade la siguiente línea en el archivo `Otcl` (entrada del sistema):

```
set val(rp) NOAH ;# routing protocol
```

Donde `val(rp)` definen el tipo de protocolo `adhocRouting` que se desea ocupar en la configuración del nodo móvil. Bajo la misma premisa de evaluar el desempeño en la capa MAC, no es necesario que los nodos tengan una velocidad de movimiento, y es por ello que se define el parámetro `<speed>` con un valor igual a cero. Dentro del escenario de simulación se define el radio máximo de transmisión que los clientes de la estación base son capaces de alcanzar. Esta distancia es de $250m$; que es un parámetro definido en el archivo `~ns/mobile/antenna.h` y establece los límites de la topografía del escenario en la configuración del archivo `Otcl`. Esto se hace con el fin de evitar la aparición de nodos ocultos y terminales expuestas.

```
set val(x) 500
set val(y) 500
```

```

.
.
.

# set up topography object
set topo      [new Topography]

$stopo load_flatgrid $val(x) $val(y)

```

Para la evaluación del desempeño en el estándar base, así como en la propuesta de tesis, es necesario tomar los valores establecidos en el estándar 802.11 para los diferentes tipos de paquetes:

Tamaño del paquete	8184 bits
MAC header	272 bits
PHY header	128 bits
ACK	112 bits + PHY header
RTS	160 bits + PHY header
CTS	112 bits + PHY header
Bit Rate del Canal	12 $\frac{Mbit}{s}$
BER	$1e^{-5}$
Retardo de propagación (δ)	1 μs
Slot Time	20 μs
SIFS	10 μs
ACK.Timeout	250 μs
CTS.Timeout	250 μs
CW_mín	32 slots
CW_máx	1024 slots

Tabla 5.1: Parámetros usados en DSSS

Se definió el máximo tamaño del paquete (MPDU) en 8184 bits, debido a que es una de las cuatro posibilidades de tamaño de paquete especificado en la técnica FHSS y es el tamaño máximo especificado para la técnica DSSS. Finalmente se consideró que un periodo de 60 segundos de simulación es suficiente para evaluar el desempeño del estándar, dado que es un tiempo donde la variación entre los diferentes valores de throughput es mínimo. El código correspondiente en el archivo Otcl para detener la simulación es:

```

for {set i 0} {$i < $val(nn)} {incr i} {
    $ns_ at 60.0 "$node_($i) reset";
}
$ns_ at 60.0 "stop"
$ns_ at 60.01 "puts \"NS EXITING...\" ; $ns_ halt"

```

Finalmente, el escenario de simulación se desarrolló bajo las siguientes condiciones:

- El Punto de Acceso se localiza en las coordenadas (0,0) de la topología definida.
- El Punto de acceso es el centro de las comunicaciones y tiene un radio de vista de 250m.
- La línea de vista entre los clientes y el AP es directa, es decir, no hay protocolo de ruteo.
- Todos los clientes envían paquetes únicamente al AP.
- Todos los clientes se localizan en el radio de transmisión del AP.
- La velocidad de los clientes es cero $\frac{m}{s}$.
- Se utilizan los paquetes de control RTS y CTS en cada transmisión.
- Se utiliza un generador de tráfico CBR.

5.2. Esquema de pruebas

Con el fin de comparar el desempeño de la propuesta de tesis contra el desempeño del estándar base, es necesario cuantificar los parámetros más predominantes en una red:

- El *throughput*: Se presenta como un promedio global entre las estaciones y además se presenta normalizado. Esto es, dividir el *throughput* total entre la tasa de transferencia de la red.
- El retardo promedio de acceso: Se mide desde el momento que el paquete es generado por el nodo transmisor hasta su llegada a la capa MAC del nodo receptor.
- El número de colisiones por segundo: Se presenta un promedio global de todas las estaciones en contención.

Para el cálculo del *throughput*, basta con realizar un archivo en el lenguaje *awk* para contabilizar el número de paquetes recibidos dentro de la estación base en un periodo dado. Esto se logra mediante el archivo de trazado generado durante la simulación en NS-2. Se sabe que el archivo de trazado `{nam}` sigue el formato que a continuación se muestra:

```
[event type] -t[time] -s[src node] -d[dst node] -p[pkt type] -e[pkt size]
-c[color] -i [pkt id] -a[flow id] -x{[src.port] [dst.port]}
```

Por ejemplo, si se desea extraer información del archivo `file.nam` para calcular el número de bytes de tráfico CBR/UDP en el nodo cero, se debe ejecutar el siguiente comando:

```
awk '$1=="r" && $7==0 && $17==2 && $9=="cbr"{b += $11}END{print b}'file.nam
```

Donde el comando `$1 == "r"` filtra el tipo de evento "r" para paquetes recibidos. El comando `$7 == 0` filtra los paquetes si su destino es el nodo cero. El comando `$17 == 2` filtra los paquetes del tipo 2 o CBR. Finalmente el comando `$9 == "cbr" {b + $11}` realiza la suma del número de bytes y los asigna a la variable *b*.

Para el cálculo del retardo promedio de acceso, se hace uso del identificador de paquete [pkt id]. Este identificador es único para cada paquete, por lo que se puede seguir línea a línea el seguimiento del paquete en el archivo de trazado. Esto es, desde el evento "s" que es la fuente o puesta del paquete en el buffer del nodo, hasta el evento "r", que es el instante en el cual el paquete llega a su destino. Con estos dos tiempos sólo resta hacer una diferencia y obtener el tiempo que le tomó al paquete ser enviado.

Finalmente, para el cálculo del número de colisiones se debe habilitar el trazado en la capa MAC con la siguiente línea en el archivo `Otel`

```
set MacTrace          ON
```

Después, con la ayuda del lenguaje *awk* se cuenta el número de veces que aparece la etiqueta COL dentro del archivo `{.tr}`:

```
{
if ( reason == "COL" )
    total_collisions++;
}
```

Una vez definido el conjunto de herramientas para evaluar los diferentes parámetros del desempeño en una red, es momento de establecer las variables de simulación que permitan hacer una comparación entre la propuesta y el estándar base. La técnica de slots de tiempo requiere de la definición del tamaño del slot. Para ello se realizaron pruebas desde *10ms*, *11ms*, ... *1s*, *10s*, *20s*, ..., etc. Sin embargo, únicamente se presentan las pruebas de *13ms*, *100ms* y *1s*, por situaciones de espacio. Además, este grupo de valores es un conjunto representativo de las pruebas realizadas, ya que los demás valores de slots fluctúan entre los valores seleccionados. La segunda variable es el número de nodos dentro de la red; para ello, se considero que grupos de 10, 25 y 50 nodos representan una carga ligera, normal y abundante respectivamente; La tercera variable, es el número de paquetes que cada nodo inyecta en el tiempo a la red o también llamada carga (G), por lo que se varió el número de paquetes por nodo desde 2, 4, 8, 16, 32, 64, 128, 256 y 512 paquetes por segundo y por nodo.

