



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

FACULTAD DE INGENIERÍA

**“ANÁLISIS DE LA MOVILIDAD DE TERMINALES EN
UNA
RED DE TIPO AD HOC”**

TESIS

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO EN TELECOMUNICACIONES

PRESENTA

RODRIGO ATAHUALPA SÁNCHEZ TELÉSFORO

DIRECTOR:

DR. JAVIER GÓMEZ CASTELLANOS



MÉXICO D.F., AGOSTO 2006.



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

Dedicatorias

A mi Dios, porque me permite ser y sentir humano.

*A mis padres, la **Profesora Florentina Telésforo Reyes** y el **M.I. Antonio Sánchez Pérez** como testimonio de mi infinito aprecio y agradecimiento por toda una vida de esfuerzos y sacrificios, brindándome siempre cariño y apoyo. Deseo que mi triunfo como hombre y profesionalista lo sientan como el suyo propio. Con mucho amor, admiración y respeto, ustedes son un gran ejemplo.*

*A mis abuelos **Eloisa Reyes Galicia** y **Leobarda y Rodrigo Sánchez Pérez**, ya que este logro es también producto de su esfuerzo.*

*A mi hermano e ingeniero **Hugo Antonio Sánchez Telésforo** por su apoyo oportuno.*

Agradecimientos

Al Pueblo de México, mi país, por la educación pública que he recibido como fruto de diversos movimientos sociales y culturales.

*A la **Universidad Nacional Autónoma de México**, por materializar mi educación integral y a la **Comunidad Universitaria** porque su trabajo diario enaltece a nuestra Universidad.*

*A la **Fundación General Electric**, por el apoyo económico que me brindó durante mi licenciatura.*

*Al **Dr. Javier Gómez Castellanos** por su apoyo y confianza al final de mis estudios de licenciatura con la elaboración de este trabajo de tesis. Su ejemplo estimula mi carrera profesional.*

*A mis profesores y sinodales el **Dr. Víctor Rancel Licea**, el **Dr. Miguel Moctezuma**, el **Ing. Carlos Girón** y el **Ing. Jesús Sánchez Zepeda**, por sus recomendaciones para mi Tesis y por sus valiosos consejos de vida. Mi admiración a ellos.*

*A mi amiga e ingeniera **Fernanda Hernández Sandoval**, pues su apoyo moral contribuyó a estimular mi desempeño académico. Mi vida en la Facultad de Ingeniería la recordaré junto a tí.*

*A todos mis amigos, en particular a **Carlos Villanueva**, **Luis Edgar Rodríguez**, **César Reyes**, **Bogdad Espinosa**, **Ramón Meza**, **Brenda Ocampo**, **Heber y Abiel Parra**, **Eurídice Moreno**, **Eduardo Gámez**, **Omar Solórzano**, **Thalía Reyes**, **Marcos Suárez**, **José Carlos González**, **Ricardo Pinales**, **Ricardo Miranda**, **Luis Cancino**, **Giovanni Fonseca**, **Leopoldo Aguilera**, **Jordi Morales**, **Jorge Campos**, **Hiroimi Valenzo**, **Jazmín Sánchez**, **Rosa María Fernández**, porque todos ellos me hacen mejorar como ser humano.*

Índice general

Introducción	1
Objetivo	2
1. Redes inalámbricas	4
1.1. Redes inalámbricas de área local	5
1.2. Topologías de redes inalámbricas	7
1.2.1. Redes inalámbricas con infraestructura	8
1.2.2. Redes inalámbricas sin infraestructura o de tipo ad hoc	9
1.2.3. Redes inalámbricas de sensores	10
1.2.4. Redes mesh	12
Referencias	14
2. Generalidades de redes sin infraestructura o ad hoc	15
2.1. Características de las redes ad hoc	15
2.2. Aplicaciones de las redes ad hoc	16
2.3. Descripción de protocolos de encaminamiento	18
2.3.1. Inundación de la red para la localización de usuarios	19
2.4. Algoritmos de encaminamiento reactivos o bajo demanda	20
2.4.1. Encaminamiento ad hoc por Vector Distancia sobre Demanda (AODV)	21
2.4.2. Algoritmo temporalmente ordenado (TORA)	22
2.4.3. Protocolo de fuente dinámica para redes móviles DSR	23
2.5. Utilidad del análisis de movilidad en redes ad hoc	25
Referencias	26

3. Análisis de movilidad mediante simulación	28
3.1. Generalidades del simulador Network Simulator (ns)	28
3.1.1. Esquema de simulación en ns-2	31
3.1.2. La herramienta de animación (nam) del ns-2	31
3.2. Planteamiento y justificación de las condiciones de simulación	32
3.2.1. El algoritmo Random waypoint	34
3.3. Simulación en ns-2 de una red inalámbrica con nodos móviles	35
Referencias	39
4. Resultados de las simulaciones	40
5. Análisis matemático del tiempo de permanencia	44
5.1. Planteamiento del problema	44
5.2. Aproximación por mínimos cuadrados	46
5.3. Intervalos de confianza	49
5.4. Modelo propuesto	53
5.5. Porcentaje de error relativo	56
Referencias	57
6. Conclusiones	58
Glosario	60
Siglas y acrónimos	62
Apéndices	63

Lista de figuras

<i>Figura 1.1.- Configuraciones de redes inalámbricas</i>	4
<i>Figura 1.2.- Escenarios de terminales oculta</i>	5
<i>Figura 1.3.- Escenario de terminal expuesta</i>	6
<i>Figura 1.4.- Topologías de red</i>	7
<i>Figura 1.5.- Red inalámbrica con infraestructura</i>	8
<i>Figura 1.6.- Red inalámbrica sin infraestructura o ad hoc</i>	9
<i>Figura 1.7.- Esquema de una red inalámbrica de sensores en una sala de conciertos</i>	10
<i>Figura 1.8.- Aplicaciones comerciales emergentes de una red mesh</i>	12
<i>Figura 1.9.- Jerarquía de tres niveles para una red mesh</i>	13
<i>Figura 2.1.- Despliegue de una red ad hoc móvil</i>	17
<i>Figura 2.2.- Inundación de paquetes o flooding</i>	19
<i>Figura 2.3.- Ejemplo de un descubrimiento de ruta mediante el protocolo AODV</i>	22
<i>Figura 2.4.- Gráfica acíclica dirigida definida por la altura de las terminales</i>	23
<i>Figura 2.5.- Descubrimiento de una ruta desde A hasta B con el protocolo DSR</i>	24
<i>Figura 3.1.- Dualidad entre objetos de C++ y de OTcl</i>	29
<i>Figura 3.2.- Esquema de simulación en ns-2</i>	31
<i>Figura 3.3.- Descripción del esquema de simulación</i>	32
<i>Figura 3.4.- Problema de frontera durante el movimiento de las terminales</i>	33
<i>Figura 3.5.- Movimiento random waypoint</i>	34
<i>Figura 4.1.- Escenario a los 0 [s]</i>	41
<i>Figura 4.2.- Escenario a los 150 [s]</i>	41
<i>Figura 4.3.- Escenario a los 300 [s]</i>	41
<i>Figura 4.4.- Escenario a los 400 [s]</i>	41
<i>Figura 4.5.- Resultados del escenario A</i>	43
<i>Figura 4.6.- Resultados del escenario B</i>	43
<i>Figura 5.1.- Definición de la variable aleatoria con distribución exponencial</i>	45
<i>Figura 5.2.- Valores promedio con su curva de aproximación para el escenario A</i>	48
<i>Figura 5.3.- Valores promedio con su curva de aproximación para el escenario B</i>	49

<i>Figura 5.4.- Intervalos de confianza al 95 % de los valores promedio en el escenario A</i>	52
<i>Figura 5.5.- Intervalos de confianza al 95 % de los valores promedio en el escenario B</i>	52
<i>Figura 5.6.- Curvas resultantes para el escenario A</i>	54
<i>Figura 5.7.- Curvas resultantes para el escenario B</i>	54
<i>Figura 5.8.- Diferencia del número de nodos en el escenario A</i>	55
<i>Figura 5.9.- Diferencia del número de nodos en el escenario B</i>	55
<i>Figura 5.10.- Tabulación de los errores relativos para el escenario A</i>	56
<i>Figura 5.11.- Tabulación de los errores relativos para el escenario B</i>	57

Lista de tablas

<i>Tabla 3.1.- Descripción de modelos de movimiento por entidades</i>	29
<i>Tabla 3.2.- Modelos de movilidad por entidad y grupal</i>	30
<i>Tabla 4.1.- Resultados del escenario A</i>	42
<i>Tabla 4.2.- Resultados del escenario B</i>	42
<i>Tabla 5.1.- Valores base en la aproximación por el método de mínimos cuadrados</i>	46
<i>Tabla 5.2.- Valor de coeficientes bajo la aproximación por el método de mínimos cuadrados</i>	47
<i>Tabla 5.3.- Funciones aproximadas por el método de mínimos cuadrados</i>	48
<i>Tabla 5.4.- Intervalos de confianza al 95 % para los valores promedio del número de vecinos</i>	51
<i>Tabla 5.5.- Funciones basadas en el modelo propuesto de movilidad</i>	53

Introducción

Una gran parte de las actividades productivas humanas, que tienen un alto impacto en nuestras vidas, se apoyan en sistemas electrónicos pues con ellos logramos reducir tiempo y costo de operación. El aumento en la cantidad de sistemas y de sus funciones hace imperante su interconexión, ya que la transmisión de la información que cada uno procesa, facilita nuestras actividades; el ser humano nunca tuvo una herramienta tan poderosa como lo es ésta.

Los sistemas de comunicación inalámbricos como la telefonía celular, la satelital, las redes inalámbricas de área personal, local y metropolitana cuentan con una amplia aceptación, convirtiéndose en instrumentos cotidianos. Sin duda alguna, la popularidad de los sistemas inalámbricos sobre los sistemas cableados se debe a función de libre desplazamiento de los equipos o movilidad, que ofrecen a los usuarios, pues al no requerir cables posibilita el paradigma de una comunicación en cualquier lugar y a toda hora. Es por ello que destaca, por la integración de la tecnología en nuestras actividades cotidianas, el impacto de las tecnologías de la información en la vida moderna.

Desde el punto de vista del canal de comunicación, el transporte inalámbrico de la información tiene como medio de transmisión el aire que nos rodea. Con esta premisa es posible transferir datos bajo un esquema en el que los usuarios modifiquen su posición física, que es lo que se define como capacidad de movilidad en una red de datos. Hasta recientes desarrollos la movilidad se ha convertido en una posibilidad real en las redes de datos, pues ha sido el resultado de las implementaciones como las del estándar IEEE 802.11 conocido también como WiFi.

En la actualidad se materializan dentro de las redes inalámbricas dos arquitecturas de servicio: las redes con infraestructura cuyo claro ejemplo es la arquitectura celular y las redes definidas como redes sin infraestructura o de tipo ad hoc.

Una red con infraestructura se caracteriza por utilizar una red troncal o *backbone* con comunicación cableada que enlaza a las estaciones base o puntos de acceso inalámbrico. Por su parte, el modo ad hoc está formado por dispositivos que cooperan entre ellos mismos de manera distribuida para llevar a cabo, mediante sus enlaces, la transmisión de los paquetes de información. Las redes ad hoc poseen una topología dinámica, es decir, con cambios frecuentes ya que las terminales se encuentran en movimiento. Dos nodos pueden comunicarse de forma directa entre ellos cuando se localizan dentro de sus áreas de cobertura y en caso contrario los nodos intermedios funcionarán como encaminadores de los paquetes de información. El encaminamiento de los paquetes, la gestión y el mantenimiento de las rutas son operaciones realizadas por cada una de las terminales en una red ad hoc.

La utilidad de las redes de tipo ad hoc se basa en situaciones que necesitan establecer comunicación de rápido despliegue que no cuentan con ningún tipo de infraestructura física, como estaciones base o centros de control. Esta estructura de red es imprescindible en los entornos de despliegue militar y en las operaciones de emergencia. En la actualidad como en la década de los 70's, cuando comenzó el interés por las redes inalámbricas de conmutación de paquetes, los proyectos militares rigen la actividad investigadora en este tipo de redes pues su flexibilidad y dinamismo son por excelencia idóneos para estos fines.

Las redes móviles ad hoc son conocidas como MANETs (Mobile Ad hoc Networks). Las características particulares de una MANET como su tamaño, patrón de movimiento, tipo de tráfico, ancho de banda y consumo energético, son las que determinan el uso de ciertos protocolos de red.

En una MANET, los enlaces entre los nodos origen y destino cambian frecuentemente debido a su movimiento. En años recientes, los investigadores se han enfocado en el estudio de los modelos de movilidad para las redes MANETs [1], [2]. La evaluación de los nuevos protocolos de red propuestos muestra que el comportamiento de la red cambia de forma considerable, con las características del modelo de movilidad utilizado. Recientes reportes muestran nuevos modelos de movimiento más cercanos al comportamiento real de las redes móviles, como son el modelo de movilidad que considera limitaciones espaciales [3], el que contempla obstáculos físicos en un enlace [4], hasta llegar al modelo que propone y justifica un estado de movimiento perfecto de los nodos [5]. Estas propuestas basan sus resultados en los diversos simuladores de redes pero sólo algunas presentan modelos analíticos del movimiento de las terminales.

En este trabajo presentamos un análisis de movilidad que detalla el comportamiento dinámico de las terminales en una red sin infraestructura o de tipo ad hoc para ser considerado como base en el diseño de protocolos de encaminamiento. Específicamente consideramos el comportamiento de los nodos dentro de un simulador de redes, para que a partir de los resultados obtenidos en la simulación, presentemos un modelo matemático que represente el comportamiento de las terminales en una red de tipo ad hoc.

Objetivo

Este trabajo tiene como objetivo analizar la movilidad de nodos vecinos durante una transmisión en una red móvil sin infraestructura o de tipo ad hoc con el fin de establecer un modelo matemático basado en su comportamiento dinámico dentro de un simulador de redes.

