UNIVERSIDAD NACIONAL AUTÓNOMA DE MEXICO

PROGRAMA DE MAESTRÍA Y DOCTORADO EN INGENIERIA

FACULTAD DE INGENIERÍA

"Diseño e implementación de un protocolo de localización de nodos móviles en redes con infraestructura y *ad hoc*"

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. JUAN MANUEL CERVANTES Y RAMIREZ

TUTOR:

DR. JAVIER GÓMEZ CASTELLANOS

2006

JURADO ASIGNADO:

Presidente: Dr. Victor Rangel Licea

Secretario: Dr. Miguel López Guerrero

Vocal: Dr. Javier Gómez Castellanos

1er. Suplente: Dr. Carlos Rivera Rivera

2o. Suplente: Dr. Salvador Landeros Ayala

Ciudad Universitaria, D.F.

TUTOR DE TESIS:

DR. JAVIER GÓMEZ CASTELLANOS

_____

FIRMA

Agradecimientos

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

Al ***Consejo Nacional de Ciencia y Tecnología*** *por el apoyo económico en mis estudios de Posgrado*

Al ***Dr. Javier Gómez Castellanos*** *por haber creido en mi y por su apoyo constante durante el trabajo realizado bajo su asesoría. Su correcta guía permitió la culminación exitosa de este trabajo, por lo que le reitero mi admiración y gratitud por sus enseñanzas.*

Al ***Dr. Victor Rangel Licea*** *que junto con el Dr. Gómez, iniciaron un gran grupo de investigación en redes de datos inalámbricas, marcando el inicio de investigación de gran nivel en este ramo de las telecomunicaciones en México. Mi agradecimiento por sus enseñanzas.*

A los sinodales ***Dr. Salvador Landeros Ayala, Dr. Miguel López Guerrero y Dr. Carlos Rivera Rivera*** *por el tiempo dedicado para la revisión de este trabajo, cuyas valiosas recomendaciones contribuyeron al mejoramiento de mi tesis. Mi agradecimiento y admiración*

Dedicatorias

A ***Dios****, por brindarme una gran familia y la fortaleza en los momentos difíciles.*

A mis padres, ***Benjamín Cervantes Castro y María Antonieta Ramírez de Cervantes*** *por su apoyo incondicional, por su amor y dedicación durante toda mi vida. Su guía desde mis primeros años ha alcanzado en este momento un gran logro, del cual no encuentro las palabras para expresarles todo mi amor y agradecimiento. Su esfuerzo y constancia ante la adversidad, son ejemplos que tendre presentes todos los días de mi vida. Mi misión ahora es seguir sus pasos y agradecer a Dios con cada uno de mis actos el que me haya permitido tener a los mejores padres que un hijo puede desear. Los amo*

A mi hermano el ***M.C. José Antonio Cervantes y Ramírez****, cuya tenacidad y visión para alcanzar sus metas merece toda mi admiración y respeto. Gracias por tu apoyo y tu consejo oportuno. Tu ejemplo me hace desear cada día mejor persona y mejor profesionista.*

A mi novia ***Samantha Albarrán Magdaleno****, por su amor y apoyo que me ha permitido ver la vida de forma diferente. Conocerte me ha llenado de felicidad y me ha convertido en una mejor persona. Gracias por ser tan linda*

A mi amiga ***Maribel Miranda Garcés****, por su amistad sincera y por demostrarme que el tiempo que tenemos prestado en este mundo permite no sólo perpararnos profesionalmente, sino también brindar de luz a los demás como tu lo hiciste con quienes te conocimos. Descansa en paz*

A mis compadres ***Javier Agüero Carranza, David Brena Wilson y Linda Vega García,*** *así como mi amiga* ***Audry Elorza****, por su amistad sincera a lo largo de toda la vida. A mis compañeros y amigos* ***Luis Méndez Barredo, Eduardo Cota Guajardo, José Antonio Cardozo, Noé Torres Cruz, Ivan Gómez Castellanos, Luis Alberto Bernardino, Saulo Pérez Aragón y Abraham Martínez Martínez****, por su amistad a lo largo de la carrera.*

Índice

Índice de Figuras

Índice de Tablas

I. Introducción

Las Tecnologías de la Información han presentado un gran desarrollo en los últimos años hasta convertirse en un aspecto fundamental en la vida diaria del ser humano. De manera particular, las redes de datos representan actualmente una herramienta indispensable en el ir y devenir de negocios, finanzas, investigación, información, educación y entretenimiento, donde el intercambio de informaciòn a través de redes globales como la Internet ha marcado un cambio sin precedentes en el funcionamiento del mundo entero.

Por su parte, las tecnologías que soportan este intercambio masivo de información, han ido evolucionando y se han ido adaptando de acuerdo a las necesidades de los usuarios. La tendencia actual parte de transmitir una mayor cantidad de datos en poco tiempo y, con el *boom* de la telefonía celular, se busca permitir el intercambio de información en cualquier momento y en cualquier lugar.

Sin embargo, las aplicaciones iniciales de las redes inalámbricas respondían únicamente a la necesidad de enlazar uno o más sitios de red (por lo regular edificios coorporativos o almacenes), en configuraciones como: punto a punto (conexión de dos sitios de red mediante un enlace inalámbrico en línea de vista), punto a multipunto (conectando una localidad central con localidades remotas mediante el uso de antenas omnidireccionales) y punto a multipunto a multipunto (uno de los usuarios es a su vez punto de acceso para otros usuarios).

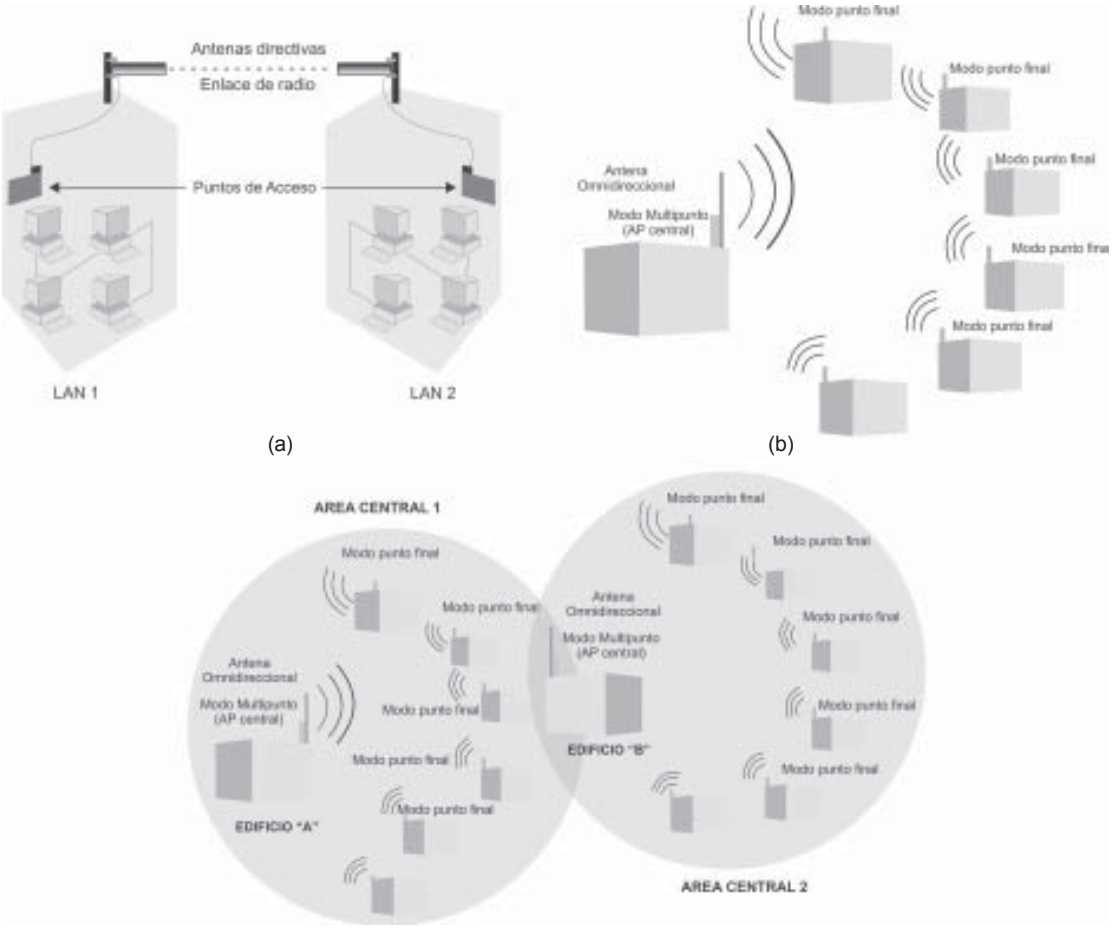

Figura 1.1.- Diferentes configuraciones de redes inalámbricas para enlazar sitios remotos (a) Punto a punto. (b) Punto a multipunto. (c) Punto a multipunto a multipunto.

Posteriormente tecnologías inalámbricas como transmisión de datos por infrarrojo y la aparición de "*bluetooth*", así como su aplicación en enseres de uso diario como televisores, componentes de audio y su final aplicación en computadoras personales, dieron una nueva perspectiva a los alcances de transmitir sin cables.

Sin embargo, el transmitir sin cables utilizando radiofrecuencia sobre el espacio libre, presenta una serie de retos para la transmisión de datos. El espectro electromagnético se encuentra actualmente saturado y en la mayoría de sus bandas requiere de licencia emitida por entidades regulatorias; las limitaciones en ancho de banda e interferencia, presentan un límite para las tasas de datos soportadas, lo cual impulsa a utilizar bandas de frecuencias más altas pero con limitaciones en el alcance de transmisión.

Por todo lo anterior, con el objeto de ofrecer una arquitectura flexible por las redes de datos, tecnologías como IEEE802.3 (*Ethernet*), se adapta en sus capas física y de acceso al medio, para utilizar el espacio libre como medio de transmisión, apareciendo el estándar IEEE802.11. Este estándar en sus inicios contemplaba únicamente tasas de datos de 1Mbps y 2Mbps, principalmente en transmisión infrarroja. Posteriormente, con el objeto de incrementar las tasas de transferencia y no depender de una comunicación en línea de vista, se optó por utilizar la banda ISM (Industrial, Científica y Médica) al no necesitar de licenciamiento.

A partir de eso aparecieron los estándares IEEE802.11b, 802.11a y 802.11g, que transmiten a 11Mbps (para "b" en la banda de los 2.45GHz) y 5Mbps (para "a" en la banda de los 5GHz y "g" en la banda de 2.45GHz). Sin embargo, para hacer posibles las transmisiones de estas tasas de datos entre varios usuarios sin interferencia, se adaptaron tecnologías utilizadas en telefonía celular tales como espectro disperso (en sus variantes de Espectro Disperso de Salto de Frecuencia - FHSS, Espectro Disperso de Secuencia Directa (DSSS) y en combinación de Multiplexaje por división de Frecuencias Ortogonales - OFDM), en conjunto con el control de acceso al medio con prevención de colisiones.

El estándar IEEE802.11 contempla dos arquitecturas de servicio: redes con infraestructura (donde se contempla la presencia de una entidad coordinadora o punto de acceso) y redes sin infraestructura (*ad-hoc*). En lo que respecta al esquema con infraestructura, su similitud con las arquitecturas celulares se debe a la habilidad del rehúso de las infraestructuras de voz existentes actualmente y de las frecuencias portadoras, así como la simplicidad del modelo debido a la presencia de una entidad central coordinadora; además permite distribuir una serie de estaciones base conectadas a la misma red de *backbone* para ofrecer servicio a un área determinada, extendiendo la capacidad de servicio de una red convencional.

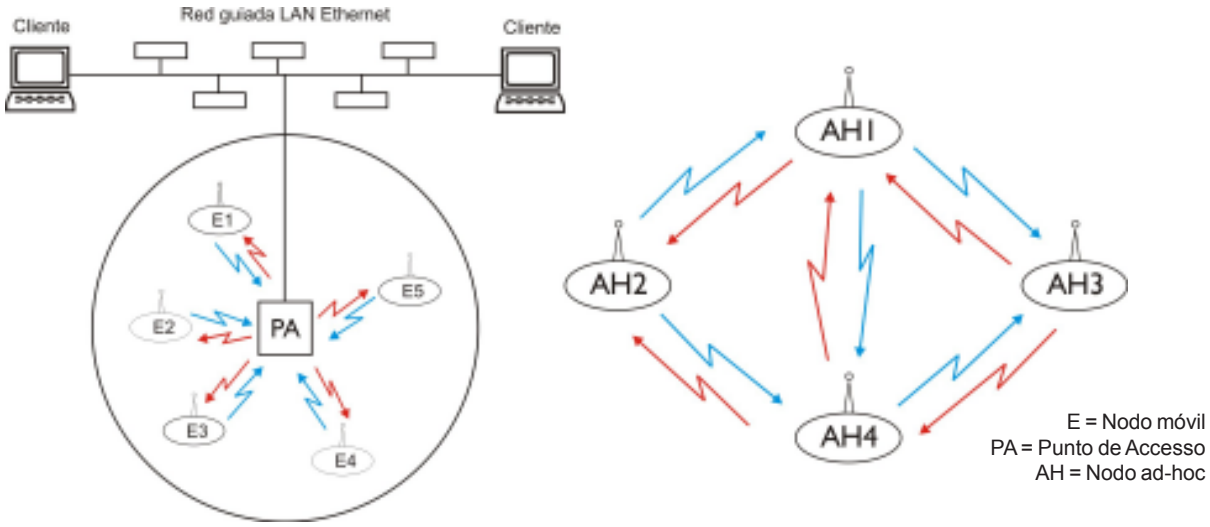

Figura 1.2.-Configuraciones de operación de redes inalámbricas bajo el estándar 802.11
(a) Modo de infraestructura (b)Modo ad-hoc

El otro esquema (*ad hoc*), tiene la peculiaridad de que no necesita de una entidad centralizadora que dirija el tráfico; los nodos pueden comunicarse entre sí, ya sea de forma directa o con la ayuda de otros nodos que relevan el tráfico salto por salto (un nodo releva un paquete no dirigido a él a otro nodo intermedio hasta alcanzar el nodo destino). Sin embargo, este esquema aunque versátil en su despliegue y operación, presenta una serie de retos adicionales en el establecimiento y mantenimiento de las conexiones.

Para el caso de redes *ad hoc* estáticas, la problemática no es tan severa, solo que si participan muchos nodos en el relevo de los datos, puede existir cierto retraso en la comunicación, que es en general proporcional al número de relevos. En cuanto a la definición de las rutas, ésta se realiza en una sola ocasión y en el caso de que alguno de los nodos intermedios deje de funcionar.

El trabajo presentado en esta tesis, se enfoca al estudio y optimización de las técnicas de descubrimiento de rutas y localización de nodos en una red *ad hoc*, de forma particular aquellas basadas en *flooding* (inundación de la red con paquetes de búsqueda) y en información de la posición geográfica.

La técnica de búsqueda de rutas por *flooding*, es la más utilizada en las redes *ad hoc*. Cuando un nodo desea iniciar una comunicación con otro en la red cuya posición es desconocida para el nodo iniciador, éste emite una petición de ruta que es recibida por todos los nodos en su rango de transmisión, la cual se propaga por toda la red hasta encontrar el nodo destino. En lo referente a los protocolos que basan su operación en la posición geográfica de sus nodos (con ayuda de un sistema de posicionamiento global GPS), se necesita intercambiar constantemente su posición entre vecinos, para que cuando se desee establecer una transferencia de datos, se encaminen los datos hacia los vecinos que tuvieron conocimiento del destino en tiempo reciente; la transmisión constante de paquetes de control, representa una utilización adicional del ancho de banda, que en caso de ser significativa impacta severamente el *throughput* de la red.

De manera particular, este consumo adicional de ancho de banda es lo que limita el tipo de servicios y aplicaciones que pueden correr sobre redes *ad hoc*. Servicios como comunicaciones multimedia (videoconferencias, transmisión de video, etc.) y la transmisión de datos de alta velocidad, no son posibles debido a que no se puede ofrecer calidad de servicio por lo incierto del estado del canal y de la ruta.

En materia de encaminamiento para redes *ad hoc*, algoritmos como DSR(*Dynamic Source Routing*)[15], AODV (*Ad hoc On Demand Distance Vector*)[13] y TORA(*Temporally-Ordered Routing Algorithm*)[14], basan su funcionamiento en *flooding* cada que se desea iniciar una comunicación entre dos nodos. Esta petición de ruta es recibida por todos los nodos dentro del área de cobertura del nodo iniciador, donde cada uno revisará sus tablas de rutas (puesto que cada nodo en una red *ad-hoc* realiza funciones de *router*) y en caso de tener una ruta hacia el destino (o bien si es el destino en sí), envía una respuesta de ruta. En caso de no conocer un "camino" hacia el destino, el nodo tendrá que retransmitir el paquete de petición de ruta agregando su dirección en el mensaje y así sucesivamente hasta alcanzar al destino, que responderá con la confirmación de ruta correspondiente, utilizando el camino por el cual llegó el paquete; pero siguiendo el orden inverso. En todo el proceso de petición, confirmación de ruta y transmisión de paquetes, los nodos "aprenden" las rutas que escuchan y en las cuales participan.

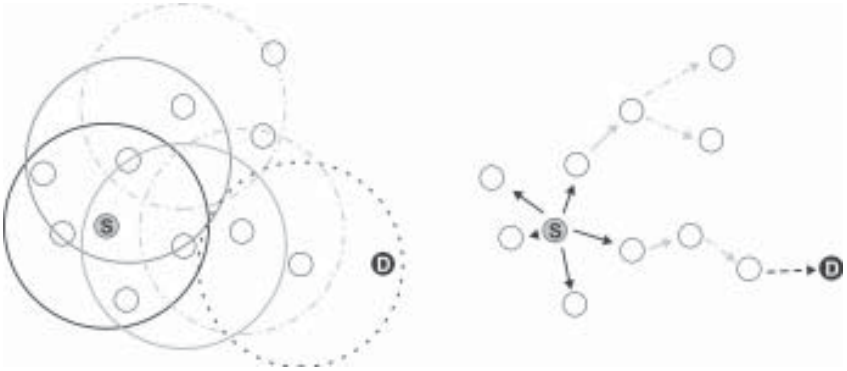

Figura 1.3.- Flooding en una red ad-hoc

Sin embargo, esta forma de descubrimiento y mantenimiento de las rutas es por demás ineficiente conforme se incrementa el tamaño de la red y se presenta una alta movilidad de los nodos. Después de estudiar varios protocolos de encaminamiento *ad hoc* (de forma particular DSR - *Dynamic Source Routing*), observamos que se genera una gran cantidad de valiosa información de rutas que es desperdiciada por estos esquemas. Cada nodo en la red que participa o "escucha" comunicaciones a su alrededor, almacena durante determinado tiempo las rutas que escucha hasta que estas caducan o bien son eliminadas a partir de la recepción y relevo de un mensaje de "error en ruta", manteniendo por ese tiempo una referencia aproximada de la posición de un nodo y de los nodos que lo circundan.

Con el objetivo de presentar una alternativa más eficiente de enrutamiento y localización de nodos, presentamos en este trabajo el protocolo *Fireworks*, cuyo funcionamiento explota de manera eficiente la información de ruta generada por su antecesor (el protocolo DSR).

El funcionamiento general del protocolo *Fireworks* se muestra en la figura 1.4. Cada nodo incluye una lista de sus vecinos a un salto cuando es fuente o destino de un paquete. Tiempo después, cuando "S" desea iniciar una nueva comunicación con "D" realiza una búsqueda acotada, buscando por el destino o alguno de los vecinos del destinatarios. En caso de encontrar una ruta al destino, se inicia inmediatamente la transferencia de datos; en caso de encontrar una ruta a uno de los vecinos, se envía un paquete unicast que al llegar al vecino dispara una nueva búsqueda acotada a cierto número de saltos preguntando por el destino.

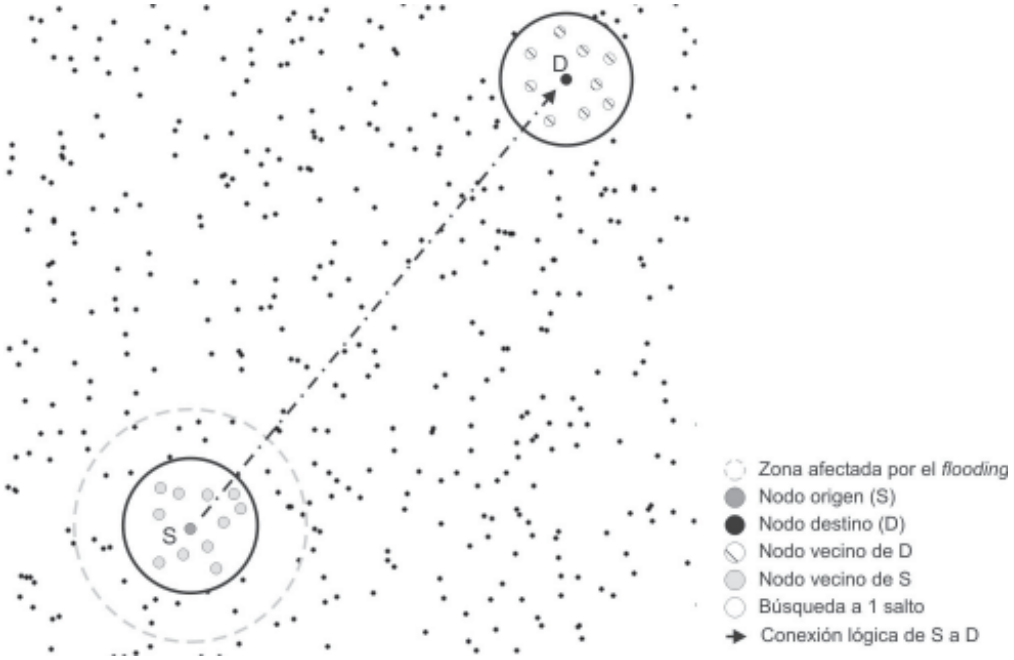

Figura 1.4.- Principio de operación de Fireworks. El nodo origen S realiza una búsqueda acotada del nodo destino D o alguno de sus vecinos a 1 salto. Al encontrar una ruta válida a D se transmiten inmediatamente los datos; en caso de encontrar una ruta a algún vecino se crea un paquete unicast al vecino D y posteriormente una nueva búsqueda a D.

Para lograr la implementación de *Fireworks*, la programación se basó en el funcionamiento de DSR, pero con los ajustes necesarios para optimizar la generación de paquetes de flooding basados en los principios propuestos de *Fireworks*. Mecanismos como la generación y envío de peticiones de ruta, así como el mecanismo de recepción de paquetes de control y formato de encabezados de paquetes se modificaron para mejorar el desempeño de DSR.

La implementación del protocolo *Fireworks* se realizó en el simulador NS2 desarrollado por la Universidad del Sur de California, debido a su estructura y la gran cantidad de elementos del modelo de comunicaciones TCP/IP que considera; también resulta favorable el carácter de tiempo discreto de su operación, permitiendo analizar paso a paso los eventos generados por el simulador con información relacionada con todos los niveles.

Posterior a la implementación, se procedió a correr diversos escenarios contabilizando y promediando el número de rutas encontradas al destino en la búsqueda acotada, vecinos alcanzados por paquetes *Fireworks* unicast y el número de ocasiones en que se encontró al nodo destino exitosamente. Como parámetro importante se contabilizó el promedio de paquetes de flooding generados por *Fireworks* y su comparación en el mismo escenario al *flooding* generado por DSR sin modificar.

Los resultados mostrados en el capítulo VI, muestran que el desempeño de *Fireworks* es superior al desempeño de DSR, siendo más eficiente en las búsquedas de nodos, generando significativamente menos flooding (hasta un 80% menos) y reduciendo así el tiempo de búsqueda y recuperación de información de ruta, siendo ésta la mayor contribución de este trabajo.

1.1 Objetivo

El presente trabajo tiene como objetivo, diseñar un protocolo de enrutamiento para redes inalámbricas con infraestructura y *ad hoc*, capaz de eficientar el uso de paquetes de control para el descubrimiento de rutas entre los nodos de la red, a partir de la información de ruteo constantemente generada e intercambiada durante las comunicaciones que ocurren en una red inalámbrica, para de esta forma utilizar de mejor manera el ancho de banda del canal.

El trabajo se organiza de la siguiente manera: En el capítulo II se presenta un resumen del estado del arte de las tecnologías de red inalámbricas, donde se revisan los inicios de esta tecnología en las redes de datos y las implementaciones mayormente utilizadas actualmente. Posteriormetne, en el capítulo III se mencionan conceptos generales de las redes sin infraestructura *ad hoc,* así como diferentes esquemas y aplicaciones de este tipo de redes. En el capítulo IV se presentan propuestas de esquemas de flooding eficiente basados en la posición geográfica de los nodos (obtenida a través de GPS) y en la información histórica de encuentros recientes; nos enfocamos principalmente el funcionamiento del protocolo *Fireworks* propuesto en este trabajo.

En el capítulo V se explica la implementación de este protocolo en el simulador NS2, bajo los esquemas de operación del protocolo Dinamyc Source Route (DSR). Finalmente en los capítulos VI y VII se presentan los resultados y su discusión, así como las conclusiones de este trabajo.

II. Estado del arte de las tecnologías de red inalámbricas

La mayoría de las redes de datos inalámbricas actuales, usan el modelo de red celular que consiste de una estación base (o también conocido como punto de acceso) hacia el cual los usuarios móviles "hablan" directamente, utilizando el espacio libre como medio de transmisión. La estación base en cambio, se conecta a Internet a través de una red de distribución.

Alguna de las razones detrás de la adopción del modelo celular para redes inalámbricas de datos, incluyen la habilidad del reuso de las infraestructuras de red de voz existentes actualmente y de las frecuencias portadoras, así como la simplicidad del modelo debido a la presencia de una entidad central coordinadora, la estación base. Sin embargo, las limitaciones de las redes inalámbricas de datos para escalarse hacia tasas de transferencia mayores y por tanto mantener el crecimiento acelerado en el número de usuarios, ocasionan que se busquen alternativas y mejoras a este modelo de red[1].

Este capítulo provee una breve descripción sobre tecnologías, topologías y estándares de redes inalámbricas, así como las necesidades actuales y tendencias de investigación.

2.1 Topologías de redes inalámbricas

Las redes inalámbricas pueden ser configuradas en alguno de los siguientes modos:

a) **Ad-Hoc:** No se utiliza ningún punto de acceso centralizado; todas las comunicaciones son cliente a cliente
b) **Red de infraestructura inalámbrica:** El punto de acceso es el agente de comunicación central dentro de las celdas
c) **LAN-LAN:** Dos o más puntos de acceso son utilizados como un enlace inalámbrico para conectar redes guiadas

A continuación se presenta una breve descripción de cada una de las topologías de red y sus principales aplicaciones.

2.1.1 Redes sin infraestructura (*ad hoc*)

Una red sin infraestructura o *ad hoc*, está compuesta únicamente de clientes que se comunican entre ellos sin requerir de la presencia de una entidad coordinadora o punto de acceso. Cuando un nodo desea comunicarse con otro dentro de la red, necesitará encontrar una ruta al destinatario, para establecer la comunicación mediante saltos entre los nodos intermedios que se encuentren entre iniciador y destino, o bien de manera directa cuando se encuentran dentro del rango correspondiente, como se muestra en la figura 2.1. Las rutas se forman de manera temporal para el caso de usuarios móviles, por lo que serán comunes los rompimientos y será necesario emprender una nueva búsqueda.

Como ejemplo de aplicaciones de las redes *ad hoc*, tenemos las comunicaciones militares en campos de batalla, comunicaciones de emergencia en caso de desastre, computación móvil bajo el estándar IEEE 802.11 y próximamente en comunicaciones personales.

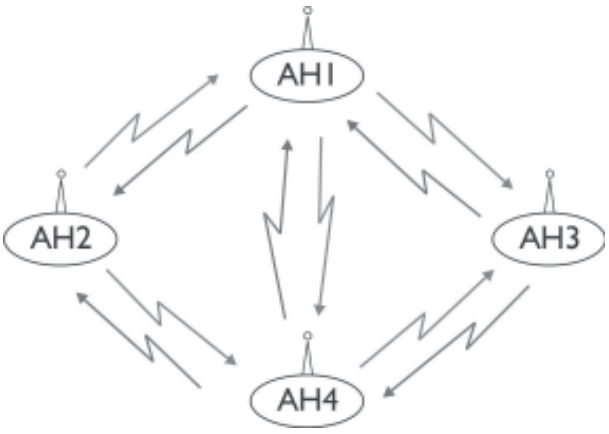

Figura 1.1.- Topología sin infraestructura (ad hoc)

2.1.2 Modo de infraestructura inalámbrica

En el modo de infraestructura inalámbrica, un punto de acceso está localizado en el centro lógico de la celda inalámbrica y se comunica con los clientes móviles dentro de la celda. Una red con infraestructura es una red que consiste de puntos de acceso y clientes inalámbricos. Típicamente, el punto de acceso está también conectado a la red Ethernet alámbrica.

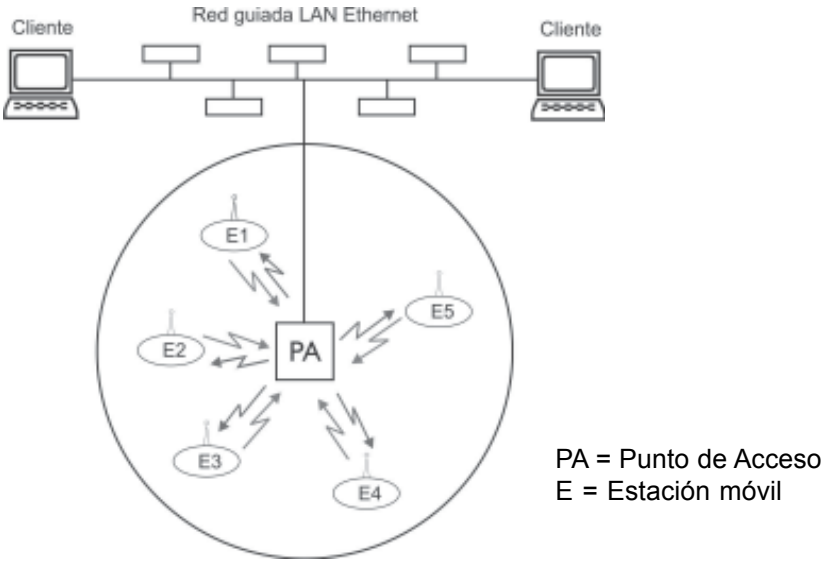

Figura 2.2.- Topología con infraestructura

Como mínimo, el punto de acceso recibe, almacena y transmite datos entre la red inalámbrica LAN y la red guiada. La figura 2.2 muestra este tipo de topología. Cabe mencionar que el número máximo de clientes que puede soportar un punto de acceso es de 250 clientes; esta cantidad puede variar según el modelo y la configuración del punto de acceso. Este número puede reducirse significativamente por varios factores, como puede ser tráfico en la red ocasionado por una gran cantidad de usuarios [3].

2.1.3 Redes de sensores

Este tipo de redes tienen la peculiaridad de que pueden crear de manera espontánea una red de intercambio de datos[4]. Cada nodo incluye en sí mismo una pequeña computadora (generalmente

constituida por un microcontrolador), interconectada a una serie de transductores (de presión, temperatura, etc.) y a un transmisor de radio que trabaja en la banda ISM. Debido a sus pequeñas dimensiones (hasta del tamaño de una moneda), el rango de transmisión es limitado (algunas decenas de metros) para menor consumo de potencia de la batería.

Este tipo de sensores se despliegan generalmente en lugares donde no pueden llegar seres humanos, debido a riesgos en las inmediaciones de la zona a monitorear o dificultad de acceso (como en estaciones petroleras, ambientes naturales poco accesibles), para recopilar información referente a algún fenómeno físico.

Estas redes trabajan bajo el principio de redes *ad-hoc*, por lo que no existe una entidad controladora de las comunicaciones. Los nodos se comunican entre sí con aquellos otros nodos que se encuentran dentro del rango de transmisión, para intercambiar información y relevarla hacia un nodo que se encargará de recopilar los datos generados y transmitirlos finalmente a una base de datos o a una red más grande. En la figura 2.3 se muestra la arquitectura de red implementada en este tipo de redes.

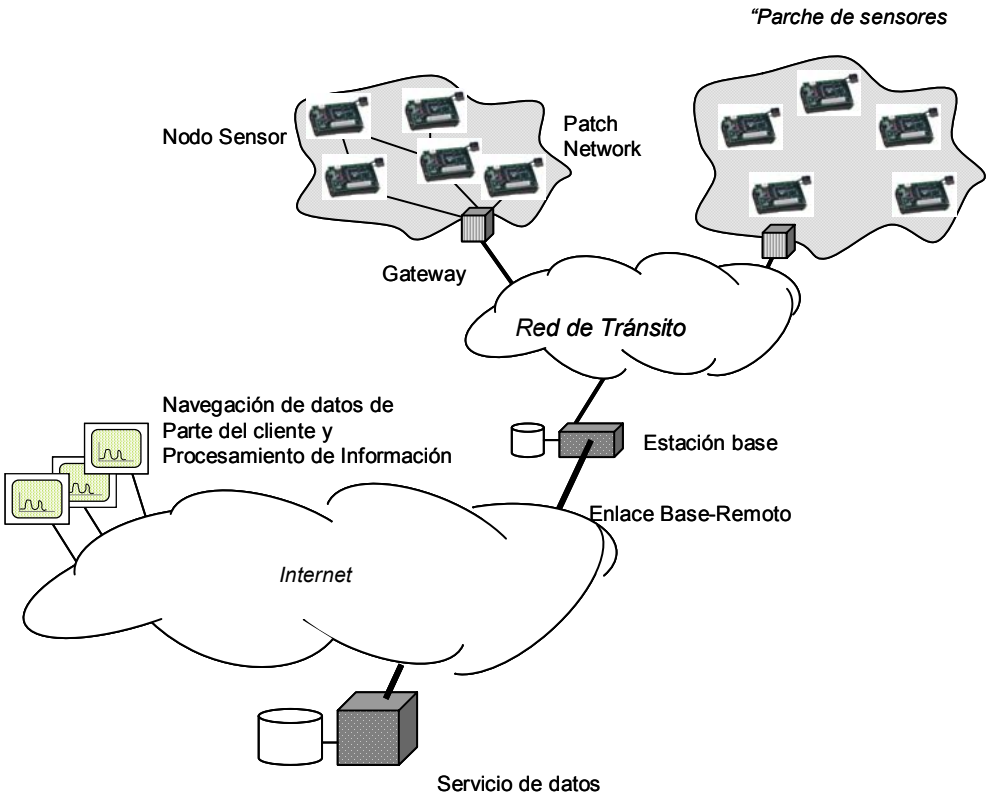

Figura 2.3.- Servicios distribuidos de red auto-organizados (redes de sensores)

2.1.4 Redes inalámbricas MESH

Los sistemas inalámbricos de comunicación industriales más utilizados son los enlaces de radio tipo celular, utilizando transmisiones punto a punto o punto a multipunto. Estos esquemas tradicionales de comunicación incluyen estructuras rígidas, requerimientos de planeación rigurosos y señales interrumpidas.

En contraste, las redes inalámbricas tipo MESH[5] son sistemas multi-saltos cuyos dispositivos se asisten unos a otros en la transmisión de paquetes a través de la red, especialmente en condiciones adversas. Esta red basada en funcionamiento *ad-hoc* puede ser desplegada en un sitio con preparación mínima y proveen un sistema flexible que puede extenderse a miles de dispositivos.

Tal como ocurre en la Internet y otras redes basadas en comunicaciones punto a punto mediante la operación de *routers*, una red MESH ofrece múltiples trayectorias redundantes a través de la red. Si un enlace falla por cualquier razón (incluyendo la introducción de interferencia de RF severa), la red automáticamente encamina los mensajes a través de trayectorias alternas.

Este tipo de redes no requieren de un administrador del sistema para indicarle como conducir un mensaje a su destino. Una red MESH es auto-organizada y no requiere de configuración manual. Debido a esto, para agregar nuevo equipo o cambiar de ubicación del mismo, es tan simple como conectarlo y encenderlo. La red descubre el nuevo nodo y automáticamente lo incorpora al sistema existente, como se muestra en la figura 2.4.

