



UNIVERSIDAD NACIONAL  
AUTÓNOMA DE  
MÉXICO

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

---

---

PROGRAMA DE MAESTRIA Y DOCTORADO EN  
INGENIERIA

FACULTAD DE INGENIERÍA

DISEÑO DE RUTAS RÁPIDAS  
EN REDES MULTISALTOS

T E S I S

QUE PARA OPTAR POR EL GRADO DE:

MAESTRO EN INGENIERIA

INGENIERÍA ELÉCTRICA - TELECOMUNICACIONES

P R E S E N T A :

ING. BARRAGÁN CONTRERAS GONZALO



TUTOR:  
DR. JAVIER GÓMEZ CASTELLANOS

2011

**JURADO ASIGNADO:**

Presidente: DR. GARCÍA UGALDE FRANCISCO  
Secretario: DR. MOCTEZUMA FLORES MIGUEL  
Vocal: DR. GÓMEZ CASTELLANOS JAVIER  
1<sup>er</sup>. Suplente: DR. RANGEL LICEA VÍCTOR  
2<sup>do</sup>. Suplente: DR. PASCOE CHALKE MICHAEL

Lugar o lugares donde se realizó la tesis:

CIUDAD UNIVERSITARIA, MÉXICO, D.F.

**TUTOR DE TESIS:**

DR. JAVIER GÓMEZ CASTELLANOS

---

**FIRMA**

# Dedicatoria

A Dios...

A mis padres: Clemente Barragán Luis y Enedina Contreras Pérez; a mis hermanos: Héctor Oscar Barragán Contreras y Jorge Humberto Barragán Contreras, por el amor incondicional que siempre me han dado, porque sin su apoyo esto no hubiera sido posible y porque no me imagino la vida sin ustedes.

A mi amigo y hermano Miguel Ángel Tizapantzi Hernández (Q.E.P.D.) que se nos adelantó en el camino, pero que sigue vivo en cada uno de nosotros.

# Agradecimientos

A la Universidad Nacional Autónoma de México por darme la formación académica y humanística que han hecho de mí lo que ahora soy, un hombre de bien.

A mi tutor el Dr. Javier Gómez Castellanos por darme la oportunidad de trabajar con él, por su comprensión, apoyo incondicional y sobre todo por sus enseñanzas.

A cada uno de mis sinodales por el esfuerzo y tiempo invertido en la revisión de esta tesis. Agradezco y valoro las correcciones realizadas a mi trabajo.

Al Posgrado de Ingeniería y a la Coordinación de Estudios de Posgrado por el apoyo económico que hizo posible mis estudios de maestría.

A los proyectos PAPIIT-106609 y CONACyT-105117.

A mis amigos/hermanos: Oscar Arana, Carlos Montes, Raymundo Tehoyotl, David Monroy, Isabel Reza, Ana Aguilar, Janisse Oviedo, Cristhian Martínez... por estar siempre.

A los que por motivos ajenos a mi no están conmigo, pero que me acompañan a cada paso y que se han vuelto imprescindibles e inolvidables.

## Resumen

En la actualidad las redes inalámbricas multisaltos basadas en el estándar IEEE 802.11 son ampliamente utilizadas en diferentes campos, desempeñando diferentes tareas. A medida que este tipo de redes se han incorporado a nuestras vidas ha sido necesario buscar la manera de otorgar servicios de mayor calidad al usuario final.

El garantizar cierta calidad de servicio en las redes inalámbricas multisaltos no es una tarea sencilla debido a las características propias de este tipo de redes. No obstante, se han propuesto distintas soluciones que buscan mejorar el desempeño de las redes inalámbricas multisaltos.

En esta tesis se hace una propuesta para garantizar calidad de servicio en las redes inalámbricas multisaltos, específicamente para las redes que utilizan el protocolo de capa MAC definido en el estándar IEEE 802.11. En la propuesta se introduce el concepto de rutas rápidas cuyo propósito es disminuir el retardo total de los paquetes de datos que llegan a su destino mediante estas rutas. La creación de las rutas rápidas se logra mediante la modificación del método de acceso al medio utilizado por los nodos de la red. Además, se plantean algunas otras estrategias para lograr disminuir el retardo total de los paquetes.

Asimismo, se describe la implementación de las rutas rápidas y demás mecanismos que componen la propuesta descrita en esta tesis, en un simulador de redes. Finalmente, se muestran y analizan los resultados obtenidos en diferentes escenarios definidos para evaluar el desempeño de la propuesta.

## **Abstract**

Currently, the multi-hop wireless networks based on the standard IEEE 802.11 are widely used in different fields, performing different tasks. As these networks have been incorporated into our lives has been necessary to find a way to provide a higher quality of service to end-users.

To guarantee a certain quality of service in multi-hop wireless networks is not an easy task due to the characteristics of such networks. However, different solutions have been proposed in order to improve the performance of multi-hop wireless networks.

In this thesis a proposal to guarantee quality of service in multi-hop wireless networks is made, specifically for the networks which use the MAC layer protocol defined in the standard IEEE 802.11. The proposal introduces the concept of fast routes aimed to reduce the total delay of the data packets that arrive to their destination via these routes. The creation of the fast routes is accomplished by modifying the medium access method used by the nodes of the network. In addition, other strategies for reducing the total delay of packets are explained.

It also describes the implementation of the fast routes and other mechanisms that compose the proposal described in this thesis, in a network simulator. Finally, are shown and analyzed the results of different scenarios defined to evaluate the performance of the proposal.

# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Definición del problema . . . . .	1
1.2. Hipótesis . . . . .	2
1.3. Objetivos . . . . .	3
1.4. Estructura de la tesis . . . . .	3
<b>2. Antecedentes</b>	<b>4</b>
2.1. Estándar IEEE 802.11 . . . . .	4
2.1.1. Función de coordinación distribuida . . . . .	5
2.2. Calidad de Servicio . . . . .	10
2.3. Resumen de capítulo . . . . .	11
<b>3. Descripción detallada del protocolo de comunicación propuesto</b>	<b>12</b>
3.1. Factores que influyen en el retardo total de los paquetes en una red multisaltos	12
3.2. Las avenidas . . . . .	13
3.2.1. Reducción del retardo de la capa MAC . . . . .	15
3.2.2. Reducción del retardo de encolamiento . . . . .	16
3.3. Avenidas con rampas . . . . .	19
3.3.1. Acceso a la avenida . . . . .	20
3.4. Problema de terminal expuesta . . . . .	22
3.4.1. Modos de operación . . . . .	24
3.5. Bajo flujo de información en la red . . . . .	26
3.6. Resumen de capítulo . . . . .	27
<b>4. Implementación de las avenidas</b>	<b>28</b>
4.1. Simulador NS-2 . . . . .	28
4.1.1. Ejemplo de un <i>script</i> Tcl utilizado en NS-2 . . . . .	29
4.2. Programación de las avenidas y rampas en NS-2 . . . . .	32
4.2.1. Nodos de la avenida . . . . .	33
4.2.2. Nodos rampa . . . . .	35
4.3. Control de la tasa de datos en la avenida . . . . .	38
4.4. Resumen de capítulo . . . . .	38

<b>5. Pruebas y resultados</b>	<b>39</b>
5.1. Experimento 1: Impacto del valor de $CW_{min}$ en el retardo total de los paquetes	39
5.2. Experimento 2: Análisis del desempeño de la avenida . . . . .	42
5.3. Experimento 3: Estudio del tráfico en la avenida al existir aleatoriedad . . . .	45
5.4. Experimento 4: Análisis del desempeño de la avenida con rampas . . . . .	47
5.5. Experimento 5: Análisis del desempeño de la avenida con rampas al existir interferencia . . . . .	48
5.6. Experimento 6: Análisis del desempeño de los nodos rampa al no existir tráfico en la avenida . . . . .	49
5.7. Experimento 7: Análisis del desempeño de la avenida y nodos rampa que contienen por el canal . . . . .	50
5.8. Resumen de capítulo . . . . .	51
<b>6. Conclusiones</b>	<b>52</b>
<b>Glosario de términos</b>	<b>54</b>

# Índice de figuras

1.1.	Red inalámbrica multisaltos. . . . .	2
2.1.	Método de acceso DCF. . . . .	7
2.2.	Procedimiento de <i>backoff</i> . . . . .	7
2.3.	Problema de terminal oculta. . . . .	8
2.4.	Método de acceso DCF con intercambio RTS/CTS. . . . .	9
3.1.	Avenida en una red inalámbrica multisaltos. . . . .	14
3.2.	Segmento de la Calzada de Tlalpan (Imagen obtenida de Google Maps). . . . .	14
3.3.	Detalles de una avenida en una red inalámbrica multisaltos. . . . .	15
3.4.	Intercambio de paquetes entre dos nodos cualesquiera de la avenida con $CW=1$ constante. . . . .	16
3.5.	Nodos que constituyen una avenida. . . . .	17
3.6.	Nodos de la avenida que pueden transmitir de manera simultánea sin interferirse mutuamente. Los nodos $A_2$ , $A_6$ y $A_{10}$ pueden transmitir mientras $A_3$ , $A_4$ , $A_5$ , $A_7$ , $A_8$ y $A_9$ permanecen inactivos. . . . .	18
3.7.	Diagrama de tiempos de la transmisión de un paquete entre el nodo A y B. . . . .	19
3.8.	Nodos de una avenida: $A_1$ - $A_{11}$ ; y nodos rampa: $R_1$ - $R_5$ . . . . .	20
3.9.	Proceso realizado por los nodos rampa para acceder a la avenida. . . . .	22
3.10.	Problema de terminal expuesta. . . . .	23
3.11.	Problema de terminal expuesta con nodos que no pertenecen a la avenida. . . . .	25
4.1.	Topología definida en el script Tcl para la simulación en NS-2. . . . .	29
5.1.	Topología de la red utilizada en NS2 . . . . .	39
5.2.	Paquetes recibidos y retardo total promedio de los paquetes. Nodos privilegiados con una $CW_{min}=3$ . . . . .	41
5.3.	Paquetes recibidos y retardo total promedio de los paquetes. Nodos privilegiados con una $CW_{min}=7$ . . . . .	41
5.4.	Topología de control utilizada en NS2. . . . .	42
5.5.	Escenario de simulación para el protocolo de comunicación propuesto, introducción de la avenida y rampas. . . . .	44
5.6.	Comparación del porcentaje y el retardo promedio de los paquetes que llegan a su destino en la conexión 4. Caso ideal, aleatorio y el del escenario de control . . . . .	45
5.7.	Porcentaje y el retardo promedio de los paquetes de la avenida que llegan a su destino en función de los paquetes por segundo creados por la aplicación. . . . .	46

5.8. Comparación del porcentaje y el retardo promedio de los paquetes que llegan a su destino de la conexión 4. Caso ideal, aleatorio y el escenario de control . . . . .	46
5.9. Comparación del porcentaje y el retardo promedio de los paquetes que llegan a su destino de las conexiones 3, 4 y 5. Caso ideal, aleatorio y el del escenario de control . . . . .	47
5.10. Comparación del porcentaje y el retardo promedio de los paquetes que llegan a su destino de cada una de las conexiones entre el protocolo propuesto y el IEEE 802.11. . . . .	48
5.11. Porcentaje y retardo promedio de los paquetes provenientes de los nodos rampa, que llegan a su destino. . . . .	50
5.12. Comparación del porcentaje y el retardo promedio de los paquetes que llegan a su destino cuando los nodos rampa acceden a la avenida al recibir un paquete de ésta y cuando estos nodos contienden por el canal de manera habitual con una $CW=1023$ . . . . .	51

# 1

## Introducción

### 1.1 Definición del problema

Las redes inalámbricas multisaltos han tenido bastante proliferación debido principalmente a la rapidez y facilidad con la que pueden ser desplegadas, al no depender de una infraestructura fija, y por la gran cantidad de escenarios donde pueden utilizarse. Ésto ha despertado el interés por buscar la manera de otorgar calidad de servicio (*Quality of Service*) (QoS) <sup>1</sup> en estas redes para poder cumplir con las necesidades que demandan ciertas aplicaciones.

Las redes inalámbricas multisaltos, como su nombre lo indica, están formadas por un conjunto de nodos que se comunican mediante múltiples enlaces inalámbricos; donde la comunicación entre ellos queda restringida por el alcance de transmisión. Un nodo puede comunicarse de manera directa con otros nodos, siempre y cuando estos se encuentren dentro de su alcance de transmisión. Con aquellos que no cumplen esta condición, la comunicación se logra vía multisaltos siempre y cuando exista una ruta para llegar a ellos. La ruta está conformada por la secuencia de nodos que establecen la comunicación através de múltiples enlaces.

En las redes inalámbricas multisaltos, cada uno de los nodos componentes de la red funcionan como *routers* que reenvían la información que no esta destinada a ellos mismos. En la Figura 1.1 se ilustra una red inalámbrica multisaltos. Así, si el nodo A deseara enviar un paquete a B lo haría de manera directa. En cambio, si el destino fuera el nodo H, uno de los posibles caminos a seguir sería el siguiente:

$$A \Rightarrow B \Rightarrow C \Rightarrow E \Rightarrow F \Rightarrow H$$

---

<sup>1</sup>Existe un glosario de términos en la página 54 de la presente tesis.

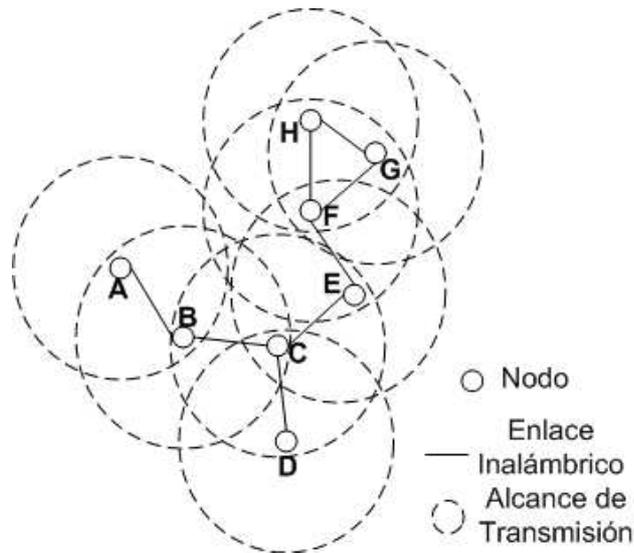


Figura 1.1: Red inalámbrica multihop.

En la literatura existen varias divisiones de las redes inalámbricas multihop como:

- \* Redes inalámbricas *Ad Hoc*: se caracterizan por la ausencia de infraestructura fija y porque la administración de las conexiones es realizada por cada uno de los nodos que las componen.
- \* Redes inalámbricas *Ad Hoc* móviles (*Mobile Ad Hoc Networks*) (MANETs): los nodos que la componen no tienen una posición fija dentro de la red, pues son libres de moverse dentro de ésta.
- \* Redes inalámbricas *Ad Hoc* vehiculares (*Vehicular Ad Hoc Networks*) (VANETs): son redes MANET donde los nodos que la componen son vehículos.
- \* Redes inalámbricas de sensores (*Wireless Sensor Networks*) (WSNs): nombradas así por el tipo de tareas que desempeñan y distinguidas por las restricciones de *hardware* que tienen los nodos de la red.

Para el análisis que se realiza en este estudio, independientemente de la clasificación que se haga, lo que importa es la forma en la que se comunican los nodos dentro de la red.

En esta tesis se consideran las redes inalámbricas multihop que utilizan el estándar IEEE 802.11 [1] (de ahora en adelante el estándar). Se ha elegido este estándar por su amplio uso en este tipo de redes.

## 1.2 Hipótesis

La hipótesis principal de esta tesis es que se pueden crear rutas rápidas dentro de una red multihop. Estas rutas rápidas deben de tener la capacidad de transportar los paquetes a su

destino, o acercarlos a él, con un menor retardo total que el que existiría si la comunicación se realizara mediante rutas tradicionales.

El éxito en la creación de las rutas rápidas reside en realizar las modificaciones necesarias en el protocolo de control de acceso al medio (*Media Access Control*) (MAC) definido en el estándar.

### **1.3 Objetivos**

El objetivo de la tesis es definir un protocolo de comunicación basado en el concepto de rutas rápidas y comprobar su comportamiento mediante el uso de un simulador.

### **1.4 Estructura de la tesis**

La tesis esta organizada de la siguiente manera:

- \* Capítulo 2: presenta los conceptos y mecanismos definidos en el estándar necesarios para el entendimiento del presente estudio, además, se exponen aspectos importantes acerca de QoS.
- \* Capítulo 3: describe cada una de las ideas que componen el protocolo de comunicación propuesto.
- \* Capítulo 4: detalla la forma en que se realizó la implementación del protocolo propuesto en el simulador.
- \* Capítulo 5: presenta los resultados de las simulaciones para los diferentes escenarios planteados y se analizan los resultados obtenidos.
- \* Capítulo 6: presenta las conclusiones generales.

# 2

## Antecedentes

### 2.1 Estándar IEEE 802.11

En el estándar se define lo relacionado con las 2 capas inferiores del modelo de referencia de interconexión de sistemas abiertos (*Open System Interconnection*) (OSI), es decir, la capa de enlace de datos y la capa física. Aunque en el estándar se detallan una gran cantidad de métodos, formatos y mecanismos, para fines de este trabajo se ha prestado mayor atención a la manera en como los nodos tienen acceso al medio inalámbrico.

En el estándar se especifican los bloques componentes de la arquitectura de red iniciando por el conjunto de servicios básicos (*Basic Service Set*) (BSS). Un BSS no es más que un conjunto de estaciones, denominadas en el estándar como STAs, y que son el elemento esencial de toda la arquitectura.

En función de los elementos que componen la topología de la red se tienen los siguientes modos de operación:

- \* Redes con infraestructura.
- \* Redes *ad hoc*.

En las redes con infraestructura existe un nodo que centraliza la información, denominado en el estándar como punto de acceso (*Access Point*) (AP), con quien los nodos deben asociarse para pertenecer al BSS.

Mediante el AP los nodos tienen acceso al sistema de distribución (*Distribution System*) (DS) por medio del cual se amplía la red al conectar un conjunto de BSSs. Así, los nodos que componen los diferentes BSSs envían sus datos a su respectivo AP para llegar al DS y

viceversa. A la extensión de la red, es decir, la conexión de varios BSSs se le conoce como conjunto de servicios extendido (*Extended Service Set*) (ESS). Los nodos pertenecientes a un ESS pueden comunicarse entre ellos y cambiar de BSS.

Por otro lado, en las redes *ad hoc*, referidas en el estándar como un conjunto de servicios básico independiente (*Independent Basic Service Set*) (IBSS), no existen APs que centralicen la información. En este modo de operación es necesario que los nodos puedan comunicarse de manera directa unos con otros, lo que está restringido por el alcance de transmisión de los mismos.

En el estándar se definen tres métodos básicos de acceso al medio que son: la función de coordinación distribuida (*Distributed Coordination Function*) (DCF), la función de coordinación puntual (*Point Coordination Function*) (PCF) y la función de coordinación híbrida (*Hybrid Coordination Function*) (HCF). No obstante, el método de acceso utilizado en las redes inalámbricas multsaltos es DCF, por lo que los detalles de PCF y HCF han sido omitidos en este estudio.

### 2.1.1 Función de coordinación distribuida

DCF es el método básico de acceso definido en el estándar y está basado en el mecanismo de acceso múltiple con detección de portadora y evasión de colisiones (*Carrier Sense Multiple Access with Collision Avoidance*) (CSMA/CA) y un contador en retroceso con crecimiento exponencial binario (*Binary Exponential Backoff*) (BEB).

Para describir la forma en que los nodos contienen por el canal, es necesario tener en cuenta los siguientes intervalos señalados en el estándar, conocidos como espacio entre tramas (*InterFrame Space*) (IFS):

- \* Espacio pequeño entre tramas (*Short InterFrame Space*) (SIFS) es el menor de todos los IFS y es el tiempo que se debe esperar antes de enviar un paquete de acuse de recibo (*Acknowledge*) (ACK), un paquete libre para enviar (*Clear To Send*) (CTS) u otra parte de un paquete en caso de haber sido fragmentado.
- \* Espacio entre tramas DCF (*DCF InterFrame Space*) (DIFS) es el tiempo que un nodo debe esperar antes de iniciar una transmisión en caso de que el canal esté libre y, siempre y cuando, el nodo no se encuentre en un periodo de *backoff*.
- \* Espacio entre tramas extendido (*Extended InterFrame Space*) (EIFS) es el tiempo que un nodo debe esperar antes de realizar una transmisión luego de haber recibido una trama errónea.

Los intervalos IFS tienen un valor fijo en la capa física que no depende de la velocidad de transmisión sino solamente del tipo de transmisión utilizada. Independientemente de lo anterior la relación existente entre los distintos IFS es la siguiente: EIFS > DIFS > SIFS.