La estructura de este trabajo de tesis es la siguiente: en el capítulo 1 introducimos a las redes inalámbricas y en particular a las topologías de redes inalámbricas actuales. En el capítulo 2 describimos las generalidades de las redes inalámbricas sin infraestructura detallando sus aplicaciones y sus características, así como los protocolos de encaminamiento más utilizados que justifican nuestro estudio de movilidad. En el capítulo 3 presentamos la descripción del simulador de redes que utilizamos y las condiciones de simulación que seguimos en nuestro estudio. Al capítulo 4 le corresponde la descripción de cómo desarrollamos los experimentos junto con sus resultados. En el capítulo 5 mostramos el desarrollo matemático que justifica nuestro modelo representativo del fenómeno, basado en su comportamiento dentro del simulador. En el capítulo 6 mostramos las conclusiones de nuestro análisis de movilidad. Incluimos además un glosario con términos específicos del área de redes de datos relativos a este trabajo y un apéndice donde presentamos el código fuente de los programas utilizados.

La contribución de este trabajo se materializa en el artículo: *Fireworks: A Novel Tracking Protocol for Mobile Ad-Hoc Networks*, donde incluimos sus resultados como parte de la justificación de esta propuesta de protocolo de encaminamiento para redes de tipo ad hoc y con ello constituir un punto de partida o referencia para la comunidad científica en las nuevas propuestas de análisis de redes sin infraestructura.

Referencias:

[1] T. Camp, J. Boleng, *A Survey of Mobility Models for Ad Hoc Network Research*, Wireless Communication and Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications, Vol. 2, No. 5, 2002.

[2] J. Tian, J. Hhner, C. Becker, *Graph-Based Mobility Model for Mobile Ad Hoc Network Simulation*, Proceedings of the 35th Annual Simulation Symposium, in cooperation with the IEEE Computer Society and ACM, San Diego, California. April 2002.

[3] A. Jardosh, E. B. Royer, et al., *Towards Realistic Mobility Models for Mobile Ad hoc Networks*, Mobicom 2003, Sept.14-19, San Diego, California, 2003.

[4] S. PalChaudhuri, J.-Y. Le Boudec, M. Vojnovic, *Perfect Simulation and Stationary of a Class of Mobility Models*, IEEE Infocom 2005, Miami, FL, 2005 (Infocom 2005 Best Paper Award).

[5] Mohammad Ilyas, *The handbook of ad hoc wireless networks*, Florida Atlantic University Boca Raton, EUA, CRC Press, 2003.

1. Redes inalámbricas

La principal diferencia entre los enlaces cableados y los inalámbricos es el tipo de comunicación. Los enlaces cableados por lo general ofrecen una comunicación con un mínimo de interferencias, mientras que la comunicación mediante enlaces inalámbricos posee considerables niveles de ruido y de interferencias que limitan la velocidad de transferencia de los datos o ancho de banda pues provocan tasas altas de error en los bits enviados.

Los esquemas de comunicación inalámbrica básicos son las conexiones punto a punto, punto a multipunto y punto a multipunto a multipunto que mostramos en la figura 1.1.

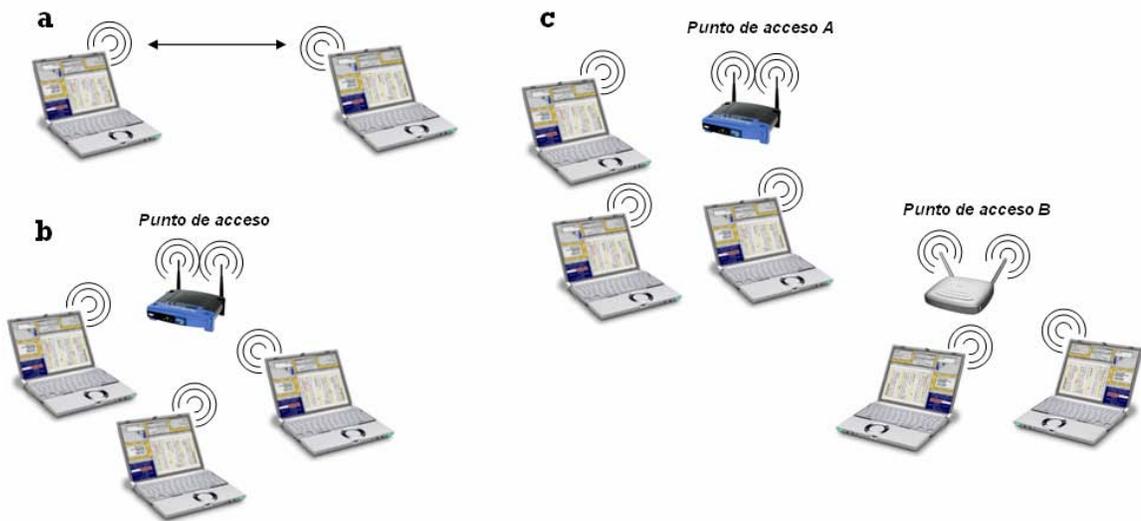


Figura 1.1.- Configuraciones de redes inalámbricas:

Esquemas punto a punto (a), punto a multipunto (b) y punto a multipunto a multipunto (c).

Una comunicación punto a punto (p2p: *peer-to-peer*) está definida por el enlace inalámbrico de sólo dos terminales por línea de vista en algunos casos. La comunicación punto a multipunto se logra con una terminal concentradora por lo general con una antena omnidireccional, que enlaza a varias terminales remotas en modo punto final. Por su lado, en la comunicación punto a multipunto a multipunto uno de los usuarios en modo punto final deja de serlo para trabajar en modo multipunto generando una nueva área de cobertura.

Las comunicaciones inalámbricas requieren de canales separados para atender a varios usuarios a la vez. Los canales pueden dividirse en diferentes frecuencias, tiempos o códigos para cada usuario, puesto que el rango de frecuencias del espectro electromagnético utilizadas limita el número de usuarios que puedan ser comunicados correctamente de forma simultánea.

Los dispositivos móviles utilizan sólo la energía almacenada por sus baterías, es por ello que el reto en el diseño de estos sistemas es el ahorro de energía. De forma teórica cualquier terminal dentro del área de cobertura podría recibir la señal de datos, de ahí que los enlaces inalámbricos son en principio inseguros. Al respecto existen técnicas de encriptación y descifrado que brindan seguridad al sistema pero que demandan recursos de ancho de banda y de energía.

1.1. Redes inalámbricas de área local

Las WLANs son redes inalámbricas de área local que ofrecen alta velocidad de transmisión dentro de un área relativamente pequeña. Su desarrollo comenzó en la mitad de la década de los 80's a partir de que la FCC (Federal Communications Commission) de Estados Unidos reguló el uso de frecuencias no licenciadas conocidas como ISM (Industrial, Scientific and Medical) quedando sujetas a interferencias significativas generadas por la cantidad de equipos que las utilizan y que hace necesario limitar los niveles de potencia de las transmisiones.

Además de satisfacer los requerimientos de una típica LAN (Local Area Network) como son la alta capacidad de transmisión, flexibilidad o conectividad completa con terminales nuevas y capacidad de transmisión por difusión o *broadcast*; las WLANs deben resolver los problemas relacionados específicamente con su transmisión inalámbrica como son los generados por su topología dinámica (creación y rompimiento de rutas) y las restricciones de ancho de banda impuestas por su medio de transmisión.

De las desventajas principales del medio de transmisión inalámbrica comparado con las transmisiones cableadas, está el incremento en la tasa de bits erróneos. El BER (Bit Error Rate) o tasa de bits erróneos en una transmisión inalámbrica llega a ser hasta diez veces mayor que el presentado en una transmisión por cable de cobre, esto debido principalmente al ruido atmosférico, obstáculos físicos presentados en la ruta de la señal, la propagación en multi-trayectorias y la interferencia con otros sistemas.

Una WLAN ofrece alta facilidad de instalación respecto a una red LAN cableada, sin embargo, es necesario considerar los factores que afectan la propagación de la señal en el ambiente de instalación. Como ejemplo tenemos a un edificio pequeño o una oficina donde es difícil precisar con exactitud los efectos causados por las paredes, ventanas, muebles y personas la propagación de las señales. Para resolver este problema, se evalúa dicha propagación mediante pruebas previas a la instalación de la WLAN.

Otro reto que presentan las WLANs es que en general, no pueden asumirse con una topología completamente conectada entre los nodos como ocurre con las redes alambreadas, pues existen los problemas conocidos como de terminal oculta y de terminal expuesta. Como se muestra en la figura 1.2., el problema de terminal oculta se refiere al caso en donde una estación A, fuera de la zona de cobertura de una terminal C, no detecta ninguna transmisión y comienza la suya. En el caso de que la terminal C hubiese estado transmitiendo, los dos paquetes transmitidos, el de A y el de C colisionarían en otras estaciones como la terminal B, que puede “escuchar” a ambas.

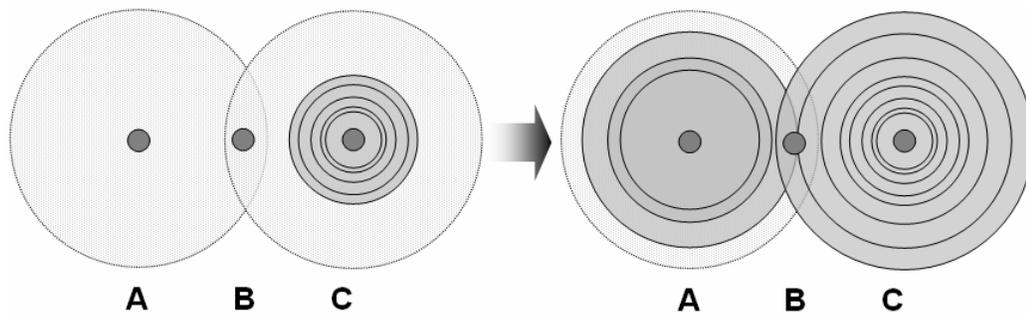


Figura 1.2.- Escenarios de terminales oculta: el nodo A es una terminal oculta para C.

Por otro lado en el problema de la terminal expuesta (figura 1.3.), una estación B emplaza su transmisión con C puesto que detecta tráfico generado por A, al mismo tiempo que C está fuera de la zona de cobertura de A. En este caso la transmisión de B podría ser recibida por C de forma satisfactoria, sin embargo esto no ocurre puesto que B emplaza su transmisión por el tráfico generado por A.

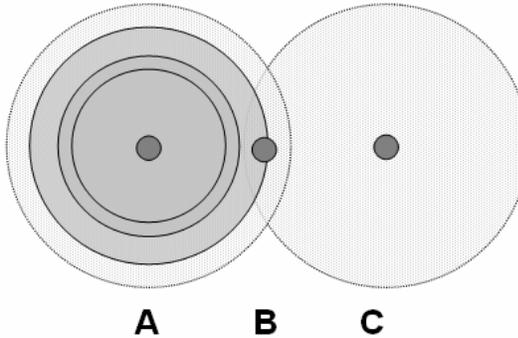


Figura 1.3.- Escenario de terminal expuesta: el nodo B está expuesto al nodo A.

La solución de éstos problemas consiste en el envío de pequeños paquetes de control denominados “petición de envío” (RTS) y “libre para enviar” (CTS) implementados en el protocolo de control de acceso al medio (MAC) de las WLAN.

Un punto importante a considerar dentro de las WLANs es el consumo de energía de las terminales. Los equipos portátiles son generalmente alimentados por baterías que tienen un tiempo finito de operación. Por lo tanto debe tenerse en consideración el diseño de mecanismos que permitan mantener el funcionamiento de la red y que minimicen el consumo de energía de las terminales.

En cuanto a la movilidad se refiere, podemos distinguir dos tipos de movilidad en una red WLAN [1]. La primera es la definida como movilidad completa, que es la capacidad de enviar y recibir información mientras se mueve la terminal dentro del área de cobertura de la red. Este tipo es difícil de lograr por los diferentes ambientes, áreas de cobertura o distancias entre los nodos de la red. Un ejemplo de este tipo de movilidad correspondería a un dispositivo manipulado por una persona de seguridad. El segundo tipo de movilidad es la portabilidad que corresponde a tener conexión con una red con sólo colocar la terminal dentro de su área de cobertura. Este es el caso de una computadora personal en una sala de conferencias que busca intercambiar archivos dentro de la WLAN.

La seguridad es otro punto a considerar en las redes WLAN. La conexión no autorizada dentro de una red cuyo medio de transmisión es el aire, demanda el uso de claves de acceso para la conexión a la red, técnicas de codificación y encriptado que lateralmente se reflejan en el tiempo de procesamiento de las terminales y por ende, en su consumo de energía.

Adicionalmente está la salud humana en el uso de este tipo de tecnologías como ocurre en general con las comunicaciones inalámbricas. Pese a que no existe un punto de acuerdo al respecto, las WLANs parecen ser, en el peor de los casos, tan inofensivas como las redes celulares pues las terminales de las WLAN utilizan radiofrecuencias con niveles de potencia de entre los 50 y los 100 [mW], que es substancialmente menor al rango entre 600 [mW] y 3 [W] utilizados por los equipos celulares.

Sobresalen dos estándares que regulan a las redes WLAN: el IEEE 802.11 y el HIPERLAN (High Performance Radio LAN). La familia de estándares HIPERLAN son promovidos por la ETSI (European Telecommunication Standard Institute) y en particular el HIPERLAN/2 establece conexiones inalámbricas de hasta 54 Mbps que se integra con las tecnologías IP, ATM y UMTS. Opera el transporte de los datos bajo calidad de servicio (QoS) y utiliza configuraciones con infraestructura y de tipo ad hoc en su implementación.