Las pruebas realizadas fueron:

1. Cálculo del throughput para 10 nodos con periodos de slots de *13ms*, *100ms* y *1s*

2. Cálculo del throughput para 25 nodos con periodos de slots de 13ms, 100ms y 1s
3. Cálculo del throughput para 50 nodos con periodos de slots de 13ms, 100ms y 1s
4. Cálculo del retardo promedio para 10 nodos con periodos de slots de 13ms, 100ms y 1s
5. Cálculo del retardo promedio para 25 nodos con periodos de slots de 13ms, 100ms y 1s
6. Cálculo del retardo promedio para 50 nodos con periodos de slots de 13ms, 100ms y 1s
7. Cálculo del número de colisiones por segundo para 10 nodos con periodos de slots de 13ms, 100ms y 1s
8. Cálculo del número de colisiones por segundo para 25 nodos con periodos de slots de 13ms, 100ms y 1s
9. Cálculo del número de colisiones por segundo para 50 nodos con periodos de slots de 13ms, 100ms y 1s

Para evaluar la propuesta que realiza un offset en la generación de paquetes, se realizaron las siguientes pruebas:

1. Cálculo del throughput para 25 nodos con periodos de 13ms, 100ms y 1s
2. Cálculo del throughput para 50 nodos con periodos de 13ms, 100ms y 1s

5.3. Resultados

La Figura 5.1 muestra la gráfica de throughput normalizado-carga para una red con diez terminales en contención. Donde el dominio *A* contiene cinco terminales y el dominio *B* tiene el resto. La gráfica muestra que tanto para el slot de 13ms, 100ms y 1s el throughput de la red está por encima del protocolo base; sin embargo, la diferencia entre el throughput del protocolo base contra la propuesta de tesis no es significativa (alrededor de 2%), debido a que la cantidad de terminales no es grande y por consiguiente la probabilidad de transmisión es alta. Se puede ver que el throughput más alto es obtenido por el slot de 13ms, esto se debe a que la cantidad de paquetes acumulados en slots pequeños son pocos y el algoritmo de retraso es capaz de administrarlos sin la necesidad de incrementar su ventana de penalización.

La Figura 5.2 muestra la gráfica de throughput normalizado-carga de una red con 25 terminales. Nuevamente se puede observar que el slot más pequeño obtuvo el mejor throughput. La gráfica muestra que la propuesta de tesis sobrepasa el throughput del protocolo base. Sólo que para este caso la diferencia

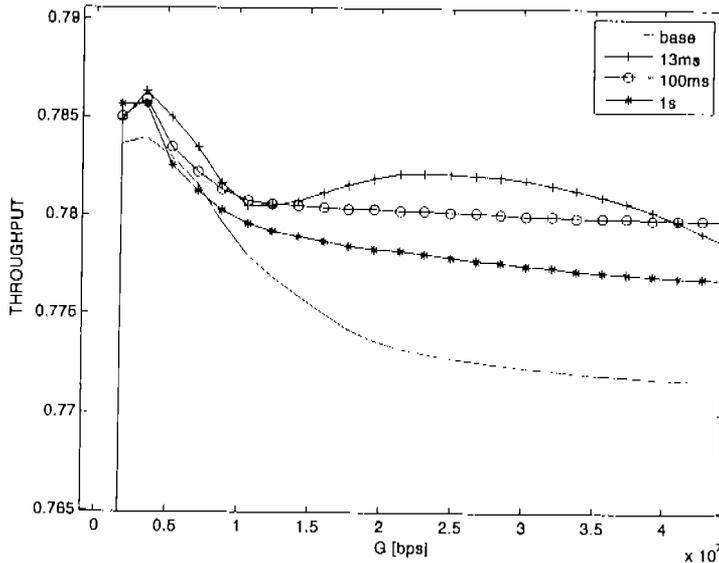


Figura 5.1: Throughput para 10 nodos

de throughput ya es significativa (alrededor de 15%). También se puede observar que para slots de más de 100ms el throughput no decae en comparación con el slot de 13ms, esto se debe a que la carga de la red aún es moderada, y slots grandes aún no almacenan una cantidad grande de paquetes que colisionen al momento del cambio de slot.

La Figura 5.3 muestra la gráfica de throughput normalizado-carga para una red con 50 terminales. La gráfica muestra que la eficiencia del estándar base cae precipitadamente alrededor del 30% con respecto a la técnica de slots. Se puede observar que mientras el uso de slots de tiempo tiene la capacidad de seguir enviando tráfico en la red el estándar base prácticamente tiene un throughput de cero. Nuevamente el slot de 13ms obtiene el mejor throughput en la red.

La Figura 5.4 muestra la gráfica de colisiones-carga para una red con 10 nodos. Se puede ver que a pesar de que la red contiene pocas terminales, existen colisiones. Que al ser tratadas a través de la propuesta de tesis, está las minimiza al rededor del 50%. Como era de esperarse el slot de 13ms tiene el menor número de colisiones, dado que tiene el mayor throughput.

La Figura 5.5 muestra la gráfica de colisiones-carga para una red con 25 terminales. Nuevamente el slot más pequeño obtiene el menor número de colisiones. Se puede ver que la técnica de slots de tiempo reduce en promedio 2000 colisiones por segundo, es esto lo que hace que el protocolo propuesto en esta tesis utilice de forma más eficiente el canal.

La Figura 5.6 muestra la gráfica de colisiones-carga para una red con 50 terminales. El uso de slots de tiempo evita un promedio de 5000 colisiones por segundo, lo que repercute en el throughput del canal. Se puede ver como el protocolo base al manejar 50 nodos minimiza la probabilidad de transmisión