Asimismo, el estándar detalla el mecanismo BEB que tiene como objetivo reducir las colisiones. El resultado del BEB se traduce en un tiempo de retraso que es obtenido de manera aleatoria y durante el cual los nodos deben esperar antes de intentar una transmisión. Más adelante se mencionan los casos específicos en los que se utiliza este mecanismo.

El tiempo que un nodo debe esperar debido al BEB esta dado de la siguiente manera:

$$\text{BackoffTime} = \text{Random}() \times \text{SlotTime}$$

Donde:

- \*  $\text{Random}()$  es un valor entero elegido en el intervalo  $[0, CW]$  siguiendo con una distribución uniforme. El valor superior del intervalo se define como ventana de contención (*Contention Window*) (CW) el cual es un valor entero que varía entre  $CW_{min}$  y  $CW_{max}$ .
- \*  $\text{SlotTime}$  es un tiempo definido por la capa física.

El valor de CW inicia en  $CW_{min}$  y se duplica cada vez que no se concreta una transmisión de manera exitosa hasta llegar a  $CW_{max}$ , el cual se mantiene hasta que se reinicia el valor de CW. El valor de CW crece de la siguiente manera por cada transmisión fallida:

$$CW = \begin{cases} 2^{i-1}CW_{min} & \text{si } 1 \leq i \leq m \\ 2^m CW_{min} & \text{si } i > m \end{cases}$$

Donde  $i$  es el número de intento de transmisión y el valor de  $m$  esta restringido por  $CW_{max}$ . Así, para la versión  $b$  del estándar donde  $CW_{min}=31$  y  $CW_{max}=1023$ , el rango de CW crecería de la siguiente forma al existir retransmisiones:

- \* Primer intento:  $[0 \ 31]$
- \* Primera retransmisión:  $[0 \ 63]$
- \* Segunda retransmisión:  $[0 \ 127]$
- \* Tercera retransmisión:  $[0 \ 255]$
- \* Cuarta retransmisión:  $[0 \ 511]$
- \* Quinta retransmisión:  $[0 \ 1023]$
- \* Sexta retransmisión:  $[0 \ 1023]$

Y de manera subsecuente se mantiene en el intervalo  $[0 \ 1023]$  hasta que se logre transmitir el paquete o se rebase el valor límite de intentos de retransmisión y el paquete sea descartado.

Cuando un nodo desea transmitir un paquete utilizando DCF, éste debe sentir el canal para verificar que está y se mantiene libre por un tiempo DIFS, o EIFS en caso de haber recibido un paquete erróneo. Si se cumple con esta condición, y no existe un tiempo de *backoff* remanente, el nodo puede hacer uso del canal e iniciar su transmisión, a lo que se le conoce como un acceso directo al canal. De otra manera, si el canal se siente ocupado durante el DIFS, se genera un tiempo aleatorio de espera mediante EB. En la Figura 2.1 se ilustra el acceso básico DCF donde se cumple con la condición de acceso directo al canal.

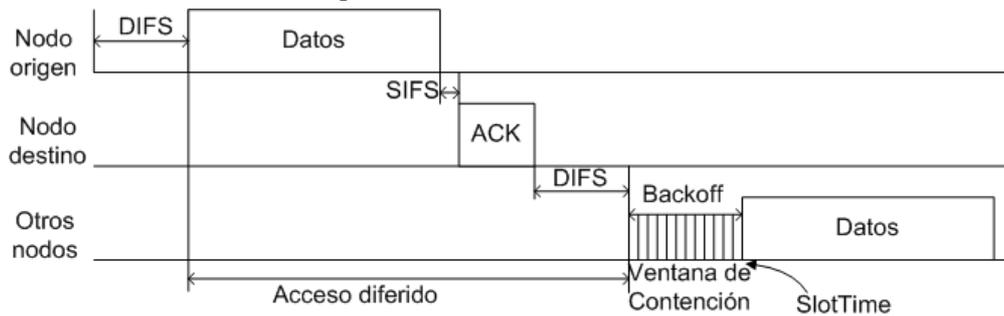


Figura 2.1: Método de acceso DCF.

Para completar una transmisión el nodo destino, después de recibir el paquete de manera correcta, envía un paquete ACK al nodo origen, como puede apreciarse en la Figura 2.1. El tiempo SIFS que un nodo espera para transmitir un paquete ACK está pensado para darle prioridad a este tipo de paquetes, pues  $SIFS < DIFS$ .

Cuando un nodo quiere transmitir un paquete pero tiene un *backoff* pendiente, el nodo debe esperar a que el canal se mantenga libre por un tiempo DIFS/EIFS para luego sentir el canal en cada SlotTime. Mientras el nodo siente el canal libre el *backoff* disminuye un SlotTime cada vez. De otra manera, si el canal se siente ocupado, el *backoff* se detiene para reanudarse después de que el canal ha estado libre nuevamente por un tiempo DIFS/EIFS. El nodo sólo podrá utilizar el canal cuando su contador de *backoff* sea igual a cero. Si el nodo logra transmitir el paquete debe iniciar un nuevo proceso de *backoff* con un valor de CW igual a  $CW_{min}$ . El procedimiento descrito se ilustra en la Figura 2.2.

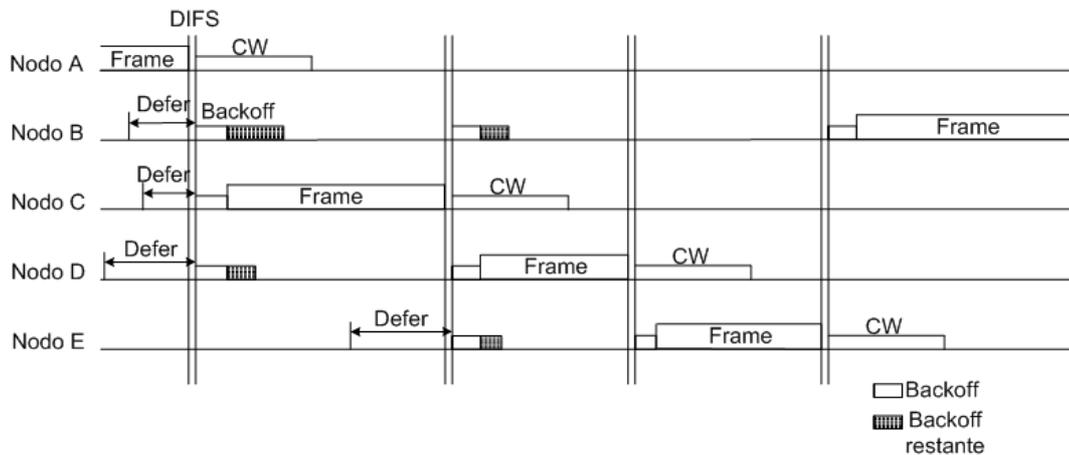


Figura 2.2: Procedimiento de *backoff*

El procedimiento de BEB esta pensado para evitar colisiones. Si se considera la posibilidad de que un conjunto de nodos sensen el canal ocupado al mismo tiempo, sin la existencia del valor aleatorio resultante de BEB una vez que el canal se detecta libre, todos los nodos intentarían utilizarlo simultáneamente, lo cual generaría colisiones. Cuando cada uno de los nodos elige un tiempo de espera aleatorio la probabilidad de que dos transmisiones o más colisionen disminuye.

Por la importancia del EB en esta tesis, se resumen los casos en los que debe utilizarse:

- \* Si en su primer intento de transmisión el nodo sensa el canal ocupado por un tiempo menor a un DIFS/EIFS.
- \* En cada intento de retransmisión.
- \* Después de un transmisión exitosa.

El estándar define otra forma de evitar colisiones y en específico las generadas por el problema de terminal oculta. La manera en como se resuelve el problema de terminal oculta usa un intercambio de paquetes solicitud de envío (*Request To Send*) (RTS)/CTS.

El problema de terminal oculta se presenta cuando existen nodos que pueden enviar información a un destino común, pero que por su posición no son capaces de detectarse mutuamente. Para entender mejor este problema observe la Figura 2.3, en el caso en el que el nodo A se encuentra enviando un paquete a B, y también C quiere comunicarse con B. El nodo C sensaría el canal y al verificar que se encuentra libre, por un tiempo DIFS/EIFS, iniciaría la transmisión del paquete provocando así una colisión con la transmisión de A en B.

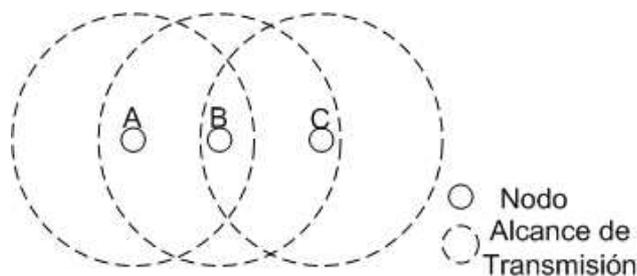


Figura 2.3: Problema de terminal oculta.

Para evitar el problema de terminal oculta, los nodos que desean transmitir un paquete hacen una reservación del canal por el tiempo que durará su envío. Ésto se logra mediante un intercambio de paquetes previo a la transferencia de los datos. En un principio el nodo origen envía un paquete RTS al nodo destino, el cual contesta con un paquete CTS luego de esperar un SIFS. Cuando el nodo origen recibe el paquete CTS envía el paquete de datos después de un intervalo SIFS, para que finalmente el destino confirme la recepción del paquete al nodo

origen, enviando el respectivo paquete ACK. En la Figura 2.4 se muestra el acceso DCF utilizando el intercambio RTS/CTS.

El problema de terminal oculta se resuelve, ya que en los paquetes RTS/CTS se especifica el tiempo que el canal estará ocupado por la transmisión de los datos. De esta manera, aunque existan nodos que no puedan detectarse entre sí, para el ejemplo de la Figura 2.3 los nodos A y C, al detectar el intercambio RTS/CTS, se les avisa que el canal será utilizado y evitarán acceder a él.

Cada nodo que escucha los paquetes RTS/CTS fija el tiempo que no puede acceder al medio en lo que el estándar denomina como vector de ubicación de red (*Network Allocation Vector*) (NAV), que no es más que un temporizador que aloja dicho valor. Este dato le proporciona al nodo algo que se conoce como sentido de portadora virtual. Para discernir si el canal está libre los nodos utilizan el sentido de portadora virtual y físico.

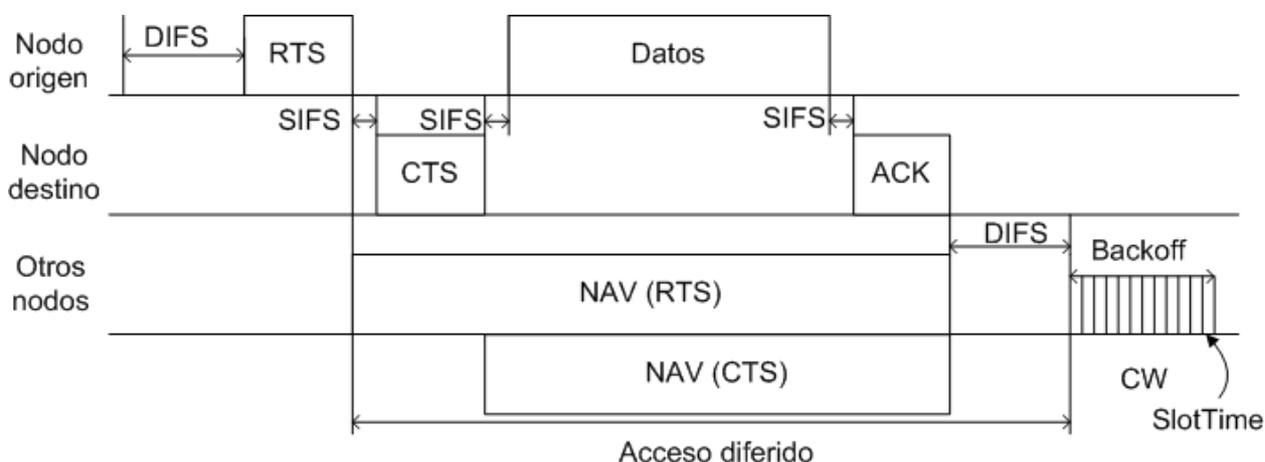


Figura 2.4: Método de acceso DCF con intercambio RTS/CTS.

Los nodos pueden utilizar el intercambio de RTS/CTS o prescindir de él. Para el caso donde se desea utilizar el intercambio RTS/CTS, se define un umbral en función del tamaño de los paquetes de datos, lo cual indica al nodo cuando es necesario iniciar tal intercambio. Se define un umbral de decisión por que el intercambio RTS/CTS tiene una mayor utilidad cuando los paquetes de datos son de gran tamaño.

Cuando los nodos no reciben la respuesta pertinente al paquete que han transmitido, lo retransmitirán hasta que logren transmitirlo correctamente o sea desechado. Los nodos inician una retransmisión en el momento que un temporizador les indica que el tiempo máximo que deben esperar por una respuesta ha expirado. Los nodos pueden definir el tiempo que esperarán por una respuesta antes de iniciar una retransmisión, al saber que:

- \* cuando se ha enviado un RTS se espera en respuesta un CTS,
- \* cuando se ha enviado un CTS se espera en respuesta un paquete de datos y

- \* cuando se ha enviado un paquete de datos se espera en respuesta un paquete ACK.

Si bien el estándar detalla una gran cantidad de cuestiones de una manera más detallada y extensa, además de las ya descritas, para efecto de esta tesis se considera suficiente con lo presentado hasta ahora.

## 2.2 Calidad de Servicio

QoS se define como la capacidad de la red para asegurar en cierta medida la entrega de los paquetes garantizando ciertos parámetros. En general, los parámetros relacionados con QoS son los siguientes:

- \* *Throughput* o ancho de banda.
- \* Retardo de los paquetes o latencia en la red.
- \* *Jitter* o variabilidad en el retardo.
- \* Pérdida de paquetes o tasa de errores.

Garantizar QoS en las redes inalámbricas multisaltos no es una tarea sencilla, dadas las características propias de este tipo de redes. Para lograr lo anterior se tiene que lidiar con factores como: un ancho de banda limitado; una topología dinámica, pues por lo regular los nodos de la red no son estáticos; restricciones de *hardware* tanto de procesamiento como de energía; problemas en los enlaces inalámbricos, pues este tipo de enlaces no son confiables, ya que el estado del canal es impredecible y variable en el tiempo.

Aún así, se ha realizado un estudio extensivo para proponer mecanismos capaces de garantizar QoS en las redes inalámbricas multisalto como se estudia en: [2], [3], [4], [5] y [6]. En general, estas propuestas se pueden clasificar de la siguiente manera:

- \* Modelos QoS: se define una arquitectura en la que se puede garantizar QoS. Un ejemplo de un modelo QoS se estudia en [7].
- \* QoS de señalización: se define la forma en que se controla el flujo de los datos en la red realizando tareas como: reserva y liberación de recursos; establecimiento, restablecimiento y cierre de las conexiones. Un ejemplo de este QoS de señalización se estudia en [8].
- \* QoS de enrutamiento: se encarga de encontrar una ruta capaz de otorgar QoS como se propone en [9].
- \* QoS de MAC: se definen mecanismos que mejoren el desempeño de la capa MAC para otorgar QoS como se describe en [10].

La labor realizada para garantizar QoS en redes inalámbricas multisaltos ha resultado tan compleja que se ha pensado en no seguir con la convención normal del modelo OSI, donde las capas son independiente unas de otras, y realizar un trabajo conjunto entre capas como se estudia en [11].

Por lo amplio del tema resultaría imposible numerar la gran cantidad de trabajos que existen acerca de QoS, pero hasta donde se tiene conocimiento no existe una propuesta como la que se hace en esta tesis.

### **2.3 Resumen de capítulo**

En este capítulo se puntualizan términos básicos utilizados en el estándar IEEE 802.11, y se explica a mayor detalle el método de acceso DCF definido en el mismo estándar. Se hace hincapié en el mecanismo BEB por la importancia que tiene para la presente tesis. Por último, se define lo que es QoS y se mencionan aspectos generales sobre este tema.

# 3

## Descripción detallada del protocolo de comunicación propuesto

No olvidemos que la presente tesis tiene como objetivo proponer un protocolo de comunicación que reduzca los tiempos de retardo total de los paquetes. Se utiliza el término retardo total para indicar el tiempo existente desde que la aplicación crea un paquete en el nodo origen y hasta que es entregado a su destino. Para disminuir el retardo total se crean rutas rápidas que de ahora en adelante se denominarán como *avenidas*.

### 3.1 Factores que influyen en el retardo total de los paquetes en una red multisaltos

Primeramente se realiza el análisis de los factores que contribuyen en el retardo total de los paquetes en la red. La definición de las *avenidas* se basa en el resultado obtenido de este análisis.

En las redes multisaltos, el retardo añadido a los paquetes de datos en cada uno de los nodos que atraviesan para alcanzar su destino se debe a:

- \* Retardo de transmisión ( $R_{Tx}$ ): se debe al tiempo que se necesita para enviar un paquete de un nodo a otro. Este retardo esta en función del tamaño del paquete y de la velocidad de transmisión principalmente.
- \* Retardo de la capa MAC ( $R_{MAC}$ ): es el retardo resultante de la contención que hacen los nodos por el uso del canal, al ser éste un medio compartido.
- \* Retardo de encolamiento ( $R_q$ ): los paquetes que llegan a un nodo son procesados para decidir que hacer con ellos. Cuando el nodo debe retransmitir el paquete verifica el estado de la cola de almacenamiento. Si la cola esta vacía se prepara el paquete para

ser enviado inmediatamente. De otra manera, el paquete tendrá que esperar su turno, definido por el tipo de cola utilizado, aumentando el retardo.

Así, el retardo añadido a un paquete en cada salto ( $R_s$ ) queda definido de la siguiente manera:

$$R_s = R_{Tx} + R_q + R_{MAC}$$

El retardo total ( $R_t$ ) depende del número de saltos ( $i$ ) que realizan los paquetes hasta llegar a su destino. Por lo que  $R_t$  se define como:

$$R_t = \sum_{n=1}^i R_{s_i}$$

Es claro que si se logra disminuir alguno de los valores de los retardos presentes en  $R_s$  se tendrá, en consecuencia, la reducción de  $R_t$ .

La definición de las *avenidas* se basa en la reducción de  $R_{MAC}$  y se complementa con el control de tráfico existente en éstas para disminuir  $R_q$ . La forma en la que operan las *avenidas* y demás mecanismos que integran al protocolo propuesto se describen a lo largo de este capítulo.

### 3.2 Las *avenidas*

Antes de describir las características de las *avenidas* se debe tener en cuenta que para cada nodo existen los siguientes alcances:

- \* Alcance de transmisión.
- \* Alcance de sensado.

Un nodo puede comunicarse de manera directa con los nodos que se encuentran dentro de su alcance de transmisión. Por otro lado, el alcance de sensado de un nodo determina con que nodos se debe contender por el canal. Por tanto, el alcance de sensado incluye al alcance de transmisión. Los nodos que se encuentran fuera del alcance de transmisión de un nodo, pero dentro de su alcance de sensado, escucharán de una manera incomprensible los paquetes que este nodo transmite. Para el desarrollo de esta tesis se consideró un alcance de transmisión y de sensado de 250m y 550m, respectivamente. Los valores de estos alcances están determinados por el modelo de propagación empleado. Se utilizó el modelo de propagación de dos rayos debido a que muestra buenos resultados para microcélulas en espacios abiertos donde existe línea de vista entre los nodos [12]. Para el cálculo de la potencia recibida, con el modelo de propagación dicho, se consideraron los valores predeterminados utilizados por el simulador empleado en esta tesis. Los valores utilizados por el simulador de manera predeterminada buscan describir el comportamiento de los dispositivos existentes en el mercado.



Para explicar los detalles de la *avenida* considere la imagen mostrada en la Figura 3.3. En esta figura se muestra una *avenida*. La *avenida* está compuesta por los nodos denotados con la letra *A* ( $A_1$ - $A_{11}$ ). También se muestran los alcances de sensado de los nodos de la *avenida* y el área que limita la intersección de estos alcances, denominada como zona de restricción. Las flechas existentes entre los nodos de la figura señalan la forma en que viajan los paquetes.