Por otro lado predomina en el mercado la familia de estándares IEEE 802.11 del IEEE (Intitue of Electric and Electronic Engineers) referido como Ethernet inalámbrico, promovido entre los fabricantes mundiales de hardware (tarjetas, adaptadores de red, puntos de acceso) y de software. El estándar IEEE 802.11 adopta técnicas utilizadas en telefonía celular como es el caso de espectro disperso (FHSS, Frequency Hopping Spread Spectrum y DSSS, Direct Sequence Spread Spectrum), técnicas de multiplexaje por división de frecuencias ortogonales (OFDM) y de control de acceso al medio con prevención de colisiones (CSMA/CA, Carrier Sense Multiple Access with Collision Avoidance)[2]. También puede ser utilizado para implementarse tanto en modo con infraestructura como en modo ad hoc. HIPERLAN/2, a diferencia del estándar IEEE 802.11, se considera aún en fase de prototipo.

1.2. Topologías de redes inalámbricas

La cantidad de formas en que pueden interconectarse los dispositivos en una red representa un reto de diseño en donde el principal problema radica en determinar lo que son diferencias superficiales y lo que corresponden a diferencias importantes de su operación, que considere criterios de respuesta, eficiencia y efectividad de la red.

La topología de red es el patrón físico o lógico que constituyen los enlaces de sus terminales determinado sólo por la forma de sus conexiones. Las distancias entre los nodos, la tasa de transmisión y la naturaleza física de la señal transmitida, no son parte de la topología de una red sin embargo, la topología puede afectar estas características.

Una serie de topologías que permiten la conexión distribuida de las terminales se presenta en la figura 1.4.; éstos son los principales esquemas de conexión de una red de datos.

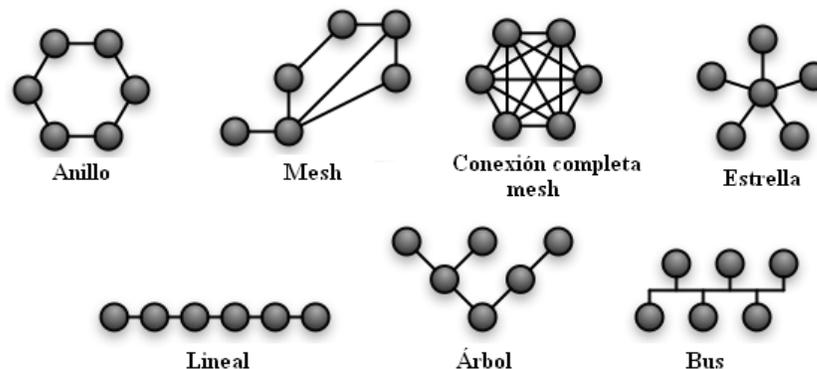


Figura 1.4.- Topologías de red.

La topología física de una red describe cómo las conexiones de las terminales en la red están organizadas (cables o señales de radio). Por su lado, la topología lógica o topología de la señal, es el término utilizado para describir cómo las terminales actúan con los recursos de la red o la forma de cómo pasan los datos a través de la red de un dispositivo al siguiente sin importar la interconexión física entre ellos. La topología lógica es parte del protocolo de la red ya que dicta cómo los datos se mueven dentro de ella. El protocolo Ethernet es el ejemplo clásico de una topología lógica en forma de bus, LocalTalk es el clásico de una topología lógica en forma de bus o de estrella y el Token Ring de IBM es el de una topología tipo anillo. Además resulta importante remarcar que una topología física no necesariamente corresponde a la misma en el nivel lógico. El ejemplo más claro es Ethernet, cuya topología lógica es de tipo bus mientras que la física es de tipo estrella.

Existen las siguientes configuraciones de redes inalámbricas móviles, cuyas características se mencionan a continuación.

1.2.1. Redes inalámbricas con infraestructura

Una red con infraestructura (figura 1.5.) se caracteriza por utilizar una red troncal fija o *backbone* con comunicación cableada que enlaza a las estaciones base o puntos de acceso inalámbricos. El típico ejemplo de una red con infraestructura es el caso celular, en donde un equipo móvil se comunica mediante ondas de radio y dentro de su área de cobertura, a un punto de acceso inalámbrico que enlaza la llamada con la red PSTN (Red Telefónica Pública Conmutada). Su mayor ventaja en comparación con una red sin infraestructura, es su capacidad de proporcionar conexión con servicios y aplicaciones que se encuentran en otras redes fijas cableadas. La figura 1.5. muestra el esquema de una red de este tipo.

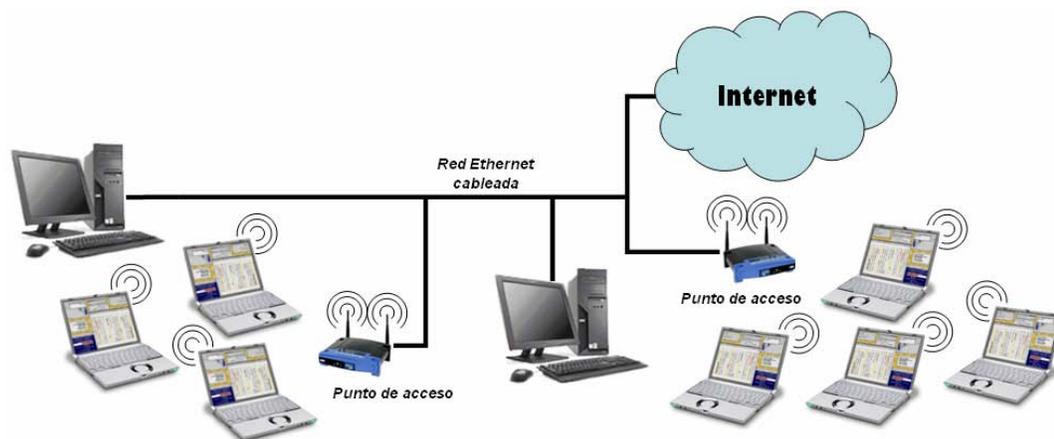


Figura 1.5.- Red inalámbrica con infraestructura.

Las estaciones base son las encargadas de realizar las funciones de encaminamiento y búsqueda de los dispositivos inalámbricos, dejando a las terminales móviles un reducido número de tareas. Al tráfico que fluye de la terminal móvil a la estación base se denomina tráfico de subida, mientras que al que fluye en dirección opuesta se le conoce como tráfico de bajada.

La presencia de varias estaciones base es común en una red con infraestructura, ya que permite aumentar la cobertura de la red inalámbrica. En este escenario, las terminales móviles pueden

moverse de una zona de cobertura a otra sin que la comunicación se interrumpa. A este proceso se le denomina *handoff*. El *handoff* es el mecanismo por el cual el tráfico del equipo móvil se redirecciona de la estación base anterior a la nueva, sin que la comunicación se interrumpa. El movimiento de las terminales de una a otra zona de cobertura (o célula para el caso celular) manteniendo la conexión lógica, se denomina *roaming* e implica que las células se traslapen apropiadamente para que los usuarios no experimenten pérdidas de conexión como se ha comentado.

El procedimiento de *handoff* es realizado tanto por una estación central de conmutación, que corresponde al esquema centralizado, como por las mismas terminales móviles en lo que se conoce como esquema descentralizado, mismo que se implementa monitoreando la intensidad de la señal en los nodos móviles. En el esquema centralizado, éste proceso lo realizan las estaciones base.

1.2.2. Redes inalámbricas sin infraestructura o de tipo ad hoc

Las redes inalámbricas sin infraestructura de manera común son conocidas como redes ad hoc y son denominadas así ya que en este caso no poseen un segmento principal cableado. La figura 1.6. muestra un ejemplo de red tipo ad hoc.

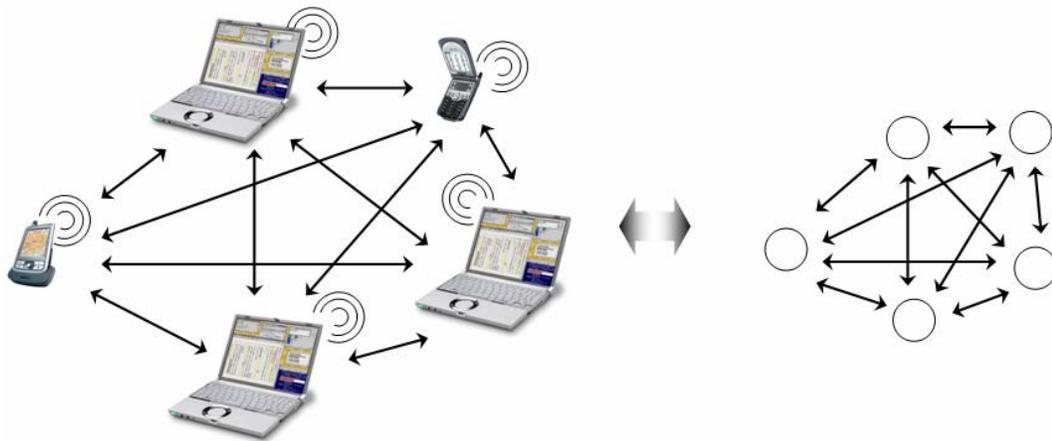


Figura 1.6.- Red inalámbrica sin infraestructura o ad hoc.

Para comunicarse, los equipos actúan como encaminadores de tráfico en la red, pues descubren y mantienen las rutas entre las terminales. Como ejemplo de aplicaciones para este tipo de redes están las operaciones de búsqueda y rescate, de emergencia, en las reuniones y convenciones donde las personas desean compartir información de forma rápida, en la adquisición de datos de operaciones en terrenos inhóspitos o simplemente en aplicaciones de despliegue inmediato [3], [4].

El concepto básico de redes ad hoc nace en los umbrales del desarrollo de Internet, durante la guerra fría, pues se visualiza una red distribuida de terminales capaz de operar incluso bajo un cierto número de conexiones inoperantes o terminales destruidas. Esta misma idea de operación distribuida está presente en las redes ad hoc inalámbricas o MANETs (Mobile Ad hoc NETWORKs), donde se tiene la característica de movilidad por el uso de una transmisión inalámbrica. De esta forma el término de red inalámbrica ad hoc establece una red sin

administración central, auto-organizada, que posee nodos móviles, con una topología que no pueden ser asumida totalmente conectada pues presenta problemas de terminal expuesta y oculta, pero sí dinámica debido a que los nodos pueden estar entre ellos, temporalmente fuera del rango de transmisión.

Las limitantes impuestas por el consumo de energía de los equipos de radiofrecuencia portátiles determinan rangos de transmisión pequeños que establecen el alcance de las terminales en una MANET. Las terminales de este tipo de redes están diseñadas para encaminar los datos logrando la comunicación en la red a través de multisaltos (*multihopping*) que determina su topología física variable con el tiempo.

En términos generales, las características de topología dinámica y de comunicaciones multisalto hacen del diseño y operación de los sistemas MANET un verdadero reto. Algunos estándares tales como IEEE 802.11, HIPERLAN/2, Bluetooth y Home RF incorporan la funcionalidad ad hoc en sus propuestas [5].

1.2.3. Redes inalámbricas de sensores

Un caso de redes ad hoc son las redes inalámbricas de sensores (WSN's). Las WSN's son redes espacialmente distribuidas de muchos equipos que poseen sensores (también conocidos como *motes*) y una o varias estaciones base que concentran el registro o monitoreo de diferentes magnitudes físicas como son: temperatura, velocidad, intensidad del sonido, vibraciones, presión, humedad, cantidad de contaminantes, entre muchos otros. Los *motes* son dispositivos pequeños y baratos, de tal forma que pueden ser desplegados en grandes cantidades, cuyos recursos de energía, memoria, velocidad de procesamiento y ancho de banda son restringidos. Cada dispositivo está equipado con una antena, un microcontrolador y una fuente de energía que generalmente es una batería. Los dispositivos se sirven entre ellos mismos para transportar los paquetes, mediante una terminal denominada puerta de enlace o estación base, al equipo que concentra y analiza el monitoreo. Un ejemplo de esta característica es el mostrado en la figura 1.7. Las redes inalámbricas de sensores trabajan bajo el principio de operación de las redes ad hoc, pues los *motes* se organizan entre ellos mismos para operar la red.

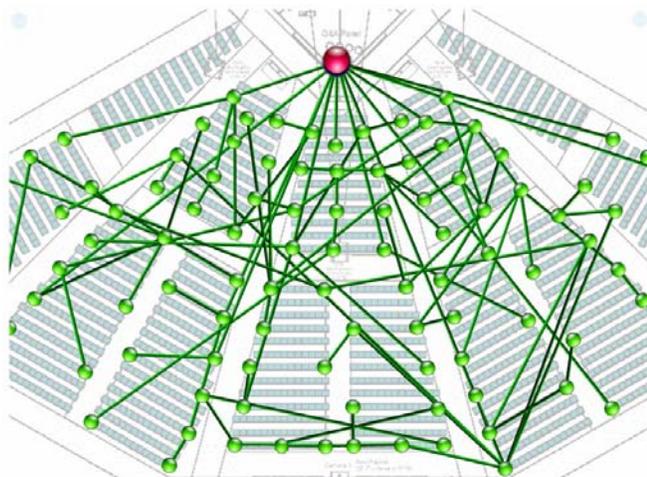


Figura 1.7.- Esquema de una red inalámbrica de sensores en una sala de conciertos. Los enlaces representan conexiones inalámbricas.

Las primeras redes de sensores fueron diseñadas con propósitos militares como es el caso del sistema SOSUS (Sound Surveillance System) durante la guerra fría. Las investigaciones actuales comenzaron alrededor de 1980 en la DARPA con el programa DSN (Distributed Sensor Networks). Actualmente su desarrollo está ligado con los avances en las comunicaciones inalámbricas, con la capacidad cada vez mayor de miniaturización y con el diseño de dispositivos de sensado de mayor alcance, precisión y menor consumo de energía.

Las redes de sensores se aplican en un diversas áreas como son en el rastreo, monitoreo o control de tráfico, en robótica, en automóviles, en el hogar, en los salones de conferencias y conciertos, en aplicaciones médicas e industriales, y son particularmente útiles en ambientes hostiles o inaccesibles para el hombre.