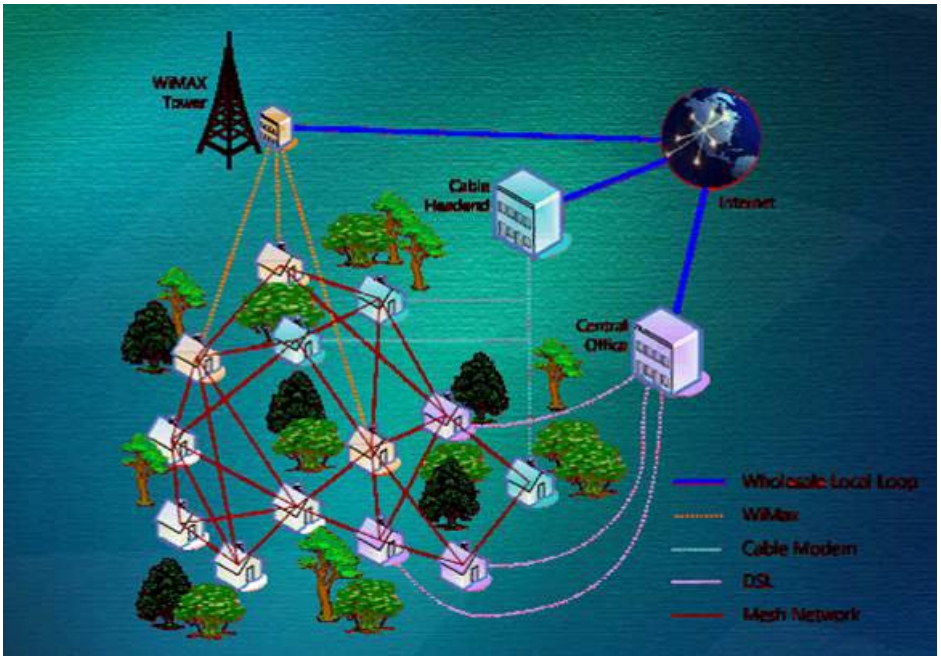

Figura 2.4.- En una red inalámbrica MESH, los nodos múltiples cooperan para relevar un mensaje a su destino. La topología MESH mejora la confiabilidad total de la red, lo cual es particularmente importante cuando se opera en ambientes industriales hostiles.

Como ejemplo de aplicación de las redes MESH tenemos a los sistemas distribuidos. Ha existido una tendencia en años recientes para colocar más inteligencia en los sistemas distribuidos. El estándar IEEE 1451 "Smart Transducer Interface for Sensor and Actuators" (Interfaz transductora inteligente para sensores y actuadores), es una prueba de ello. Una aplicación de este estándar se da de manera frecuente en sistemas de control distribuido, conformado por un "módulo de interfaz de transductor inteligente" (STIM por sus siglas en inglés) y por un "procesador de aplicación de red" (NCAP) conectado a la red. El STIM puede desempeñar mediciones locales y funciones de control dirigidas por el NCAP en cualquier punto de la red, como se muestra en la figura 2.5. De esta forma se puede mantener la temperatura de una habitación, una presión constante de un sistema hidráulico o el flujo constante de un líquido en una tubería.

La inteligencia distribuida es naturalmente ofrecida por redes inalámbricas multi-salto MESH. El control del sistema inalámbrico es distribuido a través de la red, permitiendo a los puntos inteligentes comunicarse directamente con otros puntos sin necesidad de encaminar la información a través de un punto central.

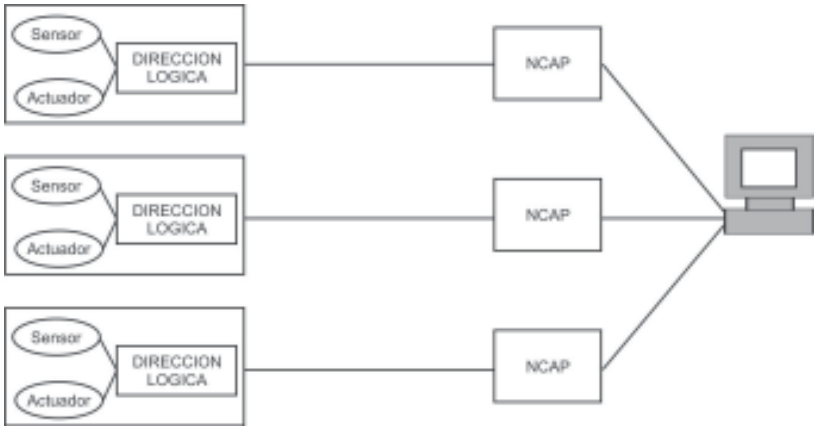

Figura 2.5.- Sistema de control distribuido operando bajo el estándar IEEE1451. Las mediciones obtenidas por los sensores remotos son analizadas por una unidad central para enviar señales a los actuadores para controlar alguna acción

2.2 El estándar IEEE 802.11

El Instituto de Ingenieros Eléctricos y Electronicos (IEEE) ratificó en 1997 el primer estándar para redes inalámbricas LANs denominado 802.11. Esta primera versión de 802.11 proponía tasas de datos de 1 Mbps y 2 Mbps, así como una variedad de métodos de señalización fundamentales y otros servicios.

El comité IEEE 802, es una autoridad mundialmente reconocida que ha establecido estándares que gobiernan la industria LAN desde hace dos décadas, por ejemplo 802.3 (Ethernet), 802.5 (Token Ring) y 802.3z 100Base-T (Fast Ethernet), entre otras. En 1997, después de siete años de trabajo, el IEEE publicó el estándar 802.11, el primer estándar internacional oficialmente aceptado para las redes inalámbricas LANs. En septiembre de 1999 ratificaron la mejora del estándar 802.11 conocido como 802.11b "tasa alta", el cual agrega dos velocidades más altas (5.5 y 11 Mbps).

Como todos los estándares IEEE 802, el estándar 802.11 se centra en los dos primeros niveles del modelo OSI, la capa física y la capa de enlace de datos (mostradas en la figura 2.6). Cualquier aplicación LAN, sistema operativo de red o protocolo, incluyendo TCP/IP y Novell NetWare, funcionará fácilmente en 802.11 tal como lo hace sobre Ethernet.

La arquitectura básica, características y servicios de 802.11b están definidos por el estándar original 802.11. La especificación 802.11b sólo afecta la capa física, adicionando tasas de datos mayores y una conectividad más robusta.

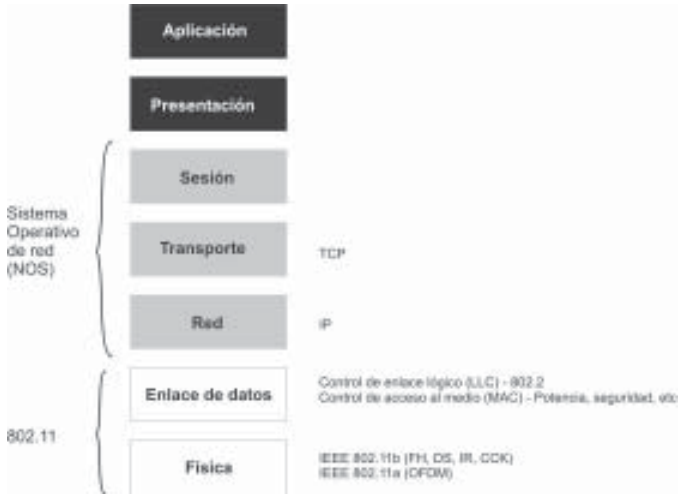

Figura 2.6.- 802.11 y el modelo OSI

2.2.1 La capa física en IEEE 802.11

Los tres tipos de tecnologías disponibles en la capa física originalmente definidos en 802.11**[6]**, incluyen dos técnicas de radio de espectro disperso y una especificación de infrarrojo difuso. El 802.11 y 802.11b operan dentro de la banda ISM de 2.4 GHz. Esta banda de frecuencias es reconocida por agencias regulatorias internacionales como la FCC (EUA), el ETSI (Europa) y el MKK (Japón) para operaciones de radio sin licencia. Por consiguiente, los productos basados en 802.11 no requieren una licencia para su uso. Las técnicas de espectro disperso, además de satisfacer los requerimientos regulatorios, incrementan la confiabilidad, mejoran el rendimiento y permiten que varios productos no relacionados compartan el espectro sin que dicha cooperación sea explícita y con una interferencia mínima.

El estándar original 802.11 define tasas de datos de 1 Mbps y 2 Mbps vía ondas de radio utilizando Espectro Disperso por salto de Frecuencia (FHSS) o Espectro Disperso de Secuencia Directa (DSSS). Es importante notar que FHSS y DSSS son mecanismos de señalización diferentes y no pueden trabajar uno con el otro.

Utilizando la técnica de salto de frecuencia, la banda de 2.4 GHz se divide en 75 subcanales de 1 MHz. El transmisor y el receptor están debidamente sincronizados en el patrón de salto, y el dato es enviado sobre una secuencia de subcanales. Cada conversación dentro de una red 802.11 ocurre sobre un patrón de salto diferente y los patrones son diseñados para minimizar la posibilidad de que dos transmisores utilicen el mismo subcanal simultáneamente.

La técnica FHSS es confiable para un diseño de radio simple, pero está limitada a velocidades no mayores de 2 Mbps. Esta limitación tiene que ver principalmente con las regulaciones de la FCC que restringen el ancho de banda del subcanal a 1 MHz. Estas regulaciones forzan a los sistemas FHSS a dispersar su uso a lo largo de la banda de 2.4 GHz, lo cual significa que deben hacerse brincoteos frecuentemente, lo que provoca una gran cantidad de brincoteos traslapados.

En contraste, la técnica de señalización de secuencia directa divide la banda de 2.4 GHz en 14 subcanales de 22 MHz cada uno. Los canales adyacentes se traslapan unos con otros parcialmente, con tres de los 14 que no se traslapan. Los datos son enviados a través de uno de esos canales de 22 MHz sin brincar a otros canales. Para compensar el ruido en un canal dado, se utiliza la técnica denominada *"chipping"*. La redundancia inherente de cada chip combinada con la dispersión de la señal a través del canal de 22 MHz provee una manera de verificación y corrección de errores; en caso de que una parte de la señal sea dañada, ésta puede ser recuperada en muchos casos, minimizando la necesidad de retransmisión.

La contribución principal del estándar 802.11b al estándar inalámbrico LAN fue la estandarización de dos nuevas velocidades soportadas en la capa física, la de 5.5 Mbps y 11 Mbps. Para lograr lo anterior, la técnica DSSS fué seleccionada como la única técnica de capa física a utilizar. Como se mencionó anteriormente, el brincoteo de frecuencia no puede soportar velocidades más altas sin la violación de las regulaciones actuales de la FCC. Lo que implica que los sistemas 802.11b interoperarán con los sistemas DSSS de 1 Mbps y 2 Mbps de 802.11, pero no funcionarán con los sistemas FHSS de 1 Mbps y 2 Mbps de 802.11.

El estándar original de 802.11 para DSSS especifica un código de *chipping* de 11 bits - llamado secuencia Barker- para codificar todos los datos enviados sobre el aire. Cada secuencia de 11 chips representa una señal de bits de datos (1 ó 0), y es convertida a símbolos, que pueden ser enviados a través del aire. Estos símbolos son transmitidos a una tasa de símbolo de 1 Mbps utilizando la técnica llamada modulación binaria en fase por desplazamiento (BPSK). En el caso de 2 Mbps, se utiliza la llamada modulación en cuadratura por desplazamiento en fase (QPSK); la cual duplica la tasa de datos disponible en BPSK, al mejorar la eficiencia en el uso del ancho de banda.

Para incrementar la tasa de datos en el estándar 802.11b se empleadan técnicas avanzadas de codificación. En lugar de dos secuencias Barker de 11 bits, 802.11b especifica un código complementario de llaveo (CCK), el cual consiste de un conjunto de 64 palabras codificadas de 8 bits.

Como un conjunto, este código de palabras tiene una propiedad matemática única que les permite ser distinguidas correctamente una de la otra en el receptor aún en la presencia de ruido sustancial e interferencia por multitrayectoria. La tasa de 5.5 Mbps utiliza CCK para codificar 4 bits por portadora, mientras que la tasa de 11 Mbps codifica 8 bits por portadora. Ambas velocidades utilizan QPSK como la técnica de modulación y señalización a 1.375 Megasímbolos por segundo y de esta manera se obtenienen tasas mayores de datos. En la tabla 2.1 se ilustran las tasas de datos y sus respectivas diferencias.

Tasa de datos(Mbps)	**Longitud del código**	**Tasa de datos(Mbps)**	**Tasa de datos(Mbps)**	**Tasa de datos(Mbps)**
1	11 (Secuencia Barker)	BPSK	1	1
2	11(Secuencia Barker)	QPSK	1	2
5.5	8(CCK)	QPSK	1.375	4
11	8(CCK)	QPSK	1.375	8

Tabla 2.1.- Especificaciones de tasa de datos en 802.11b

La figura 2.7(a) muestra la cobertura ideal para un punto de acceso inalámbrico *Roam About* de Enterasys 802.11b utilizada en ambientes abiertos (sin obstáculos). La instalación típica será en ambientes semiabiertos, como puede ser en oficinas e industrias. La cobertura ideal para ambientes semi-abiertos se muestra en la figura 2.7(b).

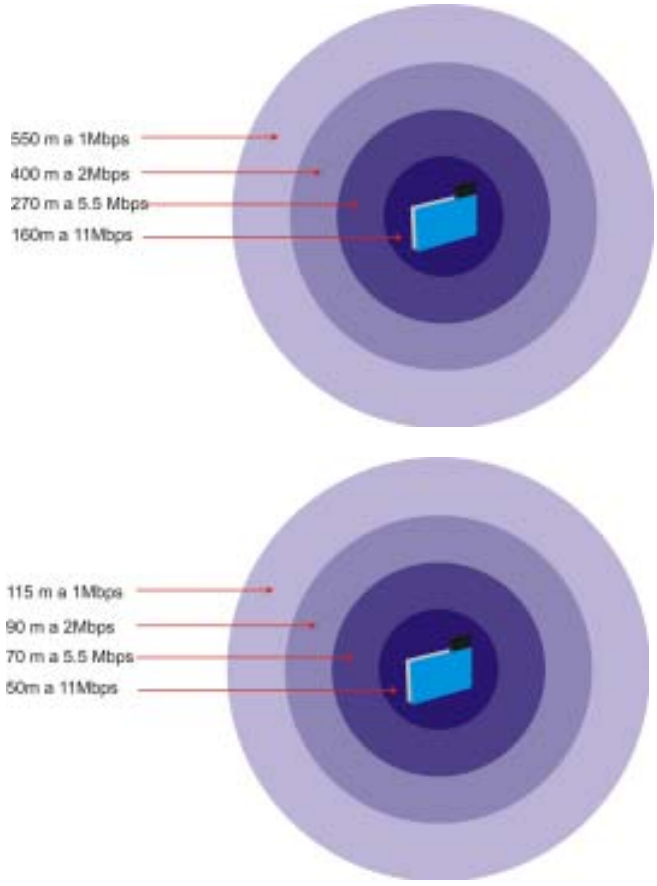

Figura 2.7(a).- [3]Cobertura ideal para ambientes abiertos bajo 802.11b.
(b).- Cobertura ideal para ambientes semi-abiertos bajo 802.11b

Para soportar ambientes con mucho ruido así como alcances extendidos, las WLANs 802.11b utilizan modulación adaptable, permitiendo que las tasas de datos se ajusten automáticamente compensando el cambio en la naturaleza del canal de radio. Idealmente, los usuarios se conectan a una tasa total de 11 Mbps. Sin embargo, cuando los dispositivos se mueven más alla del alcance óptimo de operación de 11 Mbps, o si está presente una inteferencia sustancial, los dispositivos 802.11b transmitirán a velocidades menores, llegando a velocidades de 5.5, 2 y 1 Mbps. También, si los dispositivos regresan al alcance de máxima transmisión, la velocidad de conexión regresará a una tasa de 11 Mbps. La modulació adaptable es un mecanismo de capa física transparente al usuario y a capas superiores de la pila de protocolo.

2.2.1.1 Administración de la frecuencia

El desarrollo de los mercados para servicios y equipos de comunicación inalámbrica dependen en gran parte de la disponibilidad del espectro en una banda de frecuencias apropiada, así como acuerdos entre los productores y proveedores de servicios en los estándares para las terminales inalámbricas y la infraestructura de soporte de la red. Las características de la propagación de la señal son diferentes de una banda de frecuencias a otra, lo cual implica diferencias en la cobertura de la señal.

Las redes de área local inalámbricas típicamente constan de un número de terminales estacionarias operando en ambientes de interiores, con la necesidad de interoperar con cada una de sus estaciones alámbricas e inalámbricas. Así, distintos diseños de productos han sido ofrecidos en el mercado WLAN y recientemente se ha dado un desarrollo común en las interfaces del estándar. La asignación del espectro para diferentes propósitos (redes de datos móviles y WLANs) utilizan bandas de frecuencias diferentes. Para redes de datos móviles, diseñadas para cubrir un área geográfica amplia, es necesario acceder a la banda LSMR (*Licensed Specialized Mobile Radio*), dado que debe convivir con diferentes bandas dentro del espectro radioeléctrico. Y para proveer diversos servicios se debe contar con las licencias necesarias de la comisión reguladora de comunicaciones correspondiente en las áreas del mercado a las cuales está destinado el servicio [7]. Para WLANs, la mayoría de los fabricantes han desarrollado productos para operar en la banda ISM y transmisión infrarroja (IR), de tal modo que evitan la licencia de administración de frecuencias; las diferentes bandas ISM se ilustran en la figura 2.8.

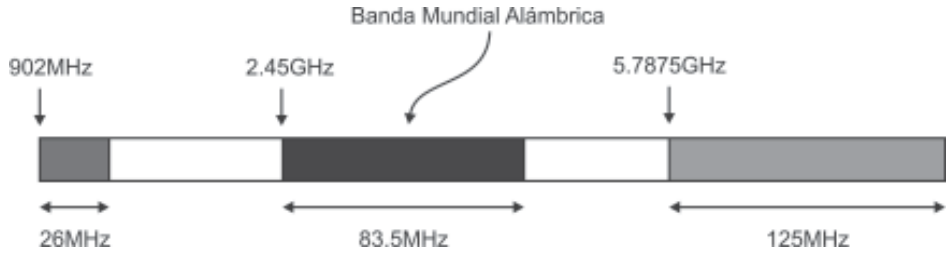

Figura 2.8.- Bandas de frecuencia de transmisión de equipo industrial, científico y médico ISM

Además, las bandas ISM están reguladas por el Instituto Europeo de Estandarización en Telecomunicaciones (ETSI), la FCC y el Instituto de Certificación e Inspección de equipo de radio (MKK-Japón). Cada agencia regulatoria del espectro determina la manera en que es utilizada la banda (por ejemplo la FCC – 11 canales, el ETSI – 13 canales y el MKK – 14 canales). La figura 2.9 muestra la banda de frecuencias sin licencia en el espectro electromagnético usado para WLANs [7].

2.2.1.2 Infrarroja (IR)

La tecnología infrarroja utiliza frecuencias muy altas, justo por debajo de la luz visible en el espectro electromagnético para transmitir datos. La tecnología IR no puede penetrar objetos opacos, entonces el mecanismo de propagación para transmisión de información es por línea de vista, y es por lo tanto

conveniente para habitaciones pequeñas. Además es más simple y barata con respecto a otras tecnologías. Esta tecnología es buena para sistemas de bajo costo, sin obstáculos y con un alcance muy limitado (aproximadamente un metro), como las redes de área personal (PAN). Todavía no es de gran utilidad en las redes inalámbricas de área local.

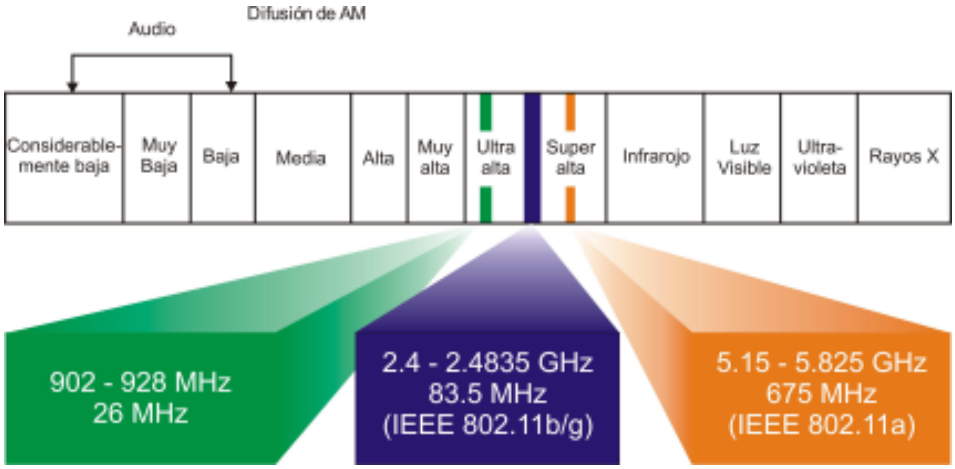

Figura 2.9.- Bandas ISM "asignadas" para WLANs

2.2.1.3 Radio de banda angosta

En radio de banda angosta, el usuario transmite y recibe información en una frecuencia específica. La banda de frecuencia es mantenida tan bajo como sea posible, justo lo suficiente para permitir el paso de la información. Cualquier diafonía entre los canales de comunicación es cuidadosamente evitada coordinando a los diferentes usuarios en diferentes canales de frecuencia. En un sistema de radio, la privacidad y ausencia de interferencia se logran utilizando bandas de frecuencias separadas. El receptor filtra todas las señales externas de radio, excepto la señal a la cual se encuentra calibrado. Esta tecnología requiere que el usuario final obtenga una licencia de algún organismo regulatorio del espectro de radio para cada sitio donde esta tecnología será distribuida.

2.2.1.4 Espectro Disperso (SS)

El espectro disperso fue desarrollado como una tecnología de RF de banda ancha para ser utilizado en misiones críticas con confiabilidad y seguridad en comunicaciones militares aunque con pérdidas de ancho de banda. En esta técnica de transmisión, el transreceptor dispersa la potencia de una señal a través de una banda más ancha de frecuencias. El proceso de dispersión provoca que la señal de datos sea menos susceptible a ruido eléctrico que las técnicas de modulación de radio convencionales. La señal transmitida utiliza un ancho de banda considerablemente más grande que el mínimo necesario para enviar la información. Sacrificando ancho de banda para ganar rendimiento señal a ruido contradice el deseo de conservar el ancho de banda. Además, si el receptor no se sintoniza a la frecuencia correcta, la señal de espectro disperso se parecerá a un ruido de fondo (como se muestra en la figura 2.10). La transmisión en radio de banda angosta y el ruido eléctrico interfieren en una pequeña porción de la señal de espectro disperso y resulta una menor interferencia y pocos errores cuando el receptor demodula la señal. Algunas de las ventajas de la modulación espectro disperso incluyen densidad espectral de baja potencia, operación de interferencia limitada, privacidad y posibilidades de acceso aleatorio. Básicamente existen dos formas de transmisión usando la técnica de espectro disperso : FHSS y DSSS[8].

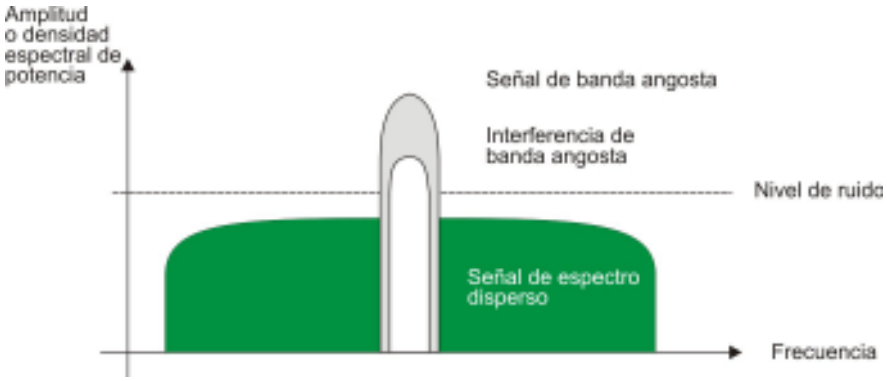

Figura 2.10.- Modulación del espectro disperso

2.2.1.4.1 Espectro disperso de brincoteo de frecuencia (FHSS)

En FHSS, la señal de datos se modula con una señal portadora de banda angosta que brinca con cierto patrón aleatorio de frecuencia a frecuencia como una función de tiempo sobre una amplia banda de frecuencias. La diversidad de frecuencias permite combatir la interferencia, lo cual se logra por medio de múltiples frecuencias, selección de código y FSK. Por ejemplo, un brincoteo de frecuencia de radio brincará a la frecuencia de portadora entre 2.4 GHz y 2.483 GHz que es la banda de frecuencia ISM de 2.4 GHz. Si se encuentra interferencia en una frecuencia, se retransmitirá la señal en un brinco subsecuente hacia otra frecuencia. Como se muestra en la figura 2.11, por ejemplo el paquete A se transmite en la ranura de 2.40 a 2.41 Ghz, en el caso de que se detecte un error de transmisión se retransmite el paquete A de acuerdo al patrón de brincoteo en alguna ranura subsecuente, y si no se detectara un error de transmisión se continua con el proceso de transmisión de paquetes B, C, E y D con la información prevista.

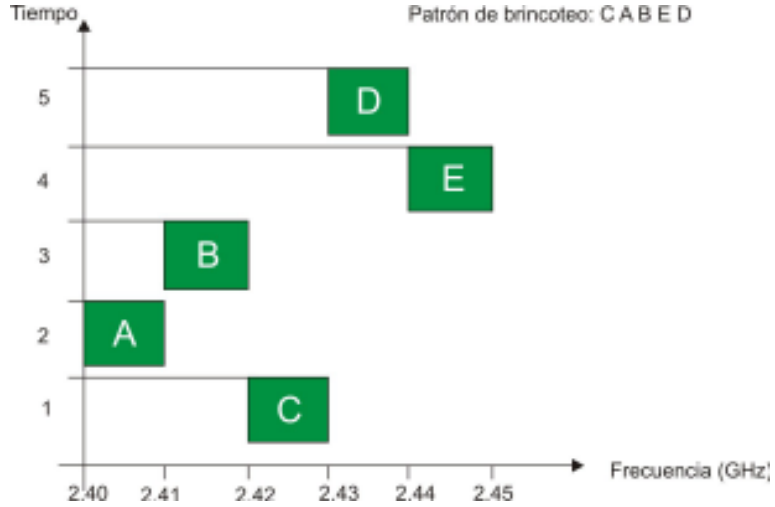

Figura 2.11.- Patrón de brincoteo de FHSS

El código de brincoteo determina la frecuencia de radio sobre la cual transmitirá y en que orden. Tanto el transmisor como el receptor conocen los patrones de brincoteo. Para recibir una señal apropiadamente, el receptor debe ser configurado con el mismo código de brincoteo y escuchar la señal que está llegando en el momento preciso y frecuencia correcta. Un receptor configurado incorrectamente recibirá señales FHSS como ruido impulsivo de corta duración. Debido a la naturaleza de esta técnica de modulación, el brincoteo de frecuencia puede llevar a cabo una transferencia de datos hasta de 2 Mbps;

tasas de datos más rápidas son susceptibles a un incremento en el número de errores. Sin embargo, la técnica de FH reduce la interferencia. Una señal de interferencia de un sistema de banda angosta afectará a una señal de SS sólo si ambas se transmiten a la misma frecuencia y al mismo tiempo. A diferencia del Multiplexaje por División de Frecuencia (FDM), donde cada canal mantiene una ranura de frecuencia durante toda la transmisión [8].

2.2.1.4.2 Espectro disperso de secuencia directa (DSSS)

La secuencia directa es más avanzada, más reconocida y más usada para espectro disperso hoy en día [1]. El proceso de DSSS se realiza efectuando una multiplicación entre una portadora de RF y una señal digital de pseudo-ruido (PN). Primero, el código de PN es modulado sobre la señal de información utilizando una de las varias técnicas de modulación (como BPSK o QPSK). Entonces un modulador balanceado doble se usa para multiplicar la portadora de RF y la señal de información modulada de PN. Este proceso provoca que la señal de RF sea reemplazada con una señal de ancho de banda amplia con el equivalente espectral de una señal de ruido.

En el dominio de la frecuencia las señales generadas por la técnica DSSS parecen ser ruido. El amplio ancho de banda provisto por el código de PN permite que la señal caiga por debajo del umbral de ruido sin pérdida de información. En la estación de transmisión DSSS combina la señal de datos con una secuencia rápida de bits de datos. El amplio ancho de banda incrementa la resistencia de la señal a la interferencia.

Un código de *chipping* es asignado para representar el "1" y "0" lógicos. Mientras la cadena de datos se transmite, en realidad es el código de chipping correspondiente el que es enviado. Una palabra código de 10-chips se usa para cada bit de datos 1. De manera similar pero invertida se usa una palabra de código de 10-chips para los bits de datos 0. La figura 2.12 muestra como en DSSS se envían cadenas específicas de chips (*chipping code*) por cada dato enviado. DSSS genera patrones de bits de redundancia para cada bit a ser transmitido.

Este patrón de bits es llamado código de *chip/chipping*. Un incremento en la longitud del chip aumentará la probabilidad de que el dato original sea recuperado, pero por lo consiguiente se requiere un mayor ancho de banda para transmitir mas chips. Si uno o más bits son dañados durante la transmisión, utilizando técnicas estadísticas en el radio, se puede recuperar el dato original sin la necesidad de retransmisión. Un receptor mal configurado observará la señal DSSS como un ruido de baja potencia y de banda ancha, y por otro lado dicha señal será ignorada o rechazada por la mayoría de los receptores de banda angosta [8].

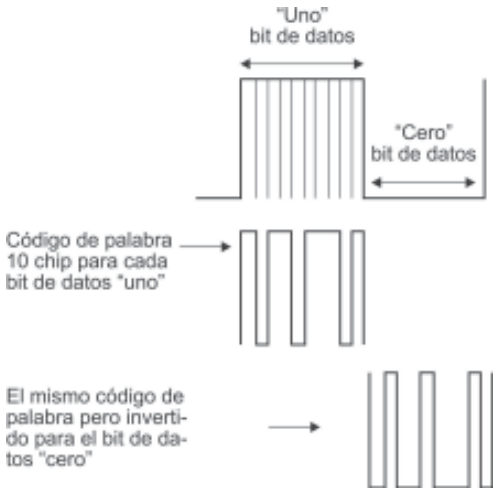

Figura 2.12.- Ejemplo de códigos de palabra de 10 chips para transmitir un uno o cero binario en DSSS.

Tanto FHSS como DSSS son utilizados en los desarrollos inalambricos, por lo cual es importante conocer sus ventajas y desventajas al momento de diseñar una WLAN. En la tabla 2.2 se muestra lo anterior.

Secuencia Directa (DS)	Brincoteo de Frecuencia (FH)
Mejor rendimiento	Inmune a la interferencia
Mayor alcance	Resistente al ECO
Escalable a frecuencias mayores a 2.4GHz	Más barato instalación más sencilla. Mayor variedad de productos así como de distribuidores

Tabla 2.2.- Secuencia directa vs brincoteo de frecuencia

2.2.1.5 Multiplexaje por división de frecuencias ortogonales (OFDM)

El multiplexaje por división de frecuencias ortogonales no es una nueva tecnología. Los ingenieros han usado esta técnica de modulación en múltiples desarrollos de tecnología de banda ancha.

Muchos de los sistemas modernos de comunicación por microondas están basados en una técnica de modulación llamada modulación digital de amplitud en cuadratura (QAM). Estos sistemas de comunicación varían en cuanto a su complejidad, pero por otro lado hay sistemas más simples que utilizan el esquema de modulación denominado modulación digital por desplazamiento en fase (PSK), el cual es relativamente fácil de desarrollar. Los sistemas PSK también tienen una tasa de datos relativamente baja. La modulación PSK no modifica la forma de onda de la señal en amplitud ni en frecuencia, pero modifica su fase.

En la modulación digital binaria por desplazamiento en fase (BPSK), las fases de la onda senoidal inician en 0 (el punto de origen del ciclo senoidal) o a 1/2 (que es 1/2 de ciclo del origen de la onda senoidal). La modulación BPSK transmite sólo 1 bit de información en cada ciclo de la onda senoidal (también llamado símbolo).

Esquemas mas complejos de modulación transmiten más de 1 bit por símbolo. La modulación digital en cuadratura por desplazamiento en fase (QPSK) es similar a BPSK. Sin embargo, en lugar de dos estados separados en fase, QPSK usa 4 (0,1/4,1/2, y 3/4) transportando 2 bits por símbolo. A diferencia de BPSK, QPSK es menos robusto. Debido a que QPSK modula solamente 2 bits por símbolo, no es muy eficiente para comunicaciones de alta velocidad. Por lo tanto, tasas de datos altas necesitan una asignación significativa del ancho de banda.

Aunque QPSK no usa cambios de estado en amplitud, esto es referido algunas veces como 4-QAM (cuatro estados de fase). Cuando se combinan cuatro niveles de amplitud con cuatro niveles de fase, se obtiene 16-QAM. 16-QAM codifica 2 bits en cambios de fase, y 2 bits en cambios de amplitud, soportando un total de 4 bits por símbolo.

Se puede expander esta modulación a 64-QAM y 256-QAM o mas allá. 64-QAM es muy popular tanto en productos alámbricos como en inalámbricos, y 256-QAM está siendo probado [9].

El multiplexaje por división de frecuencia (FDM) transmite simultáneamente múltiples señales sobre una trayectoria simple de transmisión. Cada señal viaja dentro de un intervalo de frecuencia único (una portadora), la cual es modulada por los datos. Estos datos pueden incluir cadenas de datos de voz o video. La figura 2.13(a) muestra la relación de las portadoras dentro de una asignación de frecuencias. Aunque FDM incrementa el ancho de banda, también reduce la interferencia de intersímbolos por multitrayectoria (ISI). Existe una gran ineficiencia en el uso de ancho de banda: hasta el 50 % del espectro disponible es desperdiciado en bandas de reserva, las cuales aseguran la separación entre frecuencias.

La técnica de espectro disperso OFDM utiliza QAM y un procesamiento avanzado de la señal digital, distribuyendo los datos sobre múltiples portadoras espaciadas a frecuencias precisas, como se observa en la figura 2.13(b). El espaciamiento preciso pemite la ortogonalidad que evita que los demoduladores detecten otras frecuencias que no sean las de interés. Debido a que cada portadora puede ser identificada de manera única, las bandas de reserva son eliminadas, incrementando la eficiencia de uso de espectro de frecuencia. La figura 2.14 describe la ortogonalidad de varias frecuencias de portadoras.

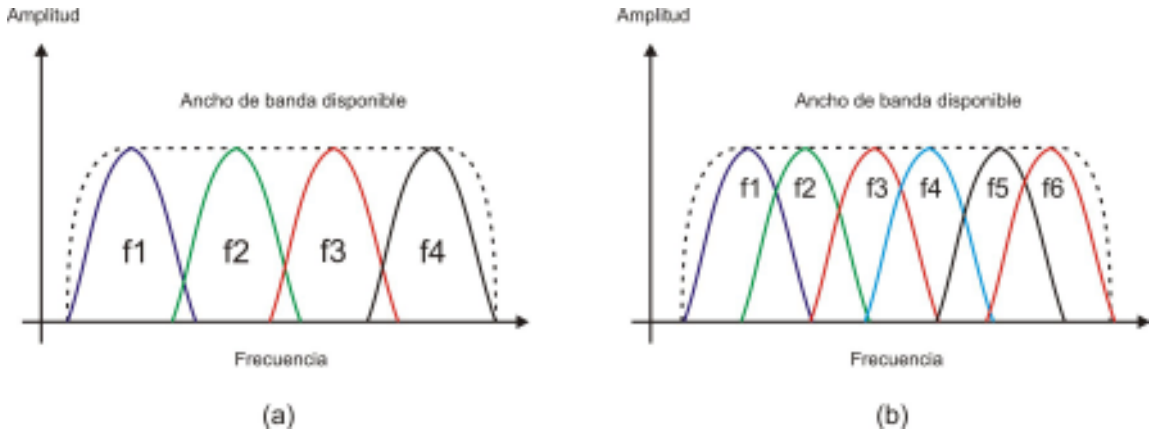

Figura 2.13.- Técnicas de transmisión de señales FDM y OFDM. a) El multiplexaje por división de frecuencia (FDM) utiliza portadoras múltiples para transmitir información agregada. Las señales viajan dentro de un intervalo único de frecuencias moduladas por los datos. (b) La técnica de espectro disperso de multiplexaje por división de frecuencias ortogonales (OFDM) permite un uso mas eficiente del ancho de banda eliminando la necesidad de utilizar bandas de reserva como en FDM

OFDM es un método especial para modulación de múltiples portadoras. Como todos los esquemas de transmisión inalámbricos, OFDM codifica datos sobre señales de radiofrecuencia. En particular, en OFDM una portadora simple de alta frecuencia es reemplazada por múltiples subportadoras, cada una operando a una frecuencia significativamente más baja. Por lo tanto, OFDM es la división de un canal simple de radio de alta frecuencia en múltiples subportadoras. El dato es transmitido en una cadena de bits en paralelo a las subportadoras, con lo cual cada una de estas cadenas de bits se modula en un subportadora separada.

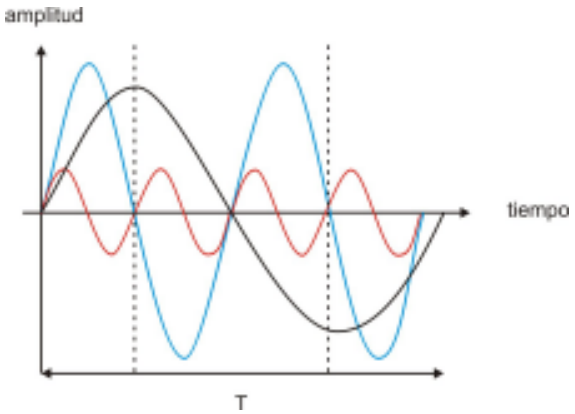

Figura 2.14.- Ortogonalidad de varias frecuencias portadoras. Frecuencia de base = 1/T, donde T = periodo de símbolo.