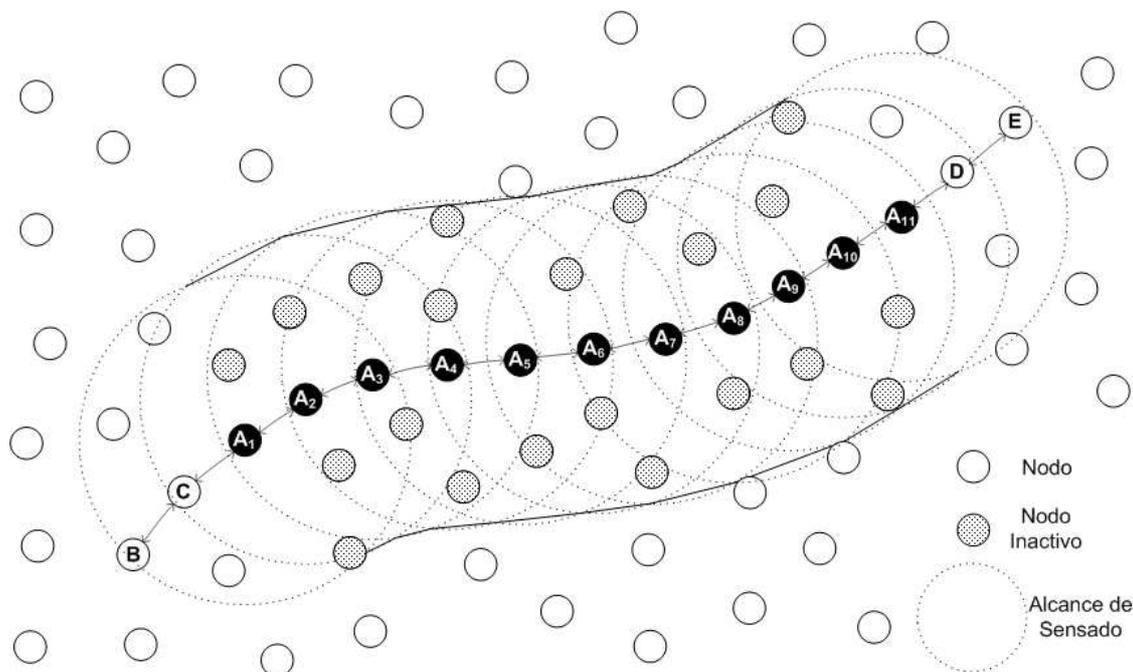


Figura 3.3: Detalles de una avenida en una red inalámbrica multisaltos.

La idea es que cuando un nodo de la red quiera transmitir un paquete que necesite llegar a su destino con un bajo retardo total, utilice las *avenidas* en lugar de enviar el paquete utilizando las rutas convencionales. Supongamos que el nodo *B* de la Figura 3.3 desea comunicarse con el nodo *E*. Si el nodo *B* necesita que los paquetes sean entregados con un menor retardo total, puede utilizar la *avenida* para completar la comunicación. Los saltos que darían los paquetes en la comunicación entre *B* y *E* serían los siguientes:

$$B \Rightarrow C \Rightarrow A_1 \Rightarrow A_2 \Rightarrow A_3 \Rightarrow A_4 \Rightarrow A_5 \Rightarrow A_6 \Rightarrow A_7 \Rightarrow A_8 \Rightarrow A_9 \Rightarrow A_{10} \Rightarrow A_{11} \Rightarrow D \Rightarrow E$$

En seguida se explica la manera en que se propone reducir el retardo total de los paquetes que viajan por las *avenidas*.

### 3.2.1 Reducción del retardo de la capa MAC

La reducción de  $R_{MAC}$  se logra restringiendo el uso del canal de manera que en la *avenida* no exista interferencia ajena a la producida por los nodos que conforman la *avenida*. Es decir, los nodos que se encuentran dentro de la zona de restricción de la *avenida* no tienen acceso al canal, como se evidencia en la Figura 3.3. Al limitar la contención, los nodos de la *avenida* pueden transmitir los paquetes de manera más rápida, pues se disminuye el retardo generado por los tiempos de *backoff* estipulados en el modo de acceso DCF.

Debido a que los nodos que se ubican en las zonas aledañas a la *avenida* no pueden utilizar el canal, solo se puede ingresar a éstas por los nodos terminales. Para el escenario mostrado en la Figura 3.3 se ingresa a la *avenida* mediante los nodos  $A_1$  y  $A_{11}$ . En esta figura se aprecia que los nodos  $B$ ,  $C$ ,  $D$  y  $E$  se encuentran dentro del rango de sentido de los nodos terminales de la *avenida*, pero si se les restringiera el uso del canal, la *avenida* resultaría inalcanzable.

Con la finalidad de seguir disminuyendo el  $R_{MAC}$  se propone que los nodos que constituyen las *avenidas* intenten una transmisión luego de sensar el canal desocupado por un tiempo de DIFS/EIFS más un SlotTime, negándoles el acceso directo al canal. Para que los nodos de la *avenida* esperen el tiempo indicado se utiliza un mecanismo BEB modificado. Esta modificación en el BEB suprime la aleatoriedad del mecanismo de tal manera que el BackoffTime siempre es de un SlotTime, sin importar el número de intento de transmisión del paquete. La Figura 3.4 ilustra la manera en que un nodo de la *avenida* realiza una transmisión.

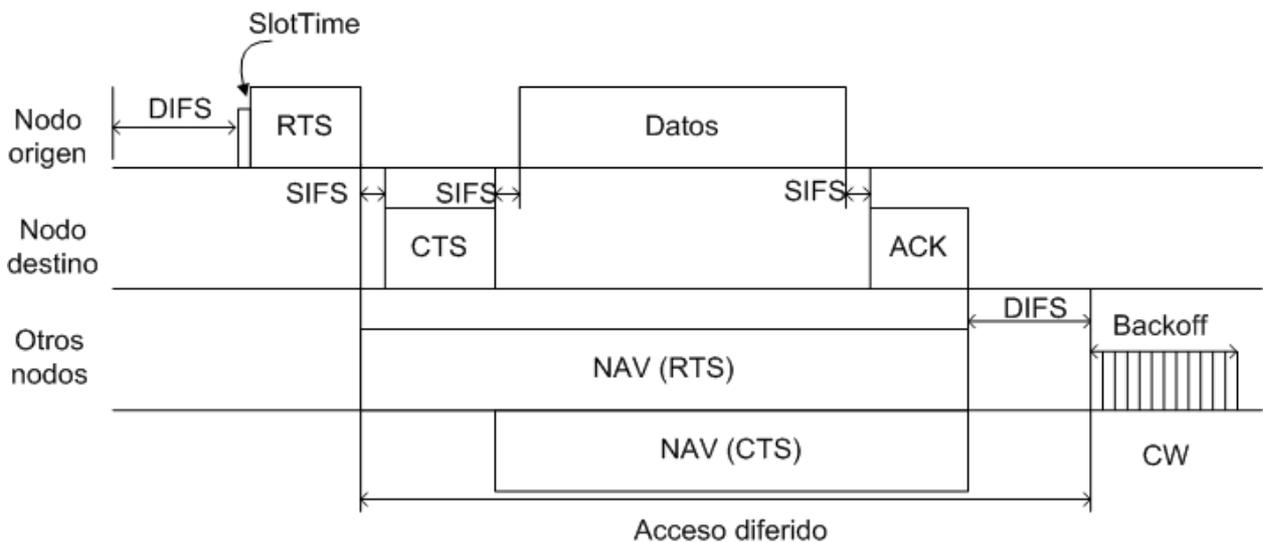


Figura 3.4: Intercambio de paquetes entre dos nodos cualesquiera de la avenida con  $CW=1$  constante.

Utilizando el mecanismo BEB modificado se disminuye el  $R_{MAC}$  ya que los tiempos de castigo resultantes del BEB no crecen exponencialmente.

### 3.2.2 Reducción del retardo de encolamiento

Buscando que el retardo de encolamiento no fuera un factor se decidió restringir el flujo de información existente en la *avenida*. La cantidad de paquetes que la *avenida* soportará debe ser tal que, dentro de lo posible, no exista encolamiento de paquetes en los nodos que la forman.

Para explicar la manera en que se propone disminuir  $R_q$  considere que la *avenida* está distribuida como se muestra en la Figura 3.5. Suponga que solo existe un paquete cuyo

origen es el nodo  $A_1$  y que tiene como destino al nodo  $A_{11}$ . Suponga, además, que el paquete se encuentra en el nodo  $A_2$ , es decir, se ha realizado de manera exitosa la transmisión  $A_1 \Rightarrow A_2$ . Los saltos restantes que hará el paquete para llegar a su destino son:

$$A_2 \Rightarrow A_3 \Rightarrow A_4 \Rightarrow A_5 \Rightarrow A_6 \Rightarrow A_7 \Rightarrow A_8 \Rightarrow A_9 \Rightarrow A_{10} \Rightarrow A_{11}$$

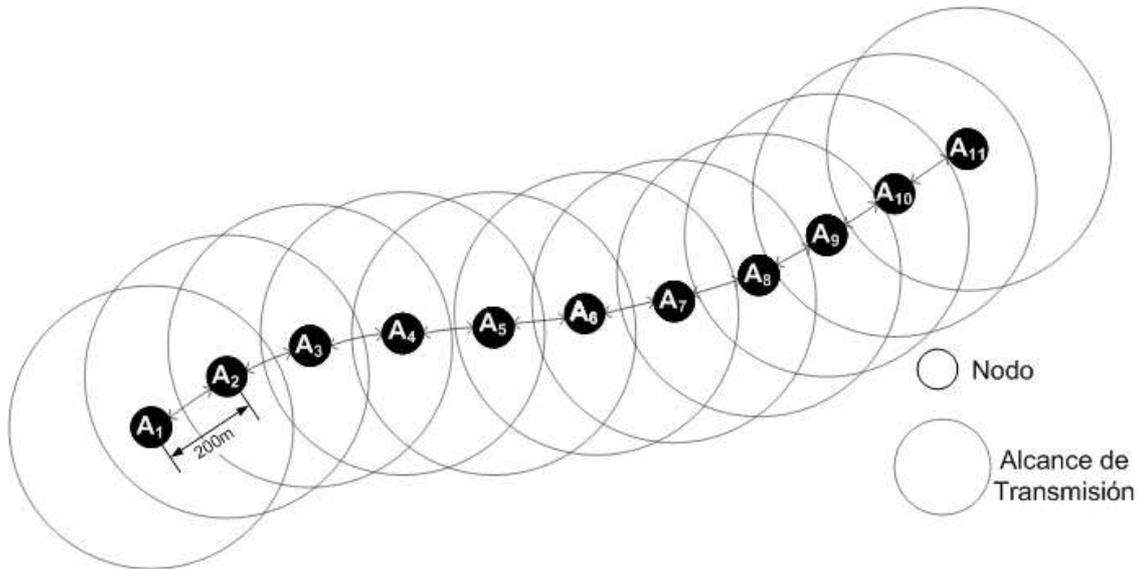


Figura 3.5: Nodos que constituyen una avenida.

Para disminuir  $R_q$  se calculó el tiempo que un nodo de la *avenida* debe esperar para utilizar el medio sin interferir con un paquete que ha transmitido previamente. El tiempo que los nodos de la *avenida* no pueden utilizar el canal determinará el valor de la tasa de datos máxima que se puede soportar en la *avenida*. Este tiempo, para el escenario de la Figura 3.5, está fijado por la respuesta a la siguiente pregunta: ¿en qué momento el nodo  $A_2$  puede comunicarse con  $A_3$  sin obstaculizar el flujo del paquete que transmitió previamente?

Para dar respuesta a la pregunta planteada se describe, siguiendo con la topología de la Figura 3.5, el proceso completo que implica la retransmisión del paquete que se encuentra en el nodo  $A_2$  al nodo  $A_3$ :

- \* Suponiendo que el nodo  $A_2$  conoce la dirección MAC del nodo  $A_3$ .  $A_2$  inicia la transmisión enviando un paquete RTS a  $A_3$ .
- \*  $A_3$  responde a  $A_2$  con un paquete CTS.
- \* Al completar el intercambio RTS/CTS,  $A_2$  envía el paquete de datos a  $A_3$ .
- \*  $A_3$  responde a  $A_2$  con el paquete ACK del paquete recibido, completando la transmisión.

Este proceso se repite en cada salto que da el paquete hasta llegar a su destino final, que para este ejemplo es el nodo  $A_{11}$ .

Conociendo lo que acontece en cada envío, se analizó en que momento  $A_2$  podría transmitir un nuevo paquete de tal manera que no interfiriera con el paquete que había enviado antes:

- \* Cuando el paquete se encuentra en  $A_3$  y se envía a  $A_4$ ,  $A_2$  sabe que el canal está ocupado por el intercambio de paquetes RTS/CTS entre  $A_3$  y  $A_4$ .
- \* Cuando el paquete se transmite de  $A_4$  a  $A_5$ ,  $A_2$  sensa el canal ocupado pues el nodo  $A_4$  se encuentra dentro de su alcance de sensado, recuerde que el alcance de sensado es mayor al de transmisión.
- \* Al darse la transmisión de  $A_5$  a  $A_6$ ,  $A_2$  podría transmitir otro paquete sin interferir con el que actualmente se encuentra en la *avenida*. Sin embargo,  $A_3$  sensaría el canal ocupado ya que  $A_6$  se encuentra dentro de su alcance de sensado.
- \* Cuando se realiza la transmisión de  $A_6$  a  $A_7$ ,  $A_2$  y  $A_3$  pueden intercambiar mensajes sin interferir con la transmisión  $A_6 \Rightarrow A_7$ .

Por lo tanto, los nodos de la *avenida* que pueden transmitir de manera simultánea están separados por tres nodos inactivos. Para la *avenida* que hemos tomado como referencia, Figura 3.6, el nodo  $A_2$  y  $A_6$  pueden transmitir de manera simultánea sin interferirse mutuamente mientras los nodos  $A_3$ ,  $A_4$  y  $A_5$  permanecen inactivos. De forma similar, el nodo  $A_{10}$  podría utilizar el canal al mismo tiempo que lo hacen los nodos  $A_2$  y  $A_6$ . Este comportamiento se muestra en la figura 3.6.

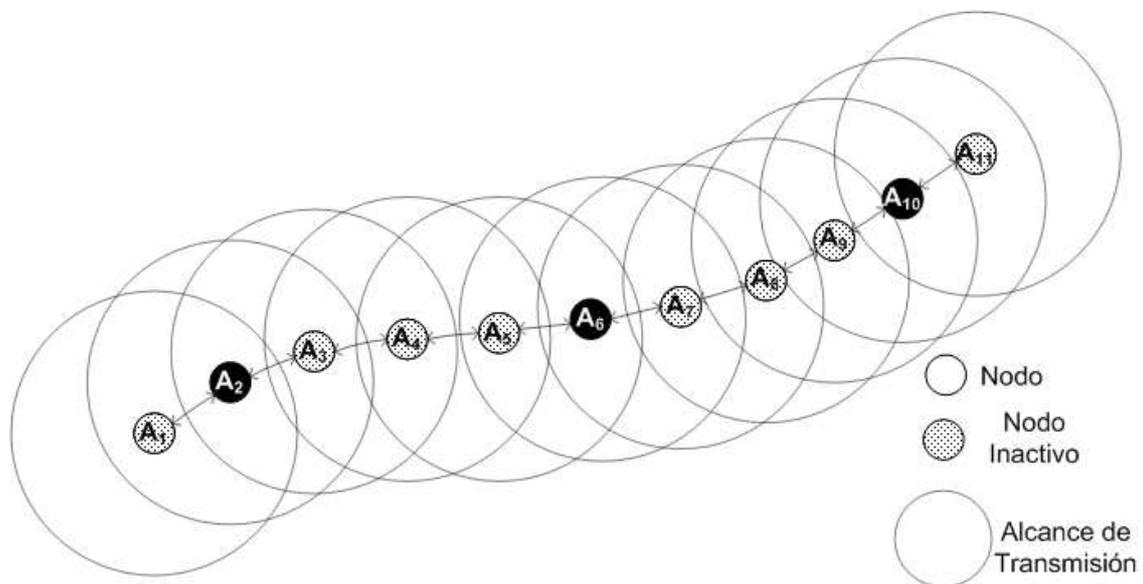


Figura 3.6: Nodos de la avenida que pueden transmitir de manera simultánea sin interferirse mutuamente. Los nodos  $A_2$ ,  $A_6$  y  $A_{10}$  pueden transmitir mientras  $A_3$ ,  $A_4$ ,  $A_5$ ,  $A_7$ ,  $A_8$  y  $A_9$  permanecen inactivos.

Fue así como se obtuvo el intervalo de tiempo que debe existir entre la transmisión de dos paquetes continuos en la avenida.

En la Figura 3.7 se presenta el diagrama de tiempos pertinente a la transmisión de un paquete de datos entre dos nodos cualesquiera ( $A_1$  y  $A_2$ ) que forman parte de la *avenida*. Se muestran los tiempos que los nodos deben esperar para usar el canal definidos en el estándar, específicamente el valor de DIFS, SIFS y CW. El tiempo de transmisión del paquete fue marcado de manera meramente ilustrativa, pues estos valores dependen directamente del tamaño del paquete y de la velocidad a la que son transmitidos.

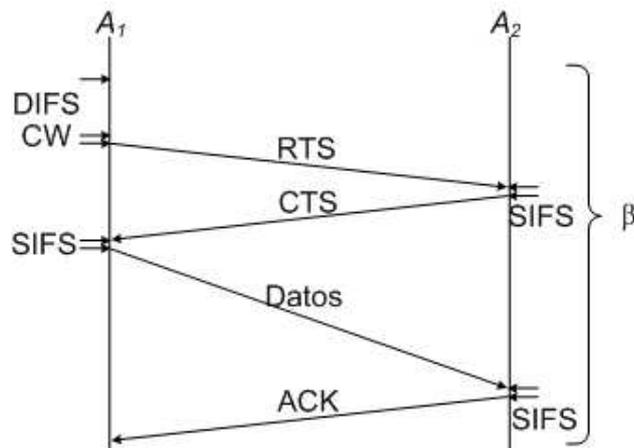


Figura 3.7: Diagrama de tiempos de la transmisión de un paquete entre el nodo A y B.

Se denominó como  $\beta$  al tiempo necesario para realizar la transmisión de un paquete de datos entre dos nodos de la *avenida*, como se señala en la Figura 3.7. Un nodo de la *avenida* inicia una transmisión que le toma un tiempo  $\beta$  para luego esperar que el paquete de tres saltos para transmitir un nuevo paquete. En consecuencia, el tiempo que un nodo debe esperar para enviar un paquete sin interferir con otro que ha transmitido previamente es de  $4\beta$ . Ésto puede corroborarse observando el ciclo de transmisiones marcado en la Figura 3.6, donde el nodo  $A_2$  tiene que esperar cuatro intercambios antes de transmitir un nuevo paquete. Estos intercambios, cada uno de los cuales tiene un valor de  $\beta$ , son:  $A_2 \Rightarrow A_3$ ,  $A_3 \Rightarrow A_4$ ,  $A_4 \Rightarrow A_5$  y  $A_5 \Rightarrow A_6$ . Así, la tasa de datos por segundo que la *avenida* puede soportar es  $\frac{1}{4\beta}$ .

Si los datos en la *avenida* muestran un comportamiento tal que no existe interferencia entre dos paquetes subsecuentes, los nodos retransmitirían el paquete recibido inmediatamente, disminuyendo a  $R_q$ .

### 3.3 Avenidas con rampas

Se definió un mecanismo para permitir que algunos de los nodos inactivos ubicados a los alrededores de la *avenida* pudieran transmitir y utilizar la *avenida*. El mecanismo definido cumple con la condición de no entorpecer el flujo de paquetes existente en la *avenida*.

Para que los nodos que se encuentran dentro de la zona de restricción de la *avenida* puedan ingresar a ésta deben tener una ubicación espacial tal que puedan comunicarse directamente

con tres nodos integrantes de la *avenida*. A los nodos que cumplen con esta característica, se les denominó como *nodos rampa*.

En la Figura 3.8 se muestra una *avenida* acompañada de su zona de restricción, los rangos de transmisión de los nodos que la componen y los *nodos rampa*, denotados con la letra *R*. Observe que cada uno de los *nodos rampa* mostrados cumplen con la característica planteada al tener la capacidad de comunicarse con tres nodos de la *avenida*. Por otro lado, los paquetes provenientes de los demás nodos de la red pueden ingresar a la *avenida* mediante los *nodos rampa*. Por la manera en que fluyen los paquetes por la *avenida*, el salto de los paquetes a los *nodos rampa* solo puede darse cuando el flujo de datos existente en la *avenida* es menor al máximo que puede soportar.