Algunos puntos de convergencia y de diferencia específicos entre las redes inalámbricas de sensores y las redes ad hoc son los siguientes [6]:

- ❖ En una red de sensores, las terminales son menos complejas que los equipos que constituyen a las redes ad hoc como las PDAs, laptops y en general otros equipos móviles que se suponen son manejados por usuarios.
- ❖ En cuanto a las características de tráfico se refiere, una WSN presenta por lo general velocidades bajas de transmisión durante largos períodos de tiempo, pero pueden tener un brusco cambio de tráfico cuando ocurre un evento repentino en el ambiente. Largos lapsos de inactividad (como pueden ser meses) se alternan con períodos cortos de alta actividad en la red (minutos o segundos) llegando al límite de su capacidad. Por su parte las MANETs se utilizan en aplicaciones más convencionales como tráfico Web y voz que tienen características de tráfico comparativamente más estudiadas.
- ❖ El principio básico de una red de sensores se basa en la existencia de cientos o miles de terminales distribuidas, cada una con mínimo de capacidad de procesamiento. De ahí que la identificación de los equipos sea un gran reto en las WSNs.
- ❖ Las terminales en una red ad hoc deben ser lo bastante confiables al momento de operar; por el contrario en una red de sensores, los *nodes* pueden llegar a encontrarse en gran cantidad que se convierten casi en irrelevantes para la operación del sistema. La calidad del servicio (QoS) para redes ad hoc está dictada por aplicaciones tradicionales como la voz o el tráfico de Internet mientras que en redes de sensores no existe tal.
- ❖ El consumo de energía de las terminales en una WSN se supone diferente al de las pertenecientes en las MANETs, pues se trata de sensores cuya capacidad no se compara con todo un equipo portátil de cómputo. Podemos considerar la ubicación de los sensores bajo condiciones hostiles para el hombre, como el factor que hace imperante el buen manejo de los tiempos de inactividad de los *nodes* para la máxima duración de sus baterías.
- ❖ La auto-organización de la red es un punto de convergencia entre las WSNs y las MANETs, pese a que las diferencias que presentan en cuanto al consumo de energía y al tráfico obligan a plantear soluciones diferentes.

1.2.4. Redes mesh

Otro caos de redes ad hoc son las redes mesh [7], formadas por una mezcla de nodos fijos y móviles que se interconectan mediante enlaces inalámbricos que puede ser desplegadas en lugares con preparación mínima y con flexibilidad de incluir a miles de terminales. Ofrecen conexión descentralizada, resistente y re-configurable.

Brindan soluciones en ambientes como en casos de emergencia, de difícil despliegue, en aplicaciones de transmisión de video a alta velocidad, transporte público y telemetría. La figura 1.8. muestra dos ejemplos de implementaciones de este tipo de redes.

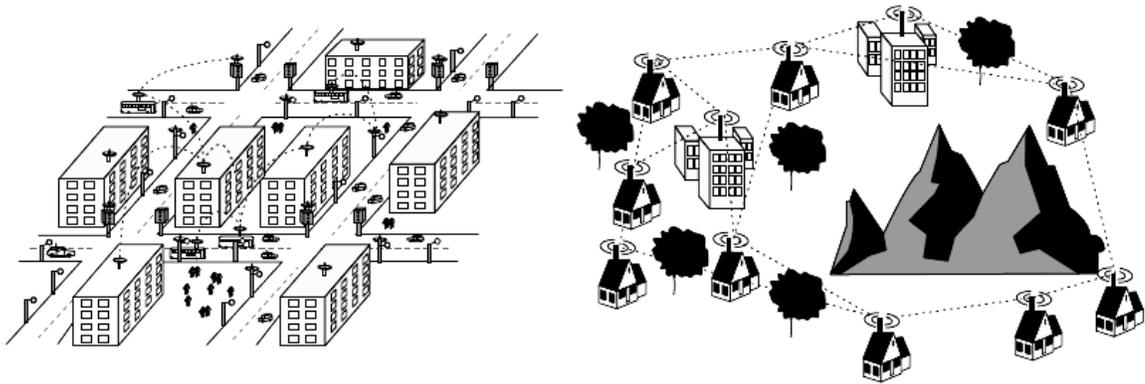


Figura 1.8.- Aplicaciones comerciales emergentes de una red mesh:

a) Sistema de transporte inteligente, b) Acceso de banda ancha residencial para regiones poco pobladas y de difícil acceso.

Representan una manera económica (comparada con las redes de infraestructura) de transmitir datos, voz o instrucciones entre las terminales, al proponer un número reducido de estaciones base que son los puntos de acceso a Internet. Las terminales de una red mesh son parte activa de la red pues actúan tanto como terminales de usuario como encaminadores de otros dispositivos, resultando fácil ampliar la cobertura de la red mesh. La comunicación se basa en la determinación de nuevas rutas mediante saltos al siguiente nodo, lo que significa que los nodos actúan como repetidores transmitiendo la información de terminal en terminal hasta alcanzar a su destino. La ruta a seguir tendría que intentar ser la más corta y con la mejor condición de ancho de banda.

Las redes mesh permanecen en operación incluso cuando un nodo desaparece o un enlace presenta problemas bajo condiciones adversas, ya que los dispositivos se asisten unos a otros en la determinación de nuevas rutas. Este tipo de redes es bastante confiable ya que siempre brindan una ruta alternativa de comunicación a sus terminales.

Al contrario de las redes ad hoc, las redes mesh pese a heredar muchos de los resultados de las MANETS, tienen aplicaciones civiles como principal eje de desarrollo. De hecho, desde su origen basan su crecimiento en situaciones cercanas a la vida cotidiana, pese a que su nacimiento ocurrió hace unos años. La adecuación de las redes mesh a la vida diaria ha estimulado los trabajos de este tipo de redes, prometiendo un gran potencial en el mercado de las tecnologías inalámbricas.

Una red mesh establece una jerarquía en su arquitectura, implementando nodos dedicados (llamados encaminadores inalámbricos) que se comunican entre ellos y que proveen transporte inalámbrico de los datos que viajan hacia otros usuarios o puntos de acceso. Estos puntos de acceso son encaminadores inalámbricos con una conexión de banda ancha cableada hacia el *backbone* de Internet. La red de los encaminadores inalámbricos constituye un *backbone* inalámbrico totalmente integrado con la red mesh que permite la conexión multisalto entre los usuarios móviles con las puertas de enlace de la red fija o *gateways*.

El enlace entre los encaminadores inalámbricos y los puntos de acceso constituye un sistema de comunicación inalámbrico denominado *backhaul*, que brinda a cada terminal una interconexión con otros usuarios móviles multisalto, de bajo costo, de alto ancho de banda y con un número limitado de puntos de acceso a Internet. De forma particular en las redes mesh, el tráfico se origina en las terminales de los usuarios que atraviesa su *backbone* inalámbrico para ser distribuido a Internet. La figura 1.9. ilustra la arquitectura de una red mesh remarcando los diferentes niveles del sistema y sus componentes.

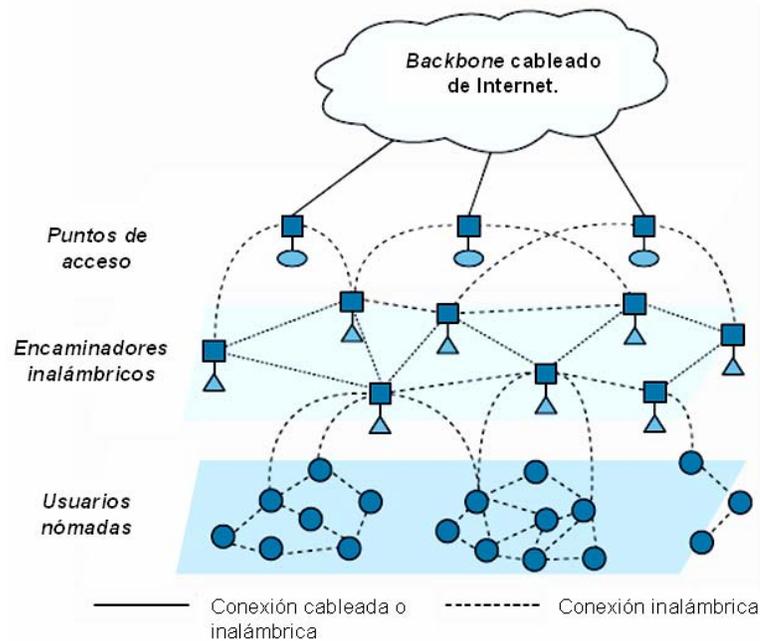


Figura 1.9.- Jerarquía de tres niveles para una red mesh.

A diferencia de las redes inalámbricas con infraestructura donde los usuarios comparten el ancho de banda que ofrece el punto de acceso, en las redes mesh a mayor número de dispositivos conectados dentro del área de cobertura, mayor será el ancho de banda disponible, considerando también que el número de saltos promedio de un enlace deberá reducirse.

Las redes mesh tiene las especificaciones de un emergente mercado para ser opciones viables en redes inalámbricas que son altamente escalables, con buena relación costo-beneficio para ofrecer soluciones de fácil conexión y alta velocidad con Internet y disponible en cualquier lugar.

Referencias:

[1]

A. Santamaría, F. J. López-Hernández, *Wireless LAN Systems*, Artech House, 1994.

[2]

P. Karn. *MACA a new channel access method for packet radio*. Proceedings of the 9th ARRL/CRRL Amateur Radio Computer Networking Conference, 1992.

[3]

Pahlavan Kaveh, Levesque Allen, *Wireless Information Networks*, Wiley series in telecommunications and signal processing, EUA., 1995.

[4]

Stojmenovic Iván, *Handbook of Wireless Networks and Mobile Computing*, John Wiley and Sons Inc., EUA, 2002.

[5]

P. Nicopolitidis, M. S. Obaidat, et al., *Wireless networks*, John Wiley and Sons, England, 2003.

[6]

Karl Holger, Willig Andreas, *Protocols and architectures for wireless sensor networks*, Hoboken, New Jersey, Wiley, 2005.

[7]

Bruno Raffaele, Conti Marco, Gregori Enrico, *Mesh Networks: Commodity Multihop Ad Hoc Networks*, IEEE Communications Magazine, National Research Council (CNR), 2005.

2. Generalidades de redes sin infraestructura o ad hoc

El punto de inicio de los estudios en redes de datos por conmutación de paquetes coincide con el interés de la DARPA (Defense Advanced Research Projects Agency) en la década de los 60's y el establecimiento de ARPANET en 1969, hoy conocido como Internet.

Actualmente, la proliferación de productos libres de licencia para las redes WLAN (Wireless Local Area Network) tiene la tendencia de incrementar la velocidad en las transmisiones. Este resultado es utilizado por la comunidad en el estudio de las redes ad hoc con el fin de implementar soluciones correctas para este tipo de redes. La comunidad que realiza estudios entorno a las redes móviles ad hoc es liderada por el grupo de trabajo MANET (Mobile Ad hoc NETWORKS) [1] creada en 1999 por el IETF (Internet Engineering Task Force).

El potencial de las aplicaciones basadas en redes ad hoc establece el reto de desarrollar protocolos eficientes para establecer rutas considerando las características propias de este tipo de redes. Por un lado, la efectividad de estos protocolos demanda que la información de la topología de la red sea lo más detallada y actualizada; Por el otro, en redes ad hoc, la topología se modifica con cierta frecuencia, lo que requiere un mayor y más frecuente intercambio de información de control entre los nodos, lo que resulta en un consumo mayor de energía que tienen siempre la limitante del tiempo de vida de las baterías en los equipos móviles.

En este capítulo describiremos algunas características y aplicaciones de las redes ad hoc para después enfocarnos en la descripción general de los protocolos de encaminamiento y centrarnos en los protocolos reactivos que justifican nuestro análisis de movilidad.

2.1. Características de las redes ad hoc

Con base en su abanico de aplicaciones, las redes ad hoc tienen características particulares. La RFC 2501 del grupo de trabajo MANET [2] establece las siguientes características relevantes de las redes AHNs (Ad Hoc Networks) [3], [4]:

- ❖ *Operación distribuida.* El concepto de ad hoc difiere de otros sistemas inalámbricos, tal como los sistemas celulares, en términos de su operación. Una red ad hoc posee terminales que tienen mismas características y responsabilidades. No existe una entidad central que controle la red, es por ello que los protocolos operan de manera distribuida.
- ❖ *Topología dinámica.* En una red inalámbrica ad hoc los nodos son libres de moverse en cualquier dirección. El hecho de que los nodos se localicen fuera del rango de transmisión de otro o que las condiciones del medio inalámbrico cambien, resulta en una topología de red dinámica.
- ❖ *Ancho de banda limitado.* Las restricciones impuestas por el canal inalámbrico como el acceso de múltiples equipos, interferencias, ruido, decaimiento de la señal y limitada disponibilidad de frecuencias, junto con los inherentes problemas del protocolo de acceso al medio, hacen que el *throughput* o cantidad de datos transmitidos en un periodo de tiempo de las terminales sea limitado.

-
-
- ❖ *Comunicación multisaltos.* Debido al decaimiento de la señal en el medio de transmisión se presenta una limitada área de cobertura. Es por ello que si un nodo S desea establecer una conexión con el nodo D fuera de su área de cobertura, necesitará encontrar una ruta al destinatario mediante saltos entre los nodos intermedios. Las rutas que se forman se mantienen de manera temporal para el caso de usuarios móviles, de ahí que será común el rompimiento de las rutas, haciendo necesario realizar nuevas búsquedas.
 - ❖ *Cambio en la calidad de los enlaces.* Esta característica está presente en todos los sistemas inalámbricos, sin embargo en la comunicación mutisaltos es importante pues la calidad de un trayecto depende de la calidad de todos los enlaces que lo constituyen.
 - ❖ *Limitada seguridad física.* Las redes inalámbricas móviles son susceptibles de ataques debido a sus posibles deficiencias en la seguridad que actualmente se aplica en las capas de enlace y de red. Por otro lado, la característica de ser una red descentralizada la hace menos vulnerable a fallas puntuales en la red. Adicionalmente, algunas redes que pueden llegar a ser grandes (decenas o cientos de nodos por área), aún necesitan mecanismos para escalar su tamaño, lo que resulta complicado siendo que muchos de los protocolos de encaminamiento presentan deficiencias durante este proceso.
 - ❖ *Dependencia del tiempo de vida de las baterías.* Pese a que ésta característica está presente en cualquier sistema inalámbrico móvil, en el caso de las redes ad hoc es de particular importancia. La dependencia de los nodos al comunicarse con los destinos cuando se encuentran fuera del rango del origen a través de otros nodos, vincula a los enlaces con el tiempo de vida de las baterías de los equipos.