El *throughput* agregado es el mismo pero la tasa de datos en cada subportadora es mucho menor. Por tanto, esto hace a cada símbolo más largo, pero prácticamente elimina el efecto de los retardos de tiempo. Un inconveniente en este modo de modulación es la necesidad de un amplificador altamente lineal, así como sus sistemas de filtraje con el objeto de evitar interferencia entre las portadoras debido a

las armónicas. Tales armónicas se generan por las no-linealidades del amplificador y son una importante fuente de interferencia en modulaciones con señales envolventes variables en el tiempo.
La transmisión en paralelo de múltiples subportadoras basadas en OFDM para WLANs operan a altas tasas de datos, tales como 54 Mbps para desarrollos IEEE 802.11a e HiperLAN2[8].

Muchos estándares alámbricos e inalámbricos han adoptado OFDM en una gran variedad de aplicaciones. Por ejemplo, OFDM es el estándar global para la tecnología digital en el lazo local conocida como Línea de Suscriptor Asimétrico Digital (ADSL), entre otras.

Un problema al desarrollar productos WLANs basados en OFDM, es el alcance limitado que permite debido a sus frecuencias altas de operación combinada con potencias de transmisión relativamente bajas. Esto provoca una dificultad para mantener una aceptable relación portadora a ruido (C/N) sobre grandes distancias. Como resultado, son necesarios un gran número de puntos de acceso para proveer coberturas de radio suficientes para soportar la movilidad en áreas locales .

2.2.2 Tecnologías de la capa de enlace de datos

La técnica CSMA/CD[34] y sus precursoras pueden ser denominadas de acceso aleatorio o de contención, puesto que no existe un tiempo preestablecido o predecible para que las estaciones trasmitan; la transmisión se realiza de manera aleatoria. Son de contención en el sentido de que las estaciones compiten para conseguir el acceso al medio.

En CSMA, una estación que desee transmitir, primero escuchará el medio para determinar si existe alguna otra transmisión en curso (sensa la portadora). Si el medio se está usando, la estación deberá esperar. En cambio, si éste se encuentra libre, la estación podrá transmitir. Puede suceder que dos o más estaciones intenten transmitir aproximadamente al mismo tiempo, en cuyo caso se producirá colisión: los datos de ambas transmisiones interferirán y no se recibirán con éxito. Para solucionar esto, las estaciones aguardan un período de tiempo razonable después de transmitir en espera de una confirmación, teniendo en consideración el retardo de propagación máximo del trayecto de ida y vuelta y el hecho de que la estación que confirma debe competir también por conseguir el medio para responder. Si no llega la confirmación, la estación supone que se ha producido una colisión y retransmite.

Podemos ver cómo esta estrategia resulta efectiva para redes en las que el tiempo de transmisión de trama es mucho mayor que el de propagación. Las colisiones sólo se producirán en el caso de que más de un usuario comience a transmitir dentro del mismo intervalo de tiempo (igual al período de propagación). Si una estación comienza a transmitir una trama y no existen colisiones durante el tiempo de propagación que transcurre desde el inicio de la transmisión del paquete hasta que alcanza a la estación más lejana, no se producirá colisión para esta trama dado que ahora todas las estaciones están enteradas de la transmisión.

La utilización máxima que se puede conseguir haciendo uso de CSMA puede superar con mucho la del ALOHA ranurado. La utilización máxima depende de la longitud de la trama y del tiempo de propagación; cuanto mayor sea la longitud de las tramas o cuanto menor sea el tiempo de propagación, mayor será la utilización.

En CSMA se necesita un algoritmo que determine qué debe hacer una estación si encuentra el medio ocupado. La técnica denominada 1-persistente es la aproximación más usual, y es la utilizada en lEEE 802.3. Una estación que desee transmitir escuchará el medio y actuará de acuerdo con las siguientes reglas:
1.- Si el medio se encuentra libre, transmite; si no se aplica la regla 2.
2.- Si el medio está ocupado, continúa escuchando hasta que el canal se detecta libre, entonces transmite inmediatamente. Se producirá unacolisión, siempre que dos o más estaciones estén en espera de transmitir. Esta técnica sólo toma medidas tras la colisión, como la espera de un tiempo aleatorio.

2.2.2.1 CSMA/CD

CSMA, aunque más eficiente que ALOHA y ALOHA ranurado, es también claramente ineficiente, Cuando colisionan dos tramas, el medio estará inutilizado mientras dure la transmisión de ambas. La capacidad desaprovechada, en comparación con el tiempo de propagación puede ser considerable para tramas largas. Este desaprovechamiento puede reducirse si una estación continúa escuchando el medio mientras dura la transmisión, lo que conduce a las siguientes reglas para la técnica CSMA/CD [34](Acceso Múltiple por Detección de Portadora con Detección de Colisones):

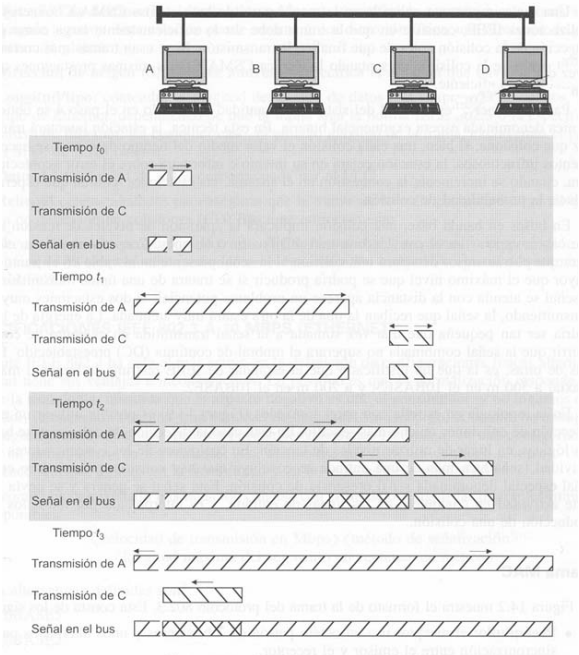

Figura 2.15.- Funcionamiento de CSMA/CD[34]

1. La estación transmite si el medio está libre, si no se aplica la regla 2.

2. Si el medio se encuentra ocupado, la estación continúa escuchando hasta que encuentra libre el canal, en cuyo caso transmite inmediatamente.

3. Si se detecta una colisión durante la transmisión, las estaciones transmiten una señal corta de alerta para asegurarse de que todas las estaciones constatan la colisión y cesan de transmitir.

4. Después de transmitir la señal de alerta se espera un intervalo de tiempo de duración aleatoria, tras el cual se intenta transmitir de nuevo (volviendo al paso 1).

La figura 2.15 ilustra la técnica para el caso de un bus en banda base. La estación A comienza a transmitir un paquete con destino D en el instante de tiempo t_0. Tanto B como C están dispuestas para transmitir en t_1, B detecta una transmisión y aplaza la suya. Sin embargo, C aún no se ha percatado de la transmisión de A (debido a que el primer bit de la transmisión de A todavía no ha alcanzado a C) y comienza a transmitir. Cuando la transmisión de A llega a C, en t_2, ésta detecta la colisión y cesa de transmitir. El efecto de la colisión se propaga hacia A, donde se detecta en un instante posterior, t_3, siendo en este momento cuando A deja de transmitir.

La capacidad desaprovechada en CSMA/CD se reduce al tiempo que se tarda en detectar la colisión. Una regla importante aplicada en la mayor parte de los sistemas CSMA/CD, incluyendo a las normalizaciones

IEEE, consiste en que la trama debe ser lo suficientemente larga como para permitir la detección de la colisión antes de que finalice la transmisión. Si se usan tramas más cortas, no se produce la detección de la colisión, presentando la técnica CSMA/CD las mismas prestaciones que el protocolo CSMA menos eficiente.

Para mantener la estabilidad del sistema, la cantidad de retardo en el paso 4 se obtiene mediante la técnica denominada espera exponencial binaria. En esta técnica, la estación intentará transmitir por cada vez que colisione, si bien, tras cada colisión el valor medio del tiempo de espera se duplica. Tras 16 intentos infructuosos, la estación cejará en su intento e informará sobre el error acontecido. De esta manera, cuando se incremente la congestión en el sistema, las estaciones tendrán que esperar más, para así reducir la probabilidad de colisión.

En buses en banda base, una colisión implicará la aparición de niveles de tensión superiores a los que cabría esperar en el caso de una transmisión sin colisiones. Consecuentemente, el estándar IEEE dicta que el transmisor detectará una colisión si la señal presente en el cable en el punto de conexión es mayor que el máximo nivel que se podría producir si se tratara de una única transmisión. Debido a que la señal se atenúa con la distancia aparece un problema potencial: si dos estaciones muy distantes están transmitiendo, la señal que reciban la una de la otra estará muy atenuada. La energía de la señal recibida podría ser tan pequeña que una vez sumada a la señal transmitida en el punto de conexión, pudiera ocurrir que la señal combinada no superara el umbral preestablecido. Esta razón, además de otras, es la que ha justificado que el estándar de IEEE restrinja la longitud máxima del cable coaxial a 500 m en el 10BASE5, y a 200 m en el 10BASE2.

2.2.2.2 CSMA/CA

Otra adaptación de CSMA conocida como CSMA con prevención de colisiones (CSMA/CA), se utiliza en la mayoría de los sistemas actuales[10]. Su principio de operación se muestra en la figura 2.16.

Como se puede apreciar, en lugar de iniciar la transmisión de una trama de manera inmediata después de que el medio se encuentre libre, el nodo espera primero un intervalo de tiempo aleatorio y sólo si el medio permanece libre después de este intervalo, empezará la transmisión. En este sentido, si otros nodos se encuentran esperando también, entonces el nodo que tenga el tiempo más corto ganará el acceso y los nodos restantes diferirán su transmisión. La eficiencia del esquema es una función del número de incrementos de tiempo y por tanto de los bits en la secuencia pseudo-aleatoria en el período máximo de tiempo de prevención de colisiones.

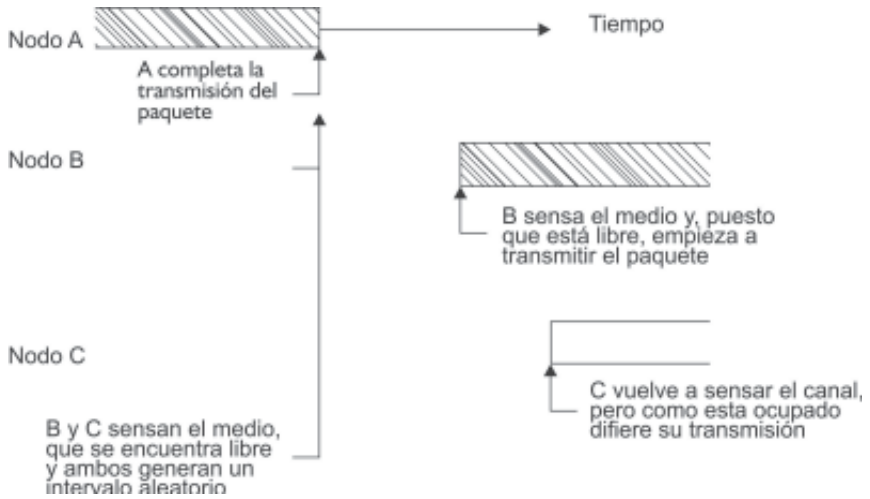

Figura 2.16.- MAC inalámbrico: protocolo CSMA/CA

Otro factor importante que debe de tomarse en cuenta cuando se utilizan comunicaciones de radio e infrarrojas, es el hecho de que no se garantiza que el nodo o destinatario de las comunicaciones se encuentre en contacto por radio con el nodo fuente. Por tanto, debe de utilizarse un procedimiento adicional de "apretón de manos" (handshake), por encima del método básico MAC a incorporarse en este protocolo, conocido también como Protocolo de MAC inalámbrica distribuida(DFW MAC).

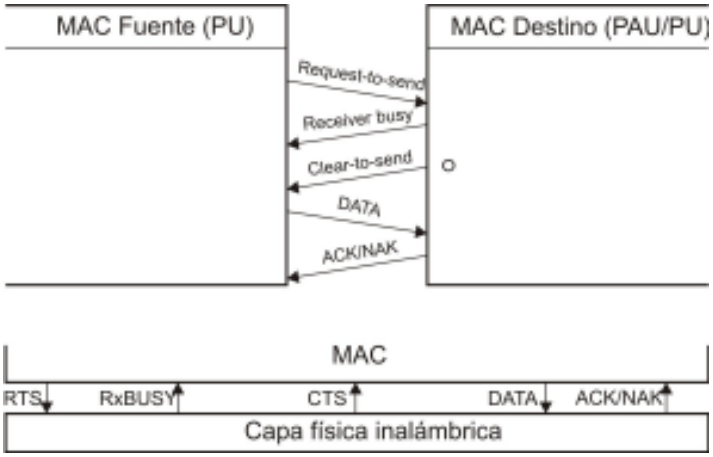

Figura 2.17.- MAC inalámbrica. Procedimiento de handshake en el protoclo DFW MAC

En el momento en que una unidad portátil necesite enviar un paquete, envía primero un paquete de "petición de envío" (RTS). Este mensaje de control contiene la dirección MAC de la fuente y destino, que al ser recibido por el destinatario, éste transmite un mensaje "libre para enviar" (CTS) con el mismo par de direcciones MAC, pero con orden inverso. De manera alternativa, si el destinatario no se encuentra preparado para recibir un paquete, regresará un paquete de "receptor ocupado" (RxBUSY). Si la respuesta es positiva, entonces la unidad que inicia la petición transmite el paquete (DATA) y, si se recibe correctamente, el destinatario regresa una confirmación positiva (ACK). Si embargo, si el paquete llega con errores, entonces se envía un mensaje de confirmación negativa (NAK) y la fuente tratará de transmitirlo nuevamente. Este procedimiento se repetirá hasta cierto número máximo de veces definidas. El diagrama que muestra el funcionamiento de este protocolo se muestra en la figura 2.17.

En el presente capítulo se analizó el estado del arte actual de las tecnologías de red inalámbrica, donde sobresalen las topologías definidas en el estándar IEE802.11, red de infraestructura inalámbrica, *ad-hoc* y la interconección LAN-LAN. De igual manera dentro de las redes *ad hoc*, los estudios actuales están orientados hacia la aplicación de res de sensores y redes MESH, para el despliegue de redes capaces de atoconfigurarse y de obtener datos donde la presencia humana no es posible o necesaria. También se definió a detalle el estándar IEEE802.11 y se estudió su evolución en busca de tasas de transferencias más altas, así como las tecnologías de acceso al medio más utilizadas actualmente.

III. Generalidades de las redes sin infraestructura (*ad hoc*)

Una red sin infraestructura o del tipo *ad hoc*, está formada por nodos inalámbricos fijos o móviles que se comunican entre sí, transportando los paquetes de información del nodo origen al nodo destino mediante saltos entre nodos intermedios. Por su naturaleza, es posible que se formen rutas múltiples entre dos nodos que se comunican [11].

Un factor importante en el desempeño de las redes del tipo *ad hoc* es la movilidad, la cual determina cambios periódicos en las rutas establecidas entre transmisor y receptor conforme pasa el tiempo. Una ruta establecida y utilizada durante cierto período, puede volverse inaccesible cuando los nodos participantes cambian su posición.

3.1 Aplicaciones de las redes *ad hoc*

Algunas características que hacen interesante la conformación de redes *ad-hoc*, son la rapidez y facilidad con la que son desplegadas y su decreciente independencia de una infraestructura fija. En diversos documentos de investigación, se plantea su uso en:

a) Redes de área personal: Teléfonos celulares conectándose con computadoras portátiles, audífonos y relojes
b) Ambientes militares: Comunicación entre soldados, tanques y aviones en el campo de batalla
c) Ambientes civiles: Red de taxis, salas de reuniones, estadios deportivos, botes e inclusive aeronaves pequeñas.
d) Operaciones de emergencia: Búsqueda y rescate, cuerpos policíacos y de bomberos.
e) Redes de sensores
f) Redes tipo MESH

3.2 Condiciones de operación

Una red que trabaja en modo sin infraestructura, posee condiciones de operación particulares en comparación con una red de datos convencional e incluso si se le compara también con una red inalámbrica con una entidad de control centralizado[11]. Una red *ad hoc* tiene las siguientes características de operación:

a) Ambiente completamente simétrico: Todos los nodos poseen capacidades y responsabilidades idénticas. Todo nodo tiene la responsabilidad de participar en la transmisión de paquetes en la red, sirviendo como *routers* al relevar los paquetes e incluso descubrir nuevas rutas para la entrega confiable de la información, contando a cambio con funciones recíprocas de los demás nodos de la red.
b) Capacidades asimétricas: Los rangos de transmisión y los radios en sí mismos pueden ser diferentes. Debido a que los equipos son móviles y poseen como fuente de poder una batería con capacidad limitada, su tiempo de vida y carga puede variar entre los nodos, cuya capacidad de procesamiento

puede ser distinta. Finalmente, debemos considerar que la velocidad con la que se mueven los usuarios varía constantemente.

c) Responsabilidades asimétricas: Sólo un conjunto de nodos puede estar participando en el relevo de paquetes y, en algunos casos, algunos nodos pueden actuar como líderes de otros nodos en vecindad (por ejemplo, facultades de cabeza de clúster).
d) Las características del tráfico puede ser diferente en las redes *ad hoc*: Debido a variaciones en la velocidad de transmisión, requerimientos de confiabilidad, transmisiones del tipo unicast, multicast y geocast.
e) Una red *ad-hoc* puede convivir con una red cuya infraestructura es centralizada: A esto se le denomina red híbrida .

Como consecuencia de algunas de las condiciones previamente mencionadas, encontramos factores que son determinantes en el funcionamiento de una red sin infraestructura. Por ejemplo en este tipo de red, los patrones de movilidad pueden ser diferentes dependiendo de la aplicación: personas esperando en una sala de un aeropuerto, grupos de servicio de taxi en una ciudad, movimientos de personal militar o bien redes de área personal. Por otro lado también varían la velocidad, dirección y patrones de movimiento, así como la uniformidad (o falta de) de la movilidad entre diferentes nodos.

3.3 Retos

Las condiciones de trabajo en una red del tipo *ad hoc*, presentan los siguientes retos

1) Rango de transmisión inalámbrica limitada: El radio integrado a una interfaz de red inalámbrica, encuentra reducido su rango de transmisión, principalmente debido a que baterías de tamaño pequeño y tiempo de vida establecido, permiten transmisiones de relativa baja potencia (también debido a regulaciones de la FCC).
2) Naturaleza de multidifusión (*broadcast*) del medio: La transmisión es "escuchada" por todos los usuarios dentro del rango de transmisión, ocasionando problemas como "terminal escondida" y "terminal expuesta" (que se analizarán a detalle más adelante).
3) Pérdidas de paquetes debidas a errores en la transmisión: Colisiones, rupturas de ruta, entre otros eventos, ocasionan que se pierdan paquetes antes de llegar a su destino.
4) Cambios de ruta inducidos por la movilidad: El movimiento de los nodos, a cierta velocidad y con una dirección aleatoria, ocasionan que aquellas rutas formadas para entregar paquetes entre dos nodos dejen de funcionar, dando como resultado una topología dinámica.
5) Seguridad: Relativa vulnerabilidad ante ataques o robo de información.

3.3.1 El problema de las terminales ocultas y expuestas

El uso de CSMA/CD , se vuelve problemático en redes inalámbricas; los radios son half-dúplex y no pueden escuchar fácilmente mientras transmiten como en Ethernet. Por lo tanto, son necesarias técnicas de resolución de colisones (CA) en lugar de detección de colisiones (CD). Principalmente nos enfrentamos a dos problemas[12]:

a) Problema de terminales ocultas: Como se muestra en la figura 3.1, debido a la distancia entre terminales y el alcance del radio de cada una, A puede enviar paquetes hacia B pero C no puede recibir de A, ocasionado por la atenuación que sufre la señal (puesto que la potencia decae

proporcional a la distancia elevada a la n, para $2 \leq n < 4$). Por tanto, C podría transmitir hacia B, sin detectar una posible transmisión simultánea de A hacia B, ocasionando una colisión.

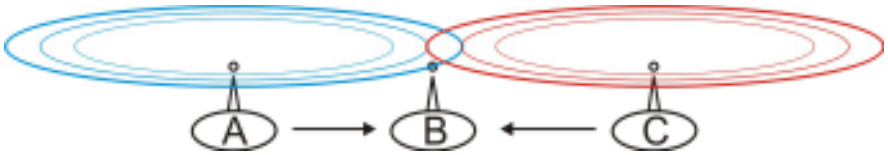

Figura 3.1 .- El nodo A es una terminal oculta para C

b) Problema de terminal expuesta: Para el caso que se muestra en la figura 3.2, la terminal B está transmitiendo a la terminal A y por su parte C transmite a D. Al dar lugar las transmisiones simultáneamente, C puede detectar que el canal está ocupado y por tanto debe esperar a su liberación. Sin embargo, A está fuera del alcance de C por lo que es innecesario la espera, ocasionando una detección incorrecta del canal. Cabe mencionar que no hay solución para el problema de la terminal expuesta.

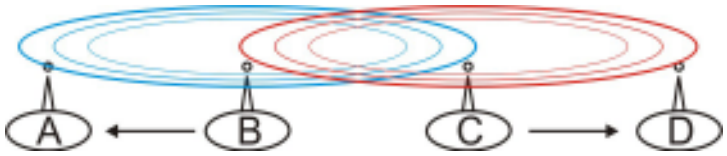

Figura 3.2 .- El nodo C está expuesto al nodo B

La solución algunos de estos problemas se trató a detalle en el apartado dedicado a la MAC 802.11 en el capítulo anterior, que consiste en el envío de pequeños paquetes de control denominados "petición de envío" (RTS por sus siglas en inglés) y "libre para enviar" (CTS).

3.4 Protocolos de encaminamiento en redes inalámbricas sin infraestructura

El encaminamiento en redes inalámbricas sin infraestructura es en muchos aspectos diferentes al enrutamiento en redes alámbricas convencionales, debido principalmente a la movilidad de los nodos causante de la ruptura y reparación periódica del enlace. Además se añaden como parámetros de desempeño la estabilidad de ruta y consumo de energía.

Muchos protocolos de enrutamiento se han propuesto para este tipo de redes en particular, algunos inventados específicamente para redes *ad- oc* y otros adaptados a partir de protocolos existentes para redes guiadas. Sin embargo, se ha probado que un protocolo en particular no trabaja bien en todos los ambientes, por lo que se han hecho intentos para proponer protocolos adaptivos.

En general, los protocolos de encaminamiento se pueden clasificar como:

a) Protocolos proactivos: Determinan rutas independientemente del patrón de tráfico. Como ejemplo tenemos protocolos basados en el estado del enlace y de vector distancia. Este tipo de protocolos presentan menor tiempo de latencia debido a la continua actualización de las rutas, pero con el costo de que se genera una mayor cantidad de paquetes de control en comparación con el número de paquetes de datos (es a lo que se le llama *overhead*).
b) Protocolos reactivos: Crean y mantienen rutas solo cuando es necesario, pero presentan mayor tiempo de latencia. Por otro lado, el *overhead* es menor que en el caso de los protocolos reactivos.
c) Protocolos híbridos: Presentan las características de los arriba mencionados de acuerdo a las circunstancias de la red.

Para definir cual de estos protocolos tendrá un mejor desempeño entre paquetes de control y paquetes de datos generados, dependerá en buena manera de los patrones de movilidad y el tráfico.

3.4.1 El concepto de inundación (*flooding*)

Cuando un nodo S desea establecer una comunicación con otro nodo D dentro de la red, aquel transmite un paquete P que puede ser de datos o de control a todos sus vecinos dentro del área de cobertura de su radio. Cada uno de estos vecinos procederá a retransmitir el paquete a sus vecinos particulares y así sucesivamente hasta alcanzar al nodo destino. Sin embargo, existe la posibilidad de que se reciban paquetes duplicados en cierto nodo, por lo cual es común utilizar números de secuencia dentro del encabezado del paquete, para que de esta forma no se presenten retransmisiones innecesarias. Cuando el paquete P alcance al nodo destino D (es decir, D es alcanzable por el nodo origen S), ya no se retransmite el paquete, recuperando la información de quiénes participaron en la ruta para ser usada posteriormente en la comunicación. Los efectos de este mecanismo se presentan en la figura 3.3.

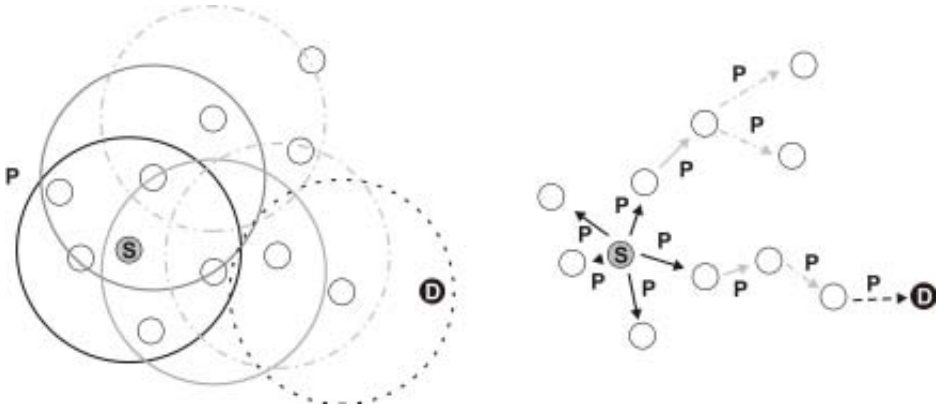

Figura 3.3.- Inundación (flooding) en una red ad hoc. (a) Transmisión y relevo del paquete P, ilustrando la cobertura del radio de cada nodo. (b) Propagación del paquete P a través de la red hasta alcanzar al destino.

En la figura 3.3, podemos observar claramente el impacto que representa la inundación (*flooding*) en una red *ad-hoc*. Primeramente, debido a la naturaleza *broadcast* del medio, cuando tenga lugar una transmisión, todos aquellos nodos dentro del área de cobertura del nodo iniciador van a recibir el paquete. Recordando el funcionamiento de la MAC 802.11 con prevención de colisiones, aquel nodo que desee transmitir deberá esperar a que se desocupe el canal cuando una transmisión ajena se encuentra en curso, de tal forma que cuando ocurra una inundación de paquetes, se presentarán retardos y hasta posibles colisiones a cualquier otra transmisión en curso, impactando la eficiencia de la red.

Por otro lado, debido a la movilidad, las rutas pierden validez, y por tanto el mecanismo de inundación para encontrar al destino deberá volver a utilizarse impactando nuevamente la eficiencia de la red. De manera general podemos encontrar las siguientes ventajas y desventajas de este mecanismo[11]:

a) Ventajas

1.- Simplicidad de operación

2.- Puede ser más eficiente que otros protocolos para el caso de bajas tasas de transferencia, en cuanto a la cantidad de mensajes adicionales (*overhead*). Este escenario puede ocurrir, por ejemplo, cuando los nodos transmiten paquetes de datos pequeños de manera poco frecuente y ocurren cambios en la topología entre transmisiones consecutivas de paquetes.

3.- Alta confiabilidad de entrega de datos (potencialmente dependiendo de las características de la red)

b) Desventajas

1.- Potencialmente alta generación de paquetes adicionales (*overhead*). Los paquetes de datos pueden entregarse a varios nodos los cuales no requieren su recepción

2.- Tiempo de latencia alto.

3.4.2 Protocolos de encaminamiento que utilizan inundación de paquetes

Muchos de los protocolos propuestos realizan inundación de paquetes de control en lugar de inundación de paquetes de datos. Estos paquetes de control tienen como objetivo el descubrimiento de rutas dentro de la red, mismas que posteriormente se utilizarán para la transmisión de la información. La creación de paquetes adicionales de control se ve compensada por la cantidad de paquetes de datos transmitidos entre inundaciones de paquetes de control consecutivas. A continuación se analizan brevemente algunos protocolos de encaminamiento que trabajan bajo este esquema.

3.4.2.1 Encaminamiento ad hoc por Vector Distancia sobre Demanda (AODV)

AODV[13] Como su nombre lo indica, este protocolo está diseñado para ser usado por nodos móviles en configuración sin infraestructura (*ad hoc*). De manera general, ofrece una rápida adaptación a condiciones dinámicas del enlace, bajo nivel de procesamiento y baja utilización de memoria, baja utilización de la red y determina rutas del tipo "*unicast*" a los destinatarios dentro de la red. Utiliza números de secuencia al destino para evitar ciclos en todo momento(incluso en el caso de entrega anómala de mensajes de control para ruteo), evitando problemas (tales como "conteo al infinito") asociados a protocolos clásicos basados en vector distancia.

3.4.2.1.1 Generalidades

El algoritmo de encaminamiento *ad hoc* por Vector Distancia sobre Demanda (AODV) permite un encaminamiento dinámico de saltos múltiples, capaz de iniciarse por sí mismo entre nodos móviles participantes que desean establecer y mantener una red del tipo *ad-hoc*. AODV permite que los nodos móviles obtengan rutas de manera rápida para nuevos destinatarios y no requiere que los nodos almacenen rutas para destinos que no participen en una comunicación activa. Este protocolo también permite que los nodos respondan a rompimientos y cambios en la topología de red en un tiempo razonable

La operación de AODV es libre de ciclos y mediante la supresión del problema de "cuenta al infinito" de Bellman-Ford, ofrece una rápida convergencia cuando sucede un cambio en la topología (típicamente, cuando un nodo se mueve en la red). Cuando un enlace se pierde, AODV se encarga de notificar a lo nodos, para que éstos sean capaces de invalidar las rutas usando el enlace perdido.

Una característica especial de AODV es el uso de un número de secuencia al destino para cada información de ruta. Este número es creado por el destinatario para que se incluya junto con cualquier información de ruta que éste envíe a los nodos iniciadores. El uso de números de secuencia asegura que no se formen ciclos y es sencillo de programar. En el caso de que se presenten dos rutas a un solo destino, se requiere que el nodo iniciador elija aquella con el número de secuencia más alto.

Los tipos de mensajes definidos por AODV son petición de ruta (Route Request - RREQ), respuesta de ruta (Route Reply - RREP) y error en ruta (RERR). Este tipo de mensajes son recibidos vía UDP y se les aplica el proceso normal de encabezado IP. De tal forma, por ejemplo, el nodo que solicita la comunicación utilizará su dirección IP como la dirección IP origen en el mensaje. Para mensajes de

broadcast, se utilizará la dirección IP definida (255.255.255.255). Lo anterior significa que tales mensajes no se envían a "ciegas".

Sin embargo, la operación de este protocolo requiere que ciertos mensajes sean diseminados de manera amplia (por ejemplo un RREQ), o tal vez a toda la red *ad-hoc*. El rango de diseminación de tales RREQs es indicado por el TTL (Time To Live – tiempo de vida) en el encabezado IP.

En el caso en que los usuarios que se desean comunicar cuenten con rutas válidas entre ellos, AODV no desempeña ninguna función. Cuando se necesita establecer una ruta a un nuevo destino, el nodo transmite un RREQ para encontrar un "camino" al destino. Una ruta puede ser determinada cuando un RREQ alcanza al destino mismo, o a algún nodo intermedio con una ruta lo suficientemente fresca al destino. Una ruta "suficientemente fresca" es una información de ruta para el destino cuyo número de secuencia asociado es al menos tan grande como el contenido en el RREQ.

La ruta se habilita mediante la transmisión de un paquete unicast RREP de regreso al nodo origen del RREQ. Cada nodo que recibió el RREQ almacena una ruta de regreso al origen, de tal forma que el RREP puede ser transmitido de forma unicast desde el destino, o desde un nodo intermedio que sea capaz de satisfacer la petición.

Los nodos monitorean el estado del enlace de los saltos siguientes en las rutas activas. Cuando se detecta un enlace caído en una ruta activa, se utiliza un mensaje de RERR para notificar a los otros nodos que ha ocurrido la pérdida de ese enlace. El mensaje de RERR indica aquellos destinos (posiblemente subredes) las cuales ya no son alcanzables mediante el enlace roto.

De esta forma, AODV es un protocolo que tiene que ver con el manejo de una tabla de ruteo. La información de esta tabla, debe de mantenerse incluso para el caso de rutas con tiempo de vida corto, las cuales se crean de manera temporal para almacenar caminos inversos hacia los nodos que originaron los RREQs.

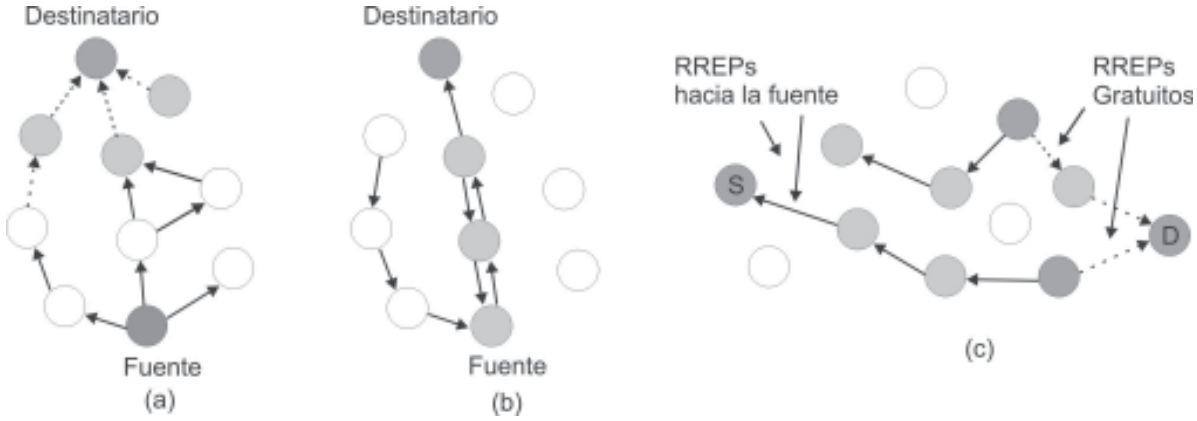

Figura 3.4.- (a) Propagación de la petición de ruta (RREQ) (b) Propagaciòn de la confirmación de ruta (RREP) y ruta formada (c) Respuesta gratuita de ruta

3.4.2.2 Algoritmo de Ruteo Temporalmente Ordenado (TORA)

El Algoritmo de encaminamiento Temporalmente Ordenado (TORA)[14] es un protocolo de adaptivo para redes multi-salto que posee los siguientes atributos:

1.- Ejecución distribuida
2.- Encaminamiento libre de lazo

3.- Encaminamiento multi-trayectoria
4.- Establecimiento y mantenimiento de ruta reactivo o proactivo
5.- Reducción de creación de paquetes adicionales (*overhead*) mediante la localización de reacción algorítmica a cambios topológicos.

TORA es un protocolo del tipo híbrido que es distribuido en el sentido en que los ruteadores sólo necesitan mantener la información relativa a nodos adyacentes (por ejemplo a un salto). Como en el caso de la aproximación de ruteo por vector distancia, TORA mantiene la información por destino principalmente. Sin embargo, TORA no ejecuta continuamente un proceso para la ruta más corta y por tanto la métrica usada para establecer la estructura de ruteo no representa una distancia. La naturaleza orientada a la conexión de la estructura de ruteo en este protocolo soporta una mezcla e ruteo reactivo y proactivo basado por destino. Durante la operación reactiva, las fuentes inician el establecimiento de rutas a un destino dado sobre demanda. Este modo de operación puede ser ventajoso en redes dinámicas con patrones de tráfico escasos, puesto que puede no ser necesario (no es deseable) mantener rutas entre todo par origen/destinatario todo el tiempo. Al mismo tiempo, destinatarios previamente seleccionados pueden realizar operaciones proactivas, asemejándose a aproximaciones de ruteo basadas en tablas. Esto permite a que las rutas sean mantenidas de forma proactiva a los destinos para los cuales el ruteo es consistente o requerido frecuentemente (por ejemplo servidores o puertas de enlace a la infraestructura guiada).

TORA está diseñado para minimizar la generación de paquetes adicionales (*overhead*) en la comunicación asociado con la adaptación a cambios topológicos de la red. La visión de los mensajes de control de TORA está enfocada principalmente a un muy pequeño conjunto de nodos cerca del cambio topológico. Un mecanismo secundario, el cual es independiente de la dinámica topológica de la red, es usado como una forma de optimización de la ruta. El diseño y flexibilidad de TORA permite que su operación sea altamente reactiva (por ejemplo complejidad baja en el tiempo) y conservación del ancho de banda (por ejemplo baja complejidad de la comunicación), haciéndola ideal para su uso en redes inalámbricas dinámicas.

3.4.2.2.1 Funcionamiento general

TORA ha sido diseñado para trabajar arriba de los mecanismos de capas inferiores o protocolos que proveen los siguientes servicios básicos entre nodos vecinos:

a) Detección del estado del enlace y descubrimiento de nodos vecinos
b) Confiabilidad, entrega ordenada de paquetes de control
c) Definición de enlaces, dirección de capa de red y mapeo
d) Seguridad mediante autenticación

Los eventos tales como la recepción de mensajes de control y cambios en la conectividad con los vecinos, disparan las reacciones algorítmicas de TORA. Una versión separada lógicamente de este protocolo se ejecuta para cada "destino" hacia el cual se requiere establecer una ruta. El término destinatario, al igual que en otros protocolos, se utiliza aquí para referirse a un destinatario para encaminamiento IP tradicional. Por tanto la ruta a un destinatario puede corresponder a la dirección individual de una interfaz en una máquina en específico (una ruta hacia un usuario) o un agregado de direcciones (una ruta de red).