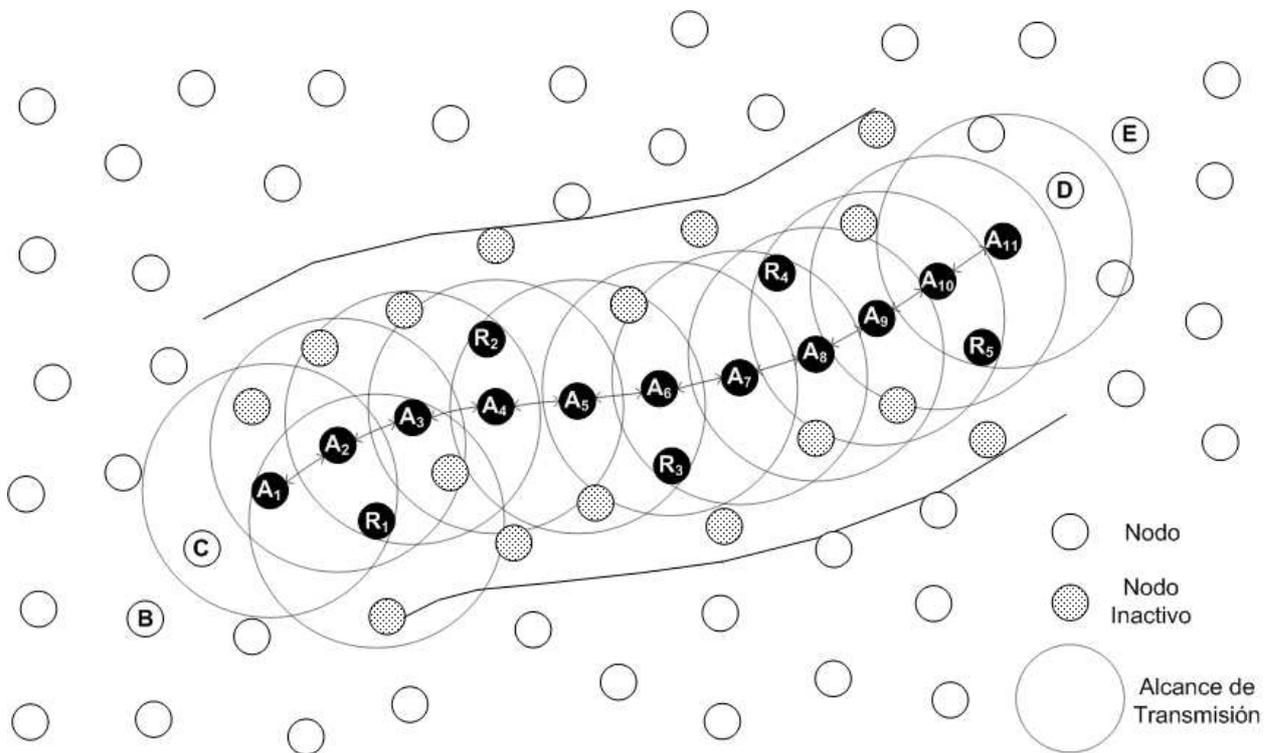


Figura 3.8: Nodos de una avenida:  $A_1$ - $A_{11}$ ; y nodos rampa:  $R_1$ - $R_5$ .

### 3.3.1 Acceso a la avenida

Para evitar que los paquetes provenientes de los *nodos rampa* afecten el flujo de datos existente en la *avenida* se propone un proceso de acceso a ésta. Para describir el proceso de acceso a la *avenida* se tomó como referencia el escenario de la Figura 3.8. Suponga que existe un paquete en el nodo  $A_1$  proveniente del nodo B y que tiene como destino el nodo E. Los saltos restantes que daría este paquete para llegar a su destino serían:

$$A_1 \Rightarrow A_2 \Rightarrow A_3 \Rightarrow A_4 \Rightarrow A_5 \Rightarrow A_6 \Rightarrow A_7 \Rightarrow A_8 \Rightarrow A_9 \Rightarrow A_{10} \Rightarrow A_{11} \Rightarrow D \Rightarrow E$$

Cuando el paquete da el salto de  $A_1 \Rightarrow A_2$  éste es escuchado por el *nodo rampa*  $R_1$  que en condiciones normales descartaría el paquete, pues al desencapsularlo se percataría que la dirección MAC destino no coincide con la de él. Sin embargo, los *nodos rampa* no hacen un filtrado MAC convencional. Cuando el nodo  $R_1$  recibe el paquete de la comunicación entre los nodos  $A_1 \Rightarrow A_2$ , verifica si en su cola de almacenamiento existe un paquete esperando a ser incorporado a la *avenida*. Si lo anterior resulta cierto, el *nodo rampa* conserva el paquete proveniente de la *avenida*, en caso contrario el paquete es descartado.

Un *nodo rampa* necesita dos paquetes para acceder a la *avenida*, uno de ellos es el recibido de un nodo de la *avenida* y el otro proviene de su cola de almacenamiento. Cuando el *nodo rampa* cuenta con estos paquetes verifica que la suma de sus tamaños no rebase la unidad máxima de transmisión (Maximum Transmission Unit) (MTU), o en su defecto, el valor definido por el administrador de la red. Si esta verificación es positiva el *nodo rampa* puede incorporar los dos paquetes a la *avenida*.

Buscando no detener el flujo de información de la *avenida*, el *nodo rampa* enviará un nuevo paquete que contenga el paquete recibido y el paquete propio que desea incorporar a la *avenida*. Una vez que el *nodo rampa* ha transmitido el nuevo paquete éste fluye de manera normal por la *avenida*.

Cuando un *nodo rampa* envía un paquete a la *avenida*, el nodo de la *avenida* que debería transmitir el paquete utilizado por el *nodo rampa* examina el paquete proveniente de este último. Si el paquete que el nodo de la *avenida* está a punto de transmitir fue transmitido por el *nodo rampa*, el envío es cancelado. Con ésto se busca evitar la duplicidad de paquetes debida a la manera en que los *nodos rampa* acceden a la *avenida*.

Para garantizar que los *nodos rampa* transmitan su paquete antes de que se presente el envío pertinente al siguiente salto en la *avenida*, se les otorgó el privilegio de tener acceso directo al medio. Es decir, los *nodos rampa* solo esperan un intervalo de DIFS/EIFS para iniciar una transmisión. El privilegio sobre el canal otorgado a los *nodos rampa* justifica el valor constante del BackoffTimer de los nodos que componen la *avenida*, y nos asegura que, siempre que se cumplan con las condiciones del mecanismo de acceso, los *nodos rampa* podrán utilizar la *avenida*.

En la Figura 3.9 se ilustra el proceso completo que sigue el *nodo rampa*  $R_1$  para poder utilizar el canal, evidenciando el comportamiento de este nodo y de los que forman parte de la *avenida*. En resumen, en el primer instante se realiza la transmisión  $A_1 \Rightarrow A_2$ , donde el paquete enviado por  $A_1$  también es recibido por  $R_1$ . El nodo  $R_1$  al tener un paquete en su cola de almacenamiento forma y transmite el nuevo paquete que contiene al paquete recibido de  $A_1$  y al que tiene almacenado. El paquete proveniente de  $R_1$  es recibido por  $A_2$  y  $A_3$ . El nodo  $A_2$  revisa el paquete, y cancela la transmisión del paquete recibido de  $A_1$  al estar contenido en el paquete de  $R_1$ . Por su parte, el nodo  $A_3$  realiza la transmisión del paquete recibido a  $A_4$ , logrando así acceder a la *avenida*.

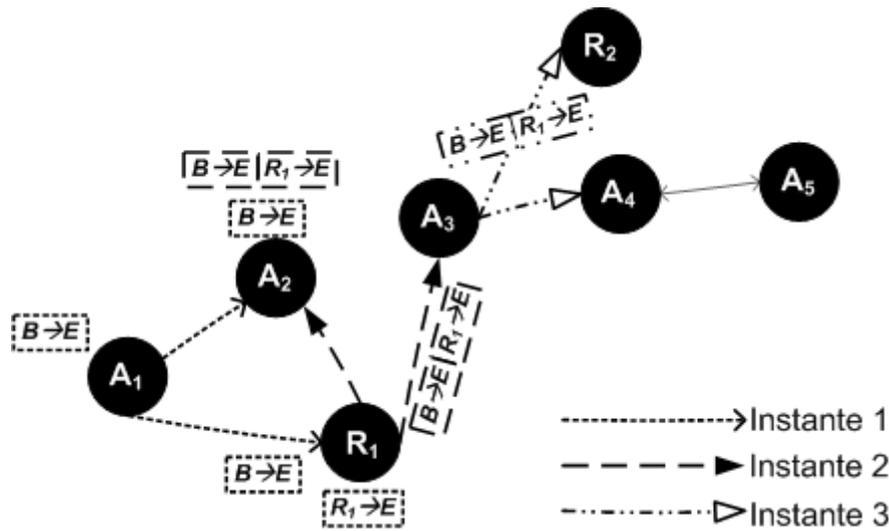


Figura 3.9: Proceso realizado por los nodos rampa para acceder a la avenida.

Para poder hacer una entrega correcta de cada uno de los paquetes utilizados en el proceso de acceso de los *nodos rampa* es necesario conservar la dirección IP de cada uno de ellos y los puertos utilizados por las aplicaciones. Esta información se encuentra contenida en el encabezado de la capa de red y de transporte, respectivamente.

Los paquetes creados por los *nodos rampa* son del tipo IP, pues su único objetivo es transportar a los paquetes que contiene. Cuando el paquete proveniente de un *nodo rampa* llega a su destino, los paquetes contenidos en él son separados para entregarlos a la conexión pertinente.

### 3.4 Problema de terminal expuesta

Para la descripción de las *avenidas* y las *rampas* se ha considerado que los paquetes que se incorporan a la avenida muestran cierta periodicidad. Es decir, los paquetes están separados de tal manera que sus transmisiones no se interfieren entre ellas sobre la *avenida*. Sin embargo, existen aplicaciones que crean paquetes de manera aleatoria, lo cual puede generar que dos paquetes en la *avenida* se interfieran entre sí. Esta característica del tráfico acentúa el problema de terminal expuesta en el protocolo propuesto en esta tesis.

El problema de terminal expuesta se presenta cuando dos transmisiones se pueden dar de manera simultánea sin provocar colisiones en los receptores, pero al sensar un nodo el canal de manera incorrecta impide el inicio de una transmisión. Tome como referencia la avenida mostrada en la Figura 3.10. En esta figura se ha marcado el alcance de sensado de los nodos  $A_2$ ,  $A_4$  y  $R_2$  para explicar el problema de terminal expuesta. Suponga el caso donde existe la transmisión  $A_4 \Rightarrow A_5$  o  $R_2 \Rightarrow A_5$ , y un paquete llega a la *avenida* por el nodo  $A_1$ . El nodo de la *avenida*  $A_1$  sensaría el canal desocupado y enviaría un paquete RTS al nodo  $A_2$ . El nodo  $A_2$  sensaría el canal ocupado pues los nodos  $A_4$  y  $R_2$  se encuentran dentro de su alcance

de sensado. El nodo  $A_2$ , al sensar de manera errónea el canal, no enviaría el paquete CTS correspondiente a  $A_1$ , aún cuando las transmisiones  $A_4 \Rightarrow A_5$  o  $R_2 \Rightarrow A_5$ , y  $A_1 \Rightarrow A_2$  pueden realizarse a la par sin provocar colisiones.

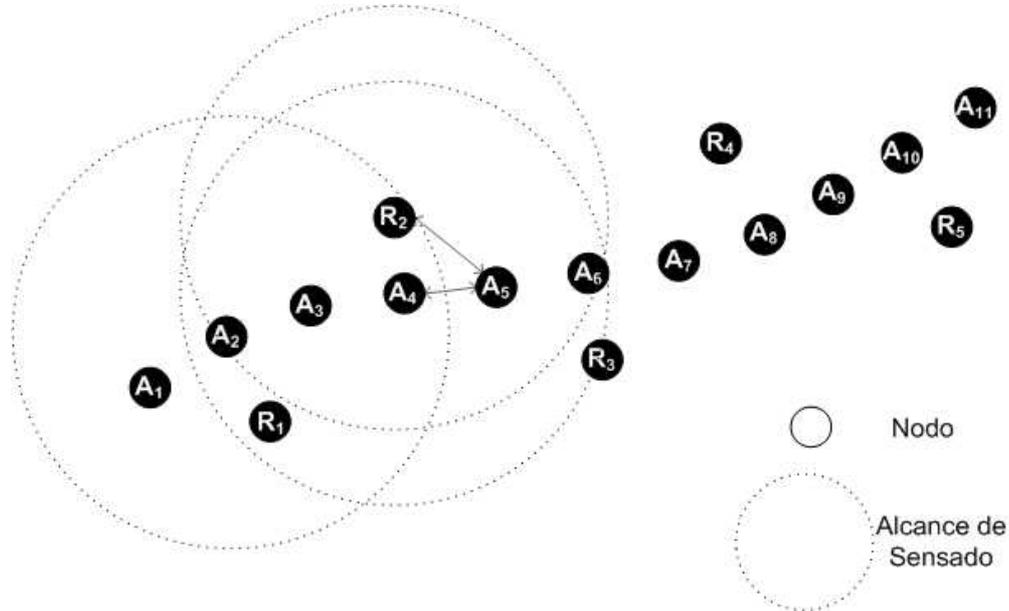


Figura 3.10: Problema de terminal expuesta.

Las consecuencias del problema de terminal expuesta se agravan para los nodos de la *avenida* pues para el escenario de la Figura 3.10, el nodo  $A_1$  no se percata de la transmisión existente en la *avenida* ( $A_4 \Rightarrow A_5$  o  $R_2 \Rightarrow A_5$ ). El nodo  $A_1$  al sensar libre el canal continuaría intentando comunicarse con el nodo  $A_2$  que no contestaría al sensar el canal ocupado. El nodo  $A_1$ , al no recibir respuesta, siempre esperaría un intervalo de tiempo DIFS mas un SlotTime para reenviar el paquete RTS. La comunicación entre  $A_4 \Rightarrow A_5$  o  $R_2 \Rightarrow A_5$  puede estar transfiriendo un paquete de datos de tamaño tal que la espera del nodo  $A_1$  resultara ser muy breve. Si el nodo  $A_1$  supera el número de intentos de retransmisión que tienen permitidos antes de que el problema de terminal expuesta se resuelva, el paquete podría ser descartado.

Cuando los nodos que forman parte de la *avenida* descartan paquetes se afecta tanto el desempeño de la *avenida* como el de los *nodos rampa*. Al descartar un paquete el número de oportunidades que tiene un *nodo rampa* para utilizar la *avenida* disminuye al igual que lo hacen el número de paquetes que son entregados a su destino. Asimismo, al no existir paquetes fluyendo por la *avenida*, pues han sido descartados, los *nodos rampa* no pueden acceder a las *avenidas*, lo cual provoca un aumento en el retardo total que experimentan los paquetes de estos nodos debido al encolamiento. Además, si las aplicaciones en los *nodos rampa* siguen generando paquetes que no pueden ser transmitidos, éstos son almacenados en la cola de espera. Otro problema se presenta cuando el nodo rampa alcanza la capacidad máxima de almacenamiento de la cola, ya que ahora los nuevos paquetes serán descartados.

Buscando atenuar el problema de terminal expuesta, se planteó que el nodo terminal de la *avenida* no proceda de la misma manera que los demás nodos que la componen. Se propone que cuando este nodo realice una retransmisión de un paquete RTS espere más tiempo que el respectivo a un SlotTime. Esta espera prolongada toma como premisa que la ausencia de respuesta por parte del nodo destino es ocasionada por un problema de terminal expuesta. De esta manera se intenta evitar que los nodos que dan acceso a la *avenida* rebasen el número máximo de intentos de retransmisión y se descarten paquetes.

Para evitar que el nodo terminal de la *avenida* espere un tiempo excesivo se plantea que para un número  $n$  de oportunidades de retransmisión se utilice un valor de CW grande en BEB. Si no se logra transmitir el paquete dentro de las primeras  $n$  oportunidades, se procede a disminuir el valor de CW buscando que el nodo que da acceso a la *avenida* no espere de manera excesiva. Para elegir el tiempo que el nodo terminal de la *avenida* debe esperar en los primeros  $n$  intentos se necesita conocer las características del tráfico en la red, por lo que es necesario realizar un análisis del mismo. Como resultado de este estudio se puede conocer el tamaño promedio de los paquetes existentes en la red que aunado a la velocidad de transmisión permite estimar el tiempo necesario para transmitir un paquete.

Basándose en la Figura 3.10, suponga que la transmisión que esta ocasionando la terminal expuesta  $A_4 \Rightarrow A_5$  o  $R_2 \Rightarrow A_5$  recién ha iniciado, y que se esta enviando un paquete de tamaño promedio. Si el nodo terminal de la *avenida* ( $A_1$ ) necesitara retransmitir el paquete RTS, al conocer el tiempo promedio que tarda la transmisión de un paquete puede valorar el tiempo que necesita esperar antes de realizar dicha retransmisión. Sin embargo, el nodo  $A_1$  no sabe con certeza cuanto tiempo falta para que la transmisión que produce el problema de terminal expuesta finalice. Debido a esto, se sugiere que el nodo terminal de la *avenida* utilice un valor de CW grande y constante, eliminando la aleatoriedad del algoritmo BEB para los primeros  $n$  intentos de retransmisión. Mientras que para los intentos restantes se utilice BEB convencional con una CW mas pequeña.

Es evidente que entre más acertado sea el análisis de tráfico mejor será la elección del tiempo de espera de los nodos terminales, o sea, los valores de CW. Es importante elegir el valor más adecuado, pues cada una de las retransmisiones efectuadas por los nodos tiene un costo en cuanto a consumo de energía se refiere.

### 3.4.1 Modos de operación

Por otro lado, piense en un escenario donde existe un nodo que, aunque se encuentra dentro del área de restricción de la *avenida*, desea utilizar el canal pero no puede ser un *nodo rampa*, generando el problema de terminal expuesta. En la Figura 3.11 se muestra un escenario donde se presenta un problema de terminal expuesta en el nodo  $A_2$  cuando se esta realizando una transmisión entre los nodos  $X \Rightarrow Y$ . Aunque los nodos que intervienen en la transmisión que produce el problema de terminal expuesta deberían estar inactivos por estar en el área de restricción, se decidió considerar el caso donde esta área no es respetada. Esta decisión se

tomó buscando que el protocolo propuesto mostrara el mejor desempeño posible en diferentes escenarios.

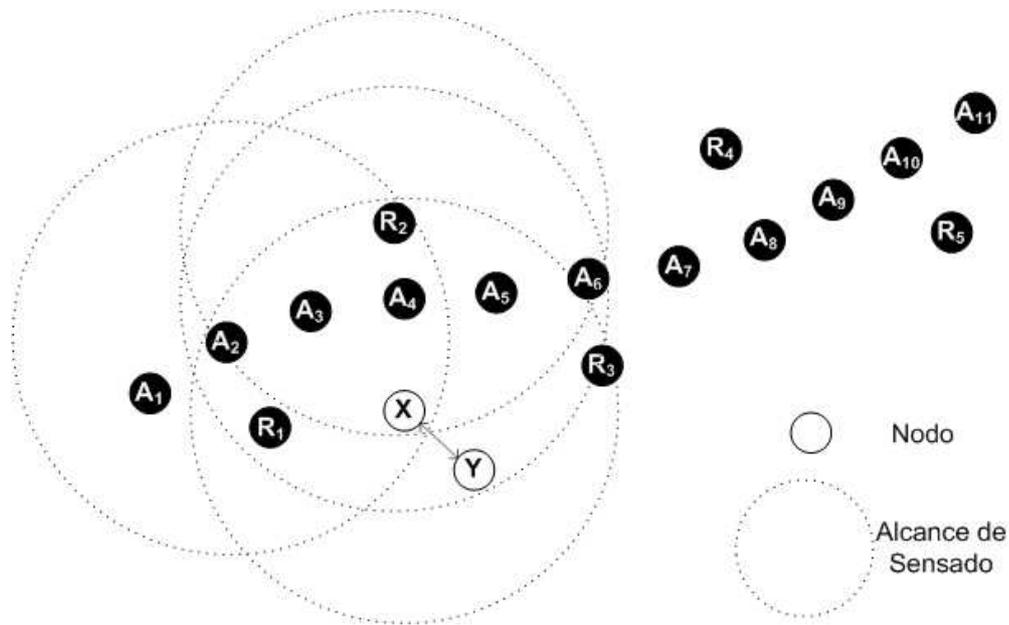


Figura 3.11: Problema de terminal expuesta con nodos que no pertenecen a la avenida.

Bajo el escenario de la Figura 3.11, la propuesta para aminorar las consecuencias debidas al problema de terminal expuesta resulta perjudicial, al estar basada en el supuesto de que después de esperar cierto tiempo la información fluye por la *avenida*. Esta suposición resulta ser falsa en este escenario ya que el problema de terminal expuesta es generado por un envío ajeno a la *avenida*, y se desconoce el tiempo que durará la comunicación  $X \Rightarrow Y$ .

Cuando existen nodos que contienen por el canal, sin ser nodos que forman parte de la *avenida* o *nodos rampas*, se puede presentar el caso donde el problema de terminal expuesta no pueda aminorarse. El nodo terminal de la *avenida* puede seguir sin recibir una respuesta del nodo con el que desea comunicarse después de esperar el tiempo estimado para que la transmisión que genera la terminal expuesta finalice. Si el problema de terminal expuesta no concluye mientras el nodo terminal de la avenida se encuentra esperando, éste descartará el paquete.

El hecho de descartar paquetes implica que aquellos que son transmitidos y llegan a su destino lo hacen con un retardo total elevado. Este retardo se debe al tiempo de encolamiento aunado a la espera resultante de la solución dada al problema de terminal expuesta. Los paquetes que desean utilizar la avenida son almacenados en la cola de espera del nodo inicial mientras éste se encuentra esperando para iniciar una transmisión. En este caso resultaría mejor omitir el tiempo extra de espera que hace el nodo terminal de la *avenida*, pues de esta manera los paquetes que no son descartados y llegan a su destino mantienen un retardo total bajo.