Todas estas características generan un conjunto de lineamientos y parámetros de desempeño que orienta el diseño de los nuevos protocolos para las redes ad hoc.

2.2. Aplicaciones de las redes ad hoc

Las redes ad hoc ofrecen sólida comunicación en una variedad de ambientes hostiles como resultado de ser auto-organizadas y de no poseer una infraestructura fija. Su conjunto de aplicaciones es diverso; desde las redes pequeñas y estáticas enfocadas en reducir su consumo de energía, hasta las redes de gran escala y alto dinamismo que requieren comunicaciones estables y de fácil despliegue. Esto hace que las redes ad hoc sean una solución de comunicación táctica en operaciones militares, de seguridad, casos de siniestros, situaciones de emergencia, en el hogar, centros de convenciones, zonas de construcción, escuelas, laboratorios, entre otros. Detalles de estas aplicaciones son:

- ❖ Conferencias y reuniones de personas con equipos portátiles que desean intercambiar archivos sin la necesidad de infraestructura adicional entre ellos.
- ❖ Ambientes domésticos para comunicar dispositivos “inteligentes” que deban intercambiar información de control de su funcionamiento.

- ❖ Operaciones de búsqueda de emergencia y rescate en ambientes inhóspitos que demandan comunicaciones rápidas y dinámicas, como el caso de zonas afectadas por terremotos u otros desastres naturales y donde la infraestructura de comunicaciones ha sido destruida.
- ❖ Acciones militares de ataque y supervivencia en el campo de batalla junto con el apoyo de sistemas satelitales de órbita baja (LOS).
- ❖ Monitoreo y medición entre sensores esparcidos en un área geográfica para formar una red ad hoc autosostenible. El consumo energético y la escalabilidad, son puntos importantes de diseño en este tipo de redes siendo necesario un tiempo promedio de vida amplio de las baterías en las terminales y facilidad para incrementar el número de terminales. Algunos casos particulares de redes de sensores bajo un modo de operación ad hoc son:
 - ✓ Para detectar cambios en el ambiente descubriendo la presencia de material peligroso como gases veneno o radiación, explosiones, etc.
 - ✓ Para monitorear tránsito vehicular en autopistas o tránsito aéreo con problemas frecuentes de congestionamiento.
 - ✓ Como herramienta para ofrecer seguridad en centros comerciales, estacionamientos, tiendas autoservicio.

En el mercado existe la serie de productos BreadCrumb® pertenecientes a la marca Rajant (www.rajant.com) que se despliegan como una red de tipo ad hoc. Enseguida mostramos en la figura 2.1. algunos equipos BreadCrumb® y un ejemplo en una zona en conflicto.

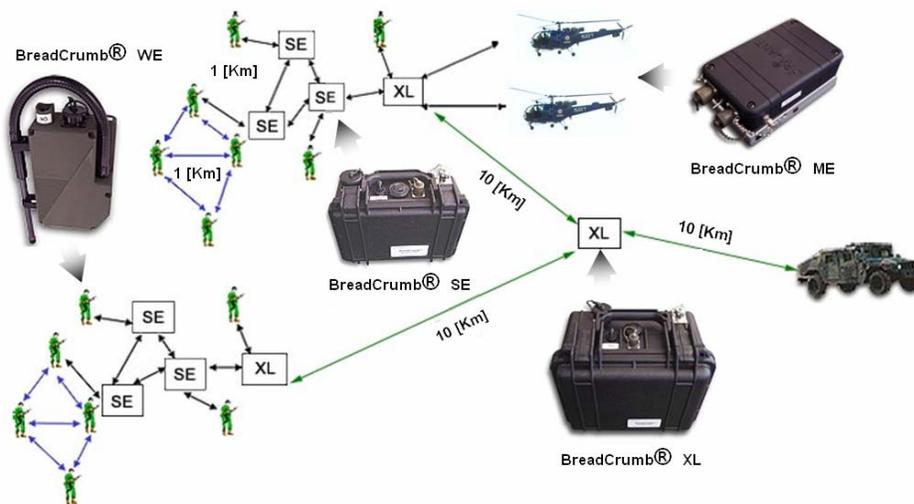


Figura 2.1.- Despliegue de una red ad hoc móvil.

Los equipos de Rajant se han utilizado en las operaciones de Estado Unidos en Irak y por las brigadas durante las acciones de rescate en la zona devastada por el tsunami en diciembre de 2004. En esa ocasión, los equipos fueron utilizados para desplegar una red de comunicación mediante voz sobre IP y teleconferencia para los damnificados y con ello facilitar las acciones de salvamento.

2.3. Descripción de protocolos de encaminamiento

El encaminamiento en redes inalámbricas sin infraestructura es diferente al encaminamiento en las redes cableadas convencionales, principalmente debido a la movilidad de las terminales que causa la ruptura y la reparación de los enlaces, en función de los parámetros de desempeño en la estabilidad de la ruta y del consumo de energía de las terminales.

Los protocolos para redes ad hoc asumen que todas las terminales poseen capacidades iguales al momento de ejecutar las tareas relacionadas con el encaminamiento tales como el descubrimiento y establecimiento de las rutas hacia nuevos nodos y su mantenimiento. Los protocolos de encaminamiento para redes ad hoc se dividen principalmente en proactivos (*table-driven*), reactivos (*on-demand*) e híbridos. Los últimos combinan las ventajas de ambas familias dependiendo de la movilidad, tamaño y del tráfico en la red ad hoc.

Los protocolos proactivos evalúan continuamente el desempeño de una red registrando las rutas disponibles para el momento en que un paquete necesite ser enviado, de esta manera la ruta a utilizar es conocida para poder ser utilizada de inmediato. Bajo este principio, todos los nodos conocen la topología de la red en todo momento independientemente de que algunas de estas rutas puedan o no ser utilizadas en alguna conexión. Para lograr esto, los nodos envían de forma consecutiva a sus vecinos los posibles cambios en la topología de la red. A medida que el número de terminales se incrementa, mayor tendrá que ser la memoria y el procesamiento en las terminales pues más largos serán los paquetes de control. La mejor ruta se obtiene haciendo uso de algoritmos que las identifican de acuerdo a una métrica establecida, misma que refleja el objetivo del operador de la red. La métrica puede ser el número de saltos, el costo, el retardo o el ancho de banda disponible. La decisión del mejor camino se hace con base en las métricas para cada enlace contenidas en la tabla de encaminamiento. Dos familias de protocolos proactivos son los algoritmos Vector de Distancia (*Vector Distance*) y Estado de los Enlaces (*Link State*).

En la técnica de encaminamiento de Vector Distancia, los dispositivos intercambian con sus vecinos su respectiva tabla de encaminamiento donde se indica el conjunto de rutas y el valor preponderante en cada una de las métricas consideradas. En este algoritmo Vector Distancia los cambios en la topología de la red se realizan de forma gradual a medida que se propaga la información relativa. El protocolo DSDV (Destination-Sequenced Distance Vector) es un ejemplo de protocolo proactivo en una red de tipo ad hoc. Por su parte en los algoritmos basados en el protocolo Estado de los Enlaces, los dispositivos envían información sobre el estado de cada uno de los enlaces que los conectan con sus vecinos, lo que les permite construir un mapa de toda la red a partir del cual es posible obtener su tabla de encaminamiento.

En contraste, los protocolos reactivos o sobre demanda (*on-demand protocols*) utilizan el procedimiento para determinar una ruta basada en una solicitud denominada bajo demanda. El descubrimiento reactivo de rutas comienza con una solicitud de llamada que utiliza inundaciones o *flooding* [5] con el fin de alcanzar al destino deseado. Es así como el descubrimiento de rutas mediante protocolos sobre demanda resulta en un menor tráfico haciéndolos más atractivos al compararlos con los protocolos proactivos. Sin embargo, la dependencia de una inundación de paquetes de los esquemas reactivos involucra un considerable tráfico de paquetes de control dentro de los ambientes ad hoc. Adicionalmente, debido al gran incremento de tráfico de control durante el descubrimiento de las rutas, el retardo generado por el proceso de descubrimiento de rutas en los protocolos reactivos puede ser significativamente alto [6].

2.3.1. Inundación de la red para la localización de usuarios

La inundación o *flooding* es la más simple de las soluciones de encaminamiento pues hace uso de la fuerza bruta sin utilizar cualquier conocimiento de la topología de la red [7]. En este caso, si un nodo A tiene un paquete para el nodo B, la terminal A simplemente hará un *broadcast* o envío por difusión del paquete en donde cada elemento distinto de B lo reenviará nuevamente, inundando literalmente la red con este paquete. Finalmente B, de estar en la red, recibirá el paquete después de varios saltos; y en caso de no estar, la red se habrá cargado de paquetes inútiles. La figura 2.2. describe el proceso de una inundación de paquetes en una red ad hoc.

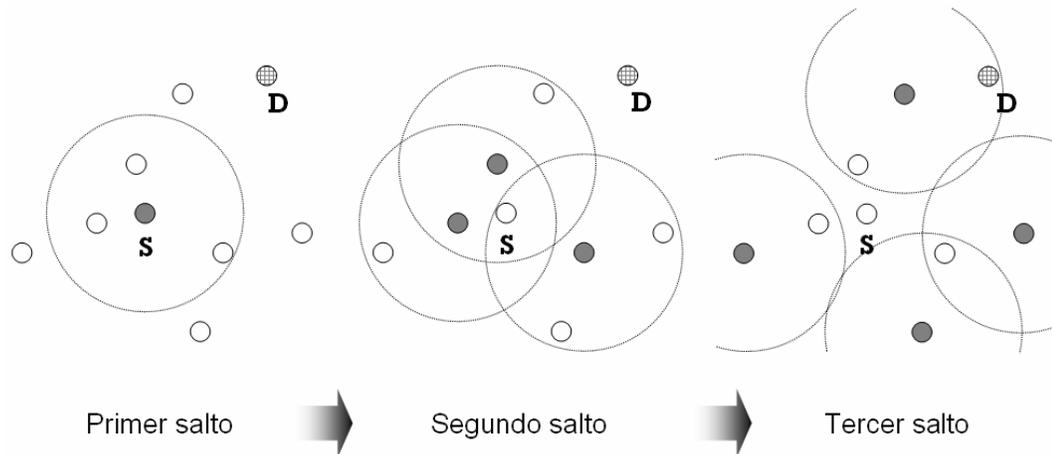


Figura 2.2.- Inundación de paquetes o *flooding*.

La técnica de *flooding* resulta muy efectiva cuando no se dispone de la información deseada en las tablas de encaminamiento, que es el caso de la inicialización del sistema o cuando se busca robustez como en las redes militares. De forma adicional resulta efectiva cuando una terminal necesita enviar un paquete a todas las estaciones conectadas a la red; es simple de implementar y no requiere alta capacidad de cómputo. Sin embargo la inundación de paquetes puede congestionar fácilmente la red debido a que un paquete genera otros más, los cuáles a su vez forman múltiples copias, produciendo un aumento exponencial del tráfico en la red ad hoc [8]. En la técnica de inundación de paquetes, los dispositivos reciben y retransmiten paquetes utilizando gran parte de sus limitados recursos en cuanto a energía y ancho de banda se refiere. Resulta entonces claro que el número de paquetes debe controlarse, para conseguir esto se utilizan varios mecanismos.

De los mecanismos más recurridos está el uso de un campo de tiempo de vida en cada paquete. Cuando el origen envía un paquete, el valor inicial de este campo se hace igual a un número muy pequeño y cada dispositivo siguiente decrementa este campo antes de retransmitir el paquete. Cuando el campo alcanza el valor cero, la terminal desecha el paquete. Idealmente el valor inicial que debe tomar este campo dependerá del número mínimo de saltos entre dos nodos más alejados de la red, lo que se conoce como diámetro de la red.

Otro método de eliminación de paquetes consiste en que cada terminal añade su identificador a la cabecera del paquete antes de transmitirlo en la inundación. Cuando una terminal recibe un paquete que contiene su identificador, lo rechazará, evitando que los paquetes entren en ciclos que consuman recursos de forma innecesaria. Una manera alterna de este método es la identificación de los paquetes por su origen, destino y número de secuencia, los cuales se

almacenan de forma temporal en la memoria de la terminal, que al momento de recibirlos decide eliminar o retransmitir el paquete.

Desde otro punto de vista, bajo el funcionamiento de MAC 802.11 con prevención de colisiones, aquella terminal que desee transmitir un paquete deberá esperar a que se desocupe el canal cuando exista una transmisión ajena, ésta es la razón por la que en una inundación de paquetes habrán retardos y o posibles colisiones con otras transmisiones en curso, situación que se refleja en la eficiencia de la red. Aunado a esta situación, debido a la movilidad de las terminales, las rutas pierden validez y por lo tanto el mecanismo de inundación para encontrar una nueva ruta impactará nuevamente en sus condiciones.

Algunos protocolos propuestos llevan a cabo la inundación de la red de paquetes de control en lugar de paquetes de datos que son más grandes. Estos paquetes de control se utilizan para descubrir las rutas dentro de la red que posteriormente serán incluidas en las tablas de encaminamiento de las terminales.

2.4. Algoritmos de encaminamiento reactivos o bajo demanda

Este tipo de algoritmos se caracterizan por eliminar las tablas convencionales de encaminamiento utilizadas por los algoritmos proactivos en donde se tiene la información de la topología de toda la red en todo momento.

Los protocolos bajo demanda establecen un camino justo antes de realizar la transmisión. Si las terminales no generan tráfico de datos entonces no hay actividad de encaminamiento, de ahí su nombre de reactivos ante una transmisión. Los procedimientos característicos de este tipo de protocolos son:

- ❖ Descubrimiento de rutas.
- ❖ Mantenimiento de rutas.
- ❖ Eliminación de rutas.