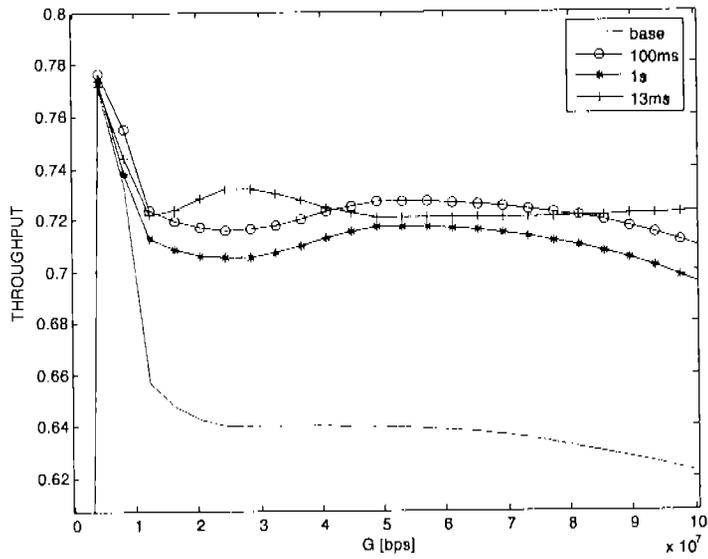


Figura 5.2: Throughput para 25 nodos

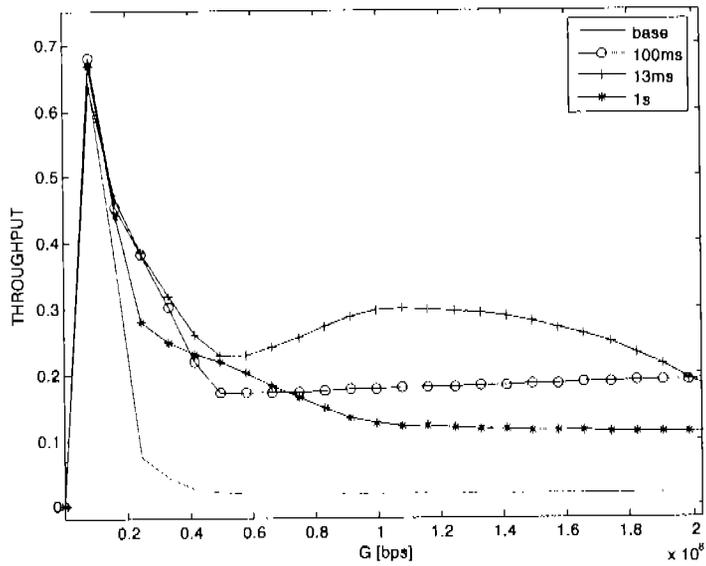


Figura 5.3: Throughput para 50 nodos

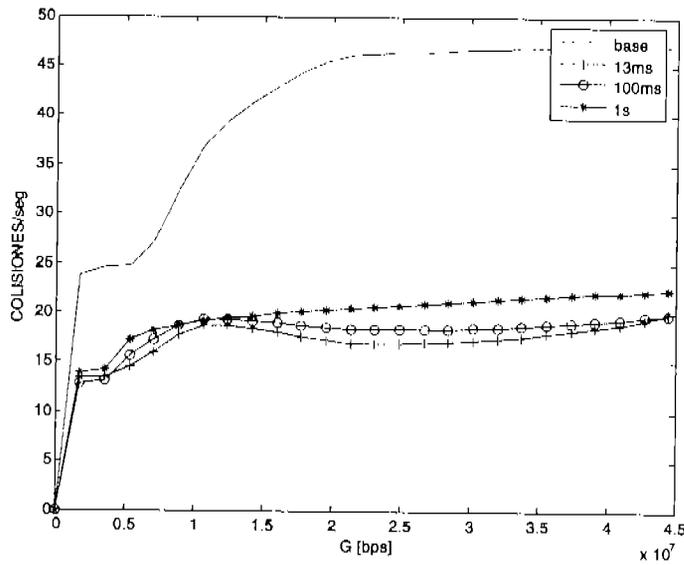


Figura 5.4: Colisiones para 10 nodos

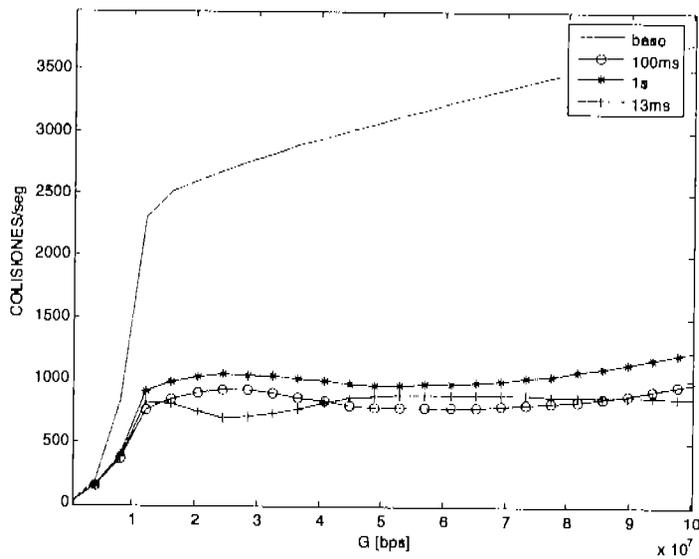


Figura 5.5: Colisiones para 25 nodos

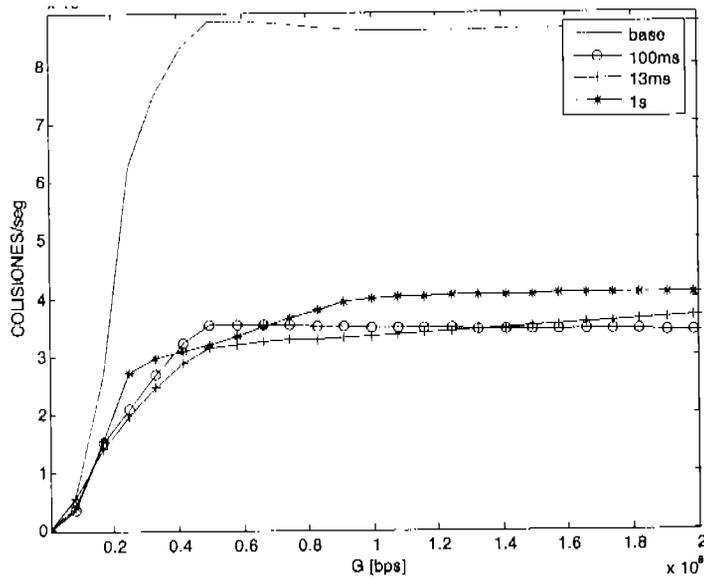


Figura 5.6: Colisiones para 50 nodos

de las estaciones, mientras que el uso de dominios garantiza que el número de colisiones se reduzca exponencialmente, incrementando la probabilidad de transmisión.

La Figura 5.7 muestra la gráfica de retardo (delay)-carga para una red con 10 terminales. Se puede observar que el retardo promedio para un número pequeño de terminales utilizando el protocolo de slots de tiempo tiene pequeñas diferencias con respecto al estándar base.

La Figura 5.8 muestra la gráfica de retardo (delay)-carga para una red con 25 terminales. La gráfica muestra que en promedio existe una diferencia de retardo de alrededor de medio segundo entre la técnica de slots y el estándar base. Si se tiene una red que implemente calidad de servicio, es a partir de este punto que la técnica de slots puede ser considerada para mejorar el tiempo de retardo promedio.

La Figura 5.9 muestra la gráfica de retardo (delay)-carga para una red con 50 terminales. La diferencia de retardo promedio entre el estándar base y la propuesta de tests ya es significativa, dado que existe un promedio de cinco segundos que la técnica de slots puede evitar. Se puede observar como nuevamente el slot que obtuvo el mejor throughput obtiene el menor retardo promedio.

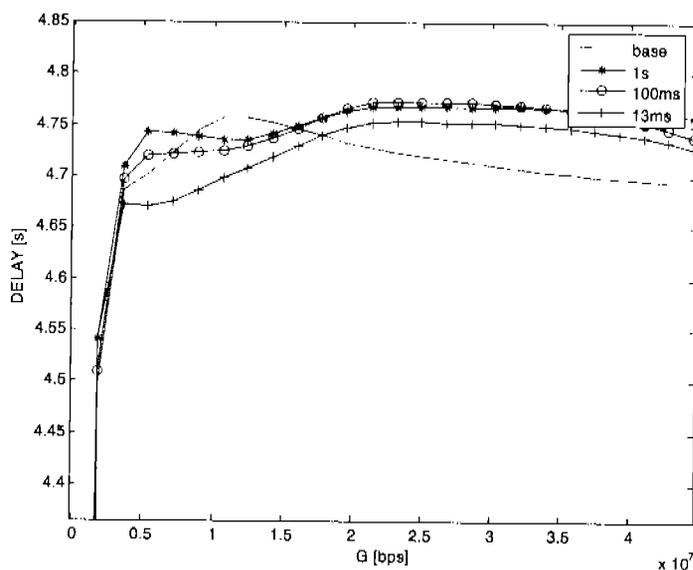


Figura 5.7: Retardo promedio para 10 nodos

5.3.1. Resultados Offset

La Figura 5.10 muestra como el uso de un offset mejora el rendimiento para slots grandes, debido a que distribuye los paquetes generados durante el slot que les pertenece el canal y evita las posibles colisiones al principio del nuevo slot. También se puede ver que para slots pequeños el uso de offset perjudica el throughput del canal, debido a que el número de colisiones al principio del slot no es alto y se desperdicia el uso libre del canal al principio de éste, donde el exponential backoff podría reducir slots de penalización.