Para que la *avenida* presentara su mejor desempeño, independientemente del escenario en el que se encuentra, se decidió que el nodo terminal de la *avenida* trabajara con dos modos de operación. Estos modos son utilizados por dicho nodo terminal en función de una estimación que hace sobre el nivel de contención existente.

Los modos en los que puede trabajar el nodo terminal de la *avenida* son:

- \* Modo de baja contención: el nodo trabaja bajo este modo al alcanzar un número determinado de transmisiones exitosas consecutivas, pues al no desechar paquetes el nodo estima que hay una baja contención por el canal.
- \* Modo de alta contención: el nodo utiliza este modo de operación al descartar un número determinado de paquetes de manera consecutiva, pues al no lograr realizar transmisiones exitosas es aceptable considerar que existe una alta contención por el medio.

El nodo terminal de la *avenida* inicia trabajando en un modo de baja contención comportándose de la forma en que se propuso para evitar desechar los paquetes por el problema de la terminal expuesta. En el modo de alta contención el nodo terminal se comporta como se definió la *avenida* de manera original, es decir, con un BackoffTimer constante igual a un SlotTime.

Lo único que tiene que hacer el nodo terminal de la *avenida* para intercambiar su modo de operación es contar el número de paquetes que descarta y los que transmite exitosamente. Estos valores se modifican mutuamente pues cuando el contador de número de paquetes descartados aumenta, el contador de paquetes transmitidos exitosamente es reiniciado a un valor igual a cero y viceversa.

Para controlar el momento en el que el nodo debe cambiar de modo de operación es necesario plantear un umbral en función del valor de las variables que llevan el conteo de los paquetes que se han transmitido o descartado. Así, cuando el umbral de paquetes transmitidos exitosamente es superado, se cambia a un modo de baja contención y de manera similar para el otro modo de operación.

### **3.5 Bajo flujo de información en la red**

Como se explicó previamente, para que un *nodo rampa* pueda acceder al canal necesita escuchar un paquete de la *avenida*, pero ¿qué pasa con los *odos rampa* cuando la tasa de datos que fluye por la *avenida* es muy baja o inexistente? Si existiera un bajo flujo de paquetes en la *avenida*, el *nodo rampa* esperaría más tiempo del que en verdad debería para transmitir sus paquetes a la *avenida*. Cuando existe un tráfico bajo en la *avenida*, el *nodo rampa* podría contender por el canal de manera normal pues el riesgo de intervenir con el flujo de datos de la *avenida* es bajo. En el peor de los casos, el *nodo rampa* esperaría por un paquete de la *avenida* que quizá no exista, por lo que su paquete jamás sería transmitido.

Cuando la tasa de datos sobre la *avenida* es muy baja se le puede otorgar a los *nodos rampa* la capacidad de contender por el canal como se define en el estándar. A cambio de poder contender por el canal de manera habitual, los *nodos rampa* pierden el acceso directo al canal y utilizan una CW grande en BEB. Además, si el *nodo rampa* recibe un paquete de la *avenida* mientras esta esperando para iniciar una transmisión normal, deberá crear y transmitir un nuevo paquete y seguir con el proceso propuesto para acceder a la *avenida*.

Para el protocolo de comunicación propuesto en esta tesis es más factible que los *nodos rampa* accedan a la *avenida* como se definió antes, a que transmitan sus paquetes de manera habitual. Intentando darle prioridad al proceso de acceso definido para los *nodos rampas*, estos nodos utilizan un valor de CW grande al contender por el canal de manera usual.

### **3.6 Resumen de capítulo**

En este capítulo se describen detalladamente las distintas ideas y soluciones que constituyen al protocolo de comunicación propuesto. Se introduce el concepto de *avenidas*, se señalan las características de los nodos que las componen, y la manera en la que fluyen los datos sobre éstas. Asimismo, se describe el comportamiento de los nodos que pueden acceder a la *avenida* conocidos como *nodos rampa*, y se define bajo que circunstancias estos nodos pueden utilizar el canal. A lo largo del capítulo se precisan los problemas generados por la definición de las *avenidas* y los *nodos rampa*, y se proponen soluciones a dichos problemas.

# 4

## Implementación de las avenidas

Para evaluar el desempeño de las avenidas se utilizó el simulador de redes (*Network Simulator version 2*) (NS-2), que fue modificado de tal manera que describiera el comportamiento descrito en el capítulo previo.

### 4.1 Simulador NS-2

El NS-2 es un simulador de redes de eventos discretos surgido como una variante del simulador REAL a finales de la década de los 80's. Este simulador fue desarrollado inicialmente por la Universidad de California conjuntamente con la Universidad de Cornell. Desde su aparición despertó un gran interés debido a la necesidad de contar con herramientas que tuvieran la capacidad de reproducir de manera detallada el comportamiento de diferentes topologías de red tanto cableadas como inalámbricas. En consecuencia, el NS-2 ha sido respaldado para su desarrollo por organizaciones como DARPA, entre muchas otras, que se han unido a este esfuerzo con el paso de los años.

El NS-2 es ampliamente utilizado debido a su vasto desarrollo, pues soporta una gran cantidad de mecanismos pertenecientes a las distintas capas del modelo OSI. A medida que se agregan o se corrigen los módulos que componen al simulador, los resultados de las simulaciones son más reales y detallados. Además, se intenta que el simulador esté a la vanguardia con los protocolos de comunicación más recientes.

El simulador cuenta con una licencia de libre distribución, hecho que lo vuelve más atractivo. Debido a lo anterior, además de permitir analizar el comportamiento de las redes de datos manipulando los mecanismos disponibles, otorga la posibilidad de implementar nuevas propuestas de comunicación. Para que el simulador muestre el comportamiento de dichas propuestas es necesario modificar los módulos existentes o agregar nuevos módulos. Por otro

lado, se debe ser consciente de que cada una de las versiones del simulador, que aunque mejoran respecto a la anterior, no están terminadas ni aseguran mostrar el comportamiento correcto.

NS-2 utiliza dos lenguajes de programación: C++ y OTcl (*Object-oriented Tool command Language*), que son utilizados conjuntamente por el simulador para realizar tareas diferentes y que se vinculan entre sí mediante TclCL (*Tool command language with CLasses*).

Los resultados de la simulación son archivos donde se señalan los eventos ocurridos tales como: la transmisión, la recepción, el reenvío o el descarte de los paquetes. Se registra además, el identificador del nodo en donde ocurrió el evento y se asenta el tiempo en el que aconteció éste. Se muestran también parámetros importantes de los paquetes originados por las distintas conexiones tales como: su tamaño en función de la capa del modelo OSI en la que se encuentran, el nodo origen y el destino de los mismos, el tipo (RTS, CTS, Datos, ACK, etc.), entre otros valores que suelen ser de interés general.

El usuario puede analizar los datos arrojados por el simulador de la manera que más le convenga en función de sus necesidades. Existen herramientas útiles para interpretar los resultados obtenidos como el animador de red (*Network AniMator*) (NAM) y XGraph. NAM permite observar el comportamiento de la red simulada de manera gráfica mientras que con XGraph se pueden generar distintas gráficas en función de los resultados obtenidos en la simulación.

#### 4.1.1 Ejemplo de un *script* Tcl utilizado en NS-2

En seguida se presenta un ejemplo típico de un *script* Tcl con la finalidad de mostrar los elementos que deben definirse para realizar una simulación en NS-2. La topología consta de 3 nodos que se comunican de manera inalámbrica mediante el protocolo definido en el estándar IEEE 802.11 como se muestra en la figura 4.1. Existe una aplicación en el nodo 0 que genera 10 paquetes por segundo cuyo destino es el nodo 2. Los nodos que componen la conexión (0 y 2) no pueden comunicarse directamente por lo que utilizan un nodo intermedio (1).

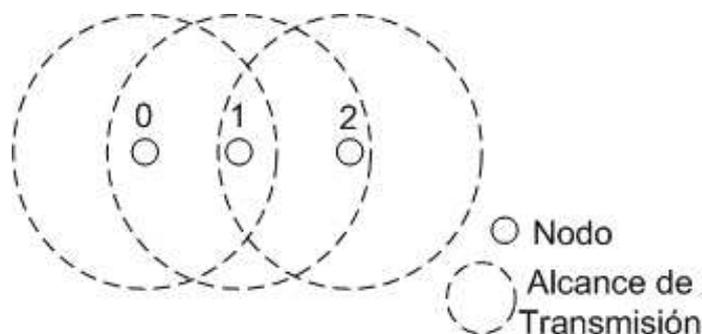


Figura 4.1: Topología definida en el *script* Tcl para la simulación en NS-2.

```

#=====
#Definición de valores importantes.
#=====
set val(chan) Channel/WirelessChannel ;# Tipo de canal de comunicación.
set val(prop) Propagation/TwoRayGround ;# Modelo de propagación.
set val(netif) Phy/WirelessPhy ;# Tipo de interfaz de red.
set val(mac) Mac/802.11 ;# Protocolo de capa MAC utilizado.
set val(ifq) Queue/DropTail/PriQueue ;# Tipo de cola.
set val(ll) LL ;# Capa de enlace.
set val(ant) Antenna/OmniAntenna ;# Modelo de antena.
set val(ifqlen) 50 ;# Número máximo de paquetes que puede almacenar la cola.
set val(nn) 3 ;# Número de nodos móviles.
set val(rp) DSR ;# Protocolo de enrutamiento.
set val(x) 500 ;# Dimension X de la topología en metros.
set val(y) 100 ;# Dimension Y de la topología en metros.
set val(time) 5.1 ;# Tiempo de simulación en segundos.
set tasa_datos 0.1 ;# Intervalo de creación de paquetes.

proc finish {} {
    global ns_ namfile analyze
    $ns_ flush-trace
    close $namfile
    close $analyze
    exec nam out.nam &
    exit 0 }

#=====
# Programa principal.
#=====
# Inicialización de variables globales.
set ns_ [new Simulator]
set analyze [open simple.tr w]
$ns_ trace-all $analyze
set namfile [open out.nam w]
$ns_ namtrace-all-wireless $namfile $val(x) $val(y)

#Configuración del escenario
set topo [new Topography]
$topo load_flatgrid $val(x) $val(y)

# Creación del número necesario de God's
create-god $val(nn)

#Configuración de los nodos
$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) \
    -topoInstance $topo \
    -channelType $val(chan) \
    -agentTrace ON \
    -routerTrace ON \
    -macTrace ON \
    -movementTrace OFF \

for { set i 0 } { $i <$val(nn) } {incr i} {
    set n($i) [$ns_ node]
    $n($i) random-motion 0 ;# Se deshabilita el movimiento de los nodos.
    $ns_ at 0.0 "$n($i) setdest 10.0 10.0 0.0" }

# Definición de la ubicación de los nodos.
$n(0) set X_ 50.0
$n(0) set Y_ 50.0
$n(0) set Z_ 0.0
$n(1) set X_ 250.0

```

```

$n(1) set Y_ 50.0
$n(1) set Z_ 0.0
$n(2) set X_ 450.0
$n(2) set Y_ 50.0
$n(2) set Z_ 0.0

# Definición de las conexiones.
#Declaración de un agente nulo para recibir los paquetes.
set null_(0) [new Agent/Null]
$ns_ attach-agent $n(2) $null_(0)

#Declaración del tipo de conexión.
set udp_(0) [new Agent/UDP]
$ns_ attach-agent $n(0) $udp_(0)

# Generador de tráfico constante.
set cbr_(0) [new Application/Traffic/CBR]
$cbr_(0) set packetSize_ 450
$cbr_(0) set interval_ $tasa_datos
$cbr_(0) attach-agent $udp_(0)
$ns_ connect $udp_(0) $null_(0)

#Inicio y final de la creación de tráfico.
$ns_ at 0.1 "$cbr_(0) start"
$ns_ at 5 "$cbr_(0) stop"

#Indicar a los nodos el fin de la simulación.
for { set i 0 } { $i <$val(nn)} {incr i}{
    $ns_ at $val(time) "$n($i) reset"}

$ns_ at $val(time) "finish" #Se llama al procedimiento finish definido al inicio.
$ns_ at $val(time) "puts \"NS EXITING...\"; $ns_ halt"

#Finalmente se inicia la simulación.
puts "Starting Simulation..."
$ns_ run

```

Al inicio del *script* se declaran algunas variables seguidas de un comentario que indican el uso que se les da. Después se define un procedimiento para manejar los archivos creados por la simulación y ejecutar el NAM. Este procedimiento, como se puede observar en las últimas líneas del *script*, es invocado en un tiempo definido por el usuario.

En el programa principal se define una variable que manejará la simulación y las encargadas de los archivos generados en la misma. El archivo *simple.tr* registrará los eventos acontecidos de una manera detallada, mientras que el *namfile* será utilizado por el NAM para desplegar los resultados en una animación.

Posteriormente se crean y configuran los elementos que participan en la simulación como: la topología, los nodos y su posición, las conexiones y sus características. Se indican los tiempos que manejarán los eventos como: el inicio y fin de la simulación y el intervalo donde la aplicación generará datos. Como se puede apreciar, los comandos utilizados por Tcl son intuitivos y no necesitan una mayor explicación.

Los eventos que se registran en el archivo *simple.tr* son: envío (s), recepción (r) y reenvío (f). En cada uno de estos eventos se marca el nodo y el tiempo en el que sucedieron, además de la capa en la que se encuentra el paquete.

## 4.2 Programación de las avenidas y rampas en NS-2

En esta sección se describen los cambios realizados al simulador para lograr que los nodos que constituyen las avenidas y los nodos rampa, mostraran el comportamiento descrito en el Capítulo 3. Debido a que la propuesta de esta tesis se basa en el protocolo definido en el estándar IEEE 802.11, la mayor parte de las modificaciones e incorporaciones de nuevos módulos se realizaron en el archivo del simulador que tiene por nombre *mac-802\_11.cc*.

Para realizar la implementación de la propuesta se debe tener en mente que cada uno de los nodos de la simulación crea una instancia de la clase definida en C++ que describe los mecanismos necesarios para simular el protocolo de capa MAC. Que cada nodo cree una instancia de la misma clase implica que siempre se accede a los mismos mecanismos y funciones.

Considerando lo anterior, en el archivo *mac-802\_11.cc* se realizó la distinción de los nodos que forman parte de la avenida, los nodos rampa y los nodos que presentan un comportamiento normal. Se muestra la la forma en que se distinguieron los nodos y la asignación del valor de CW (*cw\_*) otorgado.

---

### Inicialización de la clase MAC

---

```
1 if El nodo forma parte de la avenida then
2     control=1;
3     cw_=1;
4 else
5     if El nodo es una rampa then
6         control=2;
7         cw_=0;
8     else
9         control=0;
10        cw_=31;
11    end
12 end
```

---

Una vez diferenciados los nodos se modificaron distintos módulos que componen el archivo *mac-802\_11.cc*. Para facilitar la explicación de las modificaciones realizadas se muestra el flujo de funciones que se invocan a nivel de capa MAC en el envío y recepción de un paquete. En el flujo mostrado en seguida se ilustra el intercambio de paquetes RTS/CTS, el envío y recepción del paquete de datos, y el envío y recepción del paquete ACK. El paquete de datos es transmitido por el nodo A y tiene como destino al nodo B. Se han utilizado dos tipos de flechas para ilustrar el flujo de funciones. Las flechas  $\Rightarrow$  ilustran la manera en que las funciones se van invocando. Por su parte, las flechas  $\Uparrow$  señalan el momento en el que se inicializan los temporizadores utilizados por el simulador. Las funciones que siguen a las flechas  $\Uparrow$  son ejecutadas al expirar el temporizador activado anteriormente.

*Envío del paquete RTS.*

Nodo A:  $\text{recv}() \Rightarrow \text{send}() \Rightarrow \text{sendDATA}() \Rightarrow \text{sendRTS}() \uparrow \text{mhBackoff\_start}() \Rightarrow \text{backoffHandler}() \Rightarrow \text{check\_pktRTS}() \Rightarrow \text{transmit}() \uparrow \text{mhSend\_start}()$

*Recepción del paquete RTS y transmisión del paquete CTS.*

Nodo B:  $\text{recv}() \uparrow \text{mhRecv\_start}() \Rightarrow \text{recv\_timer} \Rightarrow \text{recvRTS}() \Rightarrow \text{sendCTS}() \Rightarrow \text{tx\_resume}() \uparrow \text{mhDefer\_start}() \Rightarrow \text{rx\_resume}() \Rightarrow \text{deferHandler}() \Rightarrow \text{check\_pktCTRL}() \Rightarrow \text{transmit}() \uparrow \text{mhSend\_start}()$

*Recepción del paquete CTS y transmisión de datos.*

Nodo A:  $\text{recv}() \uparrow \text{mhRecv\_start}() \Rightarrow \text{recv\_timer} \Rightarrow \text{recvCTS}() \Rightarrow \text{tx\_resume}() \uparrow \text{mhDefer\_start}() \Rightarrow \text{rx\_resume}() \Rightarrow \text{deferHandler}() \Rightarrow \text{check\_pktTx}() \Rightarrow \text{transmit}() \uparrow \text{mhSend\_start}()$

*Recepción de datos y transmisión del paquete ACK.*

Nodo B:  $\text{recv}() \uparrow \text{mhRecv\_start}() \Rightarrow \text{recv\_timer} \Rightarrow \text{recvDATA}() \Rightarrow \text{sendACK}() \Rightarrow \text{tx\_resume}() \uparrow \text{mhDefer\_start}() \Rightarrow \text{rx\_resume}() \Rightarrow \text{deferHandler}() \Rightarrow \text{check\_pktCTRL}() \Rightarrow \text{transmit}() \uparrow \text{mhSend\_start}() \Rightarrow \text{send\_timer}() \Rightarrow \text{tx\_resume}()$

*Recepción de ACK.*

Nodo A:  $\text{recv}() \uparrow \text{mhRecv\_start}() \Rightarrow \text{recv\_timer} \Rightarrow \text{recvACK}() \Rightarrow \text{tx\_resume}() \Rightarrow \text{rx\_resume}()$

Cada una de las funciones anteriores tienen un nombre ligado con las acciones que realizan. Basados en este diagrama se explican los cambios realizados.

#### 4.2.1 Nodos de la avenida

Los nodos de la avenida realizan el envío de paquetes de manera habitual pues el valor de CW es asignado cuando se inicializan cada uno de los nodos. En cuanto a la recepción de paquetes, solamente los nodos de la avenida que descartan los paquetes utilizados por los nodo rampa muestran un comportamiento diferente. Este comportamiento se define en la función  $\text{recv\_timer}()$ .

---

Función `recv_timer()`: Filtrado MAC de algunos nodos de la avenida.

---

```
1 if El nodo central de los 3 nodos que escucha un nodo rampa ha recibido un paquete
  then
2   if Si el paquete recibido proviene de una rampa then
3     if si el paquete recibido contiene el paquete que esta esperando a ser
      transmitido then
4       Desecha el paquete de datos;
5       if Existe un paquete RTS then
6         Desecha RTS;
7       end
8     end
9   end
10 end
```

---

Solamente los nodos terminales de la avenida necesitan de un mayor número de modificaciones al implementar la solución del problema de terminal expuesta y trabajar con dos modos de operación diferentes. Como estos comportamientos se relacionan con la retransmisión de paquetes se indica el flujo de funciones existente cuando falla un envío.

*Envío y retransmisión de un paquete RTS.*

Nodo A: `recv()`  $\Rightarrow$  `send()`  $\Rightarrow$  `sendDATA()`  $\Rightarrow$  `sendRTS()`  $\Uparrow$  `mhBackoff.start()`  $\Rightarrow$  `backoffHandler()`  $\Rightarrow$  `check_pktRTS()`  $\Rightarrow$  `transmit()`  $\Uparrow$  `mhSend.start()`  $\Rightarrow$  `send_timer()`  $\Rightarrow$  `RetransmitRTS()`  $\Rightarrow$  `tx_resume()`  $\Uparrow$  `mhBackoff.start()`  $\Rightarrow$  `backoffHandler()`  $\Rightarrow$  `check_pktRTS()`  $\Rightarrow$  `transmit()`  $\Uparrow$  `mhSend.start()`

Para lograr que el nodo terminal tuviera el comportamiento descrito en el capítulo 3 se definen parte de los mecanismos necesarios en la función `RetransmitRTS()`. En esta función se comprueba que no se halla superado el número máximo de intentos de retransmisión permitido, pues de ser así el paquete de datos, al igual que el paquete RTS, son descartados reiniciando el valor de `cw_`. Para los nodos de la avenida y los nodos rampa se respeta la CW otorgada para darles privilegios.