El descubrimiento de rutas se basa en la inundación de preguntas y respuestas entre los nodos de la red. Cuando el destino es eventualmente localizado, genera una respuesta dirigida al nodo origen. El descubrimiento de una ruta se inicia con su solicitud, cuando sea necesaria una transmisión y no se conozca ruta alguna al destino. Por otro lado el descubrimiento de rutas no es necesario en cada transmisión de paquetes, puesto que las rutas se mantienen durante un periodo de tiempo lo que permite realizar varias transmisiones sucesivas al mismo destino.

Una vez realizado el descubrimiento de una ruta, las terminales adquieren nueva información sobre el encaminamiento de los paquetes, misma que es almacenada hasta que ya no sea requerida o hasta que se decida ser eliminada.

A continuación se hace una descripción de las características y del funcionamiento de algunos protocolos de encaminamiento reactivo o bajo demanda.

2.4.1. Encaminamiento ad hoc por Vector Distancia sobre Demanda (AODV)

El protocolo AODV [9], [10] es un protocolo de encaminamiento dinámico que por medio de múltiples saltos entre nodos móviles establece y mantiene rutas en una red ad hoc utilizando un encaminamiento salto a salto. Está basado en el protocolo proactivo DSDV (Destination Sequence Distance Vector) pero difiere de éste por ser un protocolo reactivo lo que significa que realiza la solicitud de una ruta sólo cuando es necesario y no requiere que los nodos almacenen rutas hacia destinos que no participan en la comunicación activa [6].

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 participan en una comunicación activa. Este protocolo permite que los nodos respondan a rompimientos y cambios en la topología de red en un tiempo razonable.

En la operación de AODV no se presentan los ciclos reiterativos y mediante la eliminación del problema “cuenta al infinito”, brinda una rápida convergencia ante un cambio en la topología de la red que típicamente sucede cuando un nodo se mueve. Cuando se pierde un enlace, AODV se encarga de notificar a los nodos para que invaliden la ruta a través del mismo enlace perdido. AODV recurre a un número de secuencia al destino cada vez que se recaba información de la ruta que es un número creado por el destinatario incluido junto con toda la información de la ruta enviada a los nodos iniciadores de su establecimiento. Al utilizar estos números de secuencia se anulan los ciclos reiterativos, además de facilitar su programación. Cuando se tengan dos rutas a un solo destino, AODV demanda al nodo origen que elija aquella con el número de secuencia mayor.

El procedimiento para determinar una ruta válida mediante el protocolo DSDV es el siguiente: cuando un nodo S quiere mandar un paquete al nodo D, éste primero verifica si su tabla de encaminamiento posee una ruta al nodo D. Si S tiene una ruta hacia D envía el paquete al siguiente nodo perteneciente a la ruta hasta llegar a D. En caso de que no haya una ruta hacia D, S inicia el proceso de su descubrimiento. El nodo S hace una inundación en la red del mensaje RREQ (Route Request) que contiene la dirección origen, destino, número de secuencia e identificador de *broadcast*. Una vez enviado el mensaje, S inicia un contador de tiempo en espera de alguna respuesta. Los nodos reciben el mensaje RREQ y verifican primero si no ha sido recibido el mismo mensaje con anterioridad, prestando atención a la dirección de origen y al identificador de *broadcast* del mensaje. De ser el caso, descartan el mensaje, de lo contrario transmiten un mensaje de respuesta de ruta (RREP) hacia el nodo origen S.

Un nodo puede enviar un mensaje RREP en caso de tener una referencia al origen en su tabla de encaminamiento que no haya expirado, lo que le permite conocer la ruta más reciente. Para saber si una ruta conocida a un nodo intermedio es nueva, se utiliza un número secuencial de destino. A un nuevo mensaje RREQ se le asigna un nuevo número secuencial de destino más elevado. Si el nodo que envía el mensaje RREP no lo puede hacer, incrementa el número secuencial del mensaje RREQ y luego lo difunde a sus vecinos.

Cuando el destino recibe el mensaje RREQ responde con el envío de un mensaje RREP. Cuando un nodo intermedio recibe el mensaje RREP lo reenvía al destino de acuerdo a la ruta que posea en su tabla de encaminamiento, por lo que el mensaje RREP se transporta de regreso en el camino generado por RREQ. La terminal origen S comienza la transmisión de información tan pronto como el primer mensaje RREP sea recibido, para más tarde actualizar la ruta si se

descubre una mejor opción. En cualquier caso las trayectorias de ida o regreso de los mensajes son eliminadas si no se utilizan dentro de su tiempo expiración, sin embargo cada vez que son utilizadas se reinicia.

Como se muestra en la figura 2.3., el mensaje RREQ proveniente del nodo origen S es inundado a toda la red hasta alcanzar al nodo destino D (ver figura a). En su camino por la red, RREQ crea referencias temporales en las tablas de encaminamiento de todas terminales a su paso para crear el camino de regreso (ver figura b). Cuando RREQ llega al destino, este envía el mensaje RREP por la ruta de regreso a través de los nodos intermedios (ver figura c) que indican el camino al nodo que transmitió el mensaje RREP.

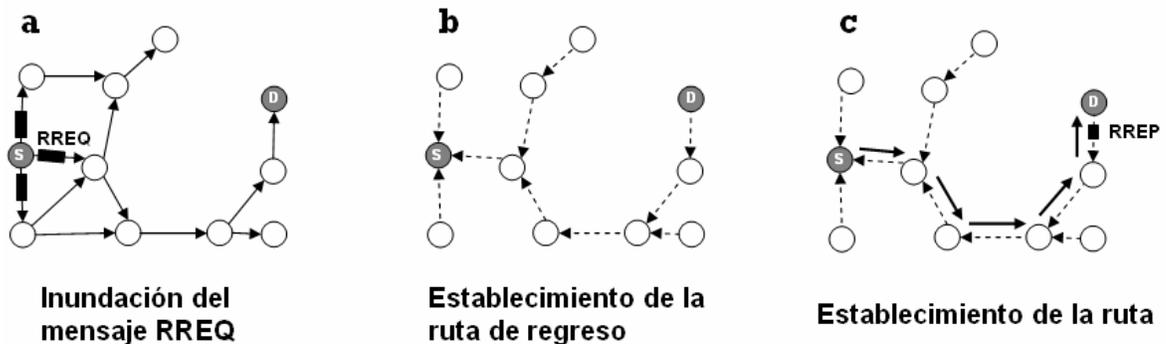


Figura 2.3.- Ejemplo de un descubrimiento de ruta mediante el protocolo AOVD.

Una ruta descubierta por el protocolo AOVD se mantiene de la siguiente forma: si una ruta deja de ser válida porque el origen se movió, éste mismo inicia la búsqueda de una nueva ruta. Si deja de existir por el movimiento del destino o de algún nodo intermedio, el último nodo de la ruta validada envía un mensaje de error RERR con destino al nodo origen para que al llegar y en caso de aún requerirla, reinicie el descubrimiento de la nueva ruta. La información sobre la vecindad de una terminal se obtiene por medio de paquetes *Hello* periódicamente transmitidos a los nodos vecinos. En el instante en que una terminal recibe un paquete *Hello*, actualiza la información de su tabla de encaminamiento asociada con el vecino que le envió este mensaje. Otra manera de mantener la información referente al estado de los vecinos de una terminal es escuchando las retransmisiones de paquetes para asegurar que él o los nodos a un salto se encuentran aún en su rango de transmisión como ocurre en la implementación de AOVD en los protocolos IEEE 802.11.

2.4.2. Algoritmo temporalmente ordenado (TORA)

El protocolo TORA [11] pertenece a la familia de los protocolos denominados como *link reversal*. TORA está diseñado para reaccionar de forma óptima a los cambios de topología y al seccionamiento de la red. Su nombre se deriva de asumir relojes sincronizados para ordenar los eventos que ocurren en la red.

El encaminamiento óptimo (menor número de saltos y mayor ancho de banda) es una cuestión secundaria considerada por el protocolo TORA, cuyo principal objetivo es encontrar rutas estables que puedan ser reparadas de forma local y rápida. TORA construye una trayectoria directa acíclica al destino o DAG (*direct acyclic graph*) que se obtiene asignando una dirección lógica a los enlaces con base en la altura o valor relativo de una métrica asociada a una terminal.

Si (i, j) es un enlace directo del DAG, i corresponde al nodo siguiente de subida (*upstream*) y j es el nodo siguiente de bajada (*downstream*). La métrica de un nodo puede pensarse de forma conceptual como la “altura del nodo”, por ejemplo, los enlaces están dirigidos desde el *router* más alto al *router* más bajo. Los enlaces entre un nodo y cualquier otro dentro de su vecindad con una altura desconocida o no definida se consideran “no direccionados” y no se utilizan en la transmisión de los paquetes. Las alturas de los nodos y las direcciones lógicas de los enlaces forman una estructura de encaminamiento libre de ciclos (acíclica) y de trayectorias múltiples en la que todas las rutas llevan hacia abajo (*downstream*) o hacia el nodo destino como se muestra en la figura 2.4., en donde se resalta el caso en el que el nodo C se encuentra más cercano al destino que el nodo B en cuanto a número de saltos, pero la altura de C es mayor que la de B.

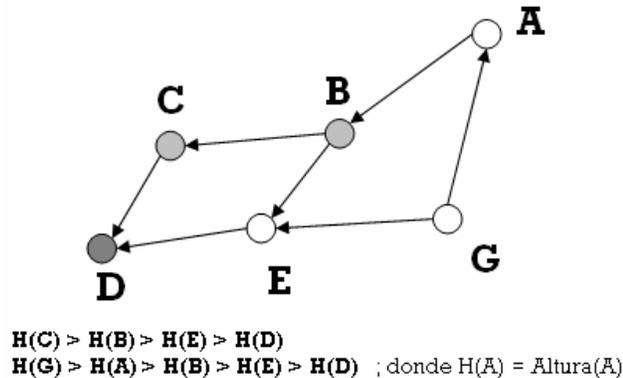


Figura 2.4.- Gráfica acíclica dirigida definida por la altura de las terminales.

La operación del protocolo TORA puede dividirse en 4 fases: descubrimiento, mantenimiento, optimización y eliminación de una ruta.

El descubrimiento de una ruta se realiza con el intercambio de paquetes de control de pregunta-respuesta (query-reply). Durante la transmisión de los paquetes de respuesta, que se realiza con su inundación, a los enlaces se les asigna una dirección lógica (de bajada o de subida) basada en la altura asignada a los nodos. Esta forma de encaminamiento puede verse como una red de tuberías por donde el agua corre por acción de la fuerza de gravedad hasta llegar al nodo destino.

El mantenimiento de la ruta consiste en adaptar la estructura de encaminamiento como respuesta a cambios en la topología de la red. Por ejemplo, si desaparece un nodo en un enlace de bajada, algunas trayectorias dirigidas pueden ya no llevar al nodo destino, este hecho dispara una secuencia de revocación del enlace causada por el reestablecimiento de las alturas de los nodos, mismas que reorientan la estructura del encaminamiento para que las nuevas rutas lleven de nuevo al destino. Cuando una red se secciona, los enlaces en la sección de la red que se ha separado, se marcan como no dirigidos con el fin de borrar las rutas no válidas. La eliminación de una ruta ocurre cuando los nodos ajustan sus alturas a cero y sus enlaces adyacentes se vuelven no dirigidos.

2.4.3. Protocolo de fuente dinámica para redes móviles DSR

El protocolo DSR [12] utiliza el denominado encaminamiento de fuente, por lo tanto, en DSR cada paquete que necesita ser encaminado lleva en su cabecera la lista ordenada de las terminales que constituyen la ruta sobre la cual será transmitido, es por ello que los nodos intermedios no

necesitan almacenar información de encaminamiento, pues los paquetes por sí mismos poseen la suficiente información para ser encaminados. Esta característica elimina el aviso periódico de rutas, así como paquetes de detección de vecinos utilizados en otros protocolos.

Por otro lado, en DSR los códigos adicionales para control y verificación de error (*overhead*) son mayores, pues cada paquete debe contener la secuencia completa de nodos incluidos en la ruta, de allí que DSR será más eficiente en el caso de redes con diámetro pequeño.

El protocolo DSR incluye los procesos de descubrimiento y mantenimiento de las rutas. Un nodo que desea establecer una conexión con otro inicia el descubrimiento de la ruta difundiendo un paquete RREQ. Este paquete lo recibe cada uno de sus vecinos que a su vez lo reenvían a sus propios vecinos. Un nodo reenvía un paquete RREQ sólo si éste no ha sido previamente recibido en este nodo o si su dirección no forma parte de la ruta. El paquete RREQ inicia un paquete RREP en el momento en que el RREQ se recibe en el nodo origen o en un nodo intermedio que conoce la ruta hacia el destino.

Hasta la llegada del mensaje RREQ al destino o a un nodo intermedio que conoce la ruta al destino, el paquete mantiene la secuencia de nodos que constituyen la ruta. Esta información se inserta para ser transportada dentro del mensaje RREP y estar disponible en el nodo origen al final del proceso. DSR incluye tanto enlaces simétricos como asimétricos, es por ello que el mensaje RREP puede ser enviado por el mismo trayecto del RREQ o, una vez generado por el nodo destino, iniciar el descubrimiento de su propio trayecto hacia el nodo origen y transportar el RREP en un nuevo mensaje RREQ. La figura 2.5. muestra el esquema de un ejemplo de descubrimiento de ruta mediante DSR del nodo A al nodo B.

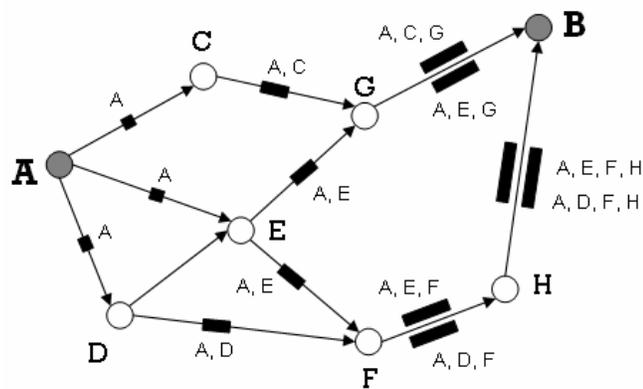


Figura 2.5.- Descubrimiento de una ruta desde A hasta B con el protocolo DSR.

Con el fin de limitar los mensajes adicionales u *overhead*, cada nodo almacena en su memoria temporal las rutas recién utilizadas o incluidas por los mensajes de control. Después de que un nodo hace la petición de una ruta, todas las posibles rutas generadas por el proceso RREQ se almacenan en su memoria temporal.

El nodo origen inicia el mantenimiento de una ruta cuando detecta un cambio en la topología de la red, pues los paquetes ya no pueden llegar al nodo destino. En esta situación el nodo origen intenta utilizar una ruta alterna en el caso que esté disponible en su memoria temporal, o reinicia el descubrimiento de una nueva ruta. El hecho de almacenar una ruta alterna en la memoria temporal evita iniciar el proceso para descubrir una nueva ruta agilizando su reestablecimiento.

Debido a que el mantenimiento de una ruta se inicia sólo cuando ocurre una falla en el enlace, DSR no utiliza transmisiones periódicas en el encaminamiento de paquetes, resultando menor señalización de control y mayor ahorro de energía en los nodos móviles.

2.5. Utilidad del análisis de movilidad en redes ad hoc

Como hemos visto, para evaluar el desempeño de los protocolos de red resulta conveniente iniciar el análisis bajo condiciones controladas en donde nuestro análisis se centra en el movimiento de las terminales. Utilizar algoritmos complejos de movimiento en una primera evaluación de los protocolos o sus propuestas puede provocar el desconocimiento de los resultados reales de la propuesta, es por ello que resulta conveniente establecer en los simuladores de redes algoritmos generales y simples que rijan el comportamiento dinámico de las redes para poder evaluar los resultados de las nuevas propuestas [13].

Por otro lado está el esfuerzo requerido en la elaboración y sustento de las propuestas; en forma específica al tiempo de desarrollo y de simulación de los algoritmos planteados, pues no siempre contamos con una alta capacidad de procesamiento en los equipos de cómputo donde se realizan las pruebas ya que los recursos de cada proyecto son limitados.

El conocimiento del patrón de movimiento a través de un modelo matemático es la forma más general de describir el fenómeno. La relación de las variables de los escenarios como la densidad de terminales, sus velocidades máximas y mínimas y los radios de conexión de los nodos son los que determinan las conexiones durante la simulación.

En la literatura encontramos el término *dwell time* o tiempo de residencia [14], [15], es una variable aleatoria que representa el tiempo total en el que una estación base mantiene un enlace de comunicación satisfactorio con una terminal móvil. El *dwell time* de una estación base depende de varios factores como son: la velocidad, la trayectoria, la potencia de transmisión, la propagación de la señal y sus interferencias. El tiempo de residencia tiene un papel importante en la evaluación y diseño de las redes inalámbricas. El tiempo en el que el canal de comunicación se mantiene ocupado puede obtenerse una vez que se determina el tiempo de residencia. El registro de este tiempo es útil cuando se desean calcular los algoritmos de señalización y en otras aplicaciones que utilizan este tiempo de residencia para conocer el desempeño de la red.

En este trabajo realizamos el análisis del tiempo de residencia dentro de una zona de cobertura circular de un nodo móvil en un escenario ideal; con una propagación isotrópica de la señal y libre de interferencias.

Referencias:

[1]

<http://www.ietf.org/html.charters/manet-charter.html>

[2]

Corson M.S., Macker J., *Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations*, Request for Comments 2501, Internet Engineering Task Force, 1999.

[3]

P. Nicopolitidis, M. S. Obaidat, et. al., *Wireless networks*, John Wiley and Sons, England, 2003.

[4]

Redondo Juan F., *Ad Hoc Networks – design and performance issues*, Helsinki University of Technology, Espoo, 2002.

[5]

Mohammad Ilyas, *The handbook of ad hoc wireless networks*, Florida Atlantic University Boca Raton, EUA, CRC Press, 2003.

[6]

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

[7]

P. Nicopolitidis, M. S. Obaidat, et al., *Wireless networks*, John Wiley and Sons, England, 2003.

[8]

León-García Alberto, Widjaja Indra, *Redes de comunicación*, Mc Graw Hill, Madrid, 2002.

[9]

Perkins C.E., *Ad Hoc On Demand Distance Vector (AOVD) Routing*, Internet draft, 1997.

[10]

Perkins C.E., Royer E.M., *Ad Hoc On Demand Distance Vector Routing*, Proceedings of the 2nd IEEE WMCSA'99, 1999.

[11]

Park V. D., Corson M. S., *A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks*, IEEE INFOCOM'97, Kobe Japan, 1997.

[12]

Johnson David B., Maltz David A., et al., *The dynamic source routing protocol for Mobile Ad-hoc Networks*, IETF MANET Working Group, Internet draft, 2000.

[13]

Xiaolong Li, Dharma P. Agrawal, Qing-An Zeng, *Impact of Mobility on the Performance of Mobile Ad Hoc Networks*, University of Cincinnati, Cincinnati, IEEE 2004.

[14]

Orlik Philip, Rappaport Stephen S., *Traffic performance and mobility modeling of cellular communications with mixed platforms and highly variable mobilities*, State University of New York, Stony Brook, New York, IEEE 1997.

[15]

Jabbari Bijan, Zhou Yong, Hillier Frederick, *Random Walk Modeling of Mobility in Wireless Networks*, George Mason University, Fairfax, VA, USA, IEEE 1998.

3. Análisis de movilidad mediante simulación

Como herramienta para el análisis de redes, los simuladores permiten implementar modelos, interpretaciones o teorías expresándolas en lenguaje simbólico con la finalidad de entender el comportamiento del sistema o de evaluar las estrategias de operación del mismo. Entre los simuladores de redes más conocidos están OPNET, BONEs, REAL, ns-2, NIST Net, PARSEC, GloMoSim, entre otros; cada uno con sus respectivas ventajas sobre el resto de ellos.

De esta forma la importancia de los simuladores de redes radica en identificar características de operación, verificar modelos y encontrar alternativas de solución junto con su justificación para ser evaluadas y definir el programa de simulación a utilizar.

En este capítulo daremos una descripción del simulador que utilizamos, el Network Simulator 2 (ns-2), además explicaremos las condiciones de los escenarios simulados y terminaremos con los detalles del procedimiento de nuestras simulaciones.

3.1. Generalidades del simulador Network Simulator (ns)

El simulador ns-2 con su versión más reciente la 2.29, recrea las condiciones específicas de una red de datos ayudando a estudiantes, académicos e investigadores de todo al mundo en el estudio de las redes de computadoras. El ns-2 es una herramienta libre de licencia que fue desarrollada en 1989 por la Universidad de California en Berkley como parte del proyecto REAL Simulator. Actualmente es desarrollado por investigadores del Network Research Group en las instalaciones del Lawrence Berkeley National Laboratory, apoyados por grupos de trabajo de la Universidad de Cininnati y por especialistas de Sun Microsystems en los Estados Unidos [1].

El simulador ns-2 está enfocado a la investigación de redes; ofrece soporte para la simulación de protocolos a nivel aplicación (HTTP, Telnet), presentación, sesión, nivel transporte (UDP, TCP, RTP, SRM), de red, de enlace y físico (MAC) [2]. En otras palabras, este simulador adiciona protocolos tanto para redes cableadas como para inalámbricas, para redes locales y satelitales, como los que soportan redes de tipo ad hoc. Su versatilidad radica en que puede ser utilizado bajo diferentes sistemas operativos como son Linux (Mandrake, Debian, Suse, Red Hat, Fedora Core), FreeBSD, SunOS, Solaris y Windows9x/2000/XP. Su carácter gratuito es la mayor justificación para ser utilizado en el desarrollo de este trabajo.

Ns-2 está basado en dos lenguajes de programación que se integran al momento de compilar las simulaciones del usuario: un simulador orientado a objetos escrito en C++ y un intérprete para OTcl (Object Tool Command Language) que es la extensión orientada a objetos de Tcl. Desde otro punto de vista, ns-2 posee dos clases de jerarquías bajo una correspondencia uno a uno entre sus elementos; una compilada en C++ y otra interpretada por OTcl. Gracias a la jerarquía compilada en C++ se logra eficiencia en la simulación y tiempos de ejecución menores, que es particularmente útil para la operación y definición detallada de los protocolos, siendo que además permite reducir tiempo de procesamiento [3], [4]. Por otro lado, el intérprete de OTcl facilita al usuario definir la topología particular de la red, los protocolos utilizados y las aplicaciones que se desean simular, así como la forma de expresar los resultados al concluir la simulación. OTcl utiliza los objetos compilados en C++ por medio de una solicitud que genera la relación del objeto OTcl con el correspondiente generado en C++, como se muestra en el esquema de la figura 3.1.

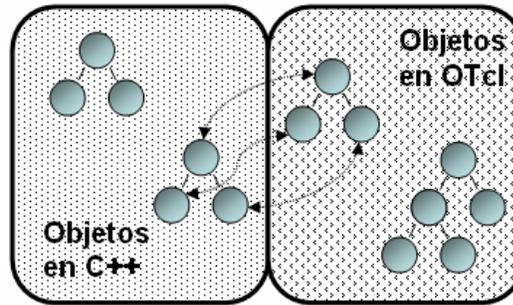


Figura 3.1.- Dualidad entre objetos de C++ y de OTcl.

Por otro lado, ns-2 es un simulador en tiempo discreto en el que existe una secuencia definida de eventos programados. Cada evento corresponde a un objeto dentro de la jerarquía de C++ con un identificador único, un tiempo reservado y una referencia al objeto que controla al evento. La lista de eventos para su ejecución sigue una estructura ordenada y se realiza uno a uno bajo el control del manejador de eventos de ns-2.

En cuanto sus funciones específicas en redes ad hoc, ns-2 ha sido enriquecido con los siguientes protocolos de encaminamiento: Destination-Sequenced Distance Vector (DSDV), Dynamic Source Routing (DSR) y Temporally-Ordered Routing Algorithm (TORA) que han sido contribuciones de la Carnegie Mellon University (CMU) y el Ad hoc On Demand Distance Vector (AODV) como contribución de la Universidad de Cincinnati. La descripción de estos protocolos fue desarrollada en el capítulo anterior.

Respecto al algoritmo que define el movimiento de las terminales, en ns-2 están definidos dos esquemas: el determinístico y el estocástico. En el esquema determinístico, el movimiento de los nodos queda del todo definido por la posición inicial, la posición final y su velocidad. En el esquema estocástico, en donde las condiciones futuras no dependen de las condiciones actuales del escenario propuesto, ns-2 incorpora modelos de movimiento por entidades [5] del tipo *random walk*, el tipo *random waypoint* [6] determinado por tiempos de espera y del tipo *random direction*.

A continuación se muestran las tablas 3.1. y 3.2. La primera contiene las características de cada modelo de movimiento mencionado y la segunda algunos modelos de movimiento por entidad y grupal, considerando que no todos están incorporados en ns-2 [7].

Random walk	<p><i>Parámetros:</i></p> <ul style="list-style-type: none"> ✓ Velocidad mínima. ✓ Velocidad máxima. ✓ Dirección entre $[0, 2\pi]$. ✓ Tiempo o distancia transversal. <p><i>Descripción:</i></p> <p>Se mueve el nodo en línea recta con una dirección establecida por un ángulo entre 0 y 2π, parando al satisfacer la condición de tiempo o distancia transversal para repetir la secuencia.</p>
--------------------	---

<p>Random waypoint</p>	<p><i>Parámetros:</i></p> <ul style="list-style-type: none"> ✓ Velocidad mínima. ✓ Velocidad máxima. ✓ Destino aleatorio (x, y). ✓ Tiempo de pausa mayor o igual a cero. <p><i>Descripción:</i></p> <p>Se mueve el nodo en línea recta en dirección de la posición aleatoria destino donde se detiene y repite la misma secuencia.</p>
<p>Random direction</p>	<p><i>Parámetros:</i></p> <ul style="list-style-type: none"> ✓ Velocidad mínima. ✓ Velocidad máxima. ✓ Dirección entre $[0, \pi]$ y al iniciar el movimiento entre $[0, 2\pi]$. ✓ Tiempo de pausa. <p><i>Descripción:</i></p> <p>Se mueve el nodo en línea recta con una dirección establecida hasta que alcanza los límites del escenario para detenerse y seleccionar una nueva dirección.</p>

Tabla 3.1.- Descripción de modelos de movimiento por entidades.

Modelos de movilidad de entidades.	Modelos de movilidad grupal
<ul style="list-style-type: none"> ✓ <i>Random Walk</i> ✓ <i>Random Waypoint</i> ✓ <i>Random Direction</i> ✓ <i>Boundless Simulation Area</i> ✓ <i>Gauss-Markov</i> ✓ Versión de probabilidad del modelo <i>Random Walk</i> ✓ <i>City Section</i> 	<ul style="list-style-type: none"> ✓ <i>Exponencial correlacionado</i> ✓ <i>Column</i> ✓ <i>Nomadic Community</i> ✓ <i>Pursue</i> ✓ <i>Reference Point Group</i>

Tabla 3.2.- Modelos de movilidad por entidad y grupal.

3.1.1. Esquema de simulación en ns-2

Una simulación en ns-2 consiste en un programa o script OTcl que contiene la topología de red estableciendo el número de nodos del escenario, el tipo de enlace inalámbrico, las dimensiones físicas del escenario, los protocolos de encaminamiento a utilizar y los nombres de los archivos de salida para el manejo de los resultados y en su caso de la animación. Adicionalmente es necesario crear los archivos *tráfico* y *escenario* que acompañen a dicha topología.

El archivo *tráfico* contiene las especificaciones del intercambio de los paquetes como son el tiempo de inicio y fin de la transmisión, la tasa de transferencia, el identificador de los nodos origen, los tiempos de pausa, entre otras. Por otro lado el archivo escenario define las posiciones iniciales, las sucesivas y las velocidades de desplazamiento de cada uno de los nodos en los tiempos discretos; un resumen con el número de posiciones destino no alcanzables y el número total de rutas y de conexiones de cada una de las terminales.

Una vez ejecutado el script OTcl o programa de simulación, ns-2 crea archivos específicos para el análisis de resultados cuyas extensiones corresponden a *.nam y *.tr o alguna otra definida de forma particular por el usuario. El esquema de la figura 3.2. resume el proceso de construcción de una simulación en ns-2.

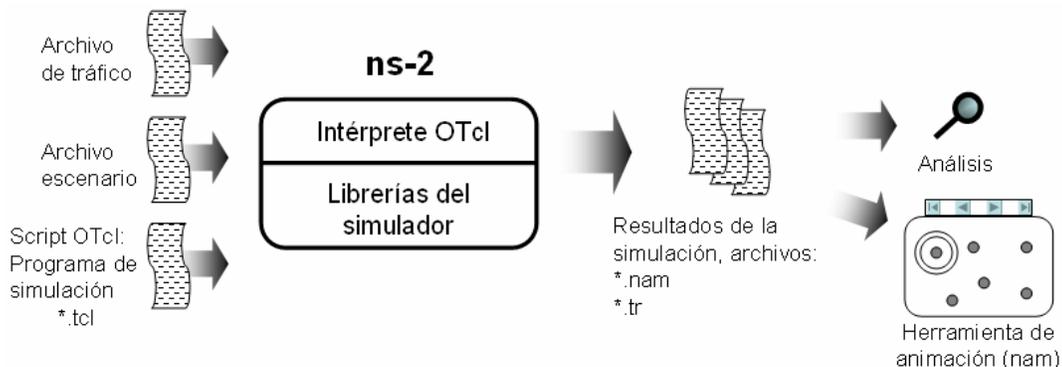


Figura 3.2.- Esquema de simulación en ns-2.

3.1.2. La herramienta de animación (nam) del ns-2

El Network Animator (nam) [8] es la herramienta de ns-2 basada en Tcl/TK que permite visualizar el comportamiento de las simulaciones de red y monitorear las rutas de los paquetes de datos como si se tratara de una observación del desempeño real de la red. Al respecto *nam* fue diseñado para plasmar información de forma sintetizada de las instrucciones de los eventos a partir de un archivo denominado *tracefile* que posee toda la descripción del funcionamiento de la red. Al ejecutar ns-2 esta información sintetizada es almacenada en un archivo cuya extensión es *.nam con el fin de ser releída por el software de animación cuantas veces sea necesario.

Para utilizar *nam* es necesario contar con el archivo denominado *tracefile*, el cuál es generado al ejecutar ns-2 y que posee la extensión *.tr . Este archivo contiene información relativa a la topología de red como son las características de los nodos, de los enlaces y de las rutas de los paquetes. La herramienta *nam* inicia leyendo el *tracefile* para dibujar la topología en una ventana

donde despliega las funciones de reproducción y alto de la simulación, velocidad de avance o de retroceso, zoom-in/zoom-out del escenario, visualización de información de los paquetes; además son posibles otras opciones como por ejemplo grabar la animación, imprimir el área de animación, ver la energía de los nodos y filtrar el tipo de paquetes a visualizar (datos, MAC, encaminamiento).

La mayor desventaja de *nam* es su imposibilidad para editar gráficamente las características desplegadas de la red siendo necesario recurrir al código de ns-2, realizar las modificaciones y ejecutar nuevamente la simulación.

3.2. Planteamiento y justificación de las condiciones de simulación

Nuestro estudio de la movilidad de las terminales o equipos móviles se basa en analizar el movimiento de un grupo específico de nodos o terminales. En escenarios cuadrados de dos dimensiones fueron colocados nodos de forma aleatoria y uniformemente distribuida. En este estudio, el objetivo es registrar el número de nodos vecinos a un nodo origen denominado S que permanecían dentro de su área de cobertura a lo largo del tiempo.

Para todos los escenarios y siguiendo las especificaciones del protocolo 802.11, utilizamos un radio de transmisión de tipo WLAN (250 metros) que determinó el área de cobertura del nodo fuente S. El valor de nuestra variable correspondió al número de nodos que fueron vecinos de S en el inicio de la simulación y que se encontraban dentro del área de cobertura en momento en que se realizó la muestra. Con esta descripción, las simulaciones corrieron bajo el esquema de la figura 3.3.

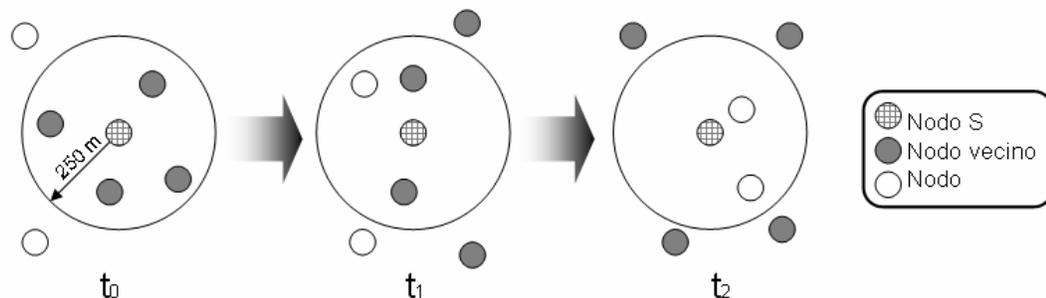


Figura 3.3.- Descripción del esquema de simulación.

Todos los nodos o terminales tenían un desplazamiento con trayectoria aleatoria descrita por el algoritmo *random waypoint* limitada por las fronteras del escenario. Otros parámetros del movimiento de los nodos fueron las velocidades mínimas y máximas que podían alcanzar. Bajo este esquema, para una velocidad deseada de 1 [m/s] utilizamos una velocidad mínima de 0.99 [m/s] y una máxima de 1.01 [m/s] para asegurar la velocidad media deseada.

Cabe resaltar que con base en artículos publicados recientemente sobre las condiciones del simulador ns-2 [9], el valor aleatorio de la velocidad entre los límites mínimo y máximo del algoritmo *random waypoint* no afecta nuestro escenario pues lo que se pretende son nodos con velocidad constante durante toda la simulación.

Para establecer las dimensiones de nuestros escenarios, primero fijamos las velocidades medias de las terminales y designamos al nodo origen S en la zona central del escenario, ya que buscamos evitar situaciones de frontera indeseadas que podrían ocurrir en los límites del escenario durante la simulación. En un escenario de 3000 por 3000 [m] y a una velocidad constante de 1 [m/s], un nodo localizado en la zona central del escenario tendría el tiempo, para llegar a una condición de frontera en el peor caso, de 1500 segundos. De esta manera, las dimensiones del escenario son establecidas con el fin de mantener el área de cobertura del nodo S equiprobable a la existencia de nodos vecinos, pues al llegar a los límites del escenario y continuar con el movimiento, el área queda truncada lo que corresponde a nuestro problema de frontera como se muestra en la figura 3.4.

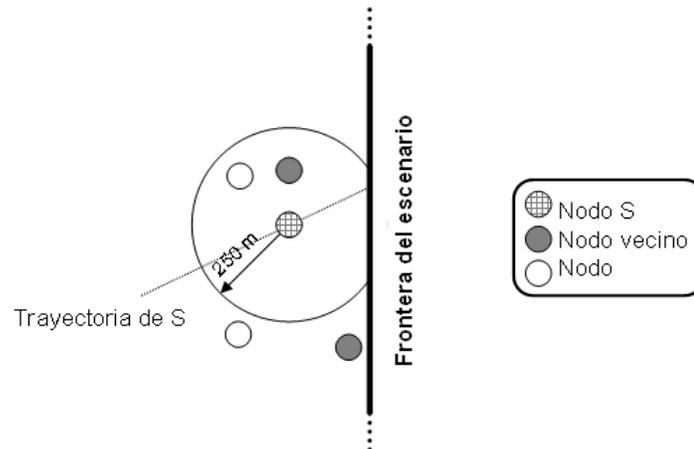


Figura 3.4.- Problema de frontera durante el movimiento de las terminales.

Bajo estas condiciones, el escenario de 3000 por 3000 [m] a un velocidad constante de 1 [m/s] ofrece el desempeño deseado dentro de los primeros 1000[s] de la simulación. En el caso de una velocidad promedio de 3[m/s] tenemos un tiempo mínimo de simulación para el peor caso de 500[s], lo bastante amplio para registrar los resultados esperados.

Por otro lado, el número de nodos fue determinado con base en una aproximación deseada del número de nodos en la vecindad de S en el tiempo cero, teniendo como restricción reducir el tiempo de simulación. Con 300 nodos, el número de terminales en la vecindad de S que corresponde a la densidad de nodos del escenario multiplicada por el área circular de cobertura de 250[m] de radio, resultó en 6.5 nodos.

Como se dijo con anterioridad los escenarios planteado fueron cuadrados y de dos dimensiones, esto significa que la dinámica de las terminales estuvo contenida en un plano, característica importante de nuestro análisis de movilidad.

3.2.1. El algoritmo *Random waypoint*

Random Waypoint (RWP) [10] es el modelo utilizado en este análisis de movilidad pues corresponde a un modelo comúnmente utilizado en la síntesis de movimiento de las redes inalámbricas móviles como las MANETs. Las características de este algoritmo de movimiento se describen enseguida:

- ❖ En el algoritmo *random waypoint* Cada nodo se mueve a lo largo de una línea en zigzag desde un punto inicial hasta el final del movimiento, como se muestra en la figura 3.5.
- ❖ Los puntos de llegada intermedia o *waypoints* están uniformemente distribuidos sobre el área donde se realiza el movimiento que en términos generales son círculos o polígonos regulares e irregulares.
- ❖ Al inicio de cada segmento del movimiento se establece una velocidad aleatoria entre los límites máximo y mínimo que se mantiene constante al recorrer dicho segmento. El caso básico es cuando la velocidad se mantiene constante durante todo el movimiento.
- ❖ De forma opcional los nodos tienen un tiempo de espera una vez que alcanzan cada *waypoint* llamado *thinking time* cuya duración es una variable aleatoria independiente y normalmente distribuida.

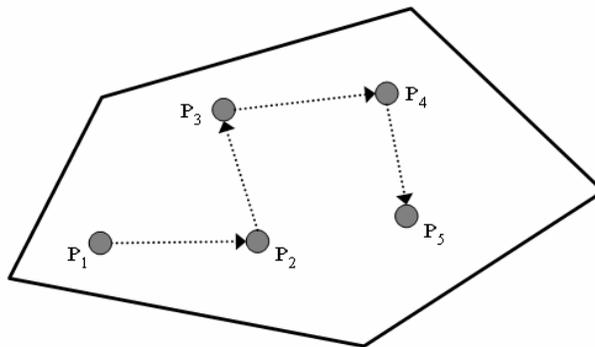


Figura 3.5.- Movimiento *random waypoint*.

En el modelo RWP los puntos waypoints están uniformemente distribuidos en un área específica. Los puntos pueden estar distribuidos sobre los límites del escenario lo que se conoce como modelo *random waypoint sobre la frontera* (RWPB). La densidad espacial de los nodos en el modelo RWPB es distinta a la del modelo RWP, pues en el caso específico de que un límite del escenario sea una línea recta, hay una alta probabilidad de que el segmento de recorrido del nodo recaiga precisamente sobre este segmento de frontera, lo que modifica la distribución de los nodos en el escenario.

3.3. Simulación en ns-2 de una red inalámbrica con nodos móviles

El primer paso para simular este tipo de redes consiste en definir el archivo ejecutable OTcl, que en nuestro caso corresponde al nombre de *movilidad.tcl* y que incluye características propias de la red inalámbrica móvil tales como: las dimensiones del escenario, el número de nodos, la capa de enlace (Link Layer), el tipo de cola de interfaz (Interface Queue), el tipo de capa MAC, el tipo de canal inalámbrico de transmisión y de recepción de señales, tipo de antenas, modelo de radiopropagación, tipo de protocolo de encaminamiento para la red ad hoc, las rutas de los archivos tráfico y escenario, tal y como se muestra enseguida:

```
# movilidad.tcl
# =====
# Se definen las opciones
# =====
set val(chan)          Channel/WirelessChannel    ;# channel type
set val(prop)          Propagation/TwoRayGround    ;# radio-propagation model
set val(netif)         Phy/WirelessPhy           ;# network interface type
set val(mac)           Mac/802_11                ;# MAC type
set val(ifq)           CMUPriQueue              ;# interface queue type
set val(ll)            LL                        ;# link layer type
set val(ant)           Antenna/OmniAntenna       ;# antenna model
set val(ifqlen)        1                        ;# max packet in ifq
set val(nn)            300                      ;# number of mobilenodes
set val(seed)          0
set val(rp)            DumbAgent                 ;# routing protocol
set val(x)             3000
set val(y)             3000
set val(sc)            "../ns-2.29/indep-utils/cmu-scen-gen/setdest/scen-3000x3000-t1000-M1.01-
m0.99-n300"
set val(cp)            "../ns-2.29/indep-utils/cmu-scen-gen/cbr-movilidad"
```

Posterior al encabezado, el archivo *movilidad.tcl* contiene las instrucciones de las variables globales donde se crea una nueva simulación y se abren los archivos de salida *movilidad.tr* y *movilidad.nam* para colocar los resultados del proceso.

```
set ns_                [new Simulator]
set tracefd            [open movilidad.tr w]
set namtrace           [open movilidad.nam w]

$ns_ trace-all $tracefd
$ns_ namtrace-all-wireless $namtrace $val(x) $val(y)
```

Enseguida se crea un objeto para la topología en donde se mantiene la