La Figura 5.11 muestra nuevamente como el uso de un offset mejora el rendimiento para slots grandes, debido a que distribuye los paquetes y evita las posibles colisiones. Nuevamente se puede ver que para slots pequeños el uso de offset perjudica el throughput del canal. También se puede ver que distribuir los paquetes generados para los nodos que durante un slot no les pertenece el canal, mejora el throughput; sin embargo, no alcanza los valores de eficiencia obtenidos por slots pequeños.

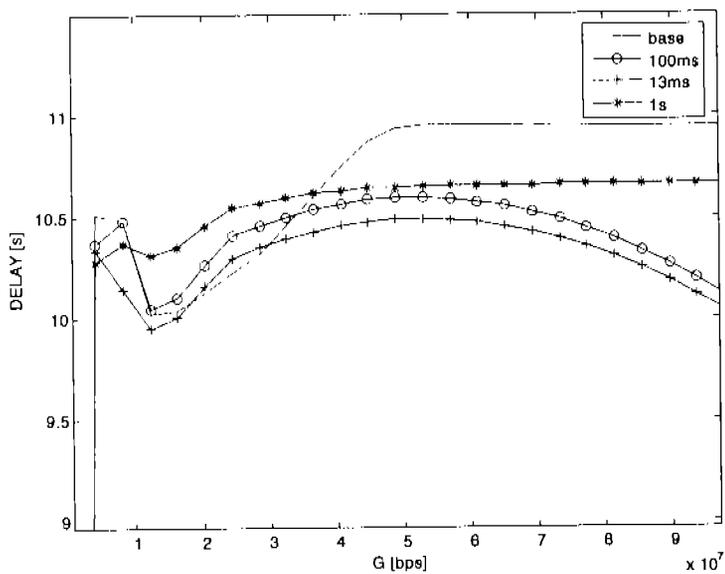


Figura 5.8: Retardo promedio para 25 nodos

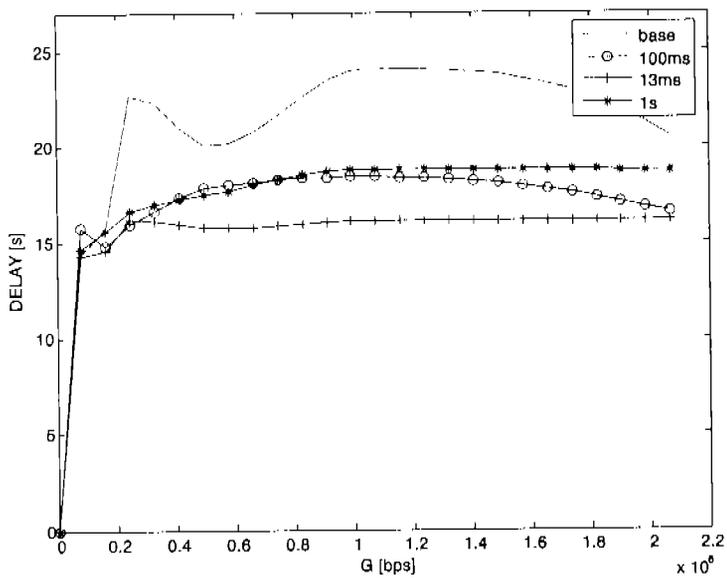


Figura 5.9: Retardo promedio para 50 nodos

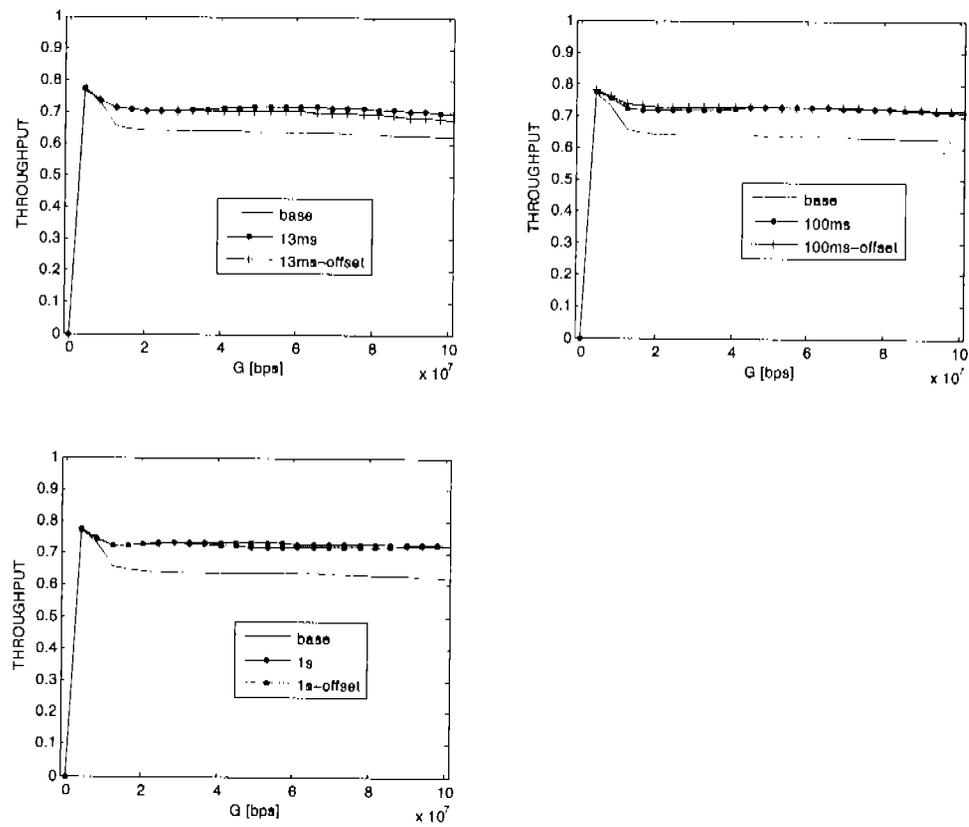


Figura 5.10: Throughput para 25 nodos con offset

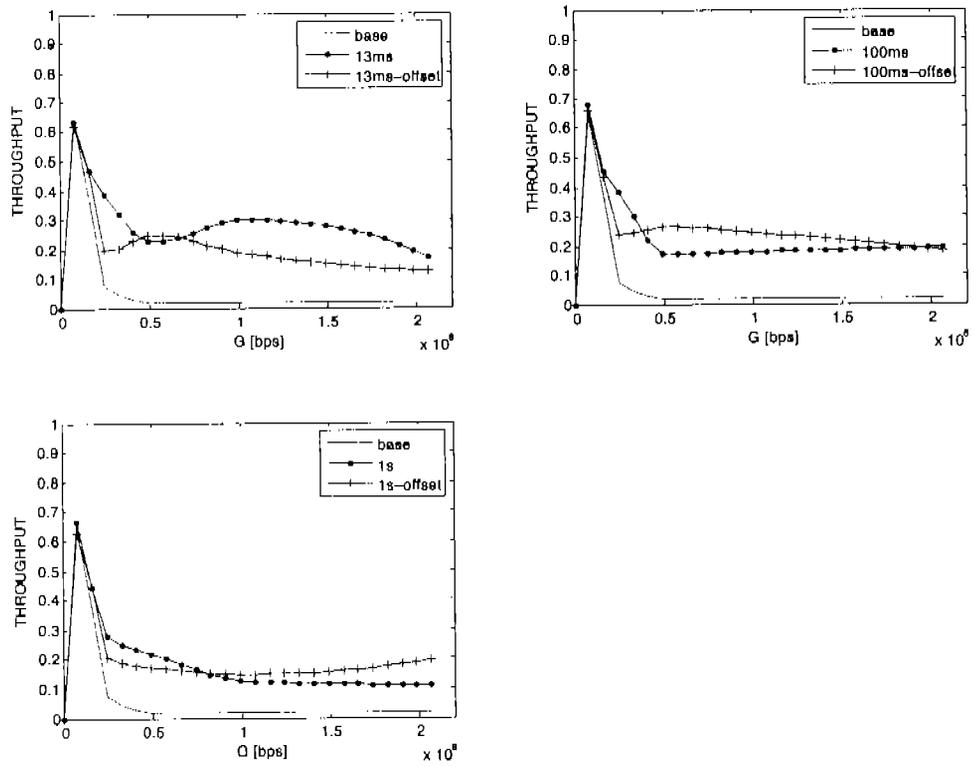


Figura 5.11: Throughput para 50 nodos con offset

Capítulo 6

Conclusiones.

Utilizar la técnica de slots de tiempo propuesta en este trabajo garantiza una alza en el throughput del canal sobre una red inalámbrica basada en el estándar 802.11 en estado de saturación. Como se vio en el capítulo de pruebas y en el modelo matemático, la eficiencia del canal depende de varios factores; sin embargo, uno de los factores predominantes es el número de terminales en contención. Al dividir la cantidad de terminales que desean acceder al medio en dos grupos llamados dominio *A* y dominio *B*, provoca que la curva de throughput-carga tienda a regresar a su contraparte lineal, y de esta manera el uso del canal se ve incrementado, permitiendo que la cantidad de paquetes correctamente recibidos aumente en gran porcentaje. Las pruebas muestran que para cargas grandes de tráfico en el estándar base, la curva throughput-carga tiende a disminuir precipitadamente, mientras al utilizar la técnica de slots de tiempo el throughput de la red tiende a aumentar hasta un 35% por encima del protocolo base. La técnica de slots de tiempo garantiza que el número promedio de colisiones disminuye alrededor del 50% con respecto al protocolo base, de hecho, es de esta manera que la técnica de slots de tiempo logra una mayor utilización del medio al evitar un porcentaje grande de colisiones. Además, uno de los problemas inherentes en los dispositivos inalámbricos es el uso de las baterías. Por lo que la utilización de la técnica de slots de tiempo reduciría alrededor de un 50% el consumo de la batería, dado que la mitad del tiempo los nodos no tienen la necesidad de enviar paquetes ni de escuchar el canal.

Las pruebas realizadas muestran que tamaños de slots pequeños producen una alza significativa en el throughput del canal, debido a que para slots pequeños la cantidad de paquetes acumulados durante el slot son pocos. Esto permite que el algoritmo de retraso binario maneje esa cantidad de paquetes sin la necesidad de aumentar la ventana de penalización a valores muy grandes. Por otro lado, se puede ver que slots por encima de 100ms mejoran el desempeño del sistema en comparación con el estándar base, pero no alcanzan los valores esperados por el modelo matemático. Esto se debe al hecho de que slots grandes almacenan una cantidad grande de paquetes y al momento de acceder al medio la probabilidad de colisión aumenta indiscutiblemente al principio del slot. Debido a este escenario, se implementó una variación del