Para gestionar el modo de contención en el que funciona el nodo terminal de la avenida se utiliza una bandera cuyo valor inicial es cero y la cual se modifica considerando 2 valores. Uno de ellos es el número de paquetes descartados consecutivamente y el otro el número de transmisiones exitosas sucesivas. Estos valores se almacenan en las variables `DrPa` (*Dropped Packets*) y `SuTx` (*Successful Transmission*). La variable `DrPa` aumenta su valor en la función `RetransmitRTS()` al cumplirse con la condición necesaria para descartar el paquete, y por ende `SuTx` toma el valor de cero. Cuando `flag` es igual a cero y `DrPa` rebasa el umbral establecido por el administrador de red, para esta tesis 2, `flag` toma un valor de uno cambiando al modo de alta contención. Por su parte la variable `SuTx` aumenta en uno cada vez que se recibe un

ACK, es decir, dentro de `recvACK()`. En `recvACK()` siempre que el valor de flag sea uno y se rebase el umbral definido, para esta tesis 15, se cambia al modo de baja contención poniendo el valor de flag en cero. Asimismo, por haber completado una transmisión de manera exitosa `DrPa` toma el valor de cero.

#### 4.2.2 Nodos rampa

Para definir el comportamiento de los nodos rampa fue necesario modificar la función `recv()`. Cuando un nodo rampa desea transmitir un paquete se verifica que el canal este libre y que las clases utilizadas en el NS-2, para controlar los tiempos que debe de esperar un nodo antes de utilizar el canal, no estén siendo ejecutadas. Una de estas clases es la encargada de simular el tiempo que el canal debe de estar libre para que un nodo pueda utilizar el canal, denominada como *DeferTimer* cuyo control se realiza mediante *mhDefer\_*. La otra clase es *BackoffTimer* controlada mediante *mhBackoff\_*, que se encarga del tiempo que un nodo espera como resultado del BEB.

Si la verificación anterior se cumple, el nodo rampa asume que no existe flujo de datos en la avenida y prepara una transmisión normal asignando el valor de la variable *cw\_* en 1023. Además, se hace una copia del paquete en *PktQ\_* para utilizarlo en el proceso para acceder a la avenida si se recibe un paquete proveniente de ella.

Cuando el canal esta ocupado o alguno de las clases se encuentra activa, se debe verificar el estado del *BackoffTimer*. Es necesario realizar esta verificación por que si la condición para asumir que no existe flujo de datos en la avenida no se cumple debido a que el *BackoffTimer* esta ocupado, el paquete debe encolarse en *pktQ\_* para decidir que hacer con él cuando el temporizador expire. Esta acción esta fundamentada en el hecho de que el *BackoffTimer* se activa luego de una transmisión exitosa. Si se ha completado una transmisión exitosa, el nodo rampa podría asumir que existe flujo en la avenida de manera errónea, pues queda la posibilidad de que el canal este libre y que el *BackoffTimer* este siendo ocupado. Cuando el *BackoffTimer* no esta activo la condición no se cumple por que el canal esta ocupado, por lo que se inicia el BEB con *cw\_=1023* y se guarda una copia del paquete que se desea transmitir en *pktQ\_*.

Del flujo de funciones mostrado con anterioridad note que para transmitir un paquete, el *BackoffTimer* es activado dentro de la función `sendRTS()`. Al momento de activar este temporizador se ha creado el paquete RTS y se ha preparado el paquete de datos que se desea transmitir. El paquete de datos a transmitir se almacena en *pktTx\_*.

Cuando un nodo rampa tienen un paquete almacenado en *pktTx\_* espera a que expire el *BackoffTimer* para transmitirlo de manera usual. Si antes de que el *BackoffTimer* expire o se tiene un *pktQ\_* y el nodo rampa recibe un paquete de la avenida, se inicia el procedimiento para crear un nuevo paquete que contenga al paquete recibido y a *pktTx\_* o *pktQ\_*. Este procedimiento se inicia en `recv_timer()`.

---

Función `recv_timer()`: Filtrado MAC de los nodos rampa e inicio del acceso a la avenida.

---

```
1 if Es un nodo rampa then
2   if El paquete recibido es de datos y proviene de la avenida then
3     if Existe pktTx_ then
4       Remueve encabezado MAC a pktTx_;
5       pktQ_=pktTx_;
6       Desecha RTS;
7     end
8     if Existe pktQ_ then
9       if Suma de los tamaños de pkt_Q y el paquete recibido < 1500 then
10        cw_=0;
11        Se invoca a sendtwopkts();
12      else
13        Desecha el paquete recibido;
14      end
15    else
16      if El paquete recibido no es para el nodo rampa then
17        Desecha el paquete recibido;
18      end
19    end
20  end
21 end
```

---

En la función `sendtwopkts()`, invocada en la línea once del pseudocódigo anterior, se elimina el encabezado MAC del paquete recibido para después acceder a la función `cloneHdrs()`. La función `cloneHdrs()` recibe como argumentos de entrada el `pktQ_` y el paquete recibido. En `cloneHdrs()` se crea el paquete IP que será ingresado a la avenida, denominado como `NPkt_`, y se especifican todos los valores del encabezado común para que llegue a su destino de manera correcta. Este encabezado es utilizado por el NS-2 para definir algunos valores que facilitan la simulación. En el encabezado común se especifica: el tamaño del paquete, el identificador único del paquete, una bandera que sirve para indicar si el paquete tiene errores, entre otros valores. Finalmente, `NPkt_` se encapsula en una trama de capa 2 donde se indica la dirección MAC origen y destino.

Ya que el NS-2 entrega los paquetes a su destino en función de la información que contiene en el encabezado común, y con la novedad de que ahora un paquete puede contener 2 o más paquetes, fue necesario declarar nuevas variables en este encabezado. Las nuevas variables fueron declaradas en el archivo `packets.h`, para conseguir que la entrega de los paquetes se diera de manera correcta.

Para almacenar la información de `pktQ_` y la del paquete recibido en el encabezado común de `NPkt_` se accede a la función `ImpValues()`. En `ImpValues()` se guarda información como:

los identificadores de los paquetes, los nodos origen y destino, los puertos que se utilizan y una bandera que indica que el paquete proviene de un nodo rampa. Una vez listo NPkt\_, se elimina el PktQ\_ y se invoca a la función `tricksend()` donde se inicia el proceso de envío. Para los nodos rampa `tricksend()` reemplaza a `send()` en el flujo de funciones.

Para el caso donde un nodo rampa recibe un paquete de la avenida creado por otro nodo rampa, se procede de manera análoga, pero esta vez se establecen los valores necesarios para identificar a los 3 paquetes que contendrá NPkt\_. Para asignar estos valores se utilizan funciones semejantes a las descritas antes. Cuando NPkt\_ está listo se invoca a la función `tricksend()` para enviar el paquete.

Debido a la modificación que se hizo a la función `recv()` para que los nodos rampa pudieran contender por el canal, fue necesario hacer lo pertinente con el `backoffHandler()`, invocado al expirar el *BackoffTimer*. Por inicio de cuentas, se verifica que el evento que desencadenó el uso del *BackoffTimer* no sea el resultado de una transmisión exitosa. Para saber si el *BackoffTimer* no fue activado en `recvACK()` se utiliza una bandera que se activa en esta función. Si no se hiciera esta verificación se caería en un error, pues el nodo rampa haría una transmisión después de un *BackoffTimer* equivocado.

Si el *BackoffTimer* no expiró al haber recibido un paquete ACK, se revisa si existe un paquete en `pktQ_`. Si el `pktQ_` no está vacío se ha esperado el tiempo suficiente para iniciar su transmisión. Para que el nodo rampa pudiera realizar una transmisión inmediata, se agregó una función nombrada `sendnow()`. La función `sendnow()` es semejante a `send()` al recurrir a `sendRTS()` y `sendDATA()`, pero difiere con ella al no activar nuevamente el *BackoffTimer*.

En caso de confirmar que el *BackoffTimer* fue desencadenado en `recvACK()` y que existe un `pktQ_` se inicia, si el canal está disponible, el *BackoffTimer* para enviar el paquete, de lo contrario se llama BEB. Para cualquier situación se utiliza un `cw_=1023` siguiendo con la idea de darle prioridad al mecanismo definido para que los nodos rampa utilicen las avenidas.

Una vez que se implementó lo descrito a lo largo de este capítulo, se dió por terminada la modificación del archivo *mac-802\_11.cc*. Las funciones y variables utilizadas fueron declaradas en la clase MAC definida en el archivo *mac.h*, especificando los valores iniciales necesarios dentro de *mac.cc*.

Para poder obtener los resultados del protocolo propuesto reflejados en el archivo de salida del simulador, se modificó el archivo *cmu-trace.cc*. Mediante estas modificaciones se reportan, en el archivo de salida, los valores individuales de cada uno de los paquetes transportados en los paquetes generados por los nodos rampa. Para lograr esto fue necesario declarar en el archivo *cmu-trace.h* un par de funciones llamadas `twopacksinone()` y `threepacksinone()` que se invocan en función de un valor que tienen los paquetes originados por los nodos rampa en el encabezado común.

### 4.3 Control de la tasa de datos en la avenida

El control de la tasa de datos que viaja por la avenida se realiza mediante la correcta definición de los generadores tráfico existentes en la topología utilizada en la simulación. El simulador cuenta con distintos generadores de tráfico predefinidos que son configurados por el usuario. A estos simuladores de aplicaciones se les indica el tipo de conexión que utilizan, el tamaño de los paquetes que crean, la cantidad de paquetes que crean por segundo, si los paquetes son creados de manera periodica o aleatoria, el tiempo inicial y final de la creación de paquetes, entre otros valores. Es necesario crea un agente que recibirá los paquetes procedente de un generador de tráfico, y adjuntarlo al nodo destino.

Se muestra un ejemplo de la configuración de una conexión UDP entre el nodo 0 y el nodo 1. La aplicación es del tipo CBR (*Constant Bit Rate*) y crea 100 paquetes de 450 bytes por segundo. Los paquetes son creados en un intervalo aleatorio de tiempo durante 4.9s.

```
# Definición de las conexiones.
#Declaración de un agente nulo para recibir los paquetes.
set null_(0) [new Agent/Null]
$ns_ attach-agent $n(1) $null_(0)
#Declaración del tipo de conexión.
set udp_(0) [new Agent/UDP]
$ns_ attach-agent $n(0) $udp_(0)
# Generador de tráfico constante.
set cbr_(0) [new Application/Traffic/CBR]
$cbr_(0) set packetSize_ 450
$cbr_(0) set interval_ 0.01
$cbr_(0) set random_ 1
$cbr_(0) attach-agent $udp_(0)
$ns_ connect $udp_(0) $null_(0)
#Inicio y final de la creación de tráfico.
$ns_ at 0.1 "$cbr_(0) start"
$ns_ at 5 "$cbr_(0) stop"
```

Finalmente, los datos obtenidos de las simulaciones fueron filtrados con *bashs* cuyas salidas sirvieron de argumento para *scripts* de Matlab, que dieron como resultado final las gráficas reportadas en esta tesis.

### 4.4 Resumen de capítulo

En este capítulo se presenta una breve introducción a NS-2. Se muestra un ejemplo de *script* TCL donde se define un sencillo escenario de simulación. Por otro lado, se estipulan las modificaciones hechas al simulador, necesarias para que éste reprodujera el comportamiento de las avenidas, los nodos rampa y demás mecanismos del protocolo. Por último, se explica la forma en que se controla el tráfico existente en las avenidas.

# 5

## Pruebas y resultados

### 5.1 Experimento 1: Impacto del valor de $CW_{min}$ en el retardo total de los paquetes

Previamente a la definición del protocolo se realizaron experimentos en los cuales se modificó el valor inicial de la CW de algunos nodos. Estas pruebas se realizaron con la finalidad de observar el impacto que tiene el valor de CW sobre el retardo total de los paquetes.

En la Figura 5.1 se muestra la topología de red utilizada para realizar estos experimentos.

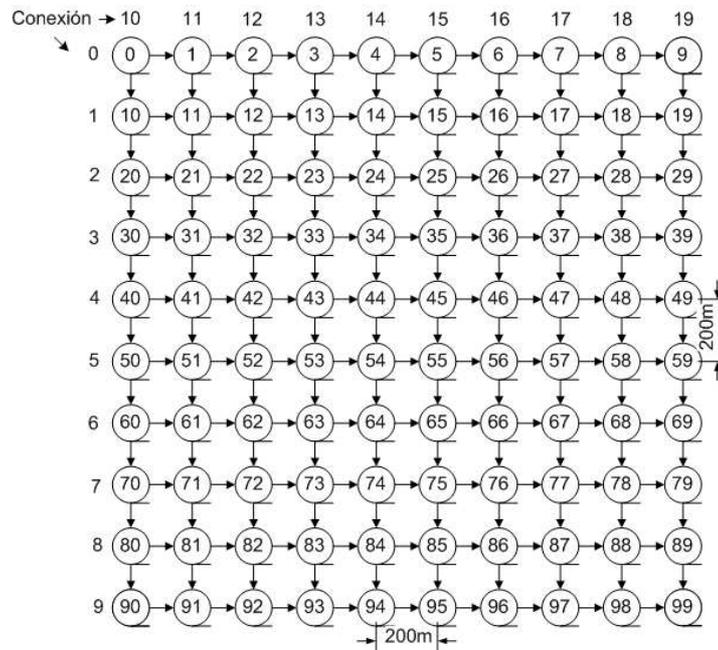


Figura 5.1: Topología de la red utilizada en NS2

La distancia entre los nodos de la topología tanto vertical como horizontal es de 200m. Debido a que el alcance de transmisión de los nodos es de 250m se asegura que se comuniquen directamente con los nodos ubicados inmediatamente arriba, abajo, a la derecha y a la izquierda. Por ejemplo, el nodo 45 puede comunicarse directamente con el nodo 35, 44, 46 y 55.

Los nodos utilizan el protocolo de capa MAC definido en el estándar IEEE 802.11. En las simulaciones se utilizó un proceso de encaminamiento estático con la finalidad de observar el comportamiento de la capa MAC. Utilizando este tipo de encaminamiento se elimina la carga de paquetes relacionados con el protocolo a nivel de red.

El tráfico que fluye en la red es originado por 20 conexiones descritas en la Tabla 5.1. Cada uno de los nodos origen de una conexión simulan una aplicación que genera tráfico a diferentes tasas de valores, utilizando el protocolo de datagrama de usuario (*User Datagram Protocol*) (UDP) como protocolo de transporte. El tamaño de los paquetes generados por las aplicaciones es de 512 bytes.

Conexión	Nodos	
	Origen	Destino
0	0	9
1	10	19
2	20	29
3	30	39
4	40	49
5	50	59
6	60	69
7	70	79
8	80	89
9	90	99
10	0	90
11	1	19
12	2	29
13	3	39
14	4	49
15	5	59
16	6	69
17	7	79
18	8	89
19	9	99

Cuadro 5.1: Conexiones existentes en la topología de red mostrada en la Figura 5.1

La forma en que los paquetes viajan dentro de la topología se ilustra, con flechas, en la figura 5.1.

Dentro de la topología de la Figura 5.1 se modificó el valor de  $CW$  para la conexión número 4, constituida por los nodos 40-49. Los nodos de esta conexión utilizan un valor de  $CW_{min}$  menor comparado con los nodos de las otras conexiones. Los nodos que no forman parte de la conexión número 4 inician el algoritmo BEB con una  $CW_{min}=31$ , mientras que los nodos que sí están involucrados con dicha conexión  $CW_{min}$  varió en diferentes simulaciones.

Bajo este escenario, se realizaron distintas simulaciones variando el número de paquetes por segundo generados por las aplicaciones y los valores iniciales de  $CW$  para la conexión privilegiada. Con los resultados de las simulaciones se generaron un par de gráficas, para cada caso, en las que se muestra el valor promedio de paquetes recibidos y el retardo total promedio de éstos en función de los paquetes por segundo generados por las aplicaciones.

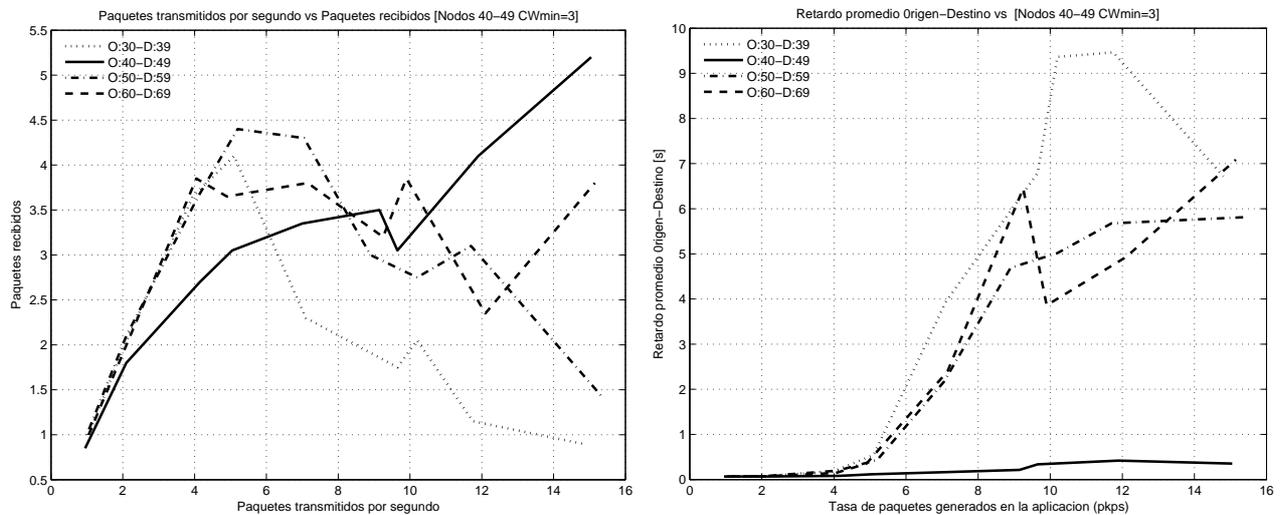


Figura 5.2: Paquetes recibidos y retardo total promedio de los paquetes. Nodos privilegiados con una  $CW_{min}=3$ .

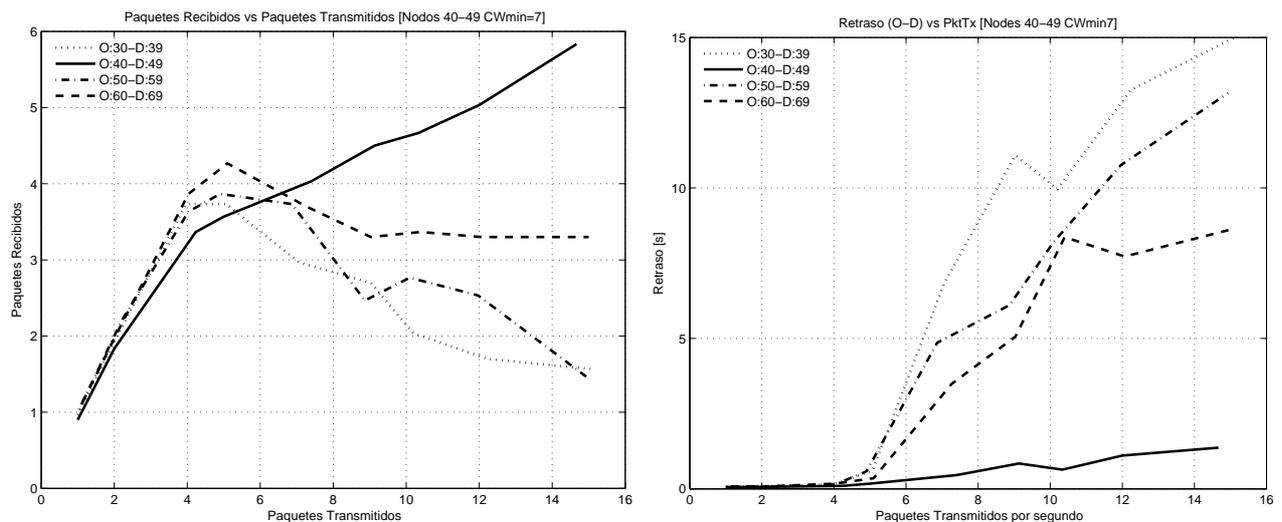


Figura 5.3: Paquetes recibidos y retardo total promedio de los paquetes. Nodos privilegiados con una  $CW_{min}=7$ .

En las Figuras 5.2 y 5.3 se reportan los resultados de las conexiones 3, 4, 5 y 6 con el propósito de observar el comportamiento de la conexión diferenciada y compararla con tres conexiones en la que los nodos tienen un  $CW_{min}=31$ . En las gráficas de las Figuras 5.2 y 5.3 la  $CW_{min}$  de la conexión número 4 fue de 3 y 7, respectivamente.

En las gráficas se aprecia que, para todas las conexiones, el número de paquetes entregados al destino es bajo debido al gran flujo de datos que existe en la red. Por otro lado, se advierte que el uso de una  $CW_{min}$  menor en la conexión 4 impacta positivamente sobre el retardo total de los paquetes, al compararlo con el de los paquetes pertenecientes a las otras conexiones.