TORA asigna direcciones a los enlaces entre nodos (*routers*) para forman una estructura de ruteo que es utilizada para transmitir los datagramas al destinatario. Un nodo asigna una dirección (de subida o bajada) al enlace con un nodo vecino basado en los valores relativos de una métrica asociada con cada *router* (nodo). La métrica mantenida por un nodo puede pensarse de manera conceptual como la "altura"

del nodo (por ejemplo, los enlaces están dirigidos desde el *router* más alto al *router* más bajo). El significado de las alturas y las asignaciones direccionales del enlace es que un nodo puede relevar datagramas de bajada. Los enlaces desde un nodo a cualquier otro en vecindad con una altura desconocida o no definida se consideran "no direccionados" y no pueden ser utilizados para relevar paquetes. Colectivamente, las alturas de los nodos (*routers*) y las asignaciones direccionales de los enlaces forman una estructura de encaminamiento libre de ciclos y de trayectorias múltiples, en la cual todas las trayectorias dirigidas llevan "hacia abajo" (*downstream*) hacia el destino, como se muestra en la figura 3.5. Note que en este ejemplo, C se encuentra más cerca al destinatario que B en términos del número de saltos, pero la métrica de la altura de C es más grande que las de B.

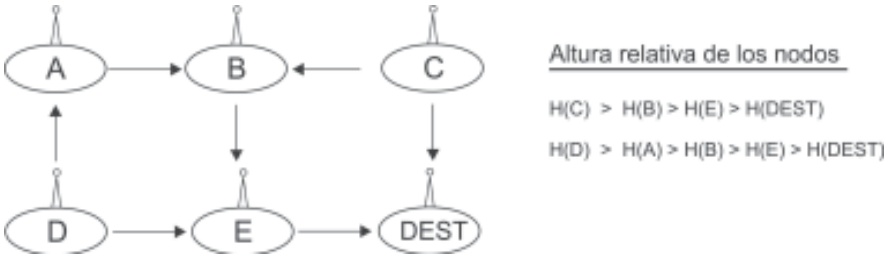

Figura 3.5.- Representación conceptual de la gráfica acíclica dirigida definida por la altura relativa de los nodos (routers) de la red

TORA puede separarse en cuatro funciones básicas: creación de rutas, mantenimiento de rutas, eliminación de rutas y optimización de rutas. La parte de creación de rutas corresponde a la selección de alturas para formar una secuencia dirigida de enlaces que llevan al destinatario en una red no dirigida o porción de la red. El mantenimiento de rutas se refiere a la adaptación de la estructura de encaminamiento en respuesta a cambios topológicos. Por ejemplo, siguiendo la pérdida de algún nodo en el último enlace de bajada, algunas trayectorias dirigidas pueden ya no llevar al destinatario. Este evento dispara una secuencia de revocaciones de enlaces (causadas por la reselección de alturas de nodos), las cuales reorientan la estructura de encaminamiento de tal forma que todas las rutas dirigidas lleven de nuevo al destinatario. En casos donde la red se particiona, los enlaces en la porción de la red que se ha particionado del destino, debe de ser marcada como no dirigida, para borrar aquellas rutas inválidas. Durante este proceso de eliminación de rutas, los nodos ajustan sus alturas a cero y sus enlaces adyacentes se vuelven no dirigidos.

Finalmente, TORA incluye un mecanismo secundario para la optimización de las rutas, en la cual los nodos (*routers*) re-seleccionan sus alturas con el objeto de mejorar la estructura de ruteo. TORA logra estas 4 funciones mediante el uso de cuatro paquetes de control: preguntar (QRY), actualizar (UPD), borrar (CLR) y optimización (OPT).

3.4.2.3 Protocolo de ruteo de fuente dinámico para redes móviles ad hoc (DSR)

DSR[15] es un protocolo de ruteo simple y eficiente diseñado para su uso específico en redes ad hoc inalámbricas de salto múltiple compuesta por nodos móviles. Este protocolo permite que la red sea completamente autoorganizada y autoconfigurable, sin la necesidad de la presencia de una infraestructura o administración y, debido a que nos basamos en este protocolo para el desarrollo de este trabajo, se explica a mayor detalle el funcionamiento de DSR.

El protocolo se compone de dos mecanismos de "descubrimiento de ruta" (*Route Discovery*) y "mantenimiento de ruta" (*Route Maintenance*), los cuales trabajan juntos para permitir a los nodos el descubrir y mantener rutas de fuente hacia destinatarios arbitrarios en la red *ad hoc*. La función general de ambos mecanismos es el siguiente:

a) Descubrimiento de ruta (*Route Discovery*): Es el mecanismo mediante el cual un nodo S que desea enviar un paquete a un nodo destino D, obtiene una ruta hacia D. Este mecanismo se utiliza sólo cuando S trata de enviar un paquete a D y todavía no se conoce una ruta a D.

b) Mantenimiento de la ruta (*Route Maintenance*): Es el mecanismo mediante el cual el nodo S es capaz de detectar, durante el uso de una ruta hacia D si la topología de red ha cambiado de tal forma que la ruta ya no sea válida. Cuando el mantenimiento de la ruta indica que una ruta se ha interrumpido, S puede intentar usar cualquier ruta alternativa hacia D, o puede invocar al descubrimiento de la ruta para encontrar un nuevo camino para los paquetes subsecuentes para D. El mantenimiento de la ruta se utiliza sólo cuando S se encuentra mandando paquetes a D.

El uso de encaminamiento de fuente permite que el ruteo de paquetes sea libre de lazo, evitando la necesidad de información de encaminamiento actualizada al momento en los nodos intermediarios a través de los cuales se transmiten los paquetes y permite a los nodos relevarlos, almacenando la ruta para uso futuro.

Todos los aspectos del protocolo operan enteramente sobre demanda, permitiendo que el sobre-encabezado del paquete se ajuste automáticamente a lo necesario para reaccionar a cambios en las rutas que están actualmente en uso. Esto quiere decir que DSR no requiere transmitir periódicamente paquetes de ningún tipo a ningún nivel dentro de la red, tales como aviso periódico de encaminamiento, verificación del estado del enlace o detección de paquetes en vecindad. Este comportamiento enteramente sobre demanda y la falta de actividad periódica permite que el número de sobre-encabezado de paquetes causado por DSR se reduzca a cero, en el caso en que todos los nodos se encuentran casi estacionarios con respecto a los demás y todas las rutas para alcanzar todos los nodos se han descubierto.

Conforme los nodos en la red se mueven entrando o dejando la misma, junto con el hecho de que las condiciones cambiantes de transmisión por el canal inalámbrico (cambio de fuentes e interferencia), todo el encaminamiento se determina y se mantiene gracias a DSR. Desde el punto de vista de la topología, el número o secuencia de saltos intermedios necesarios para alcanzar cualquier destinatario puede cambiar en cualquier momento.

Cada paquete de información enviado, porta en su encabezado la lista completa ordenada de los nodos a través de los cuales pasará el datagrama, de tal forma que para aquellos nodos que ayudan a su relevo, así como aquellos que "escuchan" cualquiera de esos paquetes, pueden fácilmente almacenar esta información de ruta para su uso futuro. Cuando se trabaja con DSR se deben de tener las siguientes consideraciones:

1.- Todos los nodos que deseen comunicarse con otros nodos dentro de la red *ad-hoc*, están destinados a participar completamente en los protocolos de la red. En particular, cada nodo participante deberá también estar destinado a relevar paquetes a otros nodos en la red

2.- El diámetro de una red *ad-hoc* es el número mínimo de saltos necesarios para que un paquete alcance al destinatario localizado en un extremo (u orilla) de la red *ad-hoc*, desde el nodo origen localizado en el otro extremo. Se asume que este diámetro será pequeño (quizás 5 o 10 saltos), pero frecuentemente mayor que

3.- Los nodos dentro de la red *ad-hoc* pueden moverse en cualquier tiempo sin aviso previo, y pueden moverse inclusive de manera contínua, pero se asume que la velocidad con que se mueven es moderada respecto a la latencia de la transmisión del paquete y el rango de transmisión inalámbrica especificado por el hardware de red en uso.

4.- El canal de comunicación inalámbrica entre cualquier par de nodos, puede por momentos trabajar de manera distinta en ambas direcciones, debido por ejemplo a patrones de propagación o interferencia de fuentes adyacentes alrededor de los nodos. Esto es, las comunicaciones inalámbricas entre cada par de nodos podrán en muchos casos ser capaces de operar de manera bidireccional; pero en ocasiones el enlace entre los nodos trabajará de manera unidireccional, permitiendo a un nodo enviar de manera exitosa paquetes al destinatario, aún cuando en el otro sentido no sea posible la comunicación.

3.4.2.3.1 Generalidades del protocolo

Cuando algún nodo S origina un nuevo paquete destinado a un nodo destino D, coloca en el encabezado del paquete una ruta fuente proporcionando la secuencia de saltos que el paquete debe de seguir en su camino a D. Normalmente S obtendrá una ruta adecuada buscando en su caché de rutas que ha adquirido previamente; en caso de que no se encuentre una ruta hacia D, se debe de inicializar el mecanismo de "descubrimiento de ruta" para encontrar un camino dinámicamente hacia el nodo destino. En este caso denominamos a S "iniciador" y a D el "objetivo" del descubrimiento de ruta.

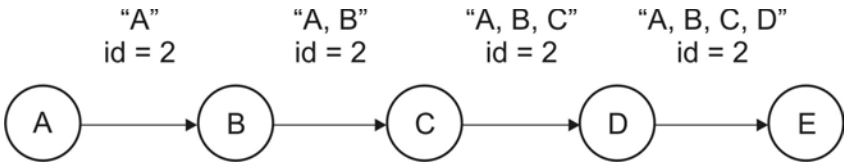

Figura 3.6.- Descubrimiento de ruta básico

Tomando como ejemplo el caso de la figura 3.6, para iniciar el proceso de descubrimiento de ruta, el nodo A transmite una "petición de ruta" (*RREQ*) como un paquete sencillo de *broadcast* local, el cual es recibido por (aproximadamente) todos los nodos que se encuentren en ese momento dentro del rango de transmisión inalámbrica de A, el cual incluye al nodo B para este caso. Cada mensaje de petición de ruta identifica al iniciador y objetivo del proceso de descubrimiento de ruta. La petición de ruta incluye también una lista de la dirección de cada nodo intermedio a través de los cuales ha sido relevado esta copia en particular del mensaje.

El proceso a seguir cuando los nodos en la periferia reciben la petición, depende de dos casos:
1.- Es el nodo destino: Regresa un mensaje de confirmación de ruta (*RREP*) al iniciador, proporcionando una copia de la ruta acumulada almacenada desde la petición. Cuando el iniciador recibe la confirmación, almacena la ruta en su tabla de ruteo para utilizarla al mandar los paquetes subsecuentes al destino.

2.- Es un nodo intermedio: El nodo agrega su propia dirección al registro en el mensaje de petición de ruta y lo propaga transmitiéndolo como un paquete de *broadcast* local (con la misma identificación de la ruta). En el ejemplo, B retransmite la petición, resultando en una copia de la petición siendo recibida por el nodo E.

Por otro lado, para mandar la confirmación de ruta hacia el iniciador, típicamente el nodo destino (E en el ejemplo) revisará su propia tabla de ruteo para encontrar una ruta de regreso a A y de encontrarla será utilizada para enviar el paquete.

En el protocolo MAC 802.11 se requiere un enlace bidireccional, por lo que el nodo E puede simplemente enviar la confirmación de ruta siguiendo el camino de regreso establecido por la petición de

ruta. De lo contrario, si están presentes enlaces unidireccionales, sería necesario que E comenzara un segundo descubrimiento de ruta para poder entregar la confirmación hacia A.

De manera adicional cuando se inicia un descubrimiento de ruta, el nodo origen guarda una copia del paquete original en un buffer local denominado "Buffer de envío" (*send buffer*). El buffer de envío contiene una copia de cada paquete que no se pudo transmitir por este nodo debido a que no se tiene en el momento una ruta de fuente para el destinatario. Cada paquete en el buffer de envío lleva consigo el tiempo en que fue colocado en el buffer y se descarta después de un tiempo de espera. Mientras el paquete permanece en el buffer de envío, el nodo debe ocasionalmente iniciar un nuevo descubrimiento de ruta para la dirección destino del paquete. Sin embargo, el nodo debe limitar la tasa a la cual debe de iniciarse una nueva búsqueda de ruta, puesto que es posible que el nodo destino no se encuentre disponible al momento.

Para evitar que se generen descubrimientos de ruta innecesarios, los nodos deben de usar un algoritmo de *exponential backoff* para limitar la tasa a la cual se inicia un nuevo descubrimiento de ruta.

Cuando se origina o se releva un paquete usando una ruta de fuente, cada nodoque transmite el paquete es responsable de confirmar que el paquete ha sido recibido por el siguiente salto a lo largo de la ruta. El paquete debe de ser retransmitido (hasta un número máximo de intentos) hasta que se reciba una confirmación de que se recibió.

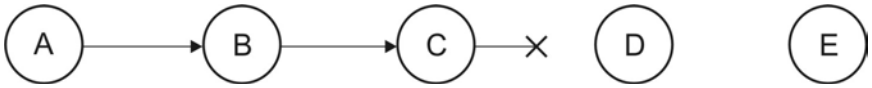

Figura 3.7.- Mantenimiento de ruta básico

En el caso de la figura 3.7, el nodo A es responsable de la recepción del paquete en B, mientras que el nodo B es responsable de que se reciba en C y así sucesivamente. En el caso en que no se reciba ninguna confirmación después de que el paquete haya sido retransmitido el número máximo de intentos en algún salto, este nodo debe de regresar un mensaje "Error de Ruta" (*Route Error*) al emisor original del paquete, identificando el enlace sobre el cual el paquete no pudo ser relevado. En el ejemplo de la figura 3.7, si C no puede entregar el paquete al siguiente salto en D, C debe regresar un mensaje de error de ruta para A, estableciendo que el enlace entre C y D se encuentra actualmente "roto". En el caso de que de parte de un protocolo de mayor nivel inicie una retransmisión (por ejemplo TCP), si A tiene en su caché de rutas otro camino hacia E (por ejemplo a partir de respuesta de ruta adicionales originados en su descubrimiento de ruta, o bien de la información de ruta escuchada de otros paquetes), puede enviar el paquete usando una nueva ruta inmediatamente. De otra forma, A debe de iniciar un nuevo descubrimiento de ruta. DSR presenta las siguientes características adicionales del proceso de descubrimiento de ruta:

1.- Almacenamiento de información de ruta "escuchada"

Un nodo retransmitiendo o "escuchando" cualquier paquete, puede agregar la información de ruta de ese paquete a su propio caché de rutas. Una limitación sin embargo, es la posible presencia de rutas unidireccionales en la red.

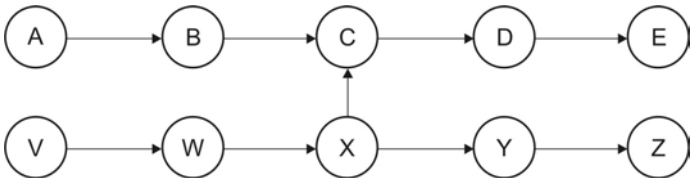

Figura 3.8.- Características adicionales del descubrimiento de ruta

En la figura 3.8, el nodo C "escucha " al paquete y la información de ruta hacia Y

2.- Responder a peticiones de ruta usando rutas almacenadas

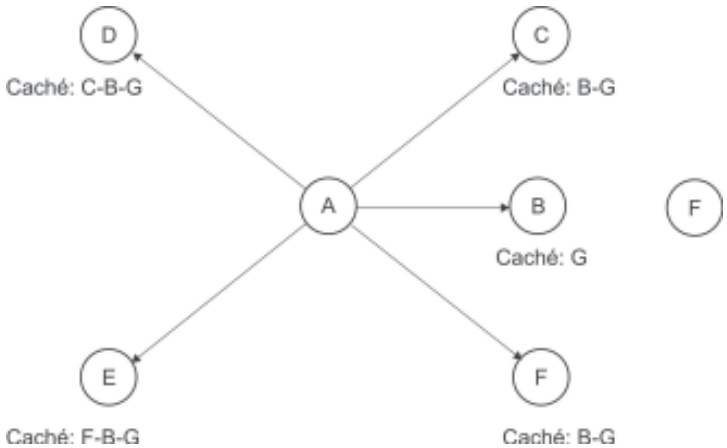

Figura 3.9.- Almacenamiento de rutas

Todo nodo que recibe una petición de ruta para la cual no es el objetivo, busca en su caché de rutas una ruta al objetivo de la petición. Si la encuentra, el nodo intermedio generalmente regresa una respuesta de ruta al iniciador en lugar de retransmitir el mensaje de petición de ruta. En la respuesta de ruta, coloca el registro de ruta para enlistar la secuencia de saltos sobre los cuales la petición de ruta fue enviada, concatenándola con su propia idea de ruta desde el nodo en si hasta el objetivo.

3.- Prevención de "tormentas" de confirmación de rutas (RREP)

La habilidad de los nodos para responder a una petición de ruta a partir de la información en su caché de rutas, puede resultar en una "tormenta" de respuesta de rutas en algunos casos. En particular, si un nodo transmite una petición de ruta para un nodo objetivo y nodos intermedios reciben la petición, si éstos tienen una dirección almacenada hacia el destinatario, tratarán de enviar una respuesta de ruta al iniciador, desperdiciando ancho de banda y posiblemente incrementando el número de colisiones en el área. Además, puede darse el caso en que las diferentes respuestas de ruta indiquen caminos de diferentes longitudes.

Si la interfaz de red se encuentra en modo promiscuo, el nodo deberá de retardar su propia respuesta de ruta por un periodo corto, mientras escucha si el nodo iniciador comienza a utilizar una ruta más corta. Esto es, ese nodo deberá retardar su envío de respuesta por un período aleatorio:

$$d = H * (h - 1 + r)$$

donde r es un número aleatorio entre 0 y 1 y H es una constante de retardo pequeña (de al menos dos veces el retardo máximo de propagación en el enlace inalámbrico) a ser introducido por salto. Este retardo vuelve aleatorio efectivamente el tiempo en el cual cada nodo envía su respuesta de ruta, con todos los nodos enviando respuestas proporcionando rutas de longitudes menores que "h" (donde "h" es la longitud de ruta en número de saltos), enviadas antes de este nodo. De la misma manera, el resto de los nodos con rutas de mayor longitud que h enviando sus respuestas después de este nodo.

Si un paquete de información recibido por el nodo durante el periodo de retardo usa una ruta de longitud menor o igual a "h", este nodo puede suponer que el iniciador del descubrimiento de ruta ha recibido una respuesta de ruta proporcionando información de una ruta igual o mejor. En este caso, este

nodo debería cancelar su temporizador de retardo para así no enviar su respuesta para la petición ruta del iniciador.

4.- Límite de número de saltos para la petición de ruta

Cada mensaje de petición de ruta contiene un "límite de saltos" que puede ser usado para limitar el número de nodos intermedios permitidos para relevar la copia de la petición. Este límite de saltos se implementa usando el campo de "tiempo de vida" (*time-to-live*) en el encabezado IP del paquete que porta la petición de ruta. Conforme la petición se transmite, este límite se decrementa y el paquete se descarta si el límite alcanza cero antes de encontrar el objetivo.

Otro uso posible del límite de saltos en una petición de ruta es el implementar una búsqueda del objetivo mediante una "expansión de anillo". Por ejemplo un nodo puede enviar una petición de ruta sin propagación; si no se recibe una respuesta de ruta , el nodo puede iniciar otra petición con un límite de 2 saltos y así sucesivamente.

Sin embargo, este tipo de búsqueda por "expansión de ruta" puede tener el efecto de incrementar la latencia promedio de un descubrimiento de ruta, puesto que se podrán necesitar intentos de descubrimiento múltiples antes de descubrir una ruta al nodo objetivo. En DSR, existen otras características adicionales de mantenimiento de ruta:

1.- Salvamento de paquete

Después de enviar un mensaje de error de ruta (*RERR*) como parte del mantenimiento de ruta, un nodo puede intentar "salvar" el paquete de datos que causó el error de ruta en lugar de descartarlo. Para intentar salvar un paquete, el nodo que envía un mensaje de error de ruta, busca en su propio caché de ruta un camino desde él al nodo destino del paquete que causó el error. Si se encuentra la ruta, el nodo puede salvar el paquete antes de regresar el error de ruta reemplazando la ruta fuente original en el paquete con la ruta que obtuvo de su registro de rutas. De esta manera, se puede evitar que un paquete sea desechado si uno de los nodos en la ruta conoce un camino alterno para el destinatario.

2.- Recorte de ruta automático

Las rutas pueden ser automáticamente recortadas si uno o más de los nodos intermediarios se vuelven innecesarios.

3.- Dispersión incrementada de mensajes de error de ruta

Cuando un nodo fuente recibe un mensaje de error de ruta para el paquete que originó. Este nodo fuente propaga este mensaje de error a sus vecinos agregándolo en su próximo descubrimiento de ruta. En este sentido, la información añeja en los registros de los nodos alrededor del nodo origen no generarán respuestas de ruta que contengan el mismo enlace inválido para el cual este nodo fuente recibió el mensaje de error de ruta.

3.4.2.3.2 Formatos de paquetes en DSR

En los campos IP se contiene:

Dirección origen Debe establecerse con la dirección IP del nodo que origina este paquete. Los nodos intermedios que vuelven a propagar el RREQ no deben de cambiar este campo.

Dirección destino Debe de ajustarse a la dirección específica de broadcast (255.255.255.255).

Límite de Saltos (TTL) Puede variar de 1 a 255, por ejemplo para implementar RREQ que no se propaguen y búsquedas de anillo expandiéndose.

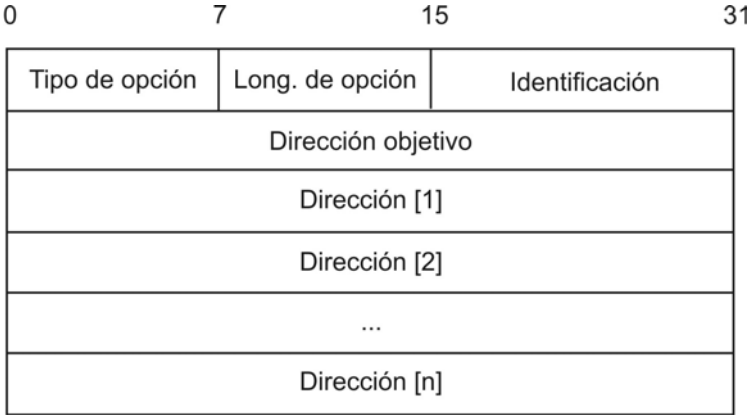

Figura 3.10.- Formato de paquete de petición de ruta (RREQ)

En los campos de la petición de ruta se tiene, como se muestra en la figura 3.10:

Tipo de Opción Los 3 bits mas significativos de este campo son iguales a 011 (valor que identifica al paquete del tipo DSR), significando que un nodo que no entienda esta opción debe de descartar el paquete

Longitud de opción Entero sin signo de 8 bits. Longitud de la opción en octetos, excluyendo los campos de tipo de opción y de longitud de opción

Identificación Un valor único generado por el iniciador (transmisor original) del RREQ. Este valor permite al nodo receptor determinar si ha recibido este paquete con anterioridad.

Dirección objetivo La dirección del nodo que es el objetivo del RREQ

Dirección [1..n] Es la dirección de los nodos participantes en la ruta. (n = (Longitud de opción -1)/4)

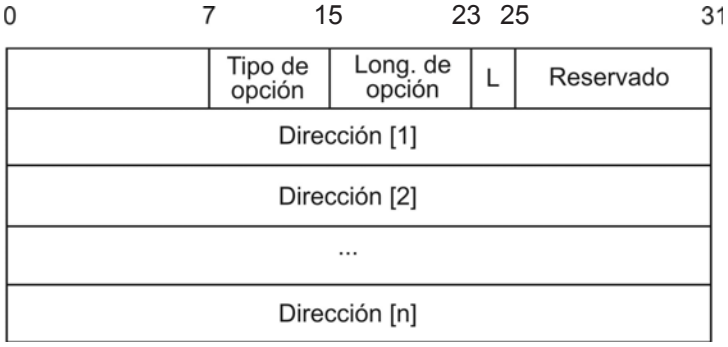

Figura 3.11.- Formato de paquete de respuesta de ruta (RREP)

Como de muestra en la figura 3.11, los campos que componen el paquete de RREP son:

Tipo de Opción	Los 3 bits mas significativos de este campo son iguales a 011, significando que un nodo que no opere bajo DSR debe de descartar el paquete
Longitud de opción	Entero sin signo de 8 bits. Longitud de la opción en octetos, excluyendo los campos de tipo de opción y de longitud de opción
Ultimo Salto Externo (L)	Habilitado para indicar que el último nodo indicado por el RREP se encuentra en realidad en una red externa a la red DSR; la secuencia exacta de saltos que llevan a ella fuera de la red DSR no están representados en el RREP.
Dirección [1..n]	Es la dirección de los nodos participantes en la ruta. El número de direcciones presentes en este campo se indica por el campo de longitud de opción, en la opción (n = (Longitud de opción -1)/4)
Reservado	Enviado como 0; ignorado en recepción

En la figura 3.12 se muestran los campos que contiene el paquete de RRER en DSR:

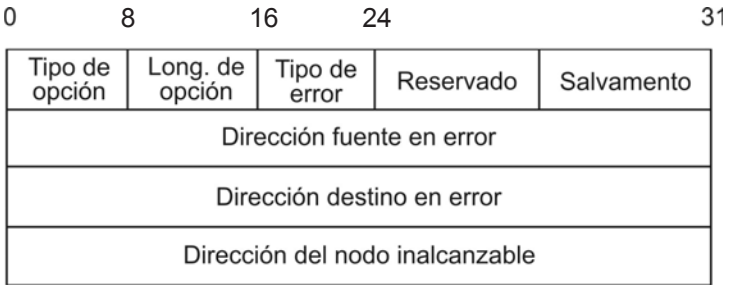

Figura 3.12.- Formato de paquete de error en ruta (RERR)

Tipo de Opción	Los 3 bits mas significativos de este campo son iguales a 011, significando que un nodo que no opere bajo DSR debe de descartar el paquete
Longitud de opción	Entero sin signo de 8 bits. Longitud de la opción en octetos, excluyendo los campos de tipo de opción y de longitud de opción
Tipo de Error	El tipo de error encontrado. Actualmente, el siguiente tipo se define: NODE_UNREACHABLE (nodo inalcanzable)
Reservado	Enviado como 0; ignorado en recepción
Salvamento	Entero de 4 bits sin signo. Copiado del campo de salvamento en el encabezado de encaminamiento DSR del paquete activando el error en ruta, incrementado por el nodo que regresa el mensaje de error
Dirección Fuente en Error	La dirección del nodo que originó el RERR (por ejemplo el nodo que intentó transmitir el paquete y descubrió la falla en el enlace)
Dirección Destino en Error	La dirección del nodo al cual el RERR debe de ser entregado
Dirección Del nodo inalcanzable	La dirección del nodo que se encontró inalcanzable (siguiente salto en el enlace que dejó de funcionar y por tanto ya no releva los paquetes)

3.5 Inundación (*flooding*) eficiente y localización de usuarios en una red ad hoc

Como se mencionó anteriormente, al mecanismo básico que se utiliza para propagar los mensajes de control se le denomina inundación (*flooding*). Con este mecanismo, un nodo transmite un mensaje a todos sus vecinos que a su vez, transmiten el mismo paquete a sus propios vecinos y así sucesivamente hasta que el mensaje se propague en toda la red. A este mecanismo también se le denomina "inundación ciega" (blind flooding)[16].

De todo lo anterior, podemos deducir que el desempeño de la inundación ciega se encuentra estrechamente relacionada con el número promedio de vecinos en una red CSMA/CA. Conforme este número se incrementa, la inundación ciega experimenta el incremento de paquetes redundantes y superfluos, de la probabilidad de colisión y por ende de la congestión en la red. Consecuentemente, el desempeño de la inundación ciega se deteriora especialmente en redes grandes y densas.

Cuando se cuenta con información sobre la topología de la red y de los nodos en vecindad, sólo un subconjunto de vecinos (por ejemplo un conjunto de nodos dominantes), deberá participar en la inundación para asegurar una búsqueda exitosa. A tal mecanismo se le denomina "inundación eficiente" (*efficient flooding*).

A continuación se presenta una breve explicación y comparación, de las diferentes propuestas de protocolos con esquemas de inundación eficiente.

3.5.1 Protocolos basados en un modelo heurístico

A la fecha se han propuesto diversos esquemas heurísticos para reducir el número de retransmisiones[17][18]. En estas propuestas cuando se recibe un paquete de *flooding*, el nodo receptor decide enviar el paquete a sus vecinos o no de acuerdo con:

1) Un esquema probabilístico donde este nodo retransmite el paquete en un tiempo aleatorio
2) Un esquema basado en la operación de un contador donde el nodo en particular retransmite si el número de paquetes duplicados recibidos es menos que un umbral (por ejemplo 2)
3) Un esquema basado en la distancia relativa al host para tomar la decisión
4) Esquema basado en la localización, que utiliza información adquirida con anterioridad de los vecinos

El desempeño de un protocolo basado en un modelo heurístico se encuentra fuertemente relacionado a una serie de parámetros bien escogidos y umbrales basados en las condiciones del ambiente de red.

3.5.2 Protocolos basados en la topología

Otro tipo de propuestas para mejorar la eficiencia del *flooding* se basan en la explotación de información topológica. Con la movilidad de los nodos y la ausencia de infraestructura bien establecida en la red *ad hoc*, la mayoría de los nodos intercambian constantemente paquetes HELLO, utilizados para compartir información topológica y se pueden clasificar de la siguiente manera:

3.5.2.1 Protocolos basados en la topología de los vecinos

Algunos protocolos propuestos proponen esquemas de cobertura de vecinos basados en la información recopilada a un salto del nodo de interés [19][20]. En estos esquemas, un nodo retransmite un paquete si el conjunto de nodos vecinos que haya retransmitido el paquete no cubre el total del conjunto de vecinos de este nodo. En otras palabras, un nodo no retransmite cuando el paquete se entrega a todos sus vecinos mediante retransmisiones de otros nodos.

El esquema de relevo multipunto (MPR) extiende el rango de información de vecinos a dos saltos a partir del nodo de interés. En este tipo de esquemas, un nodo intercambia periódicamente la lista de nodos adyacentes con sus vecinos y selecciona el subconjunto mínimo de nodos transmisores que cubren a todos los demás dentro de un rango de dos saltos, de tal manera que sólo aquellos nodos que han sido seleccionados serán aquellos que retransmitan el paquete de *flooding*.

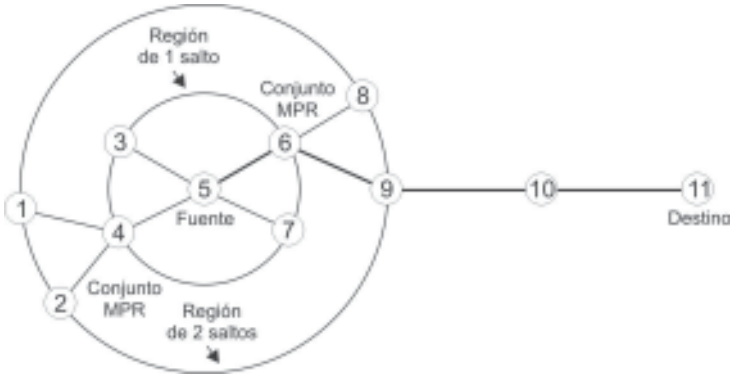

Figura 3.13.- Descubrimiento de ruta con MPR

Algunos otros esquemas también utilizan modelos heurísticos, donde se escoge un conjunto dominante de nodos basados en información topológica. Entre esas propuestas existe un algoritmo denominado "Span"[20] para seleccionar un conjunto de nodos coordinados que cubren todos los nodos a dos saltos. "Span", sin embargo es diferente a MPR principalmente en que un nodo se declara agresivamente como "nodo coordinador" (dominante) si el nodo detecta un número insuficiente de nodos vecinos dentro del conjunto específico.

3.5.2.2 Protocolos basados en árbol fuente (source-tree)

El esquema de *flooding* basado en árbol fuente (*source-tree*) utilizado en ALRP (protocolo de encaminamiento estado del enlace adaptivo – *adaptive link-state routing protocol*)[21] y TBRPF (Difusión topológica basada en transmisión por ruta inversa – *topology broadcast based on reverse-path forwarding*)[22], son otros mecanismos que utilizan información topológica.

Con el árbol fuente, un nodo retransmite el paquete de *flooding* en el caso en que no sea una "hoja" en el árbol cuya raíz es el nodo fuente. Para el caso de un protocolo con difusión topológica basada en transmisión por ruta inversa, cada nodo retransmite si reconocen que no son una "hoja" en el árbol formado por las rutas de saltos mínimos hacia el nodo fuente.

Para construir y administrar un árbol fuente, cada nodo actualiza su estado en el árbol mediante la recepción de un paquete proveniente de nodos vecinos y periódicamente desempeñan *flooding* ciego iniciado por el nodo fuente.

3.5.2.3 Protocolos basados en grupos (*clustering*)

Agrupar nodos en grupos es otro método para seleccionar nodos transmisores. A un nodo representante de cada grupo (*cluster*) se le denomina "cabeza de grupo" (*cluster head*) y a un nodo que

pertenece a más de un clúster se le denomina *gateway*. Al resto de los nodos se les denomina nodos ordinarios, como lo muestra la figura 3.14. El clúster se encuentra definido por el área de transmisión del nodo principal o cabeza de clúster, como se muestra en la figura 3.14

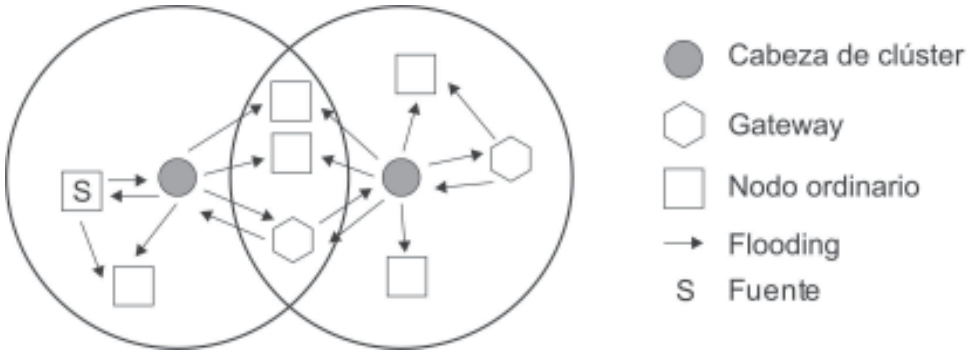

Figura 3.14.- Un ejemplo del flooding eficiente con la formación de grupos. Sólo las cabezas de clúster y gateways retransmiten y el resto de los nodos ordinarios dejan de transmitir.

Existen dos propuestas para construir una plataforma de clústeres: formación activa y pasiva de grupos. En la formación activa de clústeres, los nodos cooperan para elegir a los nodos cabeza mediante el intercambio periódico de información, formándose los grupos aunque no exista información a transmitir. Para este fin se utilizan algoritmos como son LID (*lowest ID*) y HD (*highest degree*).

Por su parte para la formación pasiva, se suspende el algoritmo hasta que empiece la transmisión de datos. Este mecanismo utiliza al tráfico saliente para propagar la "información relacionada con el grupo" (por ejemplo el estado del nodo en un grupo, la dirección IP del nodo) y colecta la información de los nodos vecinos a través de la recepción "promiscua" de paquetes. Por tanto, elimina la latencia de inicialización y el exceso de la transmisión de los paquetes de control (*overhead*) originados por la formación activa de clústeres.

3.5.3 Descubrimiento de ruta eficiente utilizando registros de encuentros y el algoritmo FRESH

Una de las propuestas más interesantes relativas a descubrimiento de rutas eficientes para localizar a un usuario en una red ad hoc, es la presentada por investigadores de la Escuela de Ciencias Computacionales y de las Comunicaciones de Laussane, Suiza[24]. En ese artículo se propone el algoritmo FRESH (FResher Encounter SearcH) en cuya operación, los nodos mantienen un registro de su más reciente encuentro con todos los demás nodos. En lugar de buscar al destino, el nodo origen busca a cualquier nodo intermedio que encontró al destino más recientemente que el nodo fuente. El nodo intermedio entonces busca por el siguiente nodo que tiene conocimiento del destino y el proceso sigue hasta que se encuentre al nodo final. Por tanto, FRESH reemplaza la búsqueda amplia en la red con una sucesión de búsquedas más pequeñas, resultando en un descubrimiento de ruta con menor costo, además de que las rutas se encuentran libres de ciclos. La propuesta se basa en que la historia de los últimos encuentros entre los nodos contiene información valiosa, aunque poco confiable acerca de la topología actual de la red (dependiendo del tiempo transcurrido).