protocolo principal que produce un desfase para los nodos que no les pertenece el medio y generan paquetes, de tal manera que los paquetes generados durante el tiempo que no les pertenece el canal sean desfasados exactamente un slot de tiempo, permitiendo distribuir de manera más eficiente el acceso al medio al evitar las colisiones en el cambio de slot o pertenencia del canal. Las pruebas hechas sobre el protocolo que ocupa el offset, reflejan un ascenso en el throughput de la red; sin embargo, no alcanzan el valor de eficiencia obtenido con slots pequeños (slots de alrededor de 13ms).

Los resultados en este trabajo tanto para el modelo matemático como para la simulación en NS se realizaron suponiendo la inexistencia de problemas como: interferencia de redes vecinas, uso de protocolos de ruteo, el problema de la terminal oculta y expuesta y la sincronización de tiempos, esta última parte es de suma importancia en los sistemas distribuidos. En una situación real se debe tomar muy en cuenta el costo que implica la sincronización y la elección del mejor protocolo que sincronice las terminales con el mínimo de señalización posible, para el correcto aprovechamiento del canal. Por tanto los valores obtenidos en las simulaciones representan los máximos alcanzables en condiciones idóneas en una red del tipo 802.11.

El uso de técnicas de slots aumenta considerablemente el desempeño en una red inalámbrica; sin embargo, este protocolo no garantiza la respuesta inmediata de sistemas que implementan calidad de servicio o es fundamental el paso de mensajes inmediatos, debido a que una terminal que deseara transmitir instantáneamente y en la línea del tiempo no le pertenece el canal, deberá esperar su turno para enviar su paquete. Esto trae como consecuencia un pequeño retardo en el envío, aunque el uso de slots pequeños minimiza este efecto. Como se pudo ver en las pruebas realizadas, la diferencia entre retardos para cargas de tráfico no muy elevadas es relativamente pequeño, mientras para cargas medianas y altas, el uso de slots de tiempo mejora el retardo promedio de acceso.

El uso del protocolo propuesto en este trabajo evita la sobrecarga innecesaria de paquetes de control para el uso del mismo, ya que en el momento de asociarse a la red, el nodo deberá preguntar si es par o impar y que periodo posee el medio en el momento de la asociación. Después de ello no existe ninguna necesidad de enviar paquetes de control que sobrecarguen la red.

A través de las pruebas realizadas se pudo observar que el dividir el tiempo en dos dominios de terminales aumenta considerablemente el desempeño del sistema, por lo que si se sigue el mismo razonamiento para valores de cuatro, ocho, . . . , dominios de tiempo se esperaría un alza significativa en cada división de dominio.

6.1. Trabajos Futuros

1. Dado que el desempeño utilizando slots de tiempo demostró ser superior a la provista por el estándar base, se propone una implementación de la propuesta de tesis en antenas comúnmente encontradas en el mercado.

2. Dada la implementación del punto anterior, se propone una medición del consumo real de energía utilizando la técnica de slots de tiempo
3. Se propone la implementación con cuatro y ocho dominios de tiempo, con el fin de comparar la ganancia de throughput con respecto a dos dominios.