Como los resultados de la conexión privilegiada fueron satisfactorios, en cuanto a la relación que existe entre el tiempo de espera resultante del BEB y el retardo total promedio de los paquetes, se procedió a definir el protocolo descrito en el capítulo 3 considerando este hecho.

## 5.2 Experimento 2: Análisis del desempeño de la avenida

Con la finalidad de poder realizar comparaciones con el protocolo propuesto en esta tesis se definió un escenario de control donde se utiliza el protocolo de capa MAC del estándar. Este escenario se muestra en la Figura 5.4.

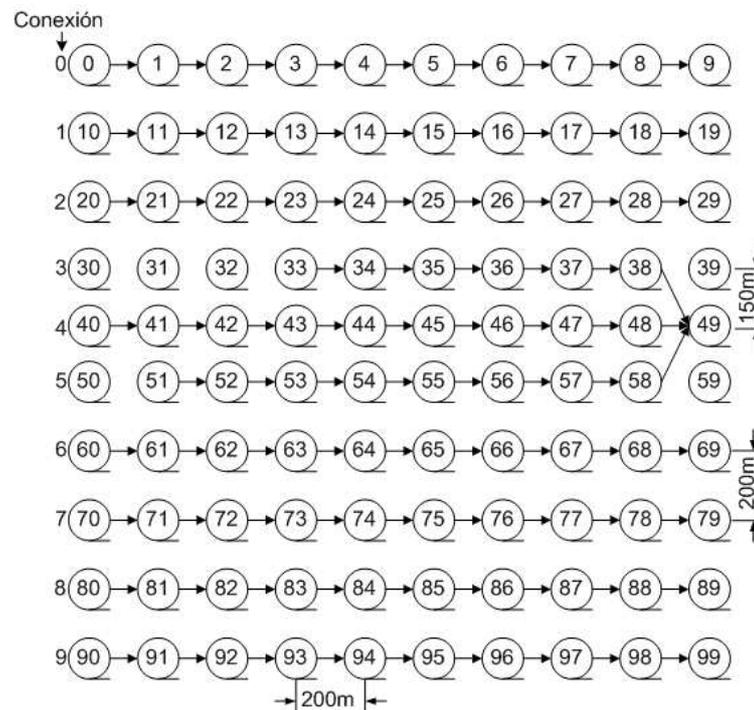


Figura 5.4: Topología de control utilizada en NS2.

Note que en la topología de la Figura 5.4 se han considerado las características que deben tener los nodos rampa en cuanto a ubicación espacial se refiere, según lo descrito en el capítulo 3. Esta particularidad se nota al observar que la distancia vertical que existe entre las líneas

de nodos marcadas con los números 3, 4 y 5 es de 150m. Para todos los demás nodos la distancia tanto vertical como horizontal es de 200m.

Se definieron 3 conexiones con un destino en común para poder comparar los resultados del escenario de control con los obtenidos cuando los nodos rampa utilizan la avenida. Todas las conexiones definidas en este escenario se resumen en la Tabla 5.2.

Conexión	Nodos	
	Origen	Destino
0	0	9
1	10	19
2	20	29
3	33	49
4	40	49
5	51	49
6	60	69
7	70	79
8	80	89
9	90	99

Cuadro 5.2: Conexiones existentes en la topología de red mostrada en la Figura 5.4

Cada nodo origen de las conexiones simulan una aplicación que genera tráfico a distintas tasas de paquetes por segundo dependiendo del experimento. Para todos los casos, los paquetes tienen un tamaño de 450 bytes y son transportados utilizando UDP. Además, la cola de almacenamiento utilizada por los nodos es del tipo *Droptail* con una capacidad de 100 paquetes.

Se utilizó un escenario igual al de control, pero en el cual se emplea el protocolo de comunicación propuesto en la presente tesis, introduciendo el uso de avenidas y rampas. En este escenario, ilustrado en la Figura 5.5, existen 2 nodos rampa, nodos 33 y 51, y una avenida constituida por los nodos 40 al 49.

Las flechas que se muestran en las Figuras 5.4 y 5.5 describen los saltos que siguen los paquetes en la red para alcanzar su destino. Los saltos que realizan los paquetes de las conexiones 3 y 5 para llegar a su destino en el escenario de control, Figura 5.4, al no haber nodos rampa difieren de los realizados cuando estos últimos nodos existen, Figura 5.5.

Para analizar el desempeño del protocolo propuesto se eligieron los siguientes parámetros de comparación:

- \* Porcentaje de paquetes que llegan a su destino. Para determinar el valor de este parámetro se obtuvo el número de paquetes creados por la aplicación y la cantidad de estos paquetes que llegaron a su destino.

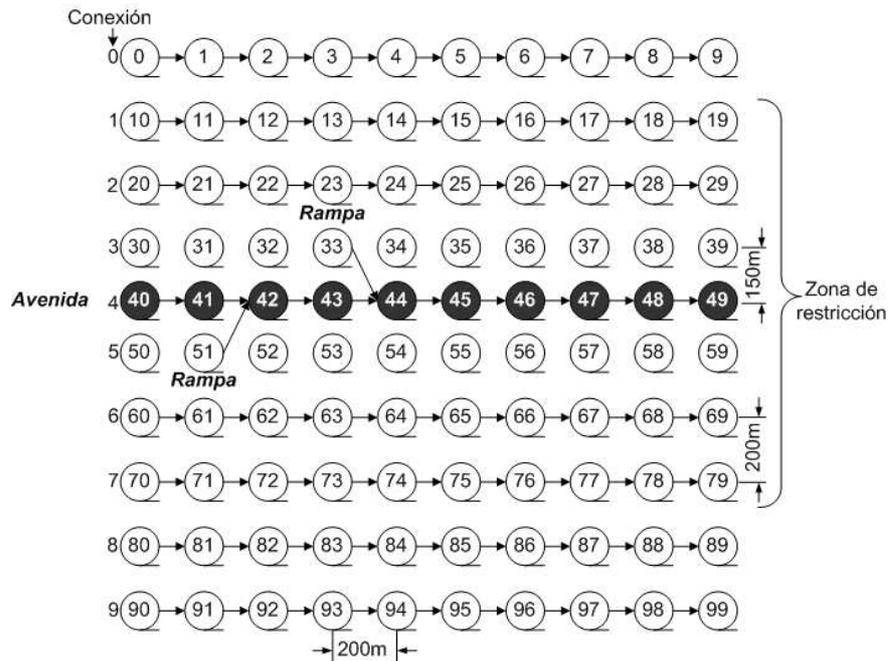


Figura 5.5: Escenario de simulación para el protocolo de comunicación propuesto, introducción de la avenida y rampas.

- \* Retardo total promedio de los paquetes. Para determinar el valor de este parámetro se calculó el promedio del retardo total de los paquetes que llegaron a su destino.

Se evaluó el caso donde el protocolo de comunicación propuesto presenta un comportamiento ideal y un caso más complejo denominado como caso aleatorio. Estos casos se describen en seguida:

- \* Caso ideal: se presenta cuando las aplicaciones existentes en la avenida y en los nodos rampa crean paquetes a intervalos de tiempo constante. Bajo estas condiciones se consigue que los nodos rampa accedan a la avenida utilizando cada paquete que fluye por ella, disminuyendo el retardo de encolamiento.
- \* Caso aleatorio: se presenta cuando las aplicaciones de la avenida y las de los nodos rampa generan paquetes de manera aleatoria. Esta aleatoriedad complica el flujo de datos por la avenida y el proceso de acceso de los nodos rampa, originando el encolamiento de los paquetes.

Para cada experimento se obtuvieron los resultados para el escenario de control y el escenario donde se utiliza la propuesta de esta tesis en el caso ideal y aleatorio.

Asimismo, como el protocolo propuesto fue concebido para funcionar en una red donde existe un flujo de datos por la avenida, se prohibió que los nodos rampa contendieran por el canal de manera normal. Por lo tanto, estos nodos solo utilizan la avenida mediante el proceso de acceso definido previamente.

En este experimento se evalúa el desempeño únicamente de la avenida, es decir, solo se activa la conexión número 4. La aplicación correspondiente a esta conexión genera datos a una tasa de 80 paquetes por segundo. Esta tasa de datos se calculó siguiendo con lo explicado en el Capítulo 3, utilizando, en todos los experimentos, una velocidad de transmisión de 2Mbps.

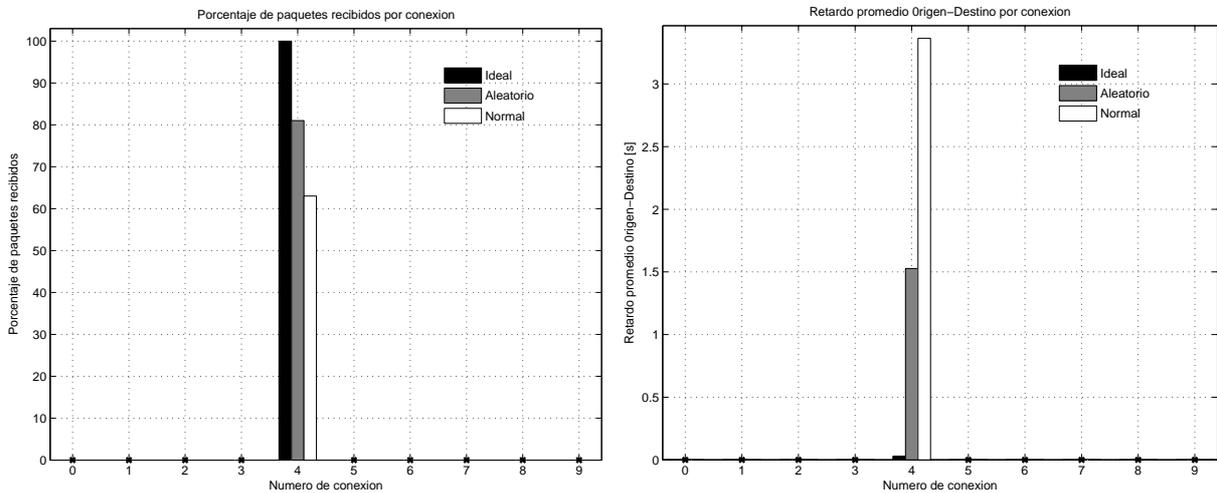


Figura 5.6: Comparación del porcentaje y el retardo promedio de los paquetes que llegan a su destino en la conexión 4. Caso ideal, aleatorio y el del escenario de control

En la gráfica de la Figura 5.6 se nota que para el escenario ideal se entregan todos los paquetes a su destino en un tiempo promedio de 28.5ms. Por otro lado, al existir aleatoriedad en la creación de los paquetes se percibe que la avenida se comporta de una manera inesperada, entregando solamente el 80 % de los paquetes con un retardo promedio total de alrededor de 1.5s.

El comportamiento imprevisible de la avenida al existir aleatoriedad, se genera por la cantidad de datos que crea la aplicación, aunado a que el nodo 40 inicia su operación utilizando el modo de baja contención. Al utilizar el modo de baja contención los paquetes generados por la aplicación deben esperar a ser transmitidos al intentar mitigar el problema de terminal expuesta. Mientras el nodo realiza esta espera los paquetes siguen siendo generados por la aplicación y se almacenan en la cola. Si el tiempo que el nodo espera para transmitir un paquete es demasiado largo, la cola de almacenamiento puede alcanzar su valor máximo y desbordar paquetes. Por lo tanto, los paquetes que se logran transmitir tienen en promedio un retardo total alto, y el porcentaje de paquetes entregados a su destino disminuye al no poder almacenarse todos en la cola del nodo. Cuando no se completan los envíos y los paquete son descartados el nodo 40 cambia de modo de operación. Este cambio se da aún cuando no existen nodos, además de los que componen la avenida, intentando utilizar el canal.

### 5.3 Experimento 3: Estudio del tráfico en la avenida al existir aleatoriedad

Por lo acontecido en el experimento anterior, se decidió valorar el desempeño de la avenida en función de la cantidad de paquetes por segundo generados por la aplicación. Mediante este experimento se puede conocer la cantidad de información que puede tolerar la avenida mientras mantiene su característica de entrega rápida al destino. Los valores en que se varió la

cantidad de paquetes por segundo generados por la aplicación fueron: 1, 5, 10, 15, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120 y 150. Los resultados obtenidos se muestran en la Figura 5.7.

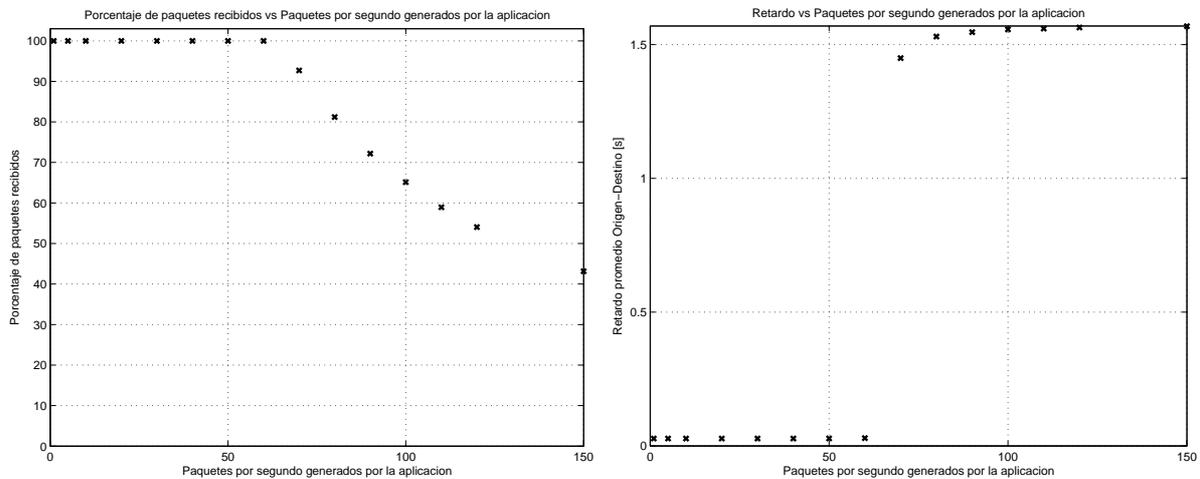


Figura 5.7: Porcentaje y el retardo promedio de los paquetes de la avenida que llegan a su destino en función de los paquetes por segundo creados por la aplicación.

En la gráfica de la Figura 5.7 se nota que para los valores iniciales, hasta 60 paquetes por segundo, el tiempo promedio de retardo total de los paquetes varía entre los 27.5 y 29.2ms, entregando todos los paquetes al destino. De 70 paquetes por segundo en adelante el comportamiento de la avenida empieza a degradarse, dando un salto drástico en el tiempo de retardo total que ronda entre 1.45 y 1.47s. Por su parte, el porcentaje de paquetes entregados al destino disminuye a medida que la tasa de generación de datos aumenta.

El máximo flujo de datos que puede existir sobre la avenida para este escenario es de 60 paquetes por segundo al existir aleatoriedad en su creación. Como en el caso ideal, la avenida soporta hasta 80 paquetes por segundo, para una tasa menor mantiene la misma característica, lo que se puede apreciar en la Figura 5.8. En la Figura 5.8 también se incluye el comportamiento del escenario de control.

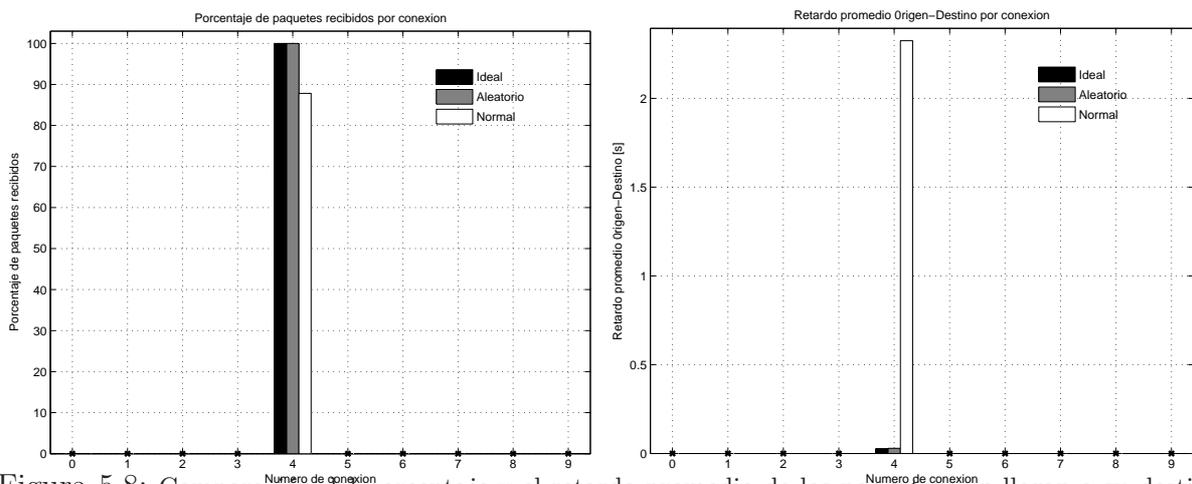


Figura 5.8: Comparación del porcentaje y el retardo promedio de los paquetes que llegan a su destino de la conexión 4. Caso ideal, aleatorio y el escenario de control

El valor del retardo total de los paquetes en la avenida para el caso ideal, y al existir aleatoriedad, es muy parecido, específicamente de 27.5ms y 29.3ms, respectivamente. En ambos casos se entregó la totalidad de los paquetes al destino. Entre tanto, en el escenario de control los paquetes presentaron un retardo total de 2.32s, con una entrega de éstos de un 87.82 %.

De los datos anteriores se tiene que el retardo total utilizando la avenida disminuye en un orden de 79 veces comparado con la forma común de actuar del estándar. De la misma manera, utilizando una avenida se incrementa la cantidad de paquetes que son entregados a su destino.

#### 5.4 Experimento 4: Análisis del desempeño de la avenida con rampas

Se procedió a obtener el comportamiento de la avenida y las rampas, activando las conexiones 3, 4 y 5. Se calculó la tasa de datos, y resultó ser de 36 paquetes por segundo. La tasa de datos disminuye por que se considera que los nodos rampa aumentan el tamaño de los paquetes que utilizan la avenida.

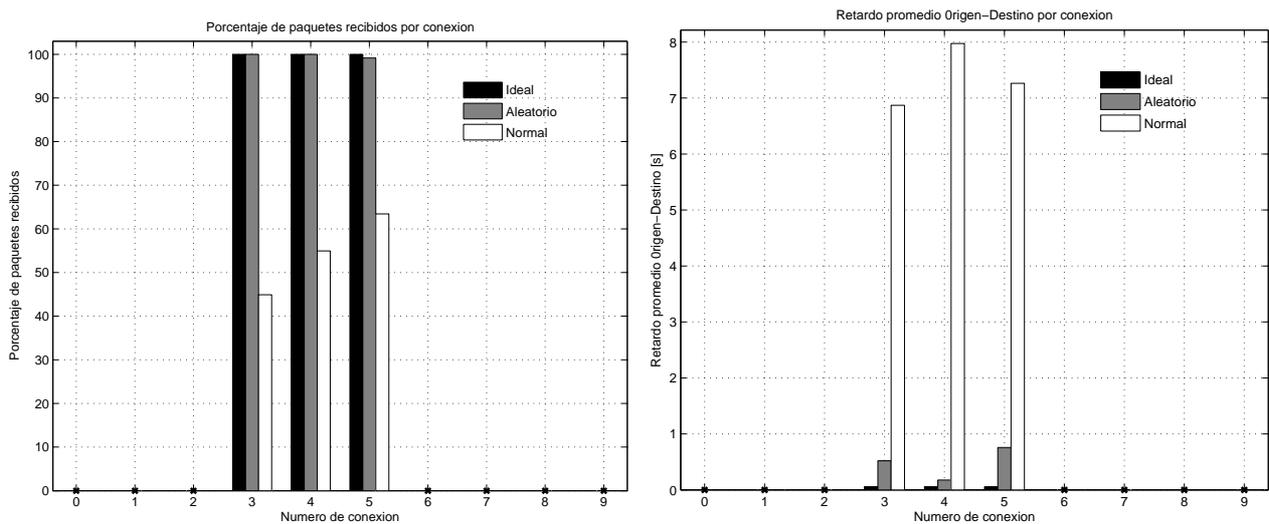


Figura 5.9: Comparación del porcentaje y el retardo promedio de los paquetes que llegan a su destino de las conexiones 3, 4 y 5. Caso ideal, aleatorio y el del escenario de control

En la Figura 5.9 puede notarse que con el protocolo propuesto, trabajando bajo condiciones ideales, el retardo total promedio es muy bajo y tiene un valor aproximado de 54 ms. Además, se entregan todos los paquetes a su destino.