Los encuentros entre dos nodos ocurren cuando ambos son vecinos a un salto. Puesto que la vecindad a un salto es dependiente de la capa de enlace, la condición exacta para un encuentro a ocurrir variará dependiendo en la tecnología de red en uso. Por ejemplo, el rango de conectividad para 802.11b puede exceder los 250 metros, mientras que para Bluetooth es de algunos metros solamente. La antigüedad

del encuentro (encounter age) de dos nodos N y M, es el tiempo que ha transcurrido desde el más reciente encuentro de N y M. Los encuentros pueden ser detectados al "escuchar" cualquier paquete de datos (ya sean regulares o paquetes "hello" enviados intencionalmente) enviado por nodos vecinos, o pueden ser detectados en la capa de enlace como en el caso de Bluetooth. Cabe mencionar que en este artículo los autores no asumen la utilización de una capa de enlace específica, puesto que lo que se propone únicamente es el algoritmo de descubrimiento de ruta, independientemente de la tecnología de red a utilizar.

Para explicar el funcionamiento de FRESH, consideremos a un nodo "S" que establece una ruta al destino "D". Denotamos como $T_{LE}(I,D)$ a la época del encuentro más reciente entre los nodos I y D, con la convención de que $T_{LE}(I,D) = \infty$ si los nodos i y d nunca se han encontrado y $T_{LE}(I,D) = 0$ si I = D.

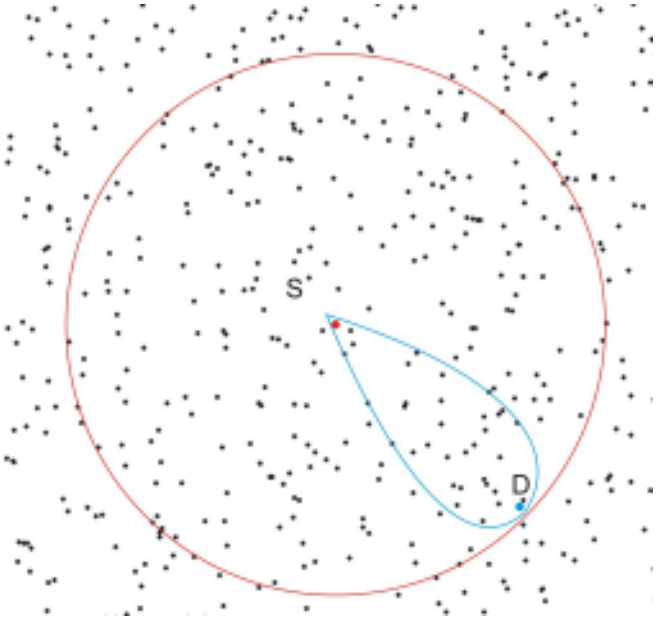

Figura 3.15.- Área cubierta por la inundación de paquetes de flooding entre el nodo origen S y el destino D. La circunferencia roja agrupa a los nodos que recibirían los paquetes en una búsqueda omnidireccional. La línea azul representa una búsqueda dirigida hacia el nodo, mediante la búsqueda de encuentros recientes.

El nodo origen "S" busca al nodo ancla más cercano A_1 tal que $T_{LE}(A_1,D) < T_{LE}(S,D)$ (este es el nodo más cercano que ha encontrado al destino más recientemente que S). El nodo a1 entonces busca a su alrededor por el nodo ancla más cercano A_2 tal que $T_{LE}(A_2,D) < T_{LE}(A_1,D)$. Posteriormente el nodo A_2 repite el proceso búsqueda de ruta cerca y el proceso se itera hasta que se alcance al destino d (para el cual $T_{LE}(D,D)=0$).

Nótese que este algoritmo no requiere conocimiento global y se basa en una implementación distribuida, debido a que cada búsqueda se define solamente en términos de las tablas locales de encuentros de los nodos. De la misma manera, el algoritmo sólo hace uso de tiempos relativos (antigüedades de encuentros) y por tanto no es necesaria la sincronización de los relojes. En la figura 3.15 se hace una comparación gráfica del desempeño de una búsqueda orientada y el flooding tradicional.

El primer criterio de desempeño es el costo de las n búsquedas ($n \geq 1$) en un descubrimiento de ruta, que claramente es menor que aquel costo de una búsqueda omnidireccional por flooding ciego. Cuando se toma en cuenta movilidad, la distancia recorrida durante un intervalo de tiempo t, se encuentra positivamente correlacionada con t. Los autores llaman a esto correlación tiempo – distancia.

Para ejemplificar lo anterior consideramos tres nodos, I, J y D. En el tiempo presente t = 0, el nodo I se encuentra a una distancia Di al nodo D; de manera similar, el nodo J se encuentra a una distancia

Dj del nodo D. La intuición detrás de FRESH es que $T_{LE}(I,D) < T_{LE}(J,D)$, entonces con una probabilidad muy alta Di < Dj. De manera coloquial tenemos; "un nodo que fue mi vecino hace 5 minutos, se encuentra probablemente más cerca que un nodo que fue mi vecino hace 5 horas".

El segundo criterio de desempeño es la calidad de las rutas. A pesar de que iteraciones sucesivas de la búsqueda de encuentros frescos en promedio nos acerca al destino, pueden no avanzar siempre en una línea recta y por tanto pueden ofrecernos no precisamente la ruta mas corta. Puesto que FRESH establece rutas a un menor costo que métodos de pasos simples, los autores consideran que se establece un compromiso entre calidad de la ruta y la reducción en el costo de la búsqueda, por lo que se debe asegurar que las rutas permanezcan lo suficientemente buenas.

3.5.4 Propuesta de *flooding* eficiente: búsquedas limitadas

En lo que se refiere a propuestas de *flooding* eficiente, existen soluciones muy variadas como se analizó en el capítulo anterior. Algunas propuestas se basan en información obtenida de forma heurística, utilizando modelos probabilísticos para retransmitir o no los paquetes de control, o bien auxiliándose en mecanismos de conteo definidos para establecer la distancia relativa al destino y explotando la información recopilada de los nodos vecinos. También se comentaron esquemas que realizan una recopilación constante de información topológica mediante el intercambio periódico de paquetes "hello" en sus diferentes configuraciones: árbol fuente (*source tree*), relevo multipunto (MPR) y la organización en grupos.

3.5.4.1 Ventajas y desventajas de los esquemas de *flooding* eficiente actuales

La finalidad de los esquemas propuestos es optimizar el costo en el descubrimiento de ruta sobre la cantidad de paquetes de datos entregados al destinatario (paquetes de control generados totales contra paquetes de datos efectivos) y por ende, maximizar el aprovechamiento del de por si escaso ancho de banda. Sin embargo, el desempeño de las diversas propuestas depende en gran manera de la aplicación y características específicas de la red *ad-hoc* en estudio. A continuación se presenta un breve análisis de las ventajas y desventajas de los esquemas analizados anteriormente.

1) **Protocolos basados en el modelo heurístico:** Esquemas tales como retransmisión aleatoria del paquete de control, contador del número de retransmisiones y recopilación de la información de la localización de usuarios, solucionan primeramente problemas como retransmisiones al infinito de paquetes de *flooding* pero no su total propagación en toda la red. El uso de tiempos de espera aleatorios para el envío del paquete de control, puede incrementar el tiempo de latencia en el descubrimiento de ruta y por ende en la transmisión de datos de información. Este tipo de esquemas presenta un buen desempeño para redes pequeñas (baja densidad de nodos por metro cuadrado), pero se ve reducido en casos de redes densamente pobladas y niveles altos de movilidad

2) **Protocolos basados en la topología:** La mayoría de este tipo de esquemas se basan principalmente en el intercambio constante de paquetes *HELLO* entre nodos vecinos, para recopilar y actualizar información sobre la topología actual alrededor de cada nodo, aunque varían principalmente en la estructura y en el uso de jerarquías para establecer "niveles de responsabilidades" diferenciando a los nodos.

   En el caso particular de protocolos basados en la topología de vecinos, los estudios propuestos se diferencian entre sí en cuanto al límite de saltos desde un nodo en particular en que se propagarán los paquetes *HELLO* (MPR realiza búsquedas a dos saltos por ejemplo [25]). La ventaja de este tipo de esquemas es que se mantiene actualizada la tabla de vecinos y por ende el

conocimiento de la topología a "n" saltos (dependiendo de la periodicidad del envío de los paquetes *hello*), definiéndose el número de retransmisiones necesarias para alcanzar a un grupo de nodos vecinos. El primer aspecto a definir es la periodicidad del envío de paquetes de control; por un lado si el intervalo de tiempo entre una transmisión y otra de paquetes *HELLO* es pequeño, la tabla de vecinos se actualiza constantemente incluso para índices de movilidad altos, pero con el costo de consumo de ancho de banda y un incremento en el *overhead* en la red. Por tanto este tipo de propuestas funcionan adecuadamente para densidad de nodos baja y para índices de movilidad bajos, lo cual presenta cambios de la topología pequeños; para el caso contrario la red se congestionará por la excesiva transmisión de paquetes de control y los constantes cambios en la topología para niveles de movilidad altos requerirá de de constantes actualizaciones para el buen desempeño del sistema [26].

Finalmente para los esquemas que utilizan configuración con árbol fuente (*source-tree*), se define un esquema jerárquico definido por el número de saltos desde el nodo origen (nodo raíz del árbol). Como se analizó en el capítulo anterior, se requiere también del constante intercambio de paquetes de control para que cada nodo actualice su estado actual dentro de la jerarquía del árbol, cuya propagación se ayuda de *"flooding* ciego". Nuevamente este esquema se ve impactado para densidades grandes de nodos y niveles altos de movilidad, pero resulta adecuado para redes de áreas grandes.

3) **Protocolos basados en clústeres:** Para este caso existe una gran diferencia en el desempeño de sus dos modalidades. Para el caso de formación activa de clústeres, donde constantemente se intercambia paquetes de control entre los nodos para definir al nodo "cabeza de clúster" aunque no se tenga información que transmitir, funciona bien para bajas tasas de transferencia de información pero se generan una gran cantidad de paquetes de control congestionando la red. Por otro lado, para la formación pasiva de clústeres, donde se definen los grupos de nodos sólo cuando exista información para transmitir, se genera una cantidad pequeña de paquetes de control y responde bien para densidades de nodos alta. Sin embargo se ve afectado cuando se presentan niveles altos de movilidad.

3.5.5 Otras alternativas en la búsqueda de *flooding* eficiente: el concepto de *Geocast*

El Servicio de Localización por Rejilla (*Grid Location Service* - GLS)[27], es un sistema distribuido de localización el cual sigue y establece la posición geográfica de un nodo dentro de una red *ad hoc*. GLS combinado con *flooding* geográfico (descrito adelante), permite la construcción de redes móviles *ad hoc* con un mayor número de nodos que otros estudios relativos a este tema. GLS es descentralizado y corre por sí mismo en cada nodo, de tal forma que no requiere de una infraestructura definida.

Cada nodo móvil periódicamente actualiza a otros nodos (sus servidores de localidad) con su posición actual, sin saber sus identidades actuales, asistido por un orden predefinido de identificadores de nodo y jerarquía geográfica predefinida. Las peticiones de localización de un nodo se logran utilizando el identificador de nodo predefinido y jerarquía espacial para encontrar a un servidor de localidad.

El sistema de rejilla (*Grid*), combina una infraestructura cooperativa con información de localidad o posición para implementar ruteo en una red *ad hoc* de grandes dimensiones. *Grid* utiliza *flooding* geográfico (*Geocast*) para tomar ventaja de la similitud entre proximidad física y de red. Una fuente debe de saber las posiciones geográficas de cualquier destino hacia el cual se desea comunicar y debe de etiquetar los paquetes para ese destino con su posición. Un nodo intermediario sólo debe de saber su propia posición y de los nodos en vecindad; ésta es información suficiente para retransmitir cada paquete a través del nodo vecino

que se encuentra geográficamente más cercano al destino final. A pesar de que *Grid* retransmite los paquetes basado puramente en información geográfica, los paquetes basan también su aproximación en términos de saltos restantes hacia el nodo final.

Debido a que los nodos sólo necesitan información local, en lugar del tamaño total de la red, el *flooding* geográfico es atractivo para redes a gran escala. Sin embargo, para ser útil en un contexto más amplio, un sistema basado en *Geocast* debe de proveer también un mecanismo para que las fuentes aprendan las posiciones de los destinatarios. Para preservar la escalabilidad este servicio de de localización debe permitir la ejecución de peticiones y actualizaciones utilizando sólo un puñado de mensajes. Desde luego, el servicio de localización por sí mismo debe de operar usando únicamente *Geocast*.

3.5.6 *Flooding* geográfico

El esquema de *flooding* geográfico propuesto por los autores es similar a encaminamiento Cartesiano [28]. Cada nodo determina su propia posición geográfica usando un mecanismo como un Sistema de Posicionamiento Global (*Global Position System* - GPS), que proporciona la posición en términos de latitud y longitud. Un nodo anuncia su presencia, posición y velocidad a sus vecinos (otros nodos dentro de la cobertura del radio) mediante la difusión periódica de paquetes "hello". Cada nodo mantiene una tabla de las identidades de sus vecinos actuales y sus posiciones geográficas.

El encabezado de un paquete destinado para un nodo particular, contiene la identidad del destinatario así como su posición geográfica. Cuando un nodo necesita retransmitir un paquete hacia la posición P, el nodo consulta su tabla de vecinos y escoge al usuario más cercano a P. Entonces transmite y releva el paquete al vecino hacia ese vecino, el cual por sí mismo aplica el mismo algoritmo hasta que el paquete se detiene cuando alcanza al destino. Un paquete puede también alcanzar a un nodo que no sabe acerca de ningún nodo más cercano al destino que el mismo. Este "callejón sin salida" indica que hay un "agujero" en la distribución geográfica de los nodos, como se muestra en la figura 3.16. En ese caso, la implementación descrita en este documento falla y envía un mensaje de error al nodo fuente del paquete.

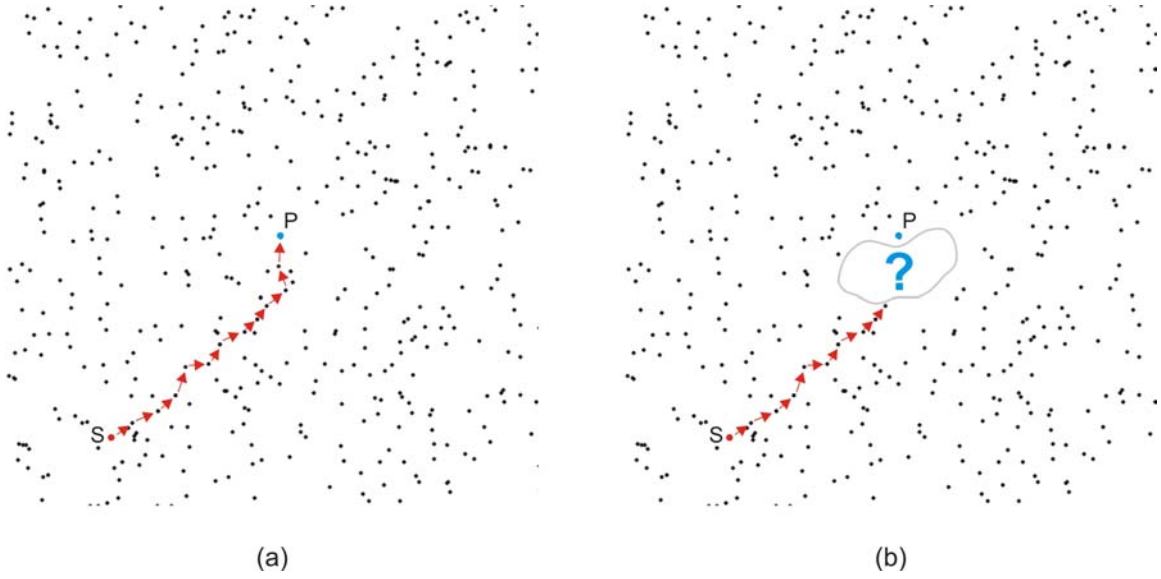

Figura 3.16.- Entrega de paquetes mediante flooding geográfico. (a) El paquete se va retransmitiendo por aquellos nodos que se encuentran más cercanos al destino en las tablas individuales de vecinos. (b) El último nodo que recibe el paquete se encuentra frente a un "agujero" en la red. Si No encuentra un nodo vecino más cercano al destino P que él mismo; genera mensaje de error.

Sin embargo, existe una propuesta para resolver el problema de los "agujeros" en este tipo de ruteo denominada GPSR (*Greedy Perimeter Stateless Routing for wireless networks*)[29], que es un sistema de ruteo geográfico que usa subgráficas planares de la gráfica principal de la red inalámbrica para encontrar rutas alrededor de los agujeros. En ese artículo [29], se simulan redes inalámbricas con 50 a 200 nodos y se demuestra que entrega una mayor cantidad de paquetes con éxito con menor *overhead* ocasionada por el protocolo de encaminamiento, que en el caso de utilizar DSR para redes con poco más de 50 nodos.

Cabe mencionar que *Grid* y otras propuestas como GPSR obtienen un mejor desempeño para redes lo suficientemente densas, de tal forma que los "agujeros" se presenten difícilmente.

3.5.6.1 Ventajas y desventajas de utilizar *flooding* geográfico

Sin duda los estudios analizados ofrecen esquemas que han demostrado ser confiables para su uso en redes inalámbricas con densidades de nodos medias y altas. El uso de un sistema de posicionamiento global (GPS) y por ende, el conocimiento previo de la localización geográfica de los nodos, ayuda a reducir de manera significativa la generación de paquetes de control para el descubrimiento de rutas.

De la misma manera, resulta conveniente y poco costoso el intercambio de información relativa a la posición y velocidad entre nodos vecinos y conformar tablas donde se almacene esta información para su uso futuro. Esta información, puede servir como complemento a protocolos de encaminamiento subyacentes (como el uso de DSR con *Grid*), para definir si un paquete de control o paquete de datos, podrá o no ser retransmitido hacia los vecinos, reduciendo así la zona afectada por el *flooding*. Otro aspecto a resaltar, es la mejora en el desempeño de estos esquemas para redes de área grande, donde han probado ser efectivos y eficientes para la entrega con bajo costo de paquetes de datos.

Sin embargo, hay puntos que pueden considerarse como desventajas acerca de estas propuestas de *flooding* eficientes. En primer lugar, aunque resulta sumamente útil el uso de un sistema de GPS (de hecho las propuestas mencionadas se basan en su totalidad en estos sistemas), actualmente resulta todavía costoso adquirir uno de estos equipos; los precios van de 150 a 250 dólares en el mercado.

Otro reto que enfrentan los sistemas basados en GPS, es la posibilidad de encontrar "huecos" entre grupos de nodos que se desean comunicar, ocasionando una interrupción en el proceso de búsqueda de ruta y su respectivo mensaje de error; además dependen de una densidad de nodos alta en la red. Las soluciones a estos problemas, como en el caso de GPSR implican un mayor consumo de recursos en el procesamiento del algoritmo para "rodear" los "huecos".

En este capítulo se estudiaron los aspectos más importantes a considerar a la hora de desplegar una red *ad-hoc*. Se definieron los retos que presenta esta topología, principalmente el rango limitado de transmisión debido a la naturaleza portátil de los equipos y por ende el uso de baterías, la naturaleza broadcast del medio; las pérdidad de paquetes debido a errores en la transmisión, los cambios de ruta inducidos por la movilidad y finalmente la relativa vulnerabilidad ante ataques o robo de información.

De igual manera se prestó especial atención al funcionamiento de protocolos de encaminamiento comúnmente utilizados en redes *ad-hoc*, haciendo incapié en la técnica utilizada para el descubrimiento de rutas mejor conocido como inundación (*flooding*). Se estudiaron a AODV, TORA y DSR, resaltando el funcionamiento de este último por el esquema de almacenamiento de rutas.

Posterior a la adopción de los protocolos de encaminamiento arriba mencionados, han aparecido propuestas que tienen una marcada tendencia a optimizar la difusión de paquetes de *flooding*, de los cuales se estudiaron a protocolos basados en arbol fuente, arquitecturas en grupos y de manera especial al protocolo de descubrimiento de ruta eficiente utilizando registros de encuentros (algoritmo FRESH). Finalmente se establecieron las ventajas y desventajas de los esquemas de *flooding* eficiente actuales.

IV. *Fireworks*: nuestra propuesta como una alternativa eficiente de *flooding*

Previo a la realización de este documento, se realizaron extensos estudios bibliográficos sobre las diferentes opciones en cuanto a encaminamiento *ad hoc* se refiere. Existen diferentes propuestas, aunque comparten varios puntos de funcionamiento en común:

1) La métrica inicial a utilizar para determinar la mejor ruta es siempre el número de saltos
2) La secuencia de búsqueda incluye el envío de una petición, respuesta y mensaje de error (en caso de presentarse)
3) El envío de los mensajes de control es a través de una transmisión UDP
4) Conforme el paquete de control se propaga mediante saltos múltiples, se va acumulando un registro de las direcciones de los nodos que participan en la transmisión y en algunos casos se pueden "aprender" de las rutas generadas por nodos vecinos

En su mayoría, los protocolos de encaminamiento que no se ayudan de sistemas GPS, tienen un desempeño aceptable para redes que cubren áreas pequeñas, debido a su principal limitante del número máximo de saltos de propagación (por lo regular 16), aunque la cantidad de paquetes de control generada en muchos casos es poco eficiente en comparación con sus contrapartes geográficos.

Sin embargo, se considera después de una serie de estudios sobre este tema, que la información generada por protocolos de encaminamiento que utilizan *flooding* ciego (por ejemplo DSR), es suficiente para optimizar su desempeño y desarrollarse una alternativa eficiente, de bajo costo de procesamiento y de inversión, capaz de responder a los cambios en la topología y de rastrear a un usuario en la red, con el cual se ha establecido una conexión en algún momento, sin la necesidad de algún dispositivo de posicionamiento adicional, extendiendo su aplicación en redes de mayor tamaño.

4.1 Estructura del protocolo

Para el desarrollo de nuestra propuesta nos basamos en el protocolo DSR (*Dynamic Source Routing Protocol*), mismo que seleccionamos debido a que el encaminamiento se realiza en base a direcciones fuente y origen, así como su capacidad de aprendizaje de rutas y el manejo de la tabla de rutas, lo cual se describirá dentro de este capítulo. Sin embargo, cabe recalcar que Fireworks se puede implementar de igual forma en AODV y TORA, con la debida adaptación en sus estructuras de datos.

Por naturaleza, DSR emplea el mecanismo de *flooding* ciego para el descubrimiento de rutas entre nodos. Sin embargo, posee dentro de su estructura una serie de mecanismos para optimizar el envío de paquetes de control que le permite una rápida convergencia a cambios en la topología, mediante la explotación de las rutas almacenadas por los nodos participantes en la comunicación. Estos mecanismos fueron mencionados en el capítulo anterior y son principalmente:

1.- Confirmación Gratuita de Ruta (*Gratuitious Route Reply*): Este mecanismo se utiliza para acortar las rutas y optimizar el número de saltos necesarios hacia el destinatario. Un nodo "escucha" un paquete

con información de ruta, para después analizar la porción todavía no utilizada de los saltos siguientes; en caso de reconocer que él mismo es un salto en la ruta o bien es el destinatario, genera un mensaje de *gratuitious route reply* hacia el nodo origen, indicándole que se pueden acortar algunos saltos de la ruta para alcanzarlo. Como ejemplo tenemos la figura 4.1, donde observamos que el nodo D ha "escuchado" un paquete de datos que se ha transmitido de B a C, para después entregarlo a D y finalmente a E. En este caso, el nodo D deberá emitir una respuesta de ruta gratuita al iniciador original del paquete, por no ser necesario el salto entre B y C para llegar al destino. A este mecanismo también se le conoce como "recorte automático de ruta" (*automatic route shortening*).

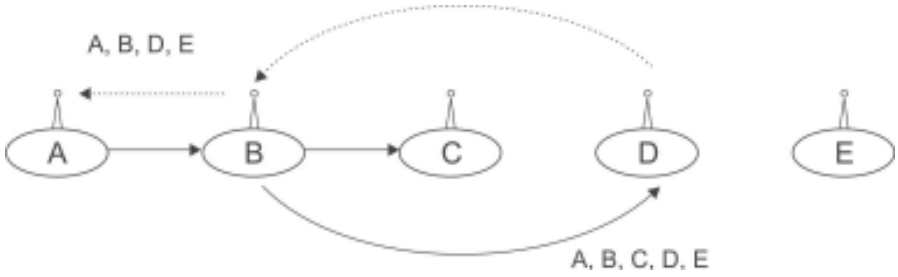

Figura 4.1.- Mecanismo de confirmación gratuita de ruta. El salto B-C no es necesario para llegar a E, por lo que D indica a A recortar la ruta (automatic route shortening)).

2.- Salvamento de paquete (*Packet Salvaging*): Una vez que se ha establecido una ruta entre dos nodos, puede ocurrir que esta enlace se rompa debido a movilidad, a congestión en un segmento del enlace o debido a que uno de los nodos deja de funcionar. Esta ruptura de ruta, en primer lugar hace necesario el descubrimiento de una nueva ruta por lo que inundar toda la red en búsqueda del destino parecería inminente. Sin embargo, después de haberse generado y propagado el mensaje de error en ruta (*route error*) cuando se rompe un enlace, cada nodo que retransmite este mensaje busca en su tabla de rutas por un nuevo camino hacia el destino. En caso de encontrar una ruta lo suficientemente fresca, envía un mensaje de salvamento de paquete (*packet salvaging*) hacia el origen para que pueda continuar la comunicación sin la necesidad de una petición de ruta y por tanto, *flooding* adicional.

3.- Almacenamiento de ruta escuchada: Todo nodo que participa en una comunicación o bien que opera en cercanías con su radio operando en modo "promiscuo", almacenará las rutas escuchadas almacenadas en peticiones y transmisiones de datos. De la misma manera las eliminará de su tabla de ruteo cuando escuche la generación de mensajes de error en esa ruta y por tanto deje de ser confiable.

Fireworks (como hemos denominado a nuestro propuesta), es un protocolo mejorado diseñado específicamente para su aplicación en redes inalámbricas del tipo *ad-hoc*, que utiliza la información de rutas generada en la red para realizar una búsqueda eficiente de los nodos destino en una comunicación.

Nuestra propuesta, realiza el descubrimiento de rutas de manera "sobre demanda"; esto quiere decir que la búsqueda de nodos se realizará cada que existan datos que se desean transmitir. De igual manera, los mecanismos utilizados para la transmisión de paquetes de control, se basan en información heurística (algoritmos previamente definidos) y en información topológica recolectada por los mismos nodos.

Fireworks opera bajo las siguientes premisas:

1) **Recopilación de información topológica de vecinos**: Cada que se establezca una conexión, los nodos origen y destino de la comunicación deberán recopilar e intercambiar la información topológica de sus vecinos a un salto, junto con la información propia de la ruta.

2) **Escuchar enlaces en proximidad**: Los adaptadores inalámbricos de red de los nodos participantes en la red *ad-hoc*, deberán estar en modo "promiscuo", de tal forma que los usuarios podrán "aprender" de rutas "escuchadas" a partir de nodos vecinos, sin participar necesariamente en ellas
3) **Reducir lo más posible el área de la red afectada por la inundación de paquetes de control**: Cuando se desee establecer una comunicación con un nodo específico, el nodo fuente buscará inicialmente no sólo al destinatario en cuestión, sino a todos sus vecinos reportados con anterioridad a "n" saltos del origen, buscando así reducir el área de propagación de los paquetes de *flooding*. En caso de no encontrar al destinatario en la primera búsqueda, deberá procederse con la inundación de la red.
4) **Los nodos deberán conservar una tabla de vecinos de los nodos origen y destino con los que han establecido una comunicación**, no así de los demás nodos que participen en la misma.

A continuación se presentan los mecanismos y conceptos que conforman al protocolo de localización *Fireworks*.

4.2 Obtención de la lista de vecinos

Durante todo el tiempo en que un nodo se mantenga operando en la red *ad-hoc* deberá, mediante el envío constante de paquetes hello transmitidos o bien mediante la habilitación de su adaptador inalámbrico de red en modo "promiscuo" (escuchando todas las comunicaciones que ocurran dentro de su rango de recepción), almacenar y actualizar una lista de aquellos nodos que se encuentren a 1 salto (vecinos), como se muestra en la figura 4.2.

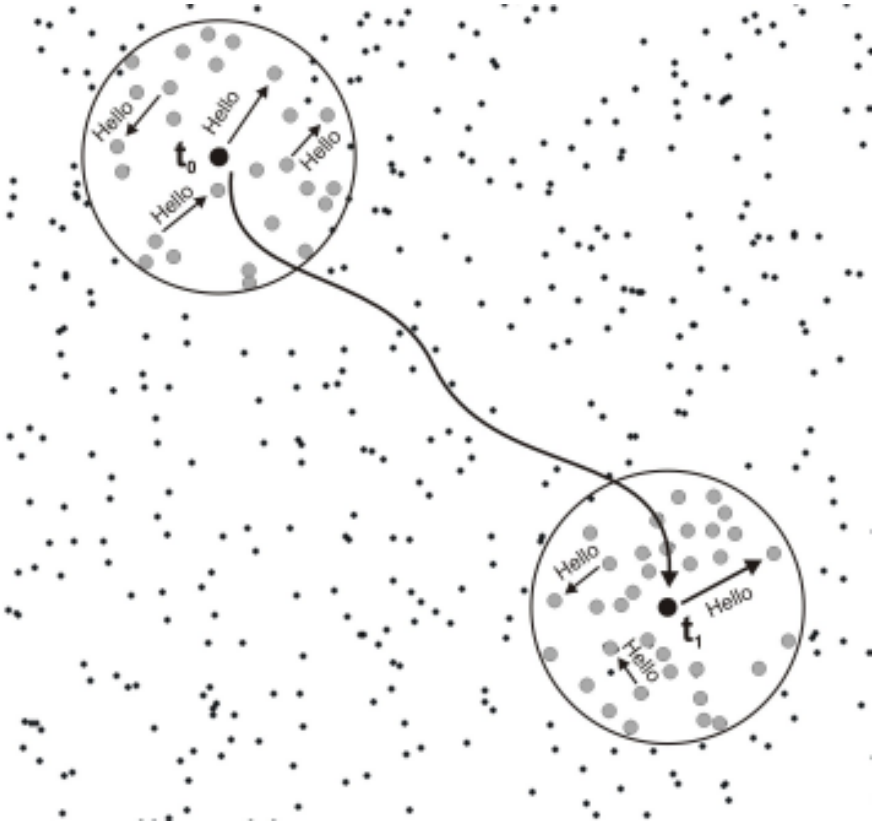

Figura 4.2.- Los nodos transmiten paquetes HELLO a 1 salto para recuperar la dirección de los nodos vecinos. Esta operación se realiza en el tiempo t_0 y de igual forma en el tiempo t_1, cuando se ha presentado un cambio de posición debido a movilidad.

Cuando un par de nodos establezcan una comunicación, deberán intercambiar una lista con las direcciones de sus nodos vecinos (lista de vecinos - LV), de tal forma que, como se muestra en la figura 4.3, el nodo I conoce los vecinos de J y viceversa.

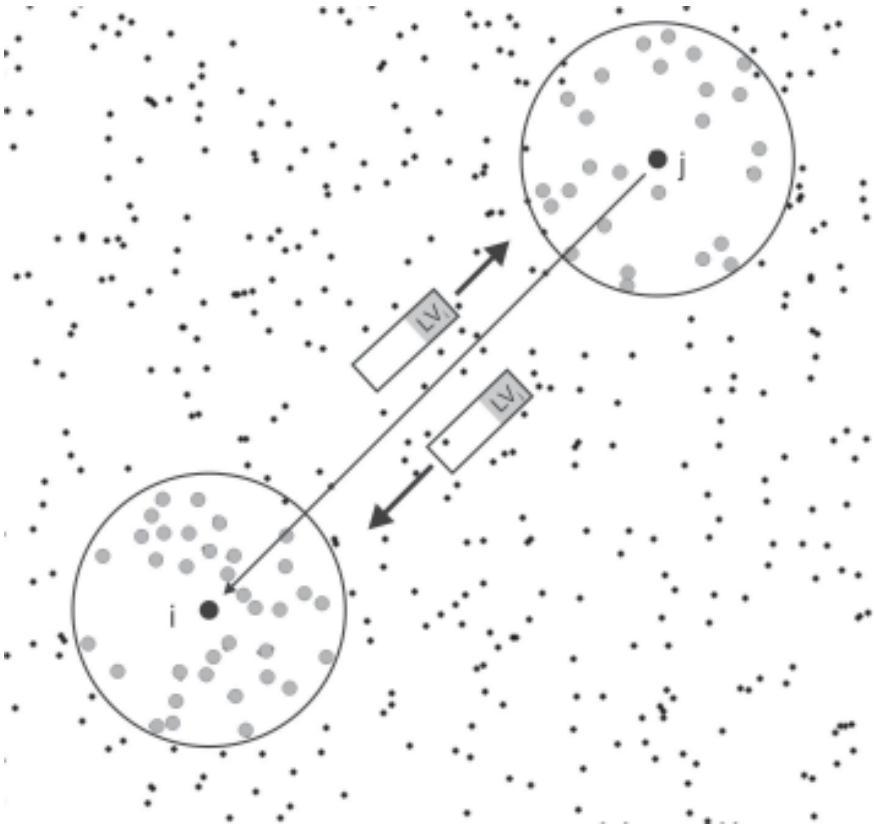

Figura 4.3.- Durante una comunicación, los nodos intercambian sus listas de vecinos correspondientes. El nodo I transmite su lista de vecinos LV_I y nodo j su lista de vecinos JV_J.

4.3 Descubrimiento de ruta

Cuando un nodo S desee comunicarse con otro dentro de la red, deberá transmitir un paquete de petición de ruta RREQ (*Route Request*) con un límite de propagación definido (por ejemplo 3 saltos), incluyendo direcciones IP de los nodos origen y nodo destino como se muestra en la figura 3.4. Todos aquellos nodos que reciban este paquete dentro del rango especificado, deberán abrir la lista de vecinos del destino (en caso de existir) y verificar si se tiene una ruta válida en su tabla de rutas locales; en el caso de que se cuente con una ruta fresca, el nodo en cuestión responderá al nodo origen con una respuesta de ruta (*Route Reply*) y se podrá iniciar la transmisión. En el caso de que no se obtenga respuesta alguna en la petición acotada inicial, se procederá a propagar a toda la red con la petición de la ruta, hasta alcanzar al destino. Esta información será propagada a un número pequeño de saltos del origen, por lo que sólo se inunda una pequeña porción de la red en cada búsqueda. Una vez que un nodo vecino de S (Vs) recibe la petición de ruta, consultará en su tabla de ruteo si se conoce un camino hacia al destino o en su defecto un camino alterno a alguno de sus vecinos, como se muestra en la figura 3.5.

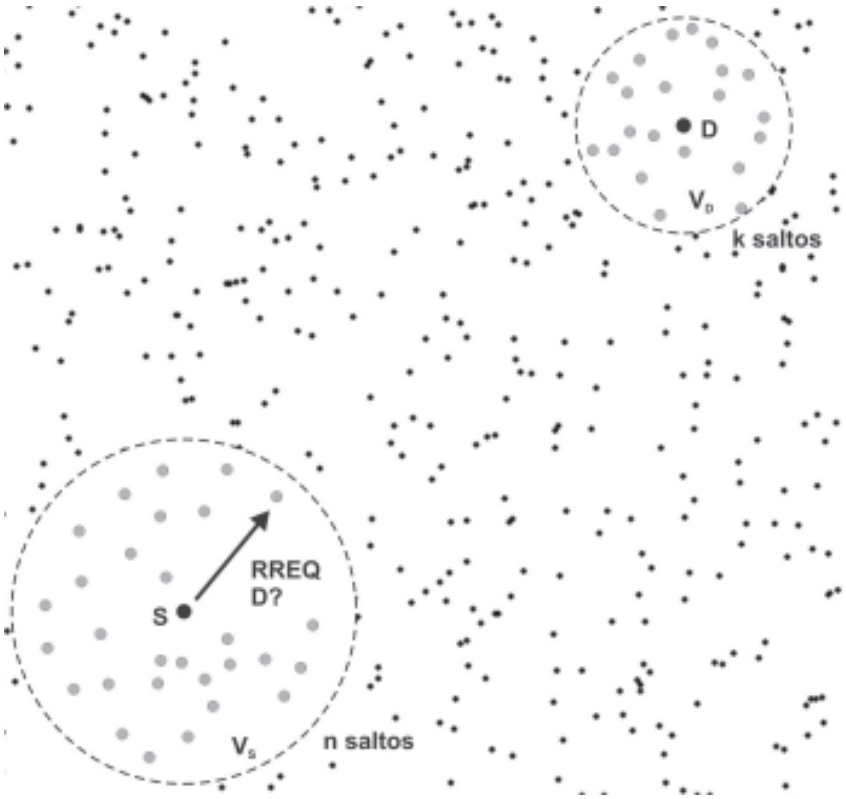

Figura 4.4.- El nodo S inicia una búsqueda acotada a "n" saltos, donde pregunta por D

Figura 4.5.- Los nodos alcanzados por el RREQ acotado a n saltos que conocen un camino hacia D o hacia uno de sus vecinos, envían a S las rutas correspondientes. K simboliza un número acotado de saltos a partir del nodo destino

El desempeño eficiente de este esquema de búsqueda de ruta, depende en gran medida en que todo nodo que participe en el relevo de un paquete (sea de datos o control) o bien que ha "escuchado" la transmisión de un paquete, pueda rescatar la información de ruta contenida en la transmisión y la agregue a su propia de tabla de ruteo. Esta información podrá provenir de registros de saltos acumulados de un RREQ, la ruta contenida en un RREP y actualizarse por cambios en la topología notificados por un mensaje de error RERR.

4.4 Procedimiento de búsqueda *Fireworks*

Junto con el mecanismo de descubrimiento de ruta básico, la innovación de nuestra propuesta consiste en que si bien no se puede alcanzar al nodo destino en sí con el RREQ inicial, pero se cuenta con una ruta lo suficientemente fresca hacia alguno de sus vecinos, podemos realizar una búsqueda adicional originada por el vecino alcanzado, al que denominaremos en este trabajo "nodo ancla" (*anchor node*), como se muestra en la figura 4.6. Esta segunda búsqueda deberá también estar geográficamente acotada, para reducir al máximo el área de la red que se verá afectada por la propagación de paquetes de control. Como es de esperarse, la búsqueda adicional se realizará utilizando *flooding* ciego a un número acotado de saltos del nodo ancla. Entonces, cuando el nodo origen reciba un RREP proveniente de algún nodo a "n" saltos de él, al reconocer que esa ruta no pertenece al nodo destino enviará un paquete ***unicast*** (denominado paquete *fireworks*) a través de la ruta definida. Cuando el paquete *fireworks* alcance al nodo vecino, éste deberá mandar un paquete de RREQ a "k" saltos buscando al nodo destino.

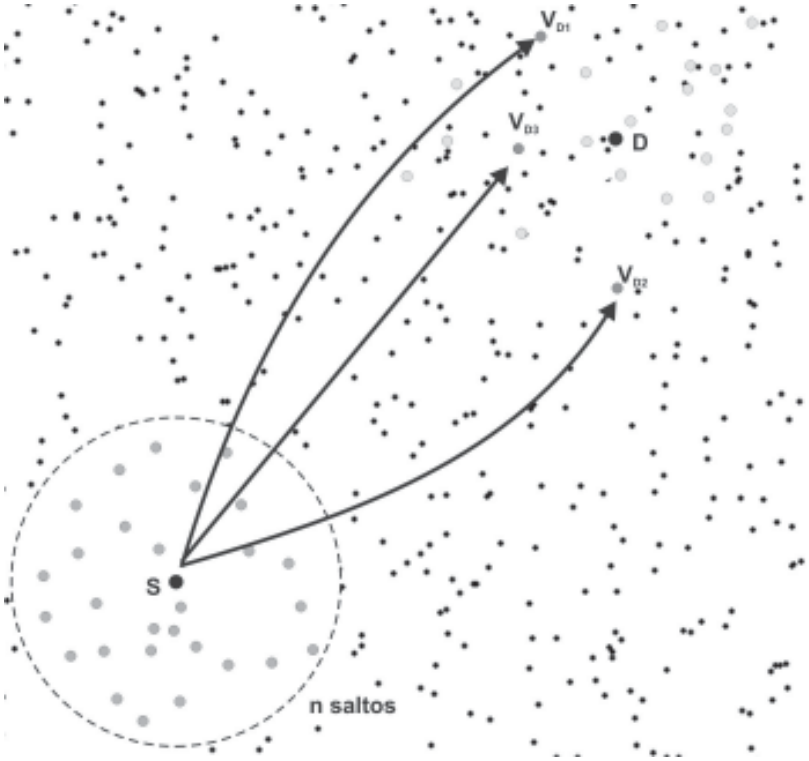

Figura 4.6.- Los nodos D y sus vecinos V_D se han desplazado con el paso del tiempo. S recuperó rutas a los vecinos V_{D1}, V_{D2} y V_{D3}, y procede a enviarles un paquete unicast de búsqueda (paquete fireworks), con la dirección del nodo D.

Cabe mencionar que la forma de propagación y búsqueda del paquete observada en el simulador, es similar al lanzamiento y explosión de los fuegos artificiales, por lo que decidimos nombrar a nuestra propuesta "Fireworks". El paquete se envía a la zona donde se espera se encuentre el destino y "explota" realizando una búsqueda limitada como se muestra en la figura 3.7.

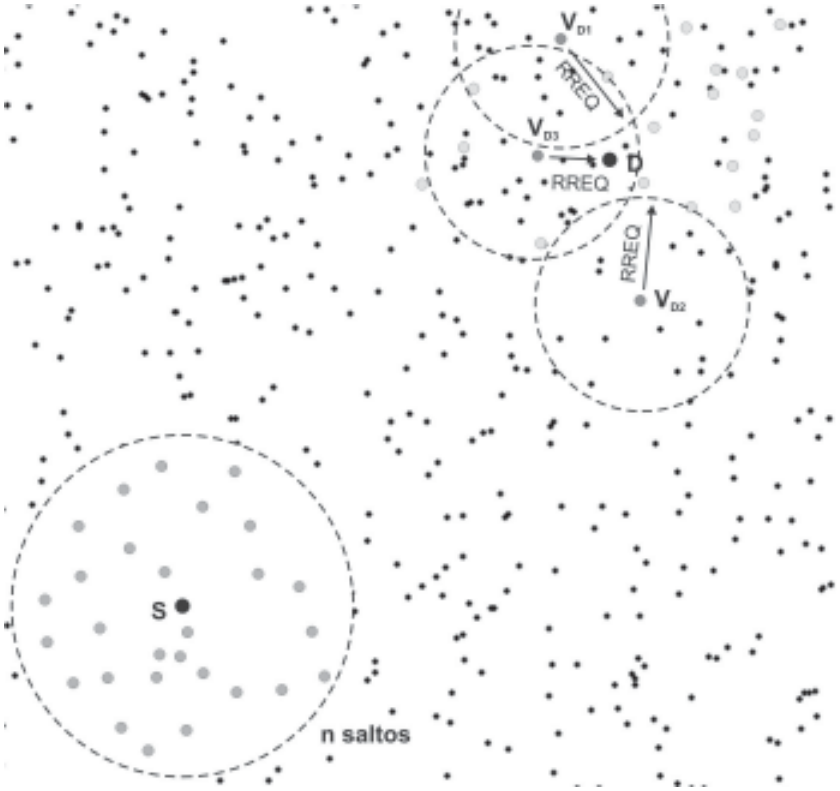

Figura 4.7.- Los nodos V_{D1}, V_{D2} y V_{D3} reciben el paquete fireworks e inician una nueva búsqueda de ruta RREQ para encontrar al nodo D a "k" saltos.

4.5 Mantenimiento de ruta

Al igual que en DSR, cuando se origine o se transmita algún paquete a través de una ruta obtenida, cada nodo que participe en la ruta es responsable de la correcta recepción del paquete en el salto siguiente en donde se deberá regresar un paquete de ACK (confirmación). En caso de no recibir la confirmación correspondiente, el nodo deberá retransmitir el paquete un número máximo de veces definido que al ser alcanzado, ocasionará la generación de un mensaje de error en ruta (*Route Error*) al emisor original del paquete. Cuando los nodos participantes en la conexión reciban este mensaje de error, deben de realizar las siguientes acciones:

1) Identificar a partir del mensaje de error cual fue el nodo que no confirmó la recepción del paquete
2) Revisar si se tiene otra ruta hacia nodo destino. En caso de ser así ejecutar el mecanismo de salvamento de paquete (*packet salvaging*), notificando a los nodos participantes que permanecen en comunicación sobre el nuevo camino que deberá tomar el paquete y concatenar la ruta alternativa con la información de ruta antigua. De lo contrario, se debe de remover la ruta de la tabla

Al utilizar el esquema propuesto por *Fireworks*, la ruta descubierta al destino puede no ser la más corta, debido a las dos búsquedas realizadas para dar con el destino. Las rutas conocidas por el vecino de S (nodo origen) y por el vecino del nodo D (destino), fueron generadas de manera separada, que al concatenarse pueden presentar un mayor número de saltos de los necesarios.

Para estos casos existe un mecanismo de recorte automático de ruta (*Automatic Route Shortening*) donde los enlaces pueden ser recortados (en lo que se refiere al número de saltos que los conforman), si uno o más de los nodos intermedios en la ruta se vuelven innecesarios. A manera de ejemplo, observemos lo sucedido en la figura 4.8:

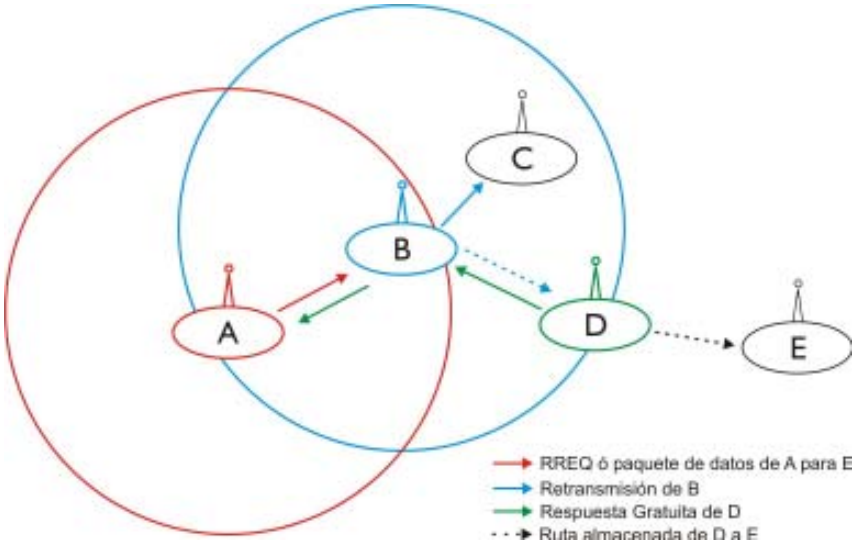

Figura 4.8.- Mecanismo de recorte automático de ruta (Automatic Route Shortening). El salto C a D es innecesario para la transmisión del paquete

Cuando se desea establecer una comunicación del nodo A al nodo E (o durante el intercambio de paquetes de datos a través de una ruta previamente establecida entre esos dos nodos), A transmite un paquete de RREQ que recibe B (o un paquete de datos), quien a su vez retransmite para C. Sin embargo el paquete es "escuchado" por D, quien posee en su tabla una ruta fresca hacia el nodo E por lo que responde con un paquete de respuesta de ruta gratuita (Gratuitious Reply) informando de la ruta al nodo A para que se inicie la comunicación o se redirija el camino a E.

4.6 Estructuras conceptuales de datos

Existen una serie de elementos importantes inherentes al protocolo, cuya definición y funcionamiento hacen posible el enrutamiento en redes sin infraestructura. Muchas de estas entidades o bien todas, están basadas en estructuras de datos ya bien definidas para redes guiadas, pero han sido adaptadas para funcionar en ambientes móviles y carentes de una entidad centralizadora.

4.6.1 Tabla de Rutas (*Route Cache*)

Toda la información de encaminamiento necesaria para un nodo que participa en una red ad hoc, usando DSR o Fireworks es almacenada en una tabla de rutas (*Route Cache*), misma que es creada y administrada de forma individual por cada nodo en la red. Todo nodo agrega información a su tabla de ruteo conforme la va "aprendiendo" o recopilando de nuevos enlaces que den lugar en la red; por ejemplo, un nodo puede aprender acerca de nuevos enlaces cuando recibe un paquete que porta ya sea una confirmación de ruta (RREP) o una petición de ruta (RREQ).

Por otro lado, todo nodo deberá remover información de su tabla cuando sea de su conocimiento que los enlaces existentes en la red *ad hoc* se encuentren fuera de servicio; por ejemplo, un nodo podrá aprender de un en lace roto cuando reciba un paquete portando un mensaje de error en ruta (*Route Error*) o a través de un mecanismo de retransmisión de la capa de enlace de datos, reportando una falla en la transmisión del paquete al siguiente salto.

La tabla de rutas trabaja bajo las siguientes premisas:

1) **La tabla de rutas debe de soportar el almacenamiento de más de una ruta a cada destino.** Durante la búsqueda de ruta para algún nodo destino, la tabla se encuentra indexada de acuerdo a la dirección del nodo destino. Cada implementación de *Fireworks* puede escoger cualquier estrategia apropiada y algoritmo para la búsqueda en su tabla de rutas, seleccionando la "mejor" ruta al destino de entre todas las rutas encontradas. Por ejemplo, un nodo puede seleccionar la ruta más corta al destino (el menor número de saltos) o puede seleccionar alguna otra métrica para seleccionar la ruta de la tabla.

   Sin embargo, si existen rutas múltiples a un solo destino, la selección de las rutas deben de dar prioridad a aquellas que no tienen activo el bit "externo" (que identifica cuando un nodo se encuentra en otra red DSR o *Fireworks* cercana a la de interés), prefiriendo aquellas rutas que llevan directamente al nodo destino.

2) **La implementación de la tabla de rutas debe de proveer una capacidad definida o bien el tamaño en memoria puede ser variable**. Cada implementación de *Fireworks* en cada nodo puede escoger una política apropiada para la administración de las entradas en su tabla de ruteo; por ejemplo en el caso en que la capacidad de la ruta esté limitada, se requerirá escoger cuáles rutas deberán permanecer almacenadas y cuales no, utilizando esquemas como el de "la ruta usada más recientemente".

3) **Cualquier organización de la estructura de datos adecuada consistente con su especificación, puede ser usada para implementar la tabla de rutas en cualquier nodo**. Por ejemplo, son posibles dos tipos de organización:

   a) En Fireworks y DSR, la ruta que se regresa en cada respuesta de ruta (RREP) recibida por el iniciador de la comunicación, representa un camino completo (una secuencia de enlaces) que llevan al nodo destino. Mediante el almacenamiento separado de estas trayectorias, se puede formar una organización de la tabla de ruteo. Una tabla de enlaces resulta simple de implementarse y garantiza fácilmente que todas las rutas estén libres de lazo (ciclos), puesto que cada ruta individual proveniente de un RREP o un RREQ, inherentemente se encuentra libre de este problema. Para la búsqueda de ruta, el nodo transmisor simplemente buscará en su tabla por cualquier enlace (o prefijo de enlace) que lleve al nodo destino.
   b) Alternativamente, se puede utilizar una organización de "tabla de enlaces", donde cada enlace individual en las rutas entregadas por los paquetes de RREP (o bien aprendidas de los encabezados de paquetes escuchados), se agregan a una estructura unificada de datos gráfica, que define el punto de vista actual del nodo de la topología de la red. Para buscar una ruta en la tabla, el nodo transmisor deberá de usar un algoritmo más complejo como el algoritmo de la ruta más corta de Dijkstra, para encontrar una trayectoria a través de la gráfica al nodo destino.

La elección de la organización de la estructura de datos de la tabla de rutas, corresponde a cada nodo y afecta únicamente a su desempeño; cualquier opción razonable de organización para la tabla no afecta lo correcto de la misma o su interoperabilidad.

4.6.2 Tabla de peticiones de ruta

La tabla de peticiones de ruta almacena la información acerca de los RREQ que fueron recientemente originados y transmitidos por un nodo en particular. La tabla se encuentra indexada con direcciones IP.

La información que se almacena en la tabla es la siguiente:

- El tiempo en que este nodo originó un descubrimiento de ruta para un nodo destino
- El número de RREQ consecutivos iniciados por este objetivo, recibiendo una respuesta de ruta válida con una ruta a ese destino
- La cantidad de tiempo restante antes del cual este nodo pudiera realizar un nuevo intento de petición de ruta para ese nodo destino
- El campo de tiempo de vida (TTL) usado en el encabezado IP para la última búsqueda de ruta iniciada por este nodo

En suma, la tabla de peticiones de ruta de un nodo también almacena la siguiente información acerca de los nodos iniciadores de los cuales este nodo ha recibido peticiones de ruta:

- Un registro del tipo FIFO de tamaño de la variable REQUEST_TABLE_IDS, conteniendo el valor de identificación y dirección objetivo desde el RREQ recibido más recientemente por este nodo desde el nodo iniciador.

4.6.3 Búfer de envío (*send buffer*)

El búfer de envío de algún nodo es una cola de paquetes que no han podido ser transmitido por ese nodo, debido a que no cuenta todavía con una ruta hacia el destinatario de esos paquetes. Cada paquete en el búfer tiene grabado el tiempo en el que se coloca, y debe de removerse de la cola y descartarse SEND_BUFFER_TIMEOUT segundos. En caso de ser necesario debe de utilizarse una estrategia FIFO para liberar los paquetes antes de que expiren para prevenir el sobreflujo del búfer. Cabe señalar que es necesario que se inicie una búsqueda de ruta tan seguido como sea posible para la dirección destino de cualquier paquete residiendo en el búfer de envío.

4.6.4 Formato de paquetes en Fireworks

Al igual que en el caso de los protocolos estudiados, Fireworks tiene formatos de paquetes definidos para cada uno de sus mecanismos de ruteo.

4.6.4.1 Petición de ruta (RREQ)

En los campos IP se contienen:

Dirección origen	Debe establecerse con la dirección IP del nodo que origina este paquete. Los nodos intermedios que vuelven a propagar el RREQ no deben de cambiar este campo.
Dirección destino	Debe de ajustarse a la dirección específica de broadcast (255.255.255.255).
Límite de Saltos (TTL)	Puede variar de 1 a 255, por ejemplo para implementar RREQ que no se propaguen y búsquedas de anillo expandiéndose.

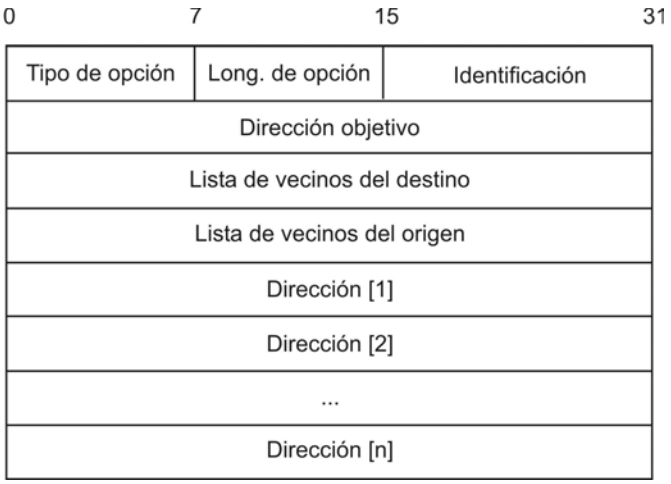

Figura 4.9.- Formato de paquete de petición de ruta (RREQ) para Fireworks

En los campos de la petición de ruta se tiene, como se muestra en la figura 4.9:

Tipo de Opción	Los 3 bits mas significativos de este campo son iguales a 011, significando que un nodo que no trabaje bajo DSR esta opción debe de descartar el paquete
Longitud de opción	Entero sin signo de 8 bits. Longitud de la opción en octetos, excluyendo los campos de tipo de opción y de longitud de opción
Identificación	Un valor único generado por el iniciador (transmisor original) del RREQ. Este valor permite al nodo receptor determinar si ha recibido a este paquete con anterioridad.
Lista de vecinos del destino	Lista de vecinos del destino obtenida durante la última comunicación con ese nodo y el nodo actual
Lista de vecinos del origen	Lista de vecinos del nodo origen
Dirección objetivo	La dirección del nodo que es el objetivo del RREQ
Dirección [1..n]	Es la dirección de los nodos participantes en la ruta

4.6.4.2 Respuesta de ruta (RREP)

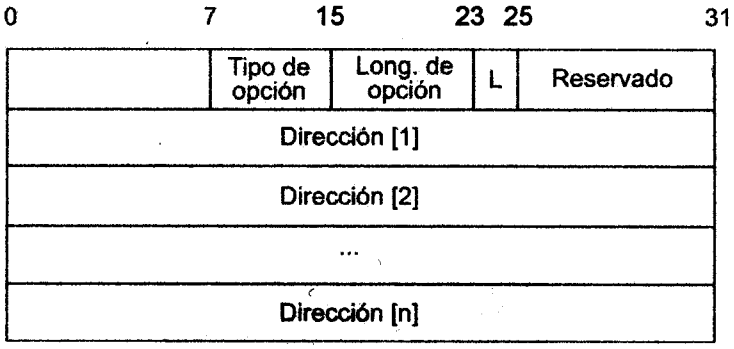

Figura 4.10.- Formato de paquete de respuesta de ruta (RREP) en Fireworks

Tipo de Opción Los 3 bits mas significativos de este campo son iguales a 011, significando que un nodo que no opere bajo DSR debe de descartar el paquete

Longitud de opción Entero sin signo de 8 bits. Longitud de la opción en octetos, excluyendo los campos de tipo de opción y de longitud de opción

Ultimo Salto Externo (L) Habilitado para indicar que el último nodo indicado por el RREP se encuentra en realidad en una red externa a la red DSR; la secuencia exacta de saltos que llevan a ella fuera de la red DSR no están representados en el RREP.

Dirección [1..n] Es la dirección de los nodos participantes en la ruta. El número de direcciones presentes en este campo se indica por el campo de longitud de opción, en la opción (n = (Longitud de opción -1)/4)

Reservado Enviado como 0; ignorado en recepción

4.6.4.2 Paquete Fireworks

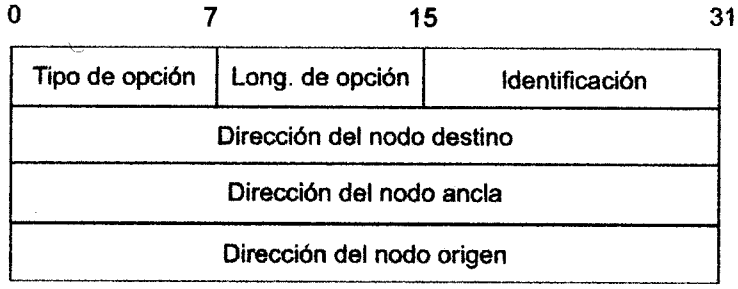

Figura 4.11.- Formato de paquete unicast Fireworks

Tipo de Opción Los 3 bits mas significativos de este campo son iguales a 011, significando que un nodo que no opere bajo DSR debe de descartar el paquete

Longitud de opción Entero sin signo de 8 bits. Longitud de la opción en octetos, excluyendo los campos de tipo de opción y de longitud de opción

Dirección del nodo destino Dirección IP del nodo objetivo de la búsqueda Fireworks

Dirección del nodo ancla Dirección IP del nodo vecino del nodo objetivo que servirá como ancla para alcanzarlo

Dirección del nodo origen Dirección IP del nodo que inició la búsqueda

4.6.4.3 Error en ruta (RRER)

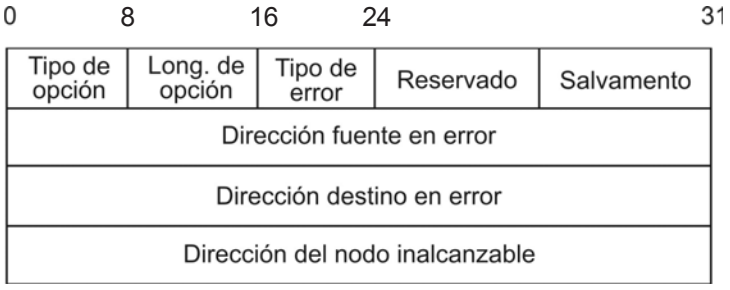

Figura 4.12.- Formato de paquete de error en ruta (RRER) en Fireworks

Tipo de Opción Los 3 bits mas significativos de este campo son iguales a 011, significando que un nodo que no entienda esta opción debe de descartar el paquete

Longitud de opción Entero sin signo de 8 bits. Longitud de la opción en octetos, excluyendo los campos de tipo de opción y de longitud de opción

Tipo de Error El tipo de error encontrado. Actualmente, el siguiente tipo se define: NODE_UNREACHABLE (nodo inalcanzable)

Reservado Enviado como 0; ignorado en recepción

Salvamento Entero de 4 bits sin signo. Copiado del campo de salvamento en el encabezado de ruteo DSR del paquete activando el error en ruta, incrementado por el nodo que regresa el mensaje de error

Dirección Fuente en Error La dirección del nodo que originó el RRER (por ejemplo el nodo que intentó transmitir el paquete y descubrió la falla en el enlace)

Dirección Destino en Error La dirección del nodo al cual el RRER debe de ser entregado

Dirección Del nodo inalcanzable	La dirección del nodo que se encontró inalcanzable (el vecino al siguiente salto al cual el nodo con la dirección fuente en error se trató de transmitir el paquete)

En el presente capítulo se explicaron a detalle los principios de operación del protocolo basado en la propagación de *flooding* eficiente denominado Fireworks. Se estableció que este protocolo opera bajo las premisas de que todo nodo en la red *ad-hoc*, mantendrá una tabla de vecinos a 1 salto obtenida mediante la propagación de paquetes de HELLO o bien mediante la recopilación de información de rutas "escuchada" alrededor del nodo, para que con esta información de rutas se inundará únicamente una porción limitada de la red con el objeto de reducir el impacto de la transmisión de paquetes de control para encontrar rutas.

Al momento de establecerse una comunicación, los nodos origen y destino intercambiarán sus listas de vecinos correspondientes. Esta información podrá ser explotada tiempo después cuando se desea establecer una nueva comunicación con el nodo destino, permitiendo enviar una petición de ruta RREQ que se propague sólo "n" saltos del origen, que contendrá la dirección del destino. Los nodos que se encuentran a "n" saltos del origen y que reciban esta petición, revisarán en sus tablas de ruteo algún camino conocido a D o a alguno de sus vecinos, para después entregar esta información al nodo origen que emitirá paquetes unicast hacia los vecinos. Los vecinos al recibir este paquete unicast (paquete fireworks) iniciarán una nueva búsqueda teniendo como objetivo al nodo destino.

La idea principal de *Fireworks* es que existe una mayor probabilidad de encontrar a varios nodos (vecinos) en un flooding acotado que encontrar a un sólo nodo (destino).

También se definieron los encabezados propios de este protocolo y el tipo de información portada por cada uno de los campos de control y de datos.

V. Implementación de *Fireworks* en el simulador Network Simulator 2 (NS2)

5.1 Generalidades del Network Simulator 2

El Network Simulator 2 es un simulador de eventos discretos orientado a la investigación de redes de computadoras. Fue creado por la Universidad del Sur de California como una variante del "Real Network Simulator" en 1989 y ha evolucionado substancialmente en los últimos años. El desarrollo de NS en 1995 fue apoyado por DARPA (Defense Advanced Research Projects Agency) a través del proyecto VINT (Virtual InterNetwork Testbed) que conforman LBL, Xerox PARC, UCB, y USC/ISI.

El simulador se enfoca en modelar diversos protocolos de red correspondientes a cada una de las capas del modelo TCP/IP (aplicación, presentación, sesión, transporte, red, enlace de datos y física) para redes guiadas e inalámbricas, incluyendo modelos que soportan simulaciones de redes del tipo *ad hoc*, redes satelitales, modelos de error para redes guiadas, entre otras características que permiten la generación de estadísticas para su posterior análisis.

NS2 es utilizado principalmente para evaluar el desempeño de protocolos de red existentes y para la implementación y prueba de nuevos protocolos propuestos alrededor del mundo, así como la simulación de escenarios a gran escala no posible en experimentos reales. Debido a su ejecución de eventos, toma cada uno de ellos y los procesa de manera secuencial (ejecuta el siguiente evento hasta que el anterior haya finalizado).

5.1.1 Funcionamiento y estructura

NS[35] es un simulador orientado a objetos, escrito en C++ con un intérprete OTcl como interfaz de usuario. El simulador soporta una jerarquía de clases en C++ y una jerarquía de clases similar dentro del intérprete OTcl. Las dos jerarquías de clases están fuertemente relacionadas entre sí. La raíz de esta jerarquía es la clase TclObject. Los usuarios crean nuevos objetos de simulación a través del intérprete, los cuales son definidos dentro del mismo y son fuertemente relacionados con un objeto correspondiente en la jerarquía compilada (jerarquía en C++). En otras palabras, lo declarado en el intérprete OTcl se lleva a las clases correspondientes en el código en C++ para su posterior compilación.

NS utiliza dos lenguajes de programación debido a que el simulador tiene dos tipos diferentes de actividades a realizar. Por un lado las simulaciones detalladas de protocolos requieren un lenguaje de programación que pueda manipular eficientemente los paquetes a nivel de bytes, encabezados de paquetes e implemente algoritmos que corran sobre grandes conjuntos de información. Para estas tareas es importante el tiempo de ejecución y el tiempo de *turn-around* (tiempo de simulación, encontrar bugs, corregirlos, recompilar y volver a correr) es menos importante.

Por otro lado, una parte importante de la investigación sobre redes envuelve parámetros levemente variables o configuraciones, o rápidamente la exploración de un número de escenarios. En estos casos, el tiempo de iteración (cambiar el modelo y volverlo a correr) es más importante. Puesto que la configuración se corre una vez (al principio de la simulación), el tiempo de arranque de esta parte de la tarea es menos importante.

NS satisface ambas necesidades con 2 lenguajes, C++ y OTcl. C++ es rápido de ejecutar pero lento para cambiarlo, haciéndolo adecuado para implementaciones detalladas de protocolos. OTcl corre mucho más lento pero puede modificarse de manera más rápida (y de forma iterativa), haciéndolo ideal para configuraciones.

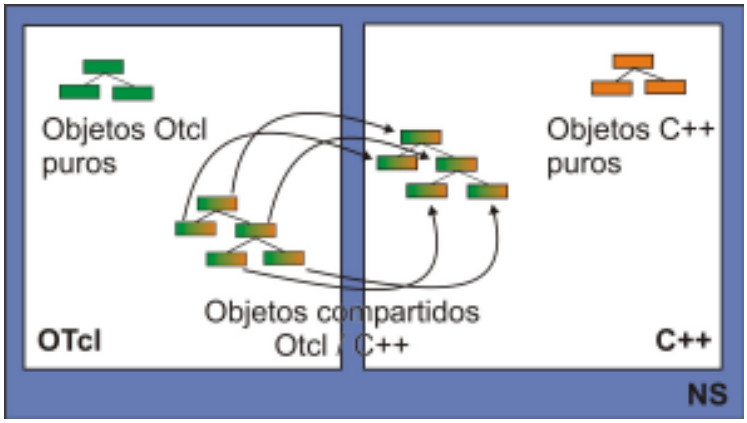

Figura 5.1.- Estructura dual del simulador NS. Se utiliza OTcl para la configuración (asignación de valores a las variables) y C++ para obtener los resultados.

Para realizar las simulaciones, el usuario debe definir una topología de red (definiendo el número de nodos a utilizar, tipo de enlace y área de trabajo), protocolos de encaminamiento a utilizar y finalmente los nombres de archivo para el trazado y la animación de salida.

Después de ejecutada la simulación, se generan los archivos de trazado (que debe de tener una extensión *.tr) y el archivo de animación. En el archivo de trazado se almacena la información en forma de datos discretos de los eventos que ocurren a cada paso de tiempo, desde movimiento de los nodos, como el inicio, proceso y fin de una transferencia de paquetes entre uno o más nodos. La información de trazado que se almacena en el archivo, depende del tipo de salida seleccionada por el usuario (información de la subcapa MAC, información particular de un protocolo de enrutamiento, etc.).

5.1.2 La herramienta de animación (NAM)

El Network Animator (NAM)[36] es una herramienta de animación basada en Tcl / Tk para la observación del desempeño de la simulación de red y la información generada en tiempo real referente al intercambio de paquetes de datos.

El primer paso para utilizar NAM es generar el archivo de trazado, el cual deberá contener información topológica (por ejemplo nodos, enlaces, así como información de paquetes). Usualmente el archivo de trazado es generado por NS. Durante una simulación en NS, el usuario puede indicar configuraciones en la topología, información de la disposición de nodos y trazado de paquetes utilizando eventos del simulador.

Cuando se genera el archivo de trazado se encuentra listo para ser animado por NAM. En el inicio, NAM leerá el archivo de trazado, creará una topología, abrirá una ventana, decidirá la disposición de nodos si es necesario y hará una pausa en el tiempo en el primer paquete indicado por el archivo de trazado. A través de su interfaz de usuario, NAM provee control sobre varios aspectos de la animación.

5.1.3 Simulando redes inalámbricas con nodos móviles

Cuando se trabaja con redes inalámbricas con nodos móviles, la declaración de la topología cambia primeramente en lo que respecta a los cambios de posición de los nodos, después al canal a utilizar y

finalmente el establecimiento de las conexiones (que solo varía en la forma que se declara el canal de comunicaciones que será el espacio libre

Para el caso de una red inalámbrica del tipo infraestructura y *ad-hoc*, se debe especificar el uso del canal inalámbrico y se puede considerar algún modelo de propagación. Los modelos de propagación se utilizan para predecir la potencia de la señal recibida para cada paquete. En la capa física de cada nodo inalámbrico, está especificado un umbral de recepción; cuando se recibe un paquete si la potencia de la señal es menor que el umbral de recepción, se marca como error y se tira de parte de la capa MAC. En NS se manejan 3 modelos de propagación: propagación en el espacio libre, propagación de dos rayos y modelo de propagación de desvanecimientos lentos (*shadowing*).

5.1.3.1 Estructura del nodo inalámbrico móvil

A manera de ejemplo, la estructura de nodo móvil para el protocolo de ruteo DSR (mismo que vamos a utilizar para este trabajo), está definido dentro de la clase SRNode que se deriva a su vez de la clase MobileNode, que es donde se definen los nodos de este tipo. SRNode (que significa nodo para ruteo de fuente – *source routing*) no utiliza un demultiplexor de direcciones o clasificadores y todos los paquetes recibidos por el nodo se envían al agente de encaminamiento DSR por default. El agente de ruteo DSR recibe paquetes para manejarlos a través del demultiplexor de puertos y decidir relevar paquetes, enviar peticiones de ruta o respuestas de rutas para paquetes frescos. El modelo de nodo móvil para DSR se muestra en la figura 5.4:

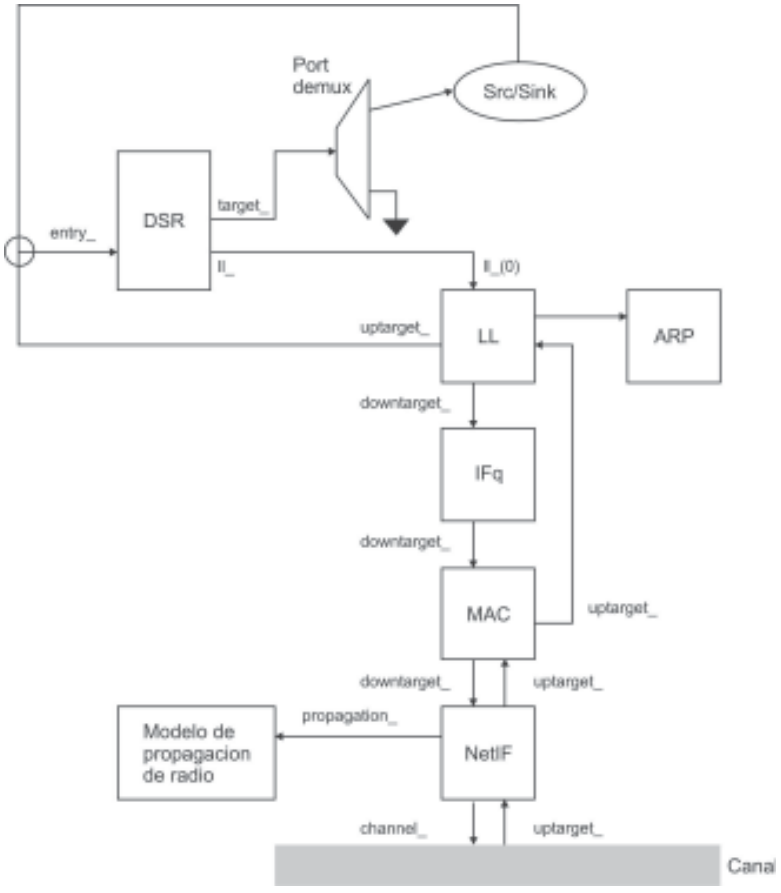

Figura 5.4.- Esquema del nodo móvil SRNode

Los elementos descritos en el esquema son los siguientes:

a) **Capa de enlace (*Link Layer - LL*):** La capa de enlace especificada para el nodo móvil, varia respecto a aquella para un nodo fijo en que posee un módulo de ARP (*Address Resolution Protocol*) conectado, el cual resuelve todas las conversiones de direcciones IP a direcciones de hardware (MAC). Normalmente para todos los paquetes salientes (hacia el canal), los paquetes son enviados hacia la capa LL por el agente de ruteo. Para los paquetes (afuera del canal), la capa MAC sube los paquetes a la LL para luego enviarse al punto de entrada de paquetes (node_entry_point).

b) **ARP:** El módulo del protocolo de resolución de direcciones recibe peticiones de la capa de enlace. Si ARP conoce la dirección de hardware para un destino, la escribirá en el encabezado MAC del paquete. De otra forma transmite en forma de *broadcast* una petición ARP y almacena temporalmente el paquete.

c) **Cola de interfaz (*Interface Queue-IFq*):** Se implementa como una cola de prioridad la cual da prioridad a los paquetes de protocolo de ruteo, insertándolos a la cabeza de la misma. Soporta un filtro sobre todos los paquetes en la cola que remueve aquellos con una dirección destino específica.

d) **Capa MAC:** Utiliza un patrón RTS/CTS/DATA/ACK para todos los paquetes *unicast* y simplemente manda datos (DATA) para todos los paquetes de *broadcast*. La implementación utiliza detección física y virtual de la portadora.

e) **Agentes Tap:** Si el protocolo MAC particular lo permite, se llevarán todos los paquetes al tap recibidos por la capa MAC, antes de que se realice el filtrado de direcciones.

f) **Interfaces de red(NetIF):** Sirve como interfaz de hardware la cual es utilizado por el nodo móvil para acceder al canal. Esta interfaz se encuentra propensa a sufrir colisiones y el modelo de propagación de radio recibe los paquetes transmitidos por otro nodo hacia el canal. La interfaz etiqueta cada paquete transmitido con la información relacionada a la transmisión, como la potencia, longitud de onda, etc.

g) **Modelo de propagación:** Utiliza atenuación del modelo de Friss ($1/r^2$) a distancias cortas y una aproximación al modelo de los 2 rayos para distancias largas ($1/r^4$).

h) **Antena y canal:** Se utiliza una antena omnidireccional con ganancia unitaria.

5.1.3.2 Generación de escenarios para nodos móvile

Al igual en el caso para redes guiadas, es necesario declarar la posición inicial de los nodos[37], pero con la particularidad de que podemos moverlos dentro del área de la red. El movimiento se define con el comando "setdest", indicando posición final a la que se quiere mover el nodo y la velocidad de movimiento (en metros por segundo), como se muestra a continuación.

```
$node_(0) set X_ 5.0
$node_(0) set Y_ 255.0
$node_(0) set Z_ 0.0

$node_(1) set X_ 205.0
$node_(1) set Y_ 255.0
$node_(1) set Z_ 0.0

$ns_ at 1.0 "$node_(0) setdest 50.5 25.0 10.0"
$ns_ at 5.0 "$node_(1) setdest 205.5 0.001 12.5"
```

en el bloque anterior se define para el nodo 0 la posición inicial (5, 255, 0) y en el segundo 1 este nodo se desplazará a la posición (50.5, 25, 0) a una velocidad de 10m/s.

Un caso especial resulta cuando se desea generar una cantidad grande de nodos, cada uno con su propio patrón de movimiento. Resultaría tedioso el generar una línea por cada movimiento que cada nodo deba realizar. Además si se desea simular un ambiente parecido a la vida real, donde los nodos sigan un patrón de movimiento aleatorio, se debe de utilizar una herramienta que pueda generar todo un escenario con el número de nodos que desee el usuario y con patrones de movimiento aleatorios. Es por eso que NS tiene un generador de escenarios llamado "setdest". Este generador se invoca de la siguiente manera:

./setdest [-n] [-p] [-s] [-t] [-x] [-y] > archivo

donde
n = número de nodos
p = tiempo de pausa en posición 1
s = velocidad (m/s)
t = tiempo de simulación
x = longitud del área de red en el eje x
y = longitud del área de red en el eje y

El resultado en el archivo que se genera con esta herramienta, es el equivalente a una serie de líneas sucesivas como las que se muestra arriba, pero con la peculiaridad de que las posiciones de los nodos se generan de acuerdo a una distribución uniforme y el movimiento sigue el siguiente patrón:

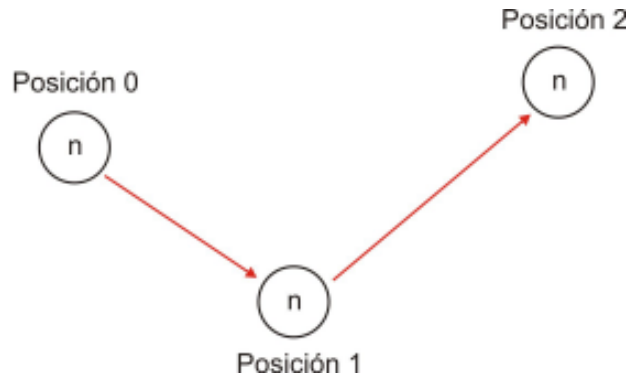

Figura 5.5.- Patrón de movimiento generado por setdest

Respecto a la figura 5.5, el nodo "n" parte de su posición inicial (posición 0) y se mueve rumbo a la posición 1, donde espera "p" segundos (indicados en el enunciado de llamado a la función) y se mueve a la posición final (posición 1) a "s" metros por segundo. Finalmente podemos llamar el archivo generado mediante la siguiente instrucción dentro del script Tcl:

```
set val(sc)          "[ruta de archivo]"
source $val(sc)
```

5.1.3.3 Generación de conexiones entre nodos móviles

Una vez que se ha generado el archivo de escenario con los nodos participantes generados en él, podemos establecer conexiones entre ellos. Una conexión puede ser del tipo TCP o CBR/UDP,

dependiendo del tipo de aplicación que genere ese tráfico y se declara de la siguiente manera:

a) Para el caso de una conexión para una aplicación donde se requiere una conexión orientada a la conexión (por ejemplo FTP), se requiere que a nivel de transporte se utilice TCP, que ofrece una comunicación confiable (debido a que se requiere de la confirmación de cada segmento enviado). En NS se declara de la siguiente manera en el script Tcl:

```
# Conexión TCP entre nodo (0) y nodo (1)

set tcp [new Agent/TCP]
$tcp set class_ 2
set sink [new Agent/TCPSink]
$ns_ attach-agent $node_(0) $tcp
$ns_ attach-agent $node_(1) $sink
$ns_ connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns_ at 10.0 "$ftp start"
```

En la declaración se define al nodo iniciador de la conexión (TCP) y al nodo destinatario de la comunicación (Sink). De igual manera se especifica el tipo de aplicación que portará la conexión y el tiempo de inicio de la transferencia.

b) Si se desea utilizar a nivel de transporte una comunicación no orientada a conexión se utiliza UDP, que tiene la peculiaridad de ser más rápido que TCP, pero no es tan confiable. En NS se define una conexión CBR (Constant Bit Rate) bajo UDP de la siguiente manera:

```
#Definimos una conexión CBR/UDP entre nodo (0) y nodo (1)

set udp_(0) [new Agent/UDP]
$ns_ attach-agent $node_(0) $udp_(0)

set null_(0) [new Agent/Null]
$ns_ attach-agent $node_(1) $null_(0)

set cbr_(0) [new Application/Traffic/CBR]
$cbr_(0) set packetSize_ 100
$cbr_(0) set interval_ 1
$cbr_(0) set maxpkts_ 1000
$cbr_(0) attach-agent $udp_(0)

$ns_ connect $udp_(0) $null_(0)
$ns_ at 100.0 "$cbr_(0) start"
$ns_ at 120.0 "$cbr_(0) stop"
```

En la declaración se define al nodo origen (UDP) y al nodo destino (null), así como el tamaño del paquete, el tiempo entre paquetes generados (interval = 1 / tasa de bit), el número máximo de paquetes a transmitir y los tiempos de inicio y término de la comunicación.

5.2 Implementación del protocolo Fireworks

Para la implementación de nuestro protocolo de búsqueda de usuarios denominado previamente Fireworks, tomamos como base el código incluido en el Network Simulator para el Ruteo de Fuente Dinámico (Dynamic Source Routing – DSR) debido a las características que posee este protocolo de enrutamiento para redes ad-hoc, mencionadas previamente en este trabajo. Entre estas características sobresalen que el ruteo se realiza en base a un nodo fuente (origen), la simplicidad de sus mecanismos de descubrimiento y mantenimiento de ruta y principalmente el hecho de que los nodos intermedios pueden almacenar la información de ruta contenida en los paquetes que retransmite y las comunicaciones que "escucha" a su alrededor, generando una cantidad información de rutas valiosa para su explotación.

De forma general el agente de ruteo DSR de NS, recibe un paquete a partir de dos fuentes posibles: capas subyacentes y la generación de una conexión nueva por capas superiores. En el caso de una nueva conexión, el agente de encaminamiento recibe el paquete para un nodo origen en particular y lo almacena en el buffer de envío (*send buffer*). Una vez almacenado se revisa dentro de la tabla de rutas del nodo si se cuenta con una ruta para poder enviar el paquete; de contar con la ruta se envía inmediatamente el paquete, de lo contrario se procede a generar una búsqueda de ruta (*route request*).

El nodo entonces envía un paquete de petición de ruta, el cual es propagado por toda la red transfiriéndose de un nodo a otro hasta alcanzar el destino. El destino a su vez responderá con una respuesta de ruta (route reply) que se propagará de regreso por el mismo camino por donde viajó la petición. Un dato importante es que durante el intercambio de información que ocurre en todo este proceso, cada nodo aprende de la información de ruta portada en cada paquete, actualizando su propia tabla de ruteo y enriqueciéndola con nuevas rutas.

El objetivo de Fireworks es explotar toda la información de ruteo generada por DSR para optimizar el área de la red que debe de ser afectada por el *flooding* y realizar un proceso de búsqueda de usuarios en la red.

A continuación se presentan los mecanismos medulares en el desempeño de Fireworks y su implementación sobre el código de DSR provisto por NS.

5.2.1 Generación de paquetes de búsqueda *unicast* y su almacenamiento en el buffer de salida – función recv( )

Dentro del código que describe las funciones del agente de ruteo DSR (dsragent.cc), existe una función denominada "recv ()" que es el punto donde DSR recibe el paquete proveniente de una aplicación superior o bien de parte de otro nodo en la red. Aquí es donde se toma la decisión si se debe de generar una petición de ruta o bien simplemente pasar el paquete al siguiente nodo en la ruta.

En base a lo que se estipuló para Fireworks en el capítulo anterior, el nodo origen de la conexión conoce quiénes eran los vecinos del nodo destino a partir de un intercambio de esta información en alguna conexión previa. En base a esto, se preparan "n" paquetes, uno por cada vecino almacenado en la tabla de vecinos estampados con las direcciones IP origen y destino correspondientes (unicast). Estos paquetes se apilan uno tras de otro dentro del buffer, el cual es revisado periódicamente por el agente de encaminamiento para enviarlos o bien buscarles una ruta en un momento dado.

A continuación se presenta la porción de código que se encarga de hacer esto, donde el nodo origen es el 699 y el destino el 366:

```
DSRAgent::recv(paquete)
/* Recibe paquete originado por una aplicación o por otro nodo y crea paquete para los "n" vecinos*/
{
```

```
        if (paquete.origen = nodo origen && paquete.destino = target)
        {
              for(contador = 0; contador < n); incrementa contador)
              {
                    unicast ————————————> se crea nuevo paquete
                    unicast.destino = vecino
                    unicast.origen = nodo origen
                    unicast.tipo = CBR
                    colocamos el paquete en el sendbuffer
                }
        }
}
```

Lo anterior, junto con las funciones nativas de DSR, trabaja como se muestra en el siguiente diagrama de flujo:

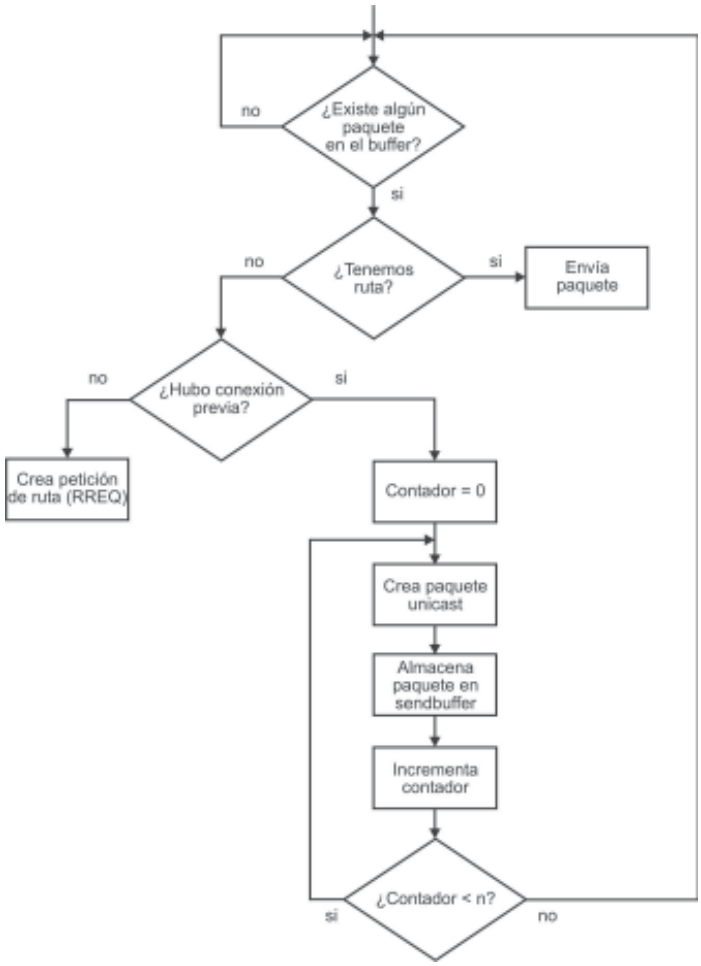

Figura 5.6.- Diagrama de flujo de la función recv ( ) modificada para Fireworks

5.2.2 Búsqueda acotada – función sendOutRtReq()

Para poder enviar la petición de ruta (route request), DSR utiliza la función sendOutRtReq( ), misma que se encarga de agregarle los encabezados correspondientes al paquete (entre ellos el número de saltos máximo de propagación y el bit bandera que lo identifica como route request). Para el caso del

paquete de búsqueda acotada de Fireworks, el número de saltos máximo de propagación puede modificarse, de acuerdo a cada implementación que se requiera. Lo anterior se logra con la siguiente función:

```
void
DSRAgent::sendOutRtReq(Paquete)
/* Revisa si el paquete es del tipo Fireworks y determina el número de saltos máximo que debe de propagarse */
{
        if (paquete = paquete Firewors)
          {
                paquete.saltos_propagacion = n  //se propaga n saltos máximo
          }
          else
          {
                paquete.saltos_propagacion = maximo //el máximo de DSR = 16 saltos
          }
}
```

El diagrama de flujo de esta función es el siguiente:

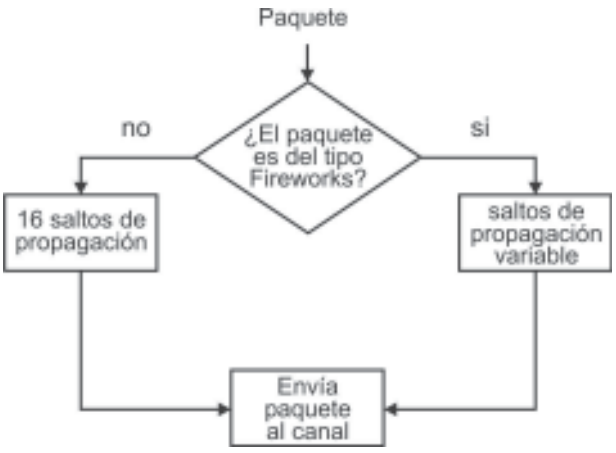

Figura 5.7.- Diagrama de flujo de la función sendOutRtReq( ) modificada para Fireworks

5.2.3 Recepción del paquete unicast – function handleRouteRequest( )

En la petición de ruta enviada por el origen, la cual contiene información del nodo origen y destino, se incluirá también una lista de los nodos vecinos del nodo destino, con la finalidad de que se pueda encontrar al destino ya sea directamente o bien mediante la localización de alguno de los vecinos en las cercanías.

Todos los nodos que reciban la petición de ruta a los "n" saltos especificados a partir del origen, abrirán el paquete y obtendrán la lista de vecinos del nodo destino. Posteriormente revisarán en su tabla de encaminamiento si se cuenta con una ruta fresca ya sea para el destino directamente o bien para alguno de sus vecinos y procederán a responder con una respuesta de ruta de regreso al nodo origen.

Estas respuestas de ruta serán recibidas por el nodo origen quien irá transmitiendo el paquete unicast de búsqueda correspondiente, uno por cada ruta recibida a cada vecino individual. Cabe señalar que el nodo origen responderá sólo a la primera confirmación de ruta a cada destino, reduciendo así la posibilidad de la generación de tormentas de paquetes y por ende una mayor congestión de la red. Lo anterior se logra mediante el siguiente bloque de código:

```
void
DSRAgent::handleRouteRequest(paquete)
    /* Si es un paquete Fireworks se revisa si se tiene ruta al destino o bien a cada uno de los vecinos incluidos en la lista */
{

        if (paquete.origen = origen && paquete.destino = target)
        {
                //Revisamos si tenemos ruta a alguno de los  "n" vecinos
                for (contador = 0; contador < n ; incrementa contador)
                {
                        if (tenemos ruta para vecino[contador])
                                contestamos con una confirmación de ruta
                        else
                        continua el ciclo
                    }

}
```

La función, junto con las modificaciones para Fireworks, siguen el siguiente diagrama de flujo:

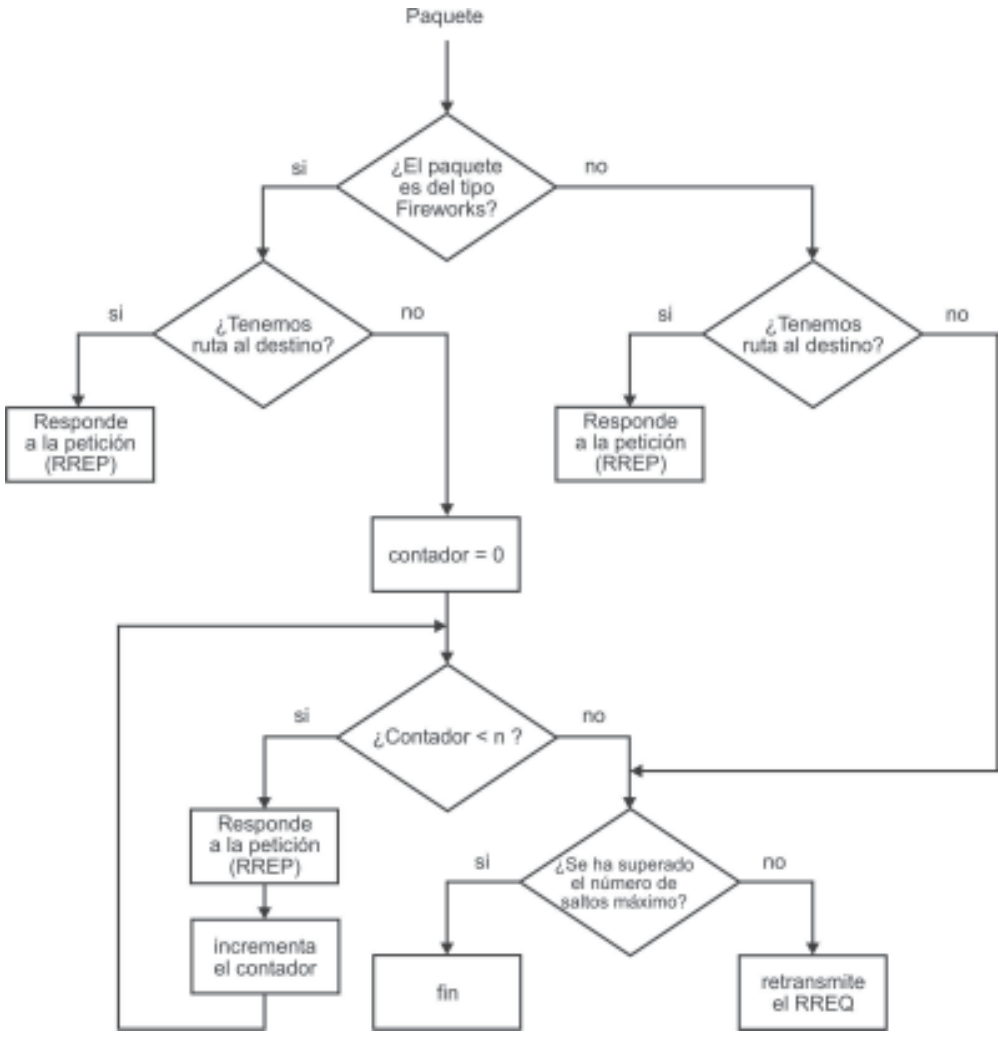

Figura 5.8.- Diagrama de flujo de la función handleRouteRequest( ) modificada para Fireworks

5.2.4 Mecanismo de búsqueda Fireworks – función handlePacketReceipt( )

Las modificaciones a esta función se encarga de que cuando un nodo registrado como vecino del nodo destino reciba un paquete *unicast* de búsqueda proveniente del nodo origen, procederá a elaborar un nuevo paquete de petición de ruta (route request) pero colocando la dirección del nodo destino y como nodo origen la propia.

Como esta función la utilizan todos los nodos cuando reciben un paquete, se define una condición donde se revisa que se inicie el mecanismo de búsqueda Fireworks sólo si el nodo que lo recibe es un vecino.

Dentro de esta función se puede especificar también el número de saltos máximo que queremos que se propague la nueva búsqueda, de acuerdo con las necesidades y circunstancias de la red (tales como movilidad por ejemplo). A continuación de presenta el bloque de código que realiza esta operación:

```
void
DSRAgent::handlePacketReceipt(Unicast)
/*Revisa el paquete unicast recibido por un vecino y emite la búsqueda Fireworks*/
{
        if (paquete.origen = origen && paquete.actual = vecino)
          {
                //Creamos el paquete de búsqueda (nueva petición de ruta) Fireworks
                fireworks.origen = actual;
                fireworks.destino = target;
                fireworks.saltos_propagación = 0; // se define saltos de propagación
          }

}
```

El diagrama de flujo correspondiente es el siguiente:

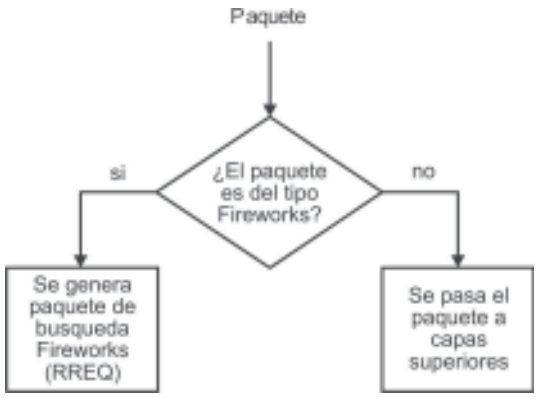

Figura 5.9.- Diagrama de flujo de la función handlePacketReceipt( ) modificada para Fireworks

En el presente capítulo se estudió inicialmente el funcionamiento del simulador NS2 (*Network Simulator 2*). Se analizó su estructura basada en C++ y OTcl, para hacerlo amigable al usuario, así como robusto y rápido para ejectuar las operaciones. Se revisó también como declarar escenarios para redes inalámbricas y modelos de movilidad. Finalmente se definió el pseudocódigo que rige el funciomamiento de *Firework*s y su implementación sobre DSR.

VI. Resultados y Discusión

6.1 Esquema de simulación

Una vez definidas las funciones que conforman a *Fireworks*, corresponde ahora establecer un escenario de simulación para observar su comportamiento dinámico y acotar su aplicación como un protocolo de encaminamiento eficiente para localizar nodos en una red *ad hoc*.

Al definir un escenario para simular una red *ad hoc* en NS2, los factores que determinan el funcionamiento de la red son:

1) Densidad de nodos en la red: Definida a partir de las dimensiones (área) de la red y número de nodos definidos en el escenario
2) Movilidad de los nodos: Velocidad a la que se mueven los nodos dentro del escenario durante el tiempo de simulación.
3) Tasa de transferencia de datos: Definida por el tamaño del paquete (en bytes) y la tasa a la que cada nodo transmitirá sus paquetes de datos y control (definido en paquetes por segundo)
4) Longitud máxima de rutas: De acuerdo a las dimensiones del escenario y la densidad de nodos definida, se tendrá un número de saltos máximo para las rutas creadas entre los nodos origen y destino seleccionados.

El análisis de los parámetros arriba señalados, ayudará a la definición de un escenario ideal para el funcionamiento eficiente del protocolo *Fireworks*, reconociendo sus ventajas y limitaciones para diferentes casos que se obtendrán al variar alguno o varios de los parámetros por esquema de simulación.

De acuerdo a las condiciones de cada escenario, se espera que *Fireworks* presente una reducción significativa de la cantidad de paquetes de *flooding* necesarios para encontrar rutas al nodo destino, mediante la explotación de la información de ruta generada durante el transcurso de las comunicaciones en una red sin infraestructura.

Por otro lado, los principales retos que enfrenta *Fireworks* son la longitud de las rutas y su respuesta a la movilidad y el tráfico generado por los nodos transmisores / receptores operando en vecindad a los nodos origen y destino.

En base a lo anterior, se correrán una serie de escenarios variando para cada caso:

a) Tasa de transferencia de datos: 1pkt/8s, 1pkt/s y 8 pkt/s, tomando como tamaño de paquete 100 bytes
b) Diferentes velocidades de movimiento: 0, 2 y 10 m/s
c) Tamaño de escenario: Pequeño (1400x1000 metros, máximo 7 saltos por ruta), Mediano (2200x1200 metros, máximo 15 saltos por ruta) y Grande (3400x3100 metros, máximo 19 saltospor ruta)

Y se contabilizarán en cada escenario:

1) Número promedio de rutas encontradas hacia alguno de los vecinos
2) Número de nodos vecinos del nodo destino alcanzados por el paquete de búsqueda *Fireworks*

3) Número de búsquedas exitosas
4) Número de paquetes de *flooding* generado y su comparación con el número de paquetes de *flooding* generados en el mismo escenario por DSR, únicamente para la conexión 11, 31, 61 y 81.

Se correrán en total 8 escenarios con diferentes patrones de disposición de nodos y patrones de movimiento, para después obtener el promedio de los parámetros a contabilizar, equivalente a correr pruebas de MonteCarlo.

En cada simulación, se dejarán correr 10, 30, 60 y 80 conexiones creadas de forma aleatoria usando *flooding* DSR a 1pkt/8s, 1pkt/s y 8pkt/s durante 80 segundos, para posteriormente iniciar la búsqueda acotada *Fireworks* y así encontrar una ruta al nodo destino. Este esquema de simulación se planteó a partir de una serie de pruebas iniciales, donde se observó que era necesario la creación de tráfico aleatorio para la recopilación de suficiente información de rutas en el escenario. Además las pruebas iniciales permitieron observar que tasa de transferencia del tráfico juega un papel importante en la entrega exitosa de los paquetes de búsqueda *Fireworks*, al observar que conforme se incrementaban las conexiones, la tasa de transferencia y tamaño de paquete, se observaba la pérdida de algunos de los paquetes en el trayecto al destino.

6.2 Resultados

A continuación se presentan los resultados obtenidos durante las simulaciones correspondientes para cada tamaño de escenario para 0, 2 y 10 m/s para una tasa de transferencia de datos de 1 pkt / s.

6.2.1 Simulaciones para escenario pequeño (1400x1000m)

Para determinar el tamaño del escenario pequeño, se optó por colocar los nodos origen y destino de tal forma que las rutas posibles determinadas por DSR, tuvieran como máximo 7 saltos (es decir, hasta 7 nodos pueden participar en el relevo de paquetes entre el origen y el destino), tomando en cuenta que el

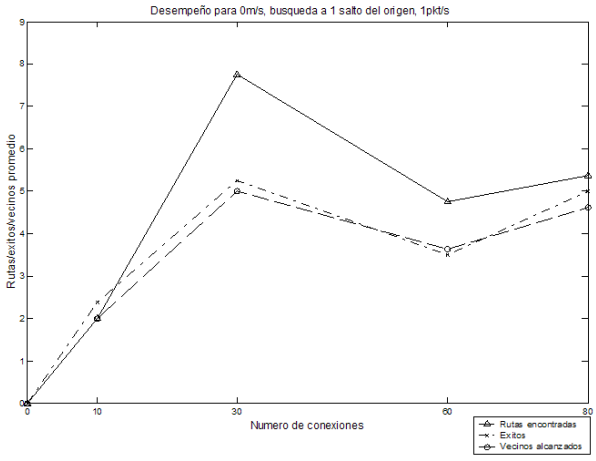

Figura 6.1.- Desempeño de Fireworks para 1pkt/s

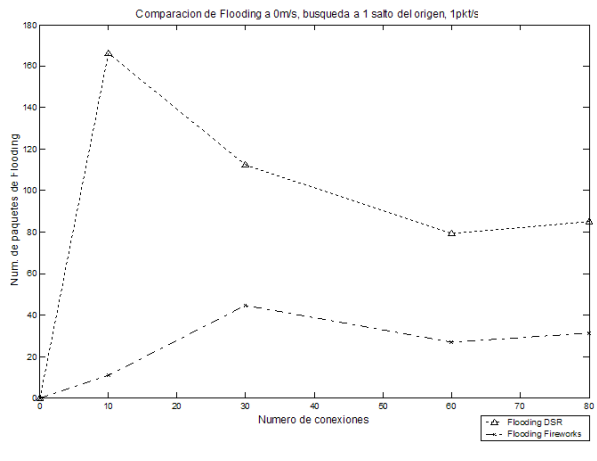

Figura 6.2.- Flooding generado por Fireworks y su comparación para el flooding generado por DSR para el mismo escenario para tráfico a 1pkt/s

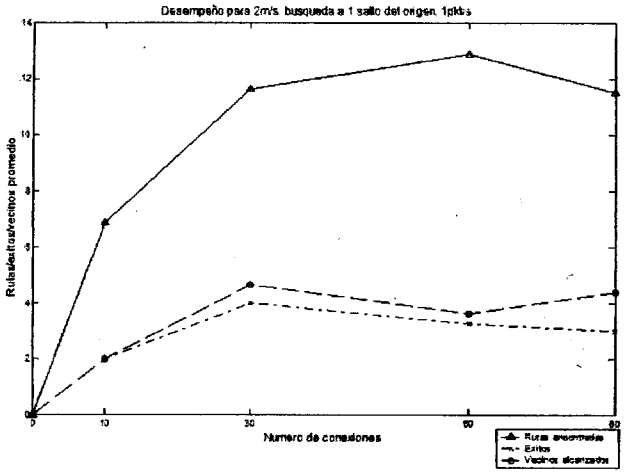

Figura 6.3.- Desempeño de Fireworks para 1pkt/s y 2m/s de movilidad

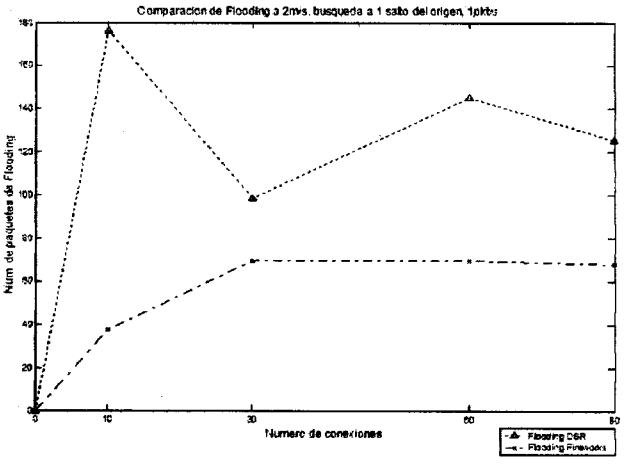

Figura 6.4.- Comparación de Flooding Fireworks y DSR para 1pkt/s y 2m/s de movilidad

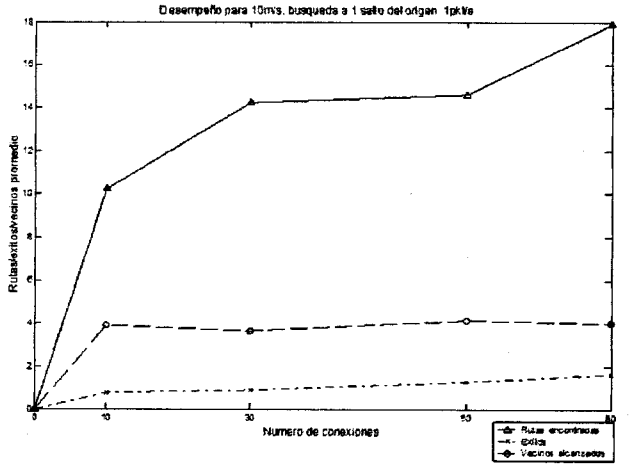

Figura 6.5.- Desempeño de Fireworks para 1pkt/s y 10m/s de movilidad

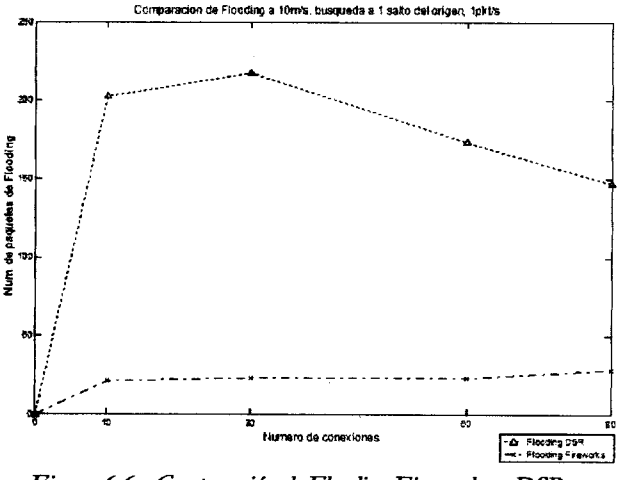

Figura 6.6.- Comparación de Flooding Fireworks y DSR para 1pkt/s y 10m/s de movilidad

número de saltos máximo considerado en el campo de TTL (*time to live*) para DSR es de 15. Los resultados obtenidos y las observaciones correspondientes se presentan a continuación.

6.2.1.1 Velocidad máxima 0m/s

A partir de los resultados obtenidos, observamos en la figura 6.1 que se obtienen hasta 8 rutas en promedio para 30 conexiones como máximo, mientras que para 60 y 80 conexiones el número de rutas encontradas disminuye. El máximo número de rutas encontradas se encuentra limitado en este caso al número de vecinos a un salto del nodo destino. Por otro lado observamos que el número promedio de éxitos alcanzados es inferior al total de rutas encontradas, dando a entender que del total de paquetes unicast que se envían para encontrar a alguno de los vecinos de los que se recuperaron rutas, algunos se pierden en el trayecto. Por ejemplo, para el caso de 30 conexiones para 8 escenarios diferentes simulados, en promedio el 21.13% de los paquetes unicast enviados por el nodo origen se pierden en el trayecto. Para el caso donde no se considera movilidad de los nodos, observamos que el factor que ocasiona la pérdida de paquetes unicast es el tráfico presente en la red.

En cuanto al *flooding* generado por *Fireworks*, observamos en la figura 6.2 que en promedio genera cuando mucho el 28% del *flooding* generado por DSR, considerando los valores máximos presentados por

DSR y *Fireworks* en esta prueba. Cabe mencionar que para el conteo del flooding se consideró únicamente la conexión adicional generada especialmente para *Fireworks* posterior a la generación y establecimiento de las conexiones 10, 30, 60 y 80.

6.2.1.2 Velocidad máxima 2m/s

Una vez analizado el caso para una red con nodos fijos, corresponde ahora hacer las mismas pruebas pero bajo un esquema de nodos móviles desplazándose hasta 2m/s de manera aleatoria.

Para el caso en que se considera una velocidad de desplazamiento máxima de 2m/s (figura 6.3), observamos que la diferencia entre rutas encontradas y éxitos conseguidos se hace mayor. De igual manera, en este caso observamos un incremento del número de vecinos alcanzados respecto de los éxitos, dando a entender que si bien se alcanzaron a cierto número de vecinos, algunos de ellos ya no encontraron al nodo destino haciendo la búsqueda Fireworks a 1 salto, esto debido a la movilidad de los nodos.

En lo que respecta al *flooding*, generado por *Fireworks*, éste no supero el 37% del flooding DSR en promedio, observando en este caso que para 30, 60 y 80 conexiones se mantiene relativamente constante, debido a que el número de rutas encontradas para el mismo número de conexiones no varía mucho, así como el número de vecinos encontrados. Para el flooding DSR observamos un decrecimiento de su valor conforme se incrementa el número de conexiones (figura 6.4), esto debido a la presencia de mecanismos de mantenimiento de ruta como el *gratuitious route reply* y por el constante intercambio de información de rutas entre los nodos, pudiéndose encontrar rutas sin necesidad de inundar toda la red.

6.2.1.3 Velocidad máxima 10m/s

Finalmente para este escenario, corresponde analizar con nodos móviles desplazándose 10m/s como velocidad máxima, corriendo las mismas pruebas que en los casos anteriores.

En este experimento se mantiene la tendencia a la alza del número de rutas encontradas conforme se incrementa el tráfico en la red (debido al incremento de la generación de información de ruta), pero su diferencia respecto al número de éxitos alcanzados y la correspondiente respecto a vecinos alcanzados se incrementa. Además el número de éxitos promedio decae de manera significativa a sus contrapartes a 0 y 2m/s, como se observa en la figura 6.5. Lo anterior sin duda se debe a que los nodos presentan una mayor movilidad y por tanto las rutas encontradas dejan de ser válidas en un menor tiempo.

En lo que respecta al *flooding* (figura 6.6), Fireworks presenta un número casi constante de paquetes de *flooding* generados, por debajo de los 25 paquetes (cerca del 12% del *flooding* generado por DSR). Lo anterior se debe primeramente a que solo un porcentaje de paquetes *unicast* alcanzan a los vecinos encontrados (22% en promedio), recorriendo los 7 saltos que tienen las rutas y el resto de esos paquetes se pierden en los primero saltos de las rutas que utilizan. Para el *flooding* DSR observamos en este caso que conforme se incrementa el tráfico este se reduce, debido a mecanismos como el *gratuitious reply* explicado en capítulos anteriores.

6.2.2 Simulaciones para escenario mediano (2200x1800)

Como siguiente paso en la evaluación del desempeño del protocolo *Fireworks*, se considera un escenario más grande de 220x1800m, donde se encontrarán distribuidos de manera aleatoria un total de 350 nodos, definido con el objetivo de que las rutas hacia el destino consistan de 16 saltos como máximo, considerando los casos sin movilidad y desplazamientos a 2 y 10m/s, para diferentes tasas de tráfico. Para observar el efecto del tráfico en el protocolo, se considera también el caso de 100 conexiones. Al igual que en el escenario pequeño, se colocaron los nodos origen y destino de tal forma que las rutas que se generen entre ellos tengan como máximo 16 saltos.

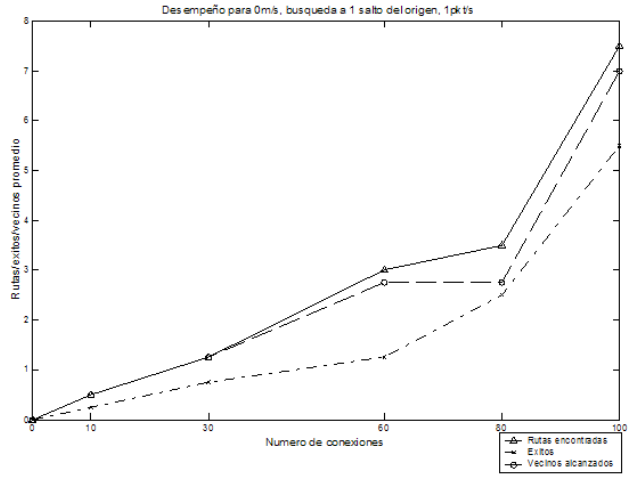

Figura 6.7.- Desempeño de Fireworks para 1pkt/s sin movilidad

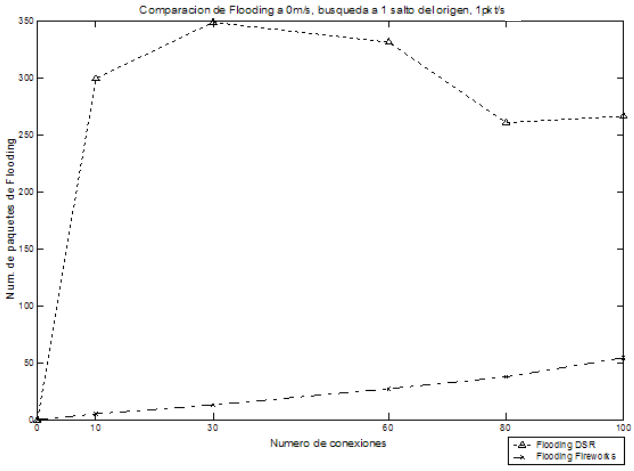

Figura 6.8.- Comparación de Flooding Fireworks y DSR para 1pkt/s y sin movilidad

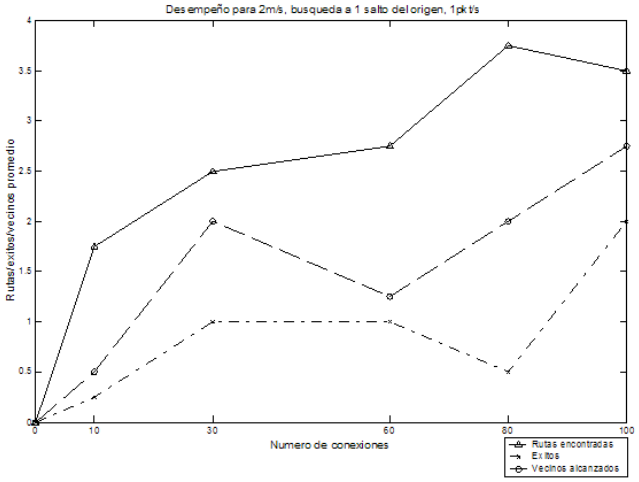

Figura 6.9.- Desempeño de Fireworks para 1pkt/s y 2m/s

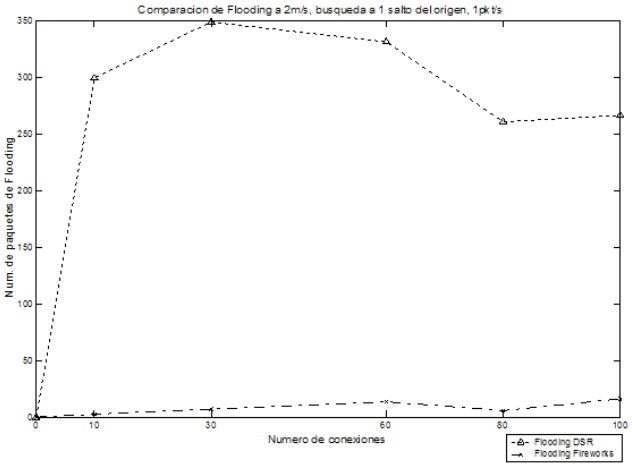

Figura 6.10.- Comparación de Flooding Fireworks y DSR para 1pkt/s y 2m/s

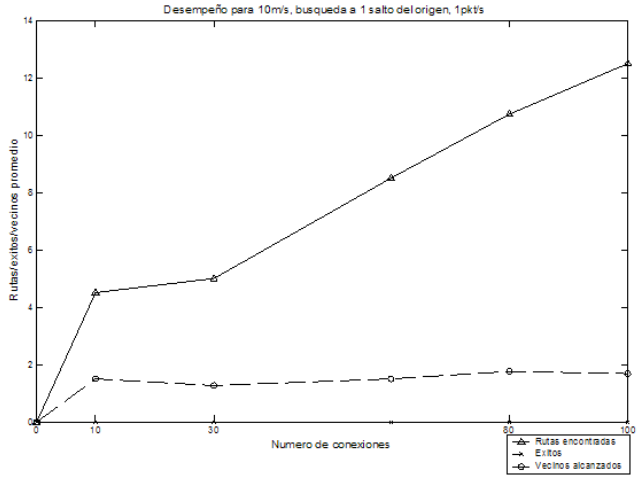

Figura 6.11.- Desempeño de Fireworks para 1pkt/s y 10m/s

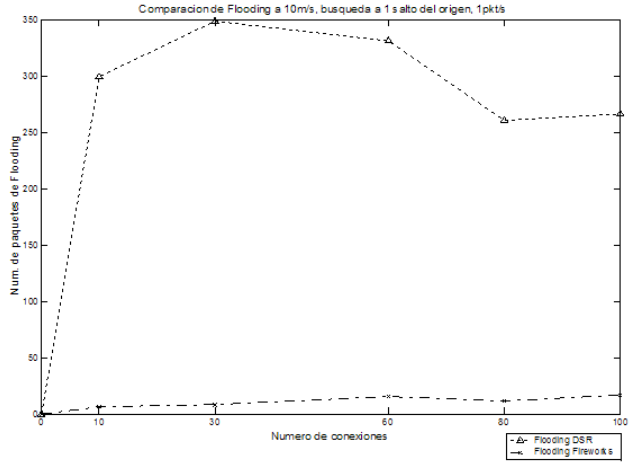

Figura 6.12.- Comparación de Flooding Fireworks y DSR para 1pkt/s y 10m/s

6.2.2.1 Velocidad máxima 0m/s

Observamos que para el escenario mediano sin movilidad, las curvas para rutas encontradas, vecinos encontrados y éxitos se incrementan conforme se incrementa el tráfico en la red, pero su promedio es menor que en el caso del escenario pequeño como observamos en la figura 6.7. Este comportamiento se debe al tamaño del escenario, donde las rutas son más largas y por lo tanto son válidas menos tiempo.

Si bien se encuentran menos rutas y se obtienen menor cantidad de éxitos que su contraparte del escenario pequeño como se observa en la figura 6.8, el *flooding Fireworks* respecto al DSR es todavía menor (16% en promedio respecto a DSR) y se incrementa levemente conforme se encuentren más rutas y por ende se envíen un mayor número de paquetes *unicast*.

6.2.2.2 Velocidad máxima 2m/s

Para una velocidad máxima de 2m/s, observamos en la figura 6.9 inicialmente un comportamiento creciente para las curvas (de manera particular para la de rutas encontradas), hasta 80 conexiones, para luego presentarse un acotamiento causado nuevamente por el número de vecinos a 1 salto en el nodo destino, y a la cantidad de rutas encontradas hacia ellos en una pequeña región del escenario (1 salto del nodo origen).

Para la comparación de *flooding* entre DSR y *Fireworks*, aunque el *flooding* del primero tiende a decrecer por sus mecanismos de compensación, su número es sumamente significativo respecto al generado por *Fireworks* (representando únicamente el 3.43% del ocasionado por DSR en promedio), ver figura 6.10.

6.2.2.2 Velocidad máxima 10m/s

Para el caso de movilidad a 10m/s, observamos en la figura 6.11 una tendencia claramente creciente en el promedio de rutas encontradas, no así en el caso de vecinos alcanzados y éxitos. Este cambio en el comportamiento se debe en gran medida al alto índice de movilidad de este escenario. Si bien se recupera gran cantidad de información de rutas debido a la constante caída y actualización de rutas, se incrementa el tráfico y las rutas duran menos tiempo, por lo que una buena cantidad de paquetes unicast se pierden (cerca del 86% en promedio) y aquellos que llegan a encontrar a un nodo vecino no terminan en una búsqueda exitosa.

El desempeño para estos niveles de movilidad, se puede mejorar levemente, si incrementamos el rango de búsqueda en el origen a 3 saltos por ejemplo (como se muestra en la figura 6.13), pero el impacto de la movilidad sigue siendo muy significativo respecto a los éxitos. Sin embargo, observamos que el *flooding* ocasionado por *Fireworks* se incrementa significativamente, por abarcar un área mayor en la búsqueda del nodo destino.

6.2.3 Simulaciones para escenario grande (3400x3100m)

Finalmente, para completar el estudio sobre *Fireworks* para diferentes tamaños de escenarios, se optó por simular en un escenario que excediera el número de saltos máximo de DSR. Para este caso se planteó un ambiente donde las rutas posibles entre origen y destino tienen como longitud máxima 19 saltos y con un total de 700 nodos en el mismo. Cabe mencionar que la prueba para 10m/s ya no se realizó debido a el bajo desempeño que presentó Fireworks para este tamaño de escenario y movilidad.Los resultados obtenidos son los siguientes:

6.2.3.1 Velocidad máxima 0m/s

Observamos que en el escenario grande, para el caso de nodos estáticos, el desempeño de *Fireworks* sigue siendo aceptable, al apreciarse un incremento en el número de rutas encontradas conforme se incrementa el número de conexiones (tráfico) en la red, así como para los éxitos logrados y los vecinos alcanzados, que en estas pruebas alcanzaron los mismos valores.

El *flooding* ocasionado por *Fireworks*, permanece menor al ocasionado por DSR representando el 17.16% del *flooding* ocasionado por este protocolo. Comparado con el porcentaje de *flooding* presentado en los escenarios pequeño y mediano, observamos que se mantiene significativamente menor comparado con el *flooding* DSR.

6.2.3.2 Velocidad máxima 2m/s

Como podemos observar en las gráficas, la presencia de movilidad afecta el desempeño de *Fireworks*. A pesar de que se encuentran rutas hacia los vecinos, pocos paquetes *unicast* logran alcanzarlos y encontrar posteriormente al destino. La principal causa de este comportamiento, es la longitud de las rutas (hasta 19 saltos) y el poco tiempo que permanecen válidas debido a la movilidad de los nodos.

El número de paquetes de *flooding* generados por *Fireworks* permanece inferior al *flooding* DSR, pero debido en gran parte a que del total de paquetes unicast enviados para alcanzar a un vecino, una buena parte se pierden en los primeros saltos del trayecto, debido al efecto de la movilidad y tráfico.

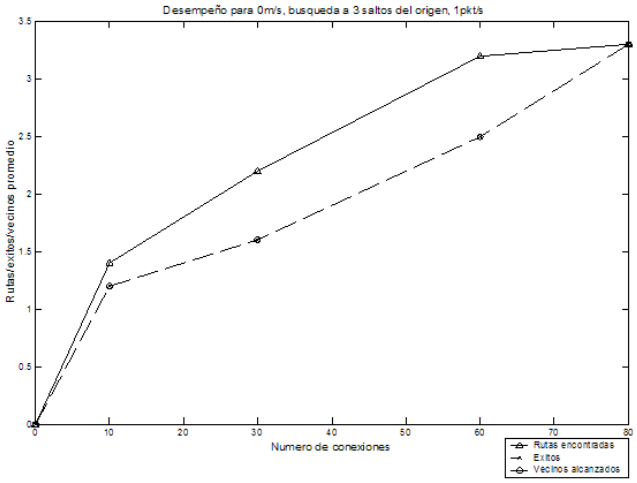

Figura 6.13.- Desempeño de Fireworks para 1pkt/s y 0m/s

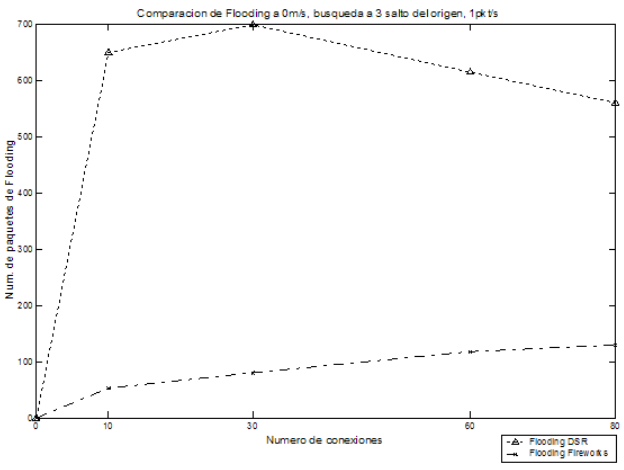

Figura 6.14.- Comparación de Flooding Fireworks y DSR para 1pkt/s y 0m/s

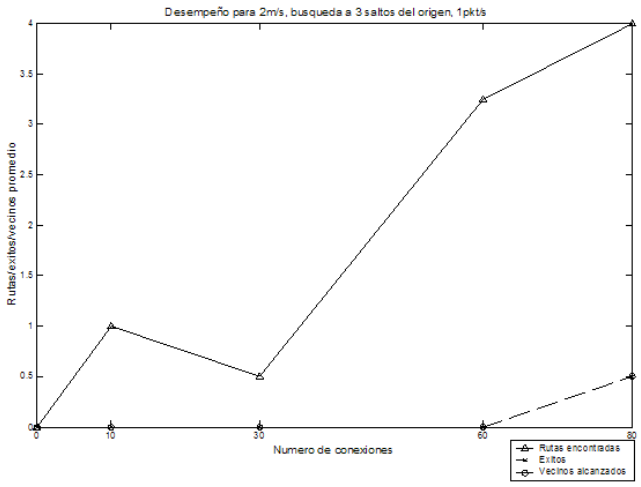

Figura 6.15.- Desempeño de Fireworks para 1pkt/s y 2m/s

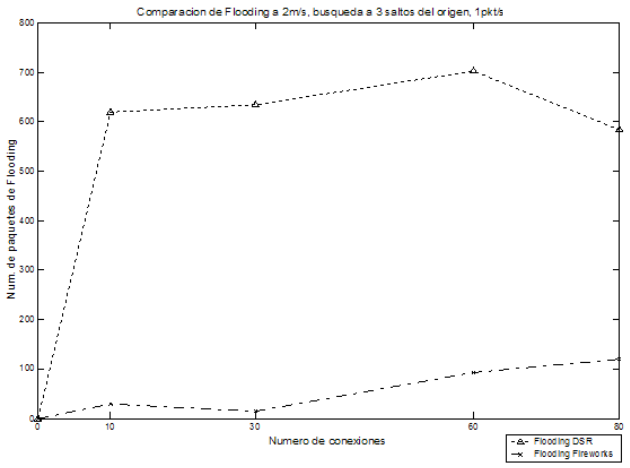

Figura 6.16.- Comparación de Flooding Fireworks y DSR para 1pkt/s y 2m/s

6.2.4 Discusión

Al observar el desempeño del protocolo *Fireworks* para el escenario pequeño, se observa que:

Conforme se incrementa el tráfico y la movilidad en el escenario, el número promedio de rutas encontradas aumenta, siendo acotado únicamente por el número de vecinos a 1 salto del destino conocidos por el nodo origen en su tabla de vecinos y la cantidad de rutas que se pueden recuperar para esos vecinos en la búsqueda acotada a un número reducido de saltos del origen.

El número promedio de vecinos alcanzados por los paquetes *unicast* presenta un incremento entre los casos de 10 y 30 conexiones presentes en el escenario, pero para mayor incremento del tráfico su valor se mantiene relativamente constante.

Por otro lado, bajo la presencia de movilidad de los nodos, el valor promedio de los vecinos alcanzados se reduce levemente, a pesar de haberse encontrado una mayor cantidad de rutas conforme se incrementa la velocidad de desplazamiento de los nodos. En este caso se observa que una cantidad de esas rutas encontradas dejan de ser válidas en poco tiempo debido a la movilidad.

El crecimiento del número promedio de éxitos (localización exitosa del nodo destino) se encuentra acotado por el incremento del tráfico en la red. Además se observó claramente que la movilidad junto con el tráfico, son responsables de la pérdida de paquetes *unicast* a nodos vecinos.

Debido al comportamiento de *Fireworks* en el escenario mediano, se observa que:

Para el caso del número promedio de rutas encontradas, observamos un claro comportamiento incremental en su valor conforme se incrementa el tráfico y la movilidad. Sin embargo, del total de rutas encontradas solo una parte ellas son válidas, como se puede concluir al observar el decremento de los valores promedio de vecinos y éxitos alcanzados conforme se incrementa la movilidad.

Se observa que el incremento de la longitud de las rutas junto con la movilidad, impacta el desempeño de *Fireworks* para el caso de este escenario mediano, al reducirse se manera importante el número de éxitos, llegando inclusive a cero para el caso de movilidad a 10m/s.

El incremento del área de búsqueda a partir del nodo origen mejora el desempeño de *Fireworks* pero a costa del incremento del número de paquetes de *flooding* y por ende del consumo de ancho de banda.

Fireworks en el escenario grande, presentó el siguiente comportamiento:

El incremento del tamaño del escenario y por ende de las rutas, aunado a una mayor velocidad de desplazamiento de los nodos involucrados en las comunicaciones, impacta de manera importante en el desempeño de *Fireworks*. Al recuperarse rutas de hasta 19 saltos para este escenario, el tiempo de validez de esta información se reduce y por ende las rutas recuperadas al ser utilizadas para enviar un paquete *unicast* a un vecino, dejan de funcionar en el trayecto ocasionando pérdidas de paquetes.

Sin duda los factores que determinan el desempeño de *Fireworks* son el tamaño del escenario, velocidad de desplazamiento de los nodos, densidad de nodos y el tráfico presente en la red. De forma particular el tamaño del escenario, densidad de nodos y la movilidad determinan la longitud de ruta, de donde resulta importante conocer el tiempo en el que esa ruta permanece funcional, permitiendo la transmisión de paquetes a través de nodos intermedios de manera exitosa al destino.

En las implementaciones de DSR (como la que se presenta en el simulador NS2 y sobre la cual se trabajó para la implementación de *Fireworks*), existe un mecanismo que define el tiempo máximo de validez de ruta (*Link-MaxLife*). [33]

Bajo el esquema de *Link-MaxLife*, cada ruta en la tabla de ruteo tiene un tiempo de vida que se determina dinámicamente por el nodo que la almacena, de acuerdo a su comportamiento pasado de ambos nodos en los extremos del enlace. Además, cuando se selecciona una ruta para un paquete a ser enviado hacia algún destino, el esquema de *Link-MaxLife* selecciona la ruta con el tiempo de vida esperado más largo (entre rutas con el mismo número de saltos). Este tiempo de vida se determina a través de un índice de estabilidad, el cual se verá reducido cada que se detecten mensajes de "error de en ruta" (*route error*)y de "error en envío" (*sendfailure*).

Por tanto, el tiempo de duración de ruta estará determinado por la movilidad y la longitud de la ruta misma y disminuirá conforme los nodos se muevan a mayor velocidad y el escenario sea más grande.

Tomando como premisa lo anterior y basándose en los resultados de las simulaciones, se observa que *Fireworks* tendrá un mejor desempeño para escenarios pequeños a medianos, donde las rutas no tendrán más de 15 saltos de longitud y bajo esquemas de movilidad inferiores a los 10m/s. Sin embargo, presenta un desempeño aceptable para redes grandes (hasta 19 saltos por ruta) para redes estáticas.

En lo que respecta al *flooding*, en los tres escenarios y para los diferentes casos de movilidad claramente se obtiene una ganancia importante al utilizar la búsqueda acotada propuesta de *Fireworks*, haciendo un uso eficiente de la información de ruta generada por su antecesor DSR, reduciendo la generación excesiva de paquetes de control en algunos casos hasta un 80%. También se establece que al incrementar el número de saltos que abarca la búsqueda inicial, se encuentran una mayor cantidad de rutas pero también se genera una mayor cantidad de *flooding*.

Para el número de rutas encontradas en los tres escenarios, se observó que conforme se incrementa la movilidad se van recolectando un mayor número de rutas. Sin embargo, solo una pequeña parte de esas rutas ayudan a localizar exitosamente al destino, al ser la mayoría de ellas obsoletas al momento de enviar los paquetes *unicast* de búsqueda, ocasionando que se pierdan en el trayecto. El hecho de que se encuentren una mayor cantidad de rutas conforme se incrementa la movilidad, obedece a que los nodos recorren un área mayor al moverse más rápido, participa en más comunicaciones y escucha muchas otras en el trayecto. Sin embargo, para niveles de movilidad altos (10m/s para nuestras pruebas), mucha de esta información de ruta que se entrega al nodo origen es errónea, al pertenecer a comunicaciones que se llevaron acabo si bien en un tiempo no muy anterior al momento de la entrega de información, si en una zona del escenario distante a la zona de la búsqueda. Por tanto en este aspecto se determina que el encontrar una gran cantidad de rutas conforme se incrementa a la movilidad, en primera instancia no ayuda a incrementar el número de éxitos en la búsqueda del nodo destino.

VII. Conclusiones

En el presente trabajo se diseñó un protocolo nuevo de descubrimiento de ruta, como una alternativa eficiente a los protocolos de enrutamiento comúnmente usados, cuya implementación y funcionamiento se evaluó en los capítulos anteriores. Se analizaron diversos escenarios donde se tomaron como variables el área de la red, el número de nodos participantes, la tasa de transferencia de datos (tráfico), número de conexiones establecidas y finalmente la movilidad de los nodos. Mediante el análisis de los resultados arrojados de las simulaciones para los diferentes escenarios, se obtienen las siguientes conclusiones:

a) **En lo que respecta al número de rutas hacia vecinos encontradas**, los factores que determinan su número máximo es el incremento del tráfico y la movilidad. Conforme un nodo se mueve a una mayor velocidad, participa en una mayor cantidad de rutas y tiene oportunidad de escuchar otras comunicaciones ocurriendo en diferentes lugares de la red. De la misma manera, cuando se envían paquetes a una tasa de transferencia mayor, se detectan más rápidamente cambios en la topología de la red, mantienendo más actualizada la tabla de rutas.

Por otro lado, el factor movilidad impacta significativamente el tiempo de duración/validez de las rutas. Al moverse los nodos a una mayor movilidad, los nodos cambian su posición con mayor rapidez, ocasionando que aquellas rutas que se encontraban funcionando en un momento dado, dejen de existir y no permitan entregar los paquetes al destinatario. Aunado a lo anterior, conforme se incrementa el tráfico se incrementan las contenciones por el uso del canal, por lo que nodos que deseen transmitir deberán esperar un tiempo dado para hacer efectivo su turno de enviar información a su destinatario e inclusive pueden presentarse colisiones, ocasionando pérdida de paquetes.

b) **En lo que respecta al número de vecinos alcanzados**, se observa un aumento en su número conforme se incrementa el tráfico y la movilidad. Sin embargo, debido a factores como movilidad y tamaño de la red, del total de rutas encontradas a vecinos, sólo una cantidad de ellas son útiles para alcanzar al nodo vecino con un paquete unicast. Lo anterior se compensa con el hecho de que se envían varios paquetes de búsqueda *fireworks* (de naturaleza *unicast*) incrementando significativamente las posibilidades de encontrar exitosamente al nodo destino, como se comprobó en los resultados obtenidos.

c) **Para el caso de éxitos alcanzados**, se concluye que su número se encuentra acotado por el incremento del tráfico en la red y la movilidad principalmente. Vale la pena recalcar que para los casos de escenarios pequeño, mediano y grande, para esquemas de movilidad de hasta 2m/s, el número de éxitos alcanzados es bastante aceptable (en particular para esquemas sin movilidad). Para el caso del escenario grande y con movilidad, el número de éxitos se reduce dramáticamente.

d) **En lo que respecta al flooding generado por Fireworks**, durante todas las pruebas el número de paquetes de control generado necesario para encontrar rutas a los destinos, se mantuvo significativamente menor en comparación con la operación de DSR en los mismos escenarios. Con esto se concluye que *Fireworks* reduce de manera muy sigificativa el *flooding* necesario para el funcionamiento de una red *ad-hoc*.

e) **En lo que respecta al tamaño de la red**, *Fireworks* demostró un gran desempeño aún en escenarios grandes (en especial para el caso de una red con nula o baja movilidad). La longitud de las rutas (la cual se incrementa conforme se incrementa el tamaño e la red) y la movilidad, limitan en cierto grado la operación de *Fireworks*, por lo que concluimos que nuesto protocolo puede ser utilizado con gran éxito para escenarios pequeños y medianos, así como para escenarios grandes con baja movilidad.

Sin duda alguna, la premisa principal bajo la cual trabaja Fireworks, dícese que existe una mayor probabilidad de encontrar a un determinado número de nodos (vecinos junto el nodo destino) que encontrar al nodo destino por sí solo, ha demostrado ser exitosa en la tarea de optimizar la inundación de las redes ad-hoc con paquetes de control para encontrar y mantener rutas.

Finalmente, con la presentación de este trabajo se presenta a *Fireworks* como una opción viable sobre protocolos de enrutamiento tales como DSR, AODV y TORA y como una alternativa de bajo costo, pero de rango limitado sobre aquellos protocolos de enrutamiento basados en información geográfica.

VIII. Glosario

A

ALOHA. Sistema de acceso al medio propuesto para coordinar un acceso arbitrario a un canal inalámbrico de comunicaciones. Fue desarrollado en los setentas por la Universidad de Hawaii. El sistema original fue utilizado para transmisiones de radio terrestres (en sus variantes sencillo y ranurado), aunque se ha implementado también en sistemas de satélites de comunicaciones.

B

Backbone. La red de *backbone* es un elemento importante de las redes en edificios empresariales. Provee un camino para el intercambio de información entre redes LAN o subredes. Una red de *backbone* puede ser un punto de convergencia de redes diferentes en un mismo edificio, en un ambiente de campus de diferentes edificios o sobre áreas amplias. Generalmente la capacidad de la red de *backbone* es más grande que las redes conectadas a él.

Bluetooth. Es el nombre código asignado para una especificación de redes inalámbricas de área personal, que ha sido desarrollada por el grupo de interés especial Bluetooth SIG. Permite que dispositivos creen de manera espontánea, redes inalámbricas dentro de áreas pequeñas. Este sistema es ampliamente implementado en computación, telefonía y diversos equipos electrónicas.

Broadcast. Tipo de transmisión de datos utilizada en redes de medio compartido (por ejemplo Ethernet), donde múltiples nodos se encuentran asociados a la misma LAN. Es un método de transmisión uno-a-muchos. Todos los dispositivos asociadas a la red que reciben paquetes de este tipo, tienen que procesarlos para definir si la información que contiene les es de utilidad.

Búfer de envío. Porción de memoria ubicada en cada nodo móvil donde se almacenan temporalmente los paquetes de datos a enviar. Su implementación puede variar en arquitecturas como FIFO (First Input First Output), LIFO (Last Input First Ouput), etc.

C

Chipping. Método utilizado en las comunicaciones de espectro disperso para representar los datos originales con una cadena de bits múltiples a partir de una secuencia pseudoaleatoria generada por el transmisor.

D

Densidad de nodos. Número de nodos móviles por unidad de área que participan en una red inalámbrica del tipo ad-hoc.

E

Enrutamiento. Proceso de descubrimiento de una ruta hacia el host destino. Puede ser complejo en redes grandes debido a la gran cantidad de destinos intermedios potenciales que un paquete debe de atravesar para llegar a su destino.

F

Flooding. Mecanismo de descubrimiento de ruta utilizado en redes de datos (inalámbricas y guiadas) para el descubrimiento de rutas hacia un destino determinado. Su propagación se realiza en forma de broadcast y se propaga hacia toda la red.

F

Flooding. Mecanismo de descubrimiento de ruta utilizado en redes de datos (inalámbricas y guiadas) para el descubrimiento de rutas hacia un destino determinado. Su propagación se realiza en forma de broadcast y se propaga hacia toda la red.

G

Gateway. Computadora u otro tipo de equipo de red que actúa como un traductor entre dos sistemas que no utilizan el mismo conjunto de protocolos de comunicaciones, estructuras de formato de datos, lenguajes y/o arquitectura.

GPS. Red satelital que ofrece los servicios de posicionamiento global e información de tiempo para navegación terrestre y marítima. El sistema consiste de 24 satélites Navstar que orbitan a una altitud de 11,000 millas.

H

Hello Packets. Paquetes de control que se envían en períodos constantes entre nodos para establecer vecindad entre ellos.

I

ISM. Banda alojada en los 2.45y 5GHz, utilizada para instrumental Industrial, Científico y Médico, que tiene como peculiaridad el que no se necesita licencia para utilizarla.

M

Mantenimiento de ruta. Mecanismo implementado en los protocolos de enrutamiento ad-hoc, que permite detectar fallas en los enlaces, identificar el nodo que la presentó y buscar caminos alternos para la transmisión de paquetes al destino. También se encarga de remover rutas obsoletas de las tablas de ruteo de los nodos.

O

Overhead. Fenómeno que se presenta en redes ad-hoc cuando el número de paquetes de control sobrepasa el número de paquetes de datos transmitidos. De igual manera este término se utiliza cuando en los paquetes que circulan en una red, los bits de encabezado (donde se definen tipo de protocolos, banderas, información de diferentes capas) es excesiva superando al número de bits determinados para portar datos.

S

Salvamento de paquete. Mecanismo por el cual, cuando se determina que un salto en un enlace ha dejado de funcionar, el nodo que detecta el error consulta en su tabla de ruteo si cuenta con una ruta adicional hacia el destino que de tenerla, transmite el paquete al destino evitando su pérdida.

T

Time-to-live. Porción del encabezado IP que determina el tiempo de vida (o saltos) de un paquete para su tránsito en una red. Cada salto por el que pase (router), este valor se decrementará en 1; cuando alcance el valor de 0, el paquete es desechado.

U

Unicast. Transmisión de un paquete de datos hacia un destino determinado, proveniente de un nodo definido en la red.

IX. Referencias

[1] *"Towards a Hybrid Nerwork Model for Wireless Packet Data Networks"*, Hung-Yun Hsieh and Raghupathy Sivakumar, the GNAN Research Group, School of Electrical and Computer Engineering, Georgia Institute of Technology. Proceedings of the Seventh International Symposium on Computers and Communications, 2002, IEEE

[2] *"Wireless Enterasys Education"*, Johnson, Keith, Enterasys Networks, 2003

[3] *"Overview of wireless networks",* Fisher Josh y Wang Rosemary , 2001

[4] http://www.eng.auburn.edu/users/lim/sensit/page2.html, noviembre 2005

[5] http://www.sensormag.com/articles/0203/38/main.html, noviembre 2005

[6] *" Wireless Information Networks"*, Pahlavan Kaveh y Levesque H. Allen, Wiley series in telecommunications and signal processing, Wiley-interscience Publication, 1995

[7] "*The ABC's of 2.4 and 5 Ghz Wireless LANs"* Dhir, Amit, Xilinx, v1.0, Agosto 1, 2001.

[8] *"802.11a More Bandwidth without the wires",* Kapp, Steve, , IEEE Internet Computing, July/Agosto. 2002, pp. 75-79

[9] *"Comunicaciones y Redes de Computadores"*, William Stallings, Sexta Edición, Prentice Hall, 2002.

[10] *"Data Communications, Computer Networks and Open Systems"*, Halsall Fred, Addison-Wesley, Cuarta edición.

[11] *"Mobile Ad Hoc Networks: Routing, MAC and Transport Issues"*, Nitin H. Vaidya, University of Illinois at Urbana-Champaign, 2001

[12] *"Redes Inalámbricas y Móviles, IEEE 802.11, WLAN"*, Javier Gómez Castellanos, Universidad Nacional Autónoma de México, 2004

[13] *"Ad hoc On-Demand Distance Vector (AODV) Routing"*, C. Perkins, Nokia Research Center, E. Belding-Royer, University of Santa Barbara, IETF Internet Draft, Julio 2003.

[14] *"Temporally-Ordered Routing Algorithm (TORA) Version 1"*, V. Park, S. Corson , Flarion Technologies, Inc., IETF Internet Draft, Julio 2001.

[15] *"The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks"*, David B. Jhonson, David A. Maltz, et al, IETF MANET Working Group, Internet-draft, Noviembre 2000

[16] *"Efficient Flooding in Ad hoc Networks: a Comparative Performance Study"*, Yunjung Yi, Mario Gerla, Taek Jin Kwon, IEEE International Conference on Communications 2003

[17] *"The broadcast storm problem in a mobile ad hoc network"*, Sze-Yao Ni, Yu-Chee Tseng, Yuh-Shyan C hen, Jang-Ping Sheu, MobiCom, 1999.

[18] *"Adaptive Approaches to Relieving Broadcast Storms in a Wireless Multihop Mobile Ad Hoc Network"*, Yu-Chee Tseng, Sze-Yao Ni, En-Yu Shih, Infocom, 2001.

[19] *"Adaptive Aproaches to Relieving Broadcast Storms in a Wireless Multihop Mobile Ad Hoc Networks"*, Yu-Chee, Sze-Yao Ni, En-Yu Shih, INFOCOM 2001

[20] *"Flooding in Wireless Ad Hoc Networks"*, H. Lim, C. Kim, IEEE Computer Communications 2000.

[21] *"An energy-efficient coordination algorithm for topology maintenance in Ad Hoc wireless networks"*, B. Chen, K.H. Jamieson, R. Morris, MOBICOM 2001

[22] *"Scalable Link-State Internet Routing"*, J.J. García-Luna-Aceves, M. Spohn, ICNP 1998

[23] *"Topology Broadcast Based on Reverse-Path Forwarding(TBRPF)"*, Internet Draft, Noviembre 2001.

[24] *"Efficient Route Discovery in Mobile Ad Hoc Networks Using Encounter Ages"*, Henri Dubois_Ferrie, Matthias Grossglauser, Martin Vetterli, School of Computer and Communications Sciences, EPFL, Laussane Switzerland.

[25] *"Multipoint relaying: An efficient technique for flooding in mobile wireless networks"*, Amir Qayyum, Laurent Viennot, Anis Laouti, INRIA report, Marzo 2000

[26] *"Efficient Flooding in Ad Hoc Networks: A comparative performance study"*, Yunjung Yi, Mario Gerla, Taek Jin Kwon, IEEE International Conference on Communications, 2003.

[27] *"A Scalable Location Service for Geographic Ad Hoc Routing"*, Jinyang Li, John Jannotti, et al, M.I.T. Laboratory for Computer Science, MOBICOM 2000, Boston MA USA

[28] *"Routing and addressing problems in large metropolitan-scale internetworks"*, Gregory G. Finn, ISI/RR-87, ISI, Marzo 1987

[29] *"GPSR: Greedy perimeter stateless routing for wireless networks"*, ACM/IEEE MobiComm, Agosto 2000

[30] *"The NS Manual"*, Kevin Fall, Kannan Varadhan, The VINT Project, UC Berkeley, LBL, USC/ISI and Xerox PARC, Julio 16, 2003

[31] *"The NAM Manual",* http://www.isi.edu/nsnam/nam/index.html

[32] *"Tutorial for the Network Simulator NS",* Marc Greis, http://www.isi.edu/nsnam/ns/tutorial/

[33] *"Caching Strategies in On-Demand Routing Protocols for Wireless Ad Hoc Networks".* Yih Chun Hu and David B. Johnson. In Proceedings of the Sixth Annual ACM International Conference on Mobile Computing and Networking, August 2000

[34] "Comunicaciones y redes de computadoras", William Stallings, Editorial Pearson, 2004