Apéndice A

Script Otcl

A.1. Definición de variables

Un nodo móvil consiste de componentes de red como la capa de enlace (LL), Queues de espera (IFQ), capa de acceso (MAC), etc. Al comienzo de una simulación inalámbrica se necesita definir el tipo de cada uno de estos componentes de red. Además, se necesita definir otros parámetros como el tipo de antena, el modelo de propagación, el tipo de protocolo de enrutamiento empleado por los nodos móviles y algunos otros que a continuación se ilustran:

```
set val(chan)           Channel/WirelessChannel    ;# channel type
set val(prop)           Propagation/TwoRayGround   ;# radio-propagation model
set val(nelink)         Phy/WirelessPhy           ;# network interface type
set val(mac)            Mac/802_11                ;# MAC type
set val(ifq)            Queue/DropTail/PriQueue   ;# interface queue type
set val(ll)             LL                        ;# link layer type
set val(ant)            Antenna/OmniAntenna       ;# antenna model
set val(ifqlen)         50                        ;# max packet in ifq
set val(nn)             99                        ;# number of mobilenodes
set val(rp)             NOAH                      ;# routing protocol
set val(cp)             "cbr-100"                ;#traffic file
set val(sc)             "escena 100-98"           ;#scenario file
set val(x)              100                       ;# scenario length
set val(y)              100                       ;#scenario width
set AgentTrace          ON                        ;# enable AgentTrace
set RouteTrace          OFF                       ;# disable RouteTrace
set MacTrace            ON                        ;#enable MacTrace
```

A.2. Organizador de eventos

Un evento es la ejecución de un procedimiento *tcl* programado para ocurrir en un tiempo determinado. El siguiente ejemplo muestra como se crea un organizador de eventos:

- Crea un organizador de evento:

```
set ns_ [new Simulator]
```

- Programación de un evento:

```
$ns at <time> <event>
<event>: cualquier comando permitido ns/tcl
ejemplo: $ns at 5.0 "finish"
```

- Inicio del organizador:

```
$ns_ run
```

A.3. Creación de la topología

En las simulaciones de NS se necesita definir la topografía de los nodos móviles, la cual se debe hacer antes de crear los nodos. Normalmente se crea una topología plana especificando el largo y ancho de la topografía:

```
set topo [new Topography]
$topo load_flatgrid $val(x) $val(y)
```

donde \$val(x), \$val(y) fueron definidos anteriormente como variables globales.

A.4. Activación de la opción de trazado

Si se activa la opción de trazado en el script OTcl, NS produce uno o más archivos de texto que contienen datos detallados de la simulación. Estos datos pueden ser útiles para el análisis de la simulación o como una entrada a una herramienta de visualización llamada Animador de redes (NAM). Para activar esta opción se utilizan los siguientes comandos:

- Trazado ns:

```
set tracefd [open wireless1-out.tr w]
$ns trace-all $tracefd
```

- Trazado NAM:

```
set namtrace [open wireless1-out.nam w]
$ns_ namtrace-all-wireless $namtrace $val(x) $val(y)
```

los comandos anteriores escribirán en los archivos que se encuentran entre corchetes los detalles de la simulación.

A.5. Configuración y creación del nodo móvil

La siguiente interfase de programación configura un nodo móvil con todos los valores definidos como: protocolo de enrutamiento, pila de red (LL, MAC, IFQ), tipo de canal, topografía, modelo de propagación, así como la activación o desactivación de diferentes tipos de trazado en diferentes niveles (router, mac, agent) para que puedan ser guardados en el archivo de trazado definido anteriormente.

```
$ns_ node-config -adhocRouting $val(rp) \
-llType $val(ll) \
-macType $val(mac) \
-ifqType $val(ifq) \
-ifqLen $val(ifqlen) \
-antType $val(ant) \
-propType $val(prop) \
-phyType $val(netif) \
-channelType $val(chan) \
-topoInstance $topo \
-agentTrace ON \
-routerTrace OFF \
-macTrace ON \
-movementTrace OFF
```

después de haber configurado el nodo móvil, el siguiente paso es crearlo, esto se realiza mediante la siguiente rutina:

```
for {set i 0} {$i < $val(nn) } {incr i} {
set node_($i) [$ns_ node]
$node_($i) random-motion 0 ;# disable random motion
}
```

A.6. Movimiento del nodo

El nodo móvil está diseñado para moverse en una topología de tres dimensiones; sin embargo la tercera dimensión normalmente no es utilizada, es decir, el nodo móvil se considera que se mueve en un terreno plano con el eje Z igual a cero. La posición inicial y los destinos futuros para un nodo móvil pueden ser definidos utilizando los siguientes comandos:

```
$node set X_ <x1>
$node set Y_ <y1>
```

```
$node set Z_ <z1>
$node at $time setdest <x2> <y2> <speed>
```

En el tiempo `$time` el nodo se empezará a mover de su posición inicial (`x1, y1`) hacia el destino (`x2, y2`) con la velocidad especificada.

A.7. Generador de tráfico

Las conexiones de tráfico TCP y CBR pueden ser configuradas entre nodos móviles utilizando un script para generar tráfico. Este script está disponible en el directorio `~ns/indep-utils/cmu-scen-gen` y se llama `cbrgen.tcl`. Este script puede ser usado para crear conexiones de tráfico CBR y TCP entre nodos móviles inalámbricos. Con el propósito de crear un archivo de conexión de tráfico, se necesita definir el tipo de conexión de tráfico (CBR o TCP), el número de nodos y el número máximo de conexiones que van a ser configurados entre ellos, una semilla aleatoria y en caso de conexiones CBR, una tasa de datos cuyo valor inverso se utiliza para calcular el intervalo de tiempo entre los paquetes CBR. La sintaxis para este comando es la siguiente:

```
ns cbrgen.tcl [-type cbr|tcp] [-nn nodes] [-seed seed]
              [-mc connections][rate rate] > [outdir/traffic-file]
```

Ahora se necesita definir el tiempo en que el programa debe detenerse, y decirle a los nodos que restablezcan sus componentes de red internos, esto se realiza con los siguientes comandos:

```
for {set i 0} {$i < $val(nn) } {incr i} {
$ns_ at <time0> "$node_($i) reset";
}
$ns_ at <time1> "stop"
$ns_ at <time2> "$ns_ halt"
proc stop {} {
global ns_ tracefd
close $tracefd
}
```

Donde `time0` es el tiempo especificado para terminar la simulación. Los nodos son restablecidos en ese instante y la función `"$ns_ halt"` se llama en `time2` un poco después de que los nodos se han restablecido. El procedimiento `stop{}` se llama para terminar los trazados y cerrar los archivos.

Apéndice B

Modelo Bidimensional de Markov

En esta sección se presenta un resumen del modelo matemático para las redes basadas en el estándar 802,11 propuesto por G. Bianchi [4].

B.0.1. Modelo Matemático

Se considera un número finito n de estaciones que desean acceder al medio. Se define un rango W de ventana en el cual trabaja el algoritmo *exponential backoff* en el estado i . El estado i se refiere a la condición por la que atraviesa el paquete, es decir, cuantas colisiones ha sufrido y el valor k que le ha asignado el algoritmo de retardo.

$$W_i = 2^i W_0 \quad i \in (0, m) \quad (\text{B.1})$$

W_0 es el valor inicial del *exponential backoff* e i representa el estado del *backoff*, el cual puede variar de 1 hasta m . El modelo bidimensional de cadenas de Markov se muestra en la Figura B.1. Los posibles estados que puede tener un paquete mientras reduce su cuanta regresiva y accede al medio se establecen a continuación:

- Reducción de un slot en la cuenta regresiva.
- Probabilidad de éxito en la transmisión de un paquete.
- Probabilidad de colisión.
- Probabilidad de alcanzar el número máximo de colisiones.

Sea P_{tr} la probabilidad que haya al menos una transmisión en un slot determinado de tiempo. Como existen n estaciones tratando de acceder al medio y cada una transmite con una probabilidad τ

$$P_{tr} = 1 - (1 - \tau)^n \quad (\text{B.2})$$

donde τ es

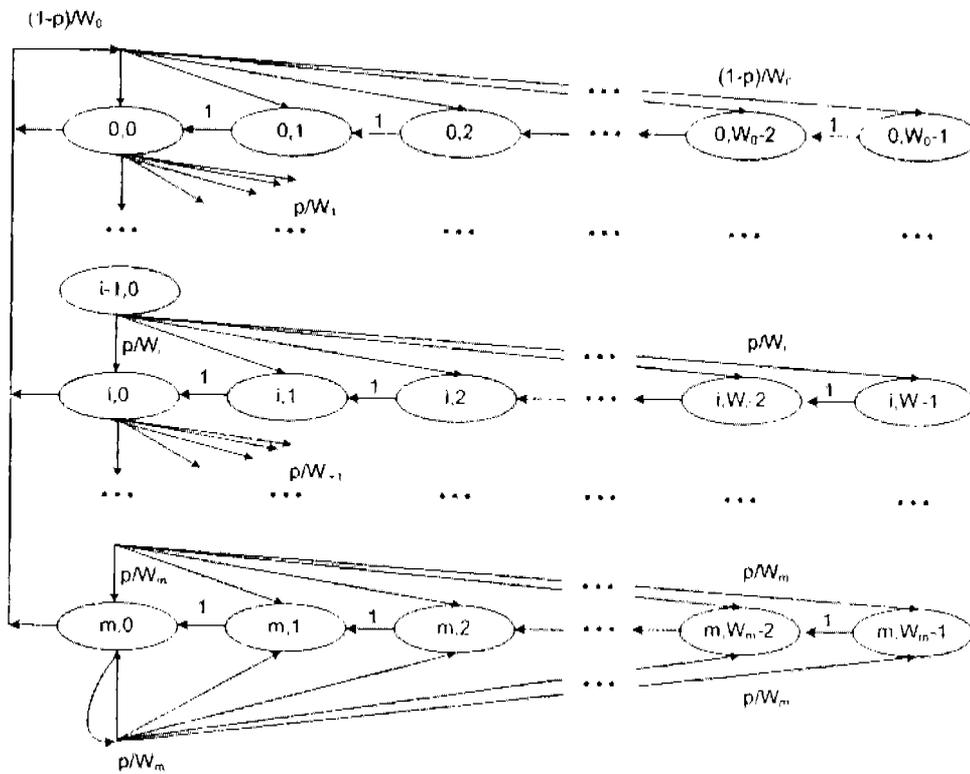


Figura B.1: Cadena de Markov

$$\tau = \frac{2(1-2p)}{(1-2p)(W_0+1) + pW_0(1-(2p)^m)} \quad (\text{B.3})$$

p es la probabilidad de que un paquete transmitido colisione, esta probabilidad está en función de que en un slot de tiempo determinado una estación ya ha transmitido y se espera que al menos una de las $n-1$ estaciones restantes también transmitan.

$$p = 1 - (1-\tau)^{n-1} \quad (\text{B.4})$$

Sea P_s la probabilidad de que una transmisión sea correcta. Esto implica que ninguna otra estación transmitió en el mismo slot de tiempo

$$P_s = \frac{n\tau(1-\tau)^{n-1}}{P_{tr}} = \frac{n\tau(1-\tau)^{n-1}}{1-(1-\tau)^n} \quad (\text{B.5})$$

La cadena bidimensional de Markov tiene las siguientes probabilidades de transición:

1. Al comienzo de cada slot, la cuenta regresiva del algoritmo de retroceso es disminuido siempre y cuando la estación haya escuchado que el canal estuvo libre por un periodo σ (tamaño de un slot), establecido en el estándar 802.11

$$P(i, k|i, k+1) = 1 \quad k \in (0, W_i - 2) \quad i \in (0, m)$$

Esto se puede leer como: la probabilidad de que se encuentre en el estado i con un valor de k en la cuenta regresiva es igual a 1, dado que estuvo en el estado i con un valor del backoff de $k+1$. En otras palabras, la probabilidad de reducir un slot en la cuenta del backoff es del 100% si el canal estuvo libre un tiempo σ . Por ello el valor de $k+1$ es reducido a k .

2. El contador del backoff es congelado cuando la estación escucha que el canal está ocupado.

$$P(i, k|i, k) = P_{tr} \quad k \in (1, W_i - 1) \quad i \in (0, m)$$

3. Después de una transmisión exitosa, el nuevo paquete a enviar, el cual estaba almacenado en el buffer, comenzará su backoff en el estado 0 y tomará un valor k entre $(1, W_0 - 1)$

$$P(0, k|i, 0) = \frac{(1-p)}{W_0} \quad k \in (0, W_0 - 1) \quad i \in (0, m)$$

4. Si existe un colisión en algún estado del backoff, para al menos dos paquetes, el backoff se incrementará, y el nuevo valor será aleatoriamente elegido dentro del rango de la nueva ventana

$$P(i, k|i-1, 0) = \frac{p}{W_i} \quad k \in (0, W_i - 1) \quad i \in (1, m)$$

el hecho de que haya colisionado el paquete, el rango de i no puede empezar en cero, dado que al menos el backoff está en el estado 1.

5. El máximo estado del backoff es m , por lo que éste no será incrementado por subsecuentes transmisiones.

$$P(m, k|m, 0) = \frac{p}{W_m} \quad k \in (0, W_m - 1)$$

Finalmente con el cálculo de la probabilidad τ de que una estación transmita exitosamente, las probabilidades P_{tr} , P_s y p , se puede definir el throughput normalizado como:

$$S = \frac{P_s P_{tr} E[P]}{(1 - P_{tr})\sigma + P_{tr} P_s T_s + P_{tr} (1 - P_s) T_c} \quad (B.6)$$

donde $E[P]$ es la longitud promedio del paquete de datos, T_s es el tiempo promedio que el canal está ocupado debido a una transmisión exitosa. T_c , es el tiempo promedio que el canal se encuentra ocupado durante una colisión. Tanto T_c como T_s depende del mecanismo de acceso al medio, ya sea el acceso básico o el acceso RTS/CTS. La ecuación B.6 calcula el throughput normalizado para un canal ideal, en el cual no existen errores y no se presenta el problema de la terminal oculta y expuesta.

Al añadirse la posibilidad de error en la transmisión de datos a la ecuación B.6, se obtiene una versión más cercana al cálculo del throughput real; a continuación se presentan las ecuaciones de probabilidad para un paquete con error.

$$P_e = 1 - (1 - BER)^{longRTS+longCTS+longDATA+longACK} \quad (B.7)$$

$$P_e^{RTS} = 1 - (1 - BER)^{longRTS} \quad (B.8)$$

$$P_e^{CTS} = ((1 - BER)^{longRTS}) * (1 - (1 - BER)^{longCTS}) \quad (B.9)$$

$$P_e^{DATA} = ((1 - BER)^{longRTS+longCTS}) * (1 - (1 - BER)^{longDATA}) \quad (B.10)$$

$$P_e^{ACK} = ((1 - BER)^{longRTS+longCTS+longDATA}) * (1 - (1 - BER)^{longACK}) \quad (B.11)$$

Las siguientes ecuaciones que se superponen en el tiempo son debidas a transmisiones con errores.

$$T_e^{RTS} = T_{RTS} + T_{CTS}^{timeout} + \delta + DIFS \quad (B.12)$$

$$T_c^{CTS} = T_{RTS} + T_{CTS} + 2\delta + SIFS + DIFS \quad (\text{B.13})$$

$$T_e^{DATA} = T_{RTS} + T_{CTS} + H + E[P] + T_{ACK}^{timeout} + 3\delta + 2SIFS + DIFS \quad (\text{B.14})$$

$$T_e^{ACK} = T_{RTS} + T_{CTS} + H + E[P] + T_{ACK} + 4\delta + 3SIFS + DIFS \quad (\text{B.15})$$

donde $H = PHY_{hdr} + MAC_{hdr}$ son los encabezados del paquete y δ es el tiempo de retardo de propagación en el medio. Finalmente la ecuación que calcula el throughput con errores y colisiones se presenta a continuación

$$S = \frac{P_s P_{tr} (1 - P_e) E[P]}{(1 - P_{tr})\sigma + P_{tr} P_s (1 - P_e) T_s + \phi + P_{tr} (1 - P_s) T_c} \quad (\text{B.16})$$

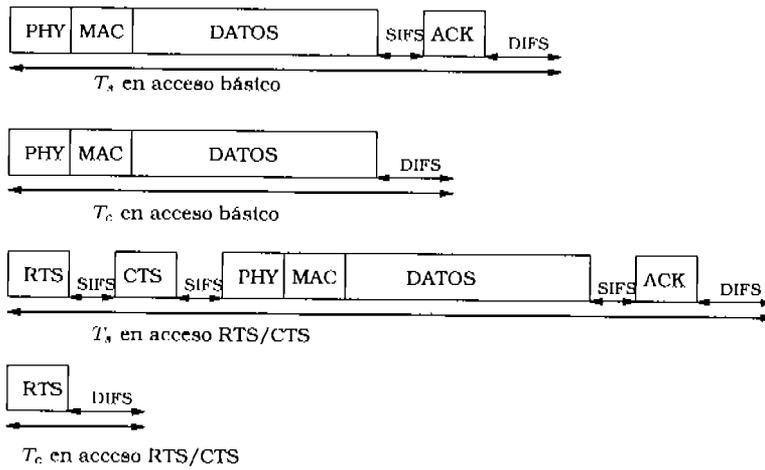
donde ϕ es

$$\phi = P_{tr} P_s (P_e^{RTS} T_e^{RTS} + P_e^{CTS} T_e^{CTS} + P_e^{DATA} T_e^{DATA} + P_e^{ACK} T_e^{ACK})$$

La ecuación B.6 como B.16 son útiles para el cálculo del throughput tanto para el acceso básico como el acceso RTS/CTS. Sólo se necesita utilizar el T_s y T_c correctos, que se muestran en la Figura B.2 y en las ecuaciones B.17, B.18.

$$\begin{cases} T_s^{bas} = H + E[P] + ACK + 2\delta + SIFS + DIFS \\ T_c^{bas} = H + E[P] + \delta + DIFS \end{cases} \quad (\text{B.17})$$

$$\begin{cases} T_s^{RTS} = RTS + CTS + H + E[P] + ACK + 4\delta + 3SIFS + DIFS \\ T_c^{RTS} = RTS + \delta + DIFS \end{cases} \quad (\text{B.18})$$

Figura B.2: T_s , T_c

Bibliografía

- [1] IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, Nov. 1997. P802.11.
- [2] Matthew S. Gast, Network Wireless 802.11, O'REILLY, 2005.
- [3] Andrew S. Tanenbaum, Redes de Computadoras, Cuarta Edición 2003.
- [4] G. Bianchi, "Performance analysis of the IEEE 802.11 distribute coordination function,"IEEE J. Select. Areas Commun., vol. 18, No. 3, Mar.2000.
- [5] G. Bianchi, "IEEE 802.11: saturation throughput analysis,"IEEE Commun. Lett., vol. 2, Dec. 1998.
- [6] Leena Chandran-Wadia, "Throughput Performance of the Distributed and Point Coordination Functions of an IEEE 802.11 Wireless LAN,"Indian Institute of Technology Bombay, Powai, Mumbai - 400 076.
- [7] Craig Sweet, Deepinder Sidhu, "Performance Analysis of the IEEE 802.11 Wireless LAN Standard,"Maryland Center for Telecommunications Research.
- [8] IEEE Draft for Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Medium Access Control (MAC) Enhancements for Quality of Service (QoS), IEEE Std 802.11e/ Draft 11.0, October 2004
- [9] Y. Kwon, Y. Fang, and H. Latchman, "A Novel MAC Protocol with Fast Collision Resolution for Wireless LANs," Proc. Infocom 2003, vol. 2, pp. 853-862, Apr. 2003.
- [10] Chonggang Wang, Bo Li, Senior Member, "A New Collision Resolution Mechanism to Enhance the Performance of IEEE 802.11 DCF,"IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, VOL. 53, NO. 4, JULY 2004.
- [11] C. Wang and W. Tang, "A Probability-Based Algorithm to Adjust Contention Window in IEEE 802.11 DCF," ICCAS: Communications, Circuits and Systems, vol. 1, pp. 418-422, 2004.
- [12] T. S. Ho and K. C. Chen, "Performance evaluation and enhancement of the CSMA/CA MAC protocol for 802.11 wireless LAN's," in Proc. IEEE PIMRC, Taipei, Taiwan, Oct. 1996, pp. 392-396.

- [13] Y. Xiao. Performance Analysis of IEEE 802.11e EDCF under Saturation Condition, Proc. IEEE ICC 2004
- [14] H. Wu, Y. Peng, K. Long, et al.. Performance of Reliable Transport Protocol over IEEE 802.11 Wireless LAN: Analysis and Enhancement, Proc. IEEE INFOCOM 2002
- [15] Gunter Bolch, Stefan Greiner, "Queueing Networks and Markov Chains", John Wiley & Sons, Inc. NewYork 1998.
- [16] "The ns manual", <http://www.isi.edu/nsnam/ns/ns-documentation.html>
- [17] "Ns by example", <http://nile.wpi.edu/NS/>
- [18] "Marc Greis's tutorial", <http://www.isi.edu/nsnam/ns/tutorial>
- [19] "The Network Simulator ns-2: workshops, tutorials, and presentations", <http://www.isi.edu/nsnam/ns/ns-tutorial/index.html>
- [20] "802.11 MAC code in ns-2", <http://www.ece.rice.edu/~jpr/ns/docs/802~11.html>
- [21] Neil Reid, Ron Seide, "802.11 (Wi-Fi) Networking Handbook", Mc Graw Hill, 2003.
- [22] J. Zhao, Z. Guo, Q. Zhang and W. Zhu. Performance Study of MAC for Service Differentiation in IEEE 802.11, Proc. IEEE Globecom 2002
- [23] Byung-Jae Kwak, Nah-Oak Song, Leonard E. Miller, "Analysis of stability and Performance of Exponential Backoff", IEEE 2003.
- [24] Tejinder S. Randhawa, Jeffrey W. Robinson, "Saturation Throughput Analysis of IEEE 802.11e Enhanced Distributed Coordination Function," IEEE J. Areas Commun., vol. 22, No.5, June 2004.
- [25] NS-2, URL <http://www-mash.cs.berkeley.edu/ns/>
- [26] NO Ad-Hoc Routing Agent (NOAH), URL <http://icapeople.epfl.ch/widmer/uwb/ns-2/noah/>