Al haber aleatoriedad se aprecia que el retardo total de los paquetes sube considerablemente tanto para la avenida como para los nodos rampa. Lo anterior se debe a los problemas que se generan al crear los paquetes sin periodicidad alguna. Sin embargo, el porcentaje de paquetes entregados a su destino alcanza practicamente el 100%. Este porcentaje de entrega muestra

que el tiempo que se hace esperar al nodo terminal de la avenida (nodo 40) para resolver el problema de terminal expuesta funciona de manera adecuada.

El valor de CW y el del número de intentos de retransmisión ( $n$ ) utilizados por el nodo 40 para mitigar el problema de terminal expuesta fueron obtenidos mediante el análisis de los paquetes en la red, y realizando simulaciones. La elección de los mejores valores de CW y  $n$  se basó en el retardo total de los paquetes y en el porcentaje de entrega de éstos. La correcta elección de dichos valores se refleja en el hecho de que el nodo 40 no descarta ningún paquete y, en consecuencia, se disminuye el retardo total de los paquetes provenientes de los nodos rampa al existir más oportunidades para acceder a la avenida.

### 5.5 Experimento 5: Análisis del desempeño de la avenida con rampas al existir interferencia

Finalmente, se activaron todas las conexiones para observar el comportamiento del protocolo propuesto en un ambiente de gran contención por el canal. De manera similar a los experimentos anteriores se presentan las gráficas comparativas en la Figura 5.10.

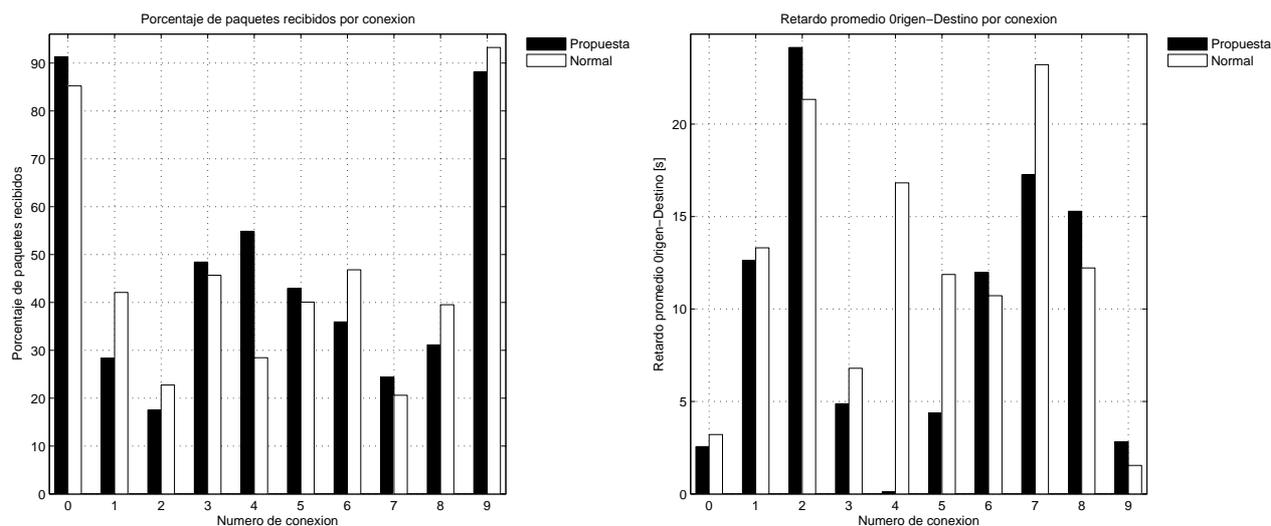


Figura 5.10: Comparación del porcentaje y el retardo promedio de los paquetes que llegan a su destino de cada una de las conexiones entre el protocolo propuesto y el IEEE 802.11.

Analizando la gráfica de la Figura 5.10 se aprecia que por el privilegio otorgado a la avenida y por la forma en que los nodos rampa acceden a ella, el porcentaje de paquetes que alcanzan su destino es mayor para las conexiones 3, 4 y 5.

Como una consecuencia del privilegio que tiene la avenida para utilizar el canal, los nodos que se encuentran dentro del área de restricción, conexiones 1, 2, 6 y 7, tienen una menor probabilidad de utilizar el medio. Al no poder acceder al canal, el porcentaje de paquetes entregados en estas conexiones se ve disminuido. No obstante, al comparar el comportamiento

de las conexiones 1, 2, 6 y 7 cuando existe la avenida y en ausencia de ella (en el escenario de control) los resultados no difieren de manera exagerada.

Al observar la gráfica de la Figura 5.10 se nota que el retardo promedio total de los paquetes pertenecientes a la avenida es drásticamente inferior comparado con el de las otras conexiones. Por su parte, los nodos rampa presentan un retardo total bajo, pero no tan marcado como el mostrado en la avenida.

Existen algunas variaciones en el comportamiento de las conexiones debidas a la topología misma. Por ejemplo, las conexiones 0 y 9 muestran un alto desempeño debido a la ubicación que tienen en la topología. Estas conexiones muestran mejores resultados que el resto de las conexiones al estar ubicadas en los extremos, donde no existe tanta contención por el uso del canal.

En la simulación de control, las conexiones 3 y 5 ubicadas en la parte central de la topología donde existe una gran contención por el medio, pareciese que cuentan con algún tipo de privilegio. Esta impresión se debe a que el retardo total de los paquetes pertenecientes a dichas conexiones es sustancialmente menor al de las otras. Si se toma en cuenta que los paquetes de estas conexiones dan menos saltos en la red para alcanzar su destino, el comportamiento mostrado es comprensible.

Los altibajos de valores manifestados en las otras conexiones presentan un comportamiento usual, pues esto esta en función de diversos factores. Algunos de estos factores son: la cantidad de nodos que contienden por el uso del canal, las veces que un nodo realiza una transmisión, las veces que se necesita retransmitir los paquetes, la cantidad de colisiones en los receptores y los paquetes descartados.

## **5.6 Experimento 6: Análisis del desempeño de los nodos rampa al no existir tráfico en la avenida**

Una vez que se evaluó el desempeño del protocolo como fue pensado originalmente se consideró el caso en el que no existe un flujo de datos sobre la avenida. Bajo este escenario los nodos rampa tienen la capacidad de contender por el medio de manera normal, es decir, pueden acceder a la avenida sin la necesidad de esperar a recibir un paquete proveniente de ella.

Para observar el desempeño de los nodos rampa sin un flujo de datos en la avenida se activaron solamente las conexiones 3 y 5 con una tasa de creación de 30 paquetes por segundo. Los resultados obtenidos se muestran en la Figura 5.11.

En las gráficas de la Figura 5.11 se aprecia que para ambas conexiones se entregan todos los paquetes con un retardo total muy bajo. Asimismo, algunos paquetes que llegan al destino fueron utilizados por el nodo rampa 33 para acceder a la avenida, pues se cumple con la

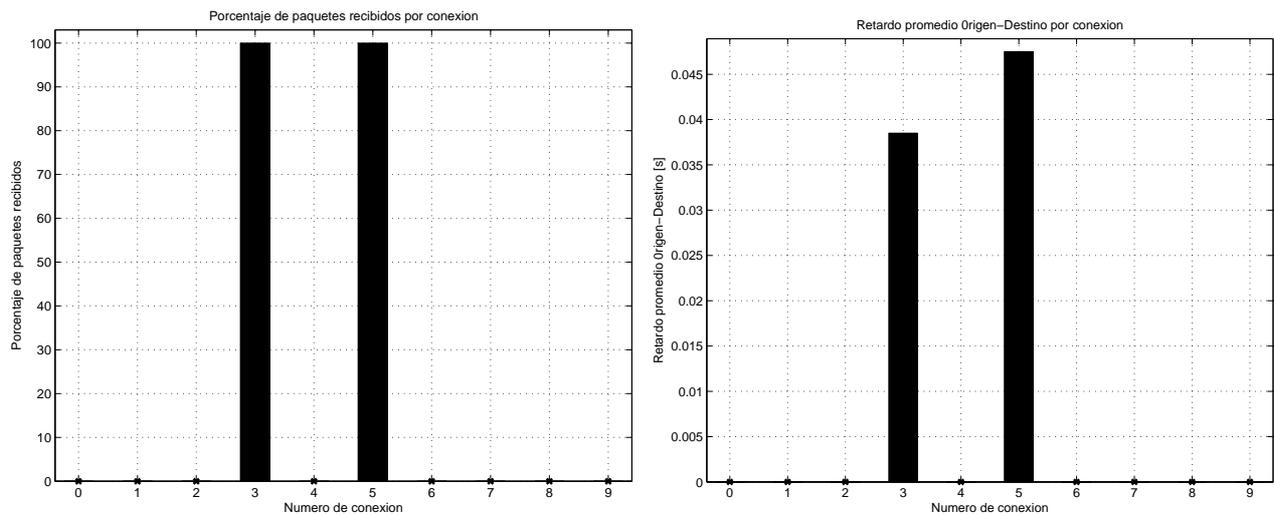


Figura 5.11: Porcentaje y retardo promedio de los paquetes provenientes de los nodos rampa, que llegan a su destino.

condición de recibir un paquete en la avenida. Los paquetes existentes en la avenida, utilizados por el nodo rampa 33, son los previamente transmitidos por el nodo rampa 51.

### 5.7 Experimento 7: Análisis del desempeño de la avenida y nodos rampa que contienen por el canal

Al dejar que los nodos rampa contengan por el canal no se puede asegurar que la avenida tenga siempre el privilegio de transmisión. Los nodos rampa pueden interferir con la avenida, pues aunque utilizan una CW grande cabe la posibilidad de que elijan un valor pequeño y utilicen el canal. Cuando existe un nodo rampa cerca del nodo inicial de la avenida, que se encuentra en un modo de baja contención, los paquetes de la avenida pueden aumentar su retardo total.

Así, la capacidad de los nodos rampa de acceder al canal puede tener un impacto negativo en el retardo total de los paquetes que viajan por la avenida. Por otro lado, los paquetes provenientes de los nodos rampa mostrarán una disminución en el retardo total, pues no tienen que esperar a cumplir con las condiciones planteadas para acceder a la avenida.

En las gráficas de la Figura 5.12 se compara el comportamiento de la avenida con 2 nodos rampa. Los escenarios difieren en la forma en que los nodos rampa acceden a la avenida. En un escenario los nodos rampa acceden a la avenida mediante el proceso antes definido, mientras que en el otro pueden contender por el canal de manera normal.

En las gráficas de este último escenario se puede observar que el retardo total de los paquetes de los nodos rampa disminuye, pero aumenta el de la avenida. Así, aunque la avenida no es la vía más rápida para entregar los paquetes, el desempeño de las tres conexiones es alto.

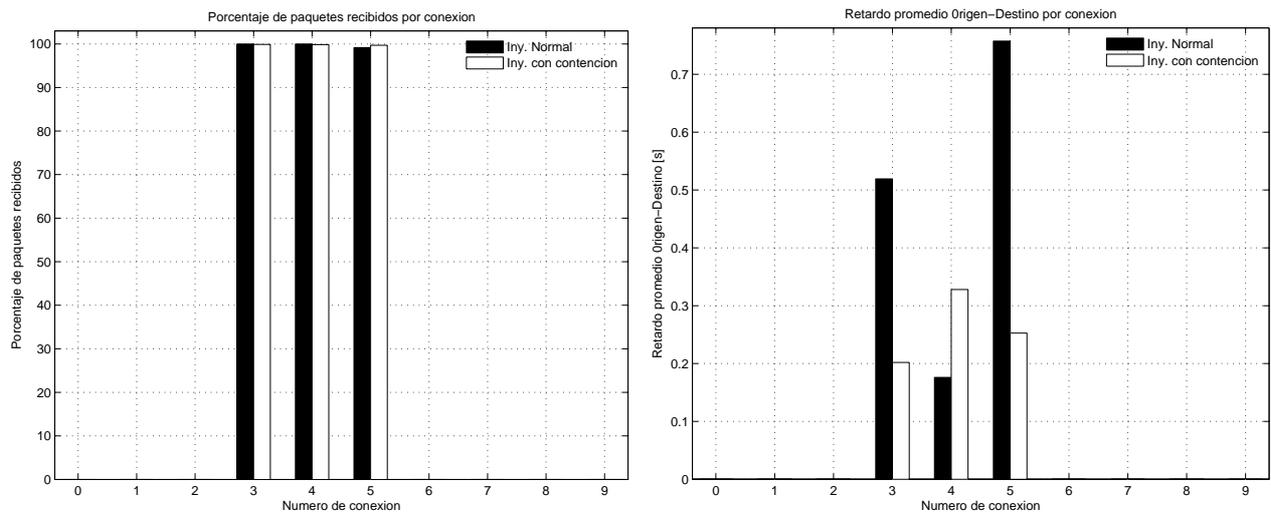


Figura 5.12: Comparación del porcentaje y el retardo promedio de los paquetes que llegan a su destino cuando los nodos rampa acceden a la avenida al recibir un paquete de ésta y cuando estos nodos contienden por el canal de manera habitual con una  $CW=1023$ .

## 5.8 Resumen de capítulo

En este capítulo se proponen diferentes experimentos para evaluar el comportamiento del protocolo de comunicación propuesto. Se describen los escenarios de cada uno de los experimentos. En cada uno de los experimentos se muestran, y se analizan los resultados obtenidos.

# 6

## Conclusiones

Con base en los resultados que se han mostrado en esta tesis, se concluye que es posible crear rutas rápidas en las redes inalámbricas multisaltos. Estas rutas rápidas pueden definirse modificando la manera en que los nodos acceden al medio, concretamente el tiempo que los nodos esperan como resultado del BEB. Además, restringiendo la tasa de datos permitida en las rutas rápidas se logra aminorar el retardo de los paquetes debido al encolamiento.

La introducción novedosa de los conceptos de *avenida* y *nodo rampa* funcionan de manera correcta, ya que los paquetes que utilizan las avenidas muestran un retardo total bajo. Además, utilizando las *avenidas* se logra entregar un alto porcentaje de paquetes a su destino. En todas las gráficas presentadas se advierte que al utilizar el protocolo de comunicación propuesto se obtienen mejores resultados que los obtenidos en el escenario donde no existen *avenidas*.

El proceso definido para que los *nodos rampa* ingresen a la *avenida*, además de no intervenir significativamente con el flujo de datos existente en ésta, evita la formación de cuellos de botella que aumentan el retardo total de los paquetes. Mediante el proceso de acceso utilizado por los *nodos rampa*, debido a la manera ordenada en que se realiza, se puede ingresar a la *avenida* sin generar un cuello de botella. Además, se puede prescindir de los nodos intermedios que se necesitan para alcanzar un destino común.

Con el protocolo de comunicación propuesto se aporta un mecanismo para garantizar QoS a datos susceptibles al retardo, garantizando la entrega de un alto porcentaje de los paquetes a su destino. Para garantizar QoS mediante el protocolo de comunicación propuesto se deben considerar las limitaciones inherentes a él, mismas que se han explicado a lo largo de esta tesis.

Asimismo, en la presente tesis se introduce el concepto de *avenidas* y *odos rampa* con lo que se plantea un nuevo escenario donde se pueden realizar mejoras. Dentro de estas mejoras se puede definir como otorgar movilidad a los nodos que constituyen las *avenidas* o como formarlas de manera dinámica, lo que otorgaría mayor flexibilidad al protocolo.

Por todo lo anterior, se concluye que el protocolo de comunicación propuesto en esta tesis cumple con el objetivo de la misma, al mantener el valor del retardo total de los paquetes muy por debajo de lo que se obtiene al utilizar el acceso al medio habitual definido por el estándar, aún en el peor escenario. Validando así la hipótesis planteada al inicio de este estudio.

# Glosario

- ACK** de acuse de recibo (*Acknowledge*). 5, 7, 9, 10, 17, 29, 32, 33, 37
- AP** punto de acceso (*Access Point*). 4, 5
- BEB** retroceso con crecimiento exponencial binario (*Binary Exponential Backoff*). 5, 6, 8, 11, 16, 24, 27, 35, 37, 41, 42, 52
- BSS** conjunto de servicios básicos (*Basic Service Set*). 4, 5
- CSMA/CA** acceso múltiple con detección de portadora y evasión de colisiones (*Carrier Sense Multiple Access with Collision Avoidance*). 5
- CTS** libre para enviar (*Clear To Send*). 5, 8, 9, 17, 18, 23, 29, 32, 33
- CW** ventana de contención (*Contention Window*). 6, 7, 19, 24, 27, 32–34, 39, 41, 42, 48, 50
- DCF** función de coordinación distribuida (*Distributed Coordination Function*). 5, 7, 9, 11, 15
- DIFS** Espacio entre tramas DCF (*DCF InteFrame Space*). 5, 7, 8, 16, 19, 21, 23
- DS** sistema de distribución (*Distribution System*). 4
- EIFS** Espacio entre tramas extendido (*Extended InteFrame Space*). 5, 7, 8, 16, 21
- ESS** conjunto de servicios extendido (*Extended Service Set*). 5
- HCF** función de coordinación híbrida (*Hybrid Coordination Function*). 5
- IBSS** conjunto de servicios básico independiente (*Independent Basic Service Set*). 5
- IFS** espacio entre tramas (*InteFrame Space*). 5
- MAC** control de acceso al medio (*Media Access Control*). 3, 10, 12, 32, 40, 42
- MANETs** Redes inalámbricas *Ad Hoc* móviles (*Mobile Ad Hoc Networks*). 2
- MTU** unidad máxima de transmisión (Maximum Transmission Unit). 21
- NAM** animador de red (*Network AniMator*). 29, 31

**NAV** vector de ubicación de red (*Network Allocation Vector*). 9

**NS-2** simulador de redes (*Network Simulator version 2*). 28, 29, 36

**OSI** interconexión de sistemas abiertos (*Open System Interconnection*). 4, 11, 28, 29

**PCF** función de coordinación puntual (*Point Coordination Function*). 5

**QoS** calidad de servicio (*Quality of Service*). 1, 10, 11, 52

**RTS** solicitud de envío (*Request To Send*). 8, 9, 17, 18, 22–24, 29, 32–35

**SIFS** Espacio pequeño entre tramas (*Short InterFrame Space*). 5, 7, 8, 19

**UDP** protocolo de datagrama de usuario (*User Datagram Protocol*). 40, 43

**VANETs** Redes inalámbricas *Ad Hoc* vehiculares (*Vehicular Ad Hoc Networks*). 2

**WSNs** Redes inalámbricas de sensores (*Wireless Sensor Networks*). 2

# Bibliografía

- [1] IEEE Std. 802.11-2007. "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications".
- [2] Xiaohua Jia, Deying Li y Dingzhu Du. "QoS Topology Control in Ad Hoc Wireless Networks".
- [3] Jangeun Jun y Mihail L. Sichitiu. "Fairness and QoS in Multihop Wireless Networks".
- [4] Chunhung Richard Lin y Jain Shing Liu. "QoS Routing in Ad Hoc Wireless Networks".
- [5] Mahasweta Sarkar y Sahitya Borra. "A QoS Enabled MAC Protocol for Wireless Ad hoc Networks with Power Control"
- [6] Hsiao-Kuang Wu y Pei-Hung Chuang. "Dynamic QoS Allocation for Multimedia Ad Hoc Wireless Networks".
- [7] Gahng-Seop Ahn, Andrew T. Campbell, Andras Veres y Li-Hsiang Sun. "SWAN: Service Differentiation in Stateless Wireless Ad Hoc Networks".
- [8] Seoung-Bum Lee y Andrew T. Campbell. "INSIGNIA: In-Band Signaling Support for QoS in Mobile Ad-Hoc Networks"
- [9] Prasun Sinha, Raghupathy Sivakumar y Vaduvur Bharghavan. "CEDAR: a Core-Extraction Distributed Ad hoc Routing algorithm"
- [10] Wu Xing-feng, Liu Yuan-an. "A Survey of WLAN QoS Systems Based on IEEE 802.11". International Journal of Computer Science and Network Security, VOL.7 No.3, Marzo 2007. p309-312.
- [11] Nityananda Sarma y Sukumar Nandi. "QoS Support in Mobile Ad Hoc Networks".
- [12] Joao Matos y Hugo Miranda. "Ad Hoc Routing Under Randomised Propagation Models".
- [13] Jian Li. "Quality of Service (QoS) Provisioning in Multihop Ad Hoc Networks". 2006. p:1-30
- [14] Qiang Ni, Lamia Romdhani y Thierry Turletti. "A Survey of QoS Enhancements for IEEE 802.11 Wireless LAN"

- [15] Shugong Xu and Tarek Saadawi. "Does the IEEE 802.11 MAC Protocol Work Well in Multihop Wireless Ad Hoc Network".
- [16] Tarun Saxena. "QoS over Multihop Wireless Networks". A Thesis Submitted for the Degree of Master of Science (Engineering) in the Faculty of Engineering, Indian Institute of Science. April 2007. p: 1-6
- [17] Kui Wu y Janelle Harms. "QoS Support in Mobile Ad Hoc Networks".
- [18] Thesis of Patrick Stüdi "Quality of Service for Mobile Ad Hoc Networks". Marzo 2003. p: 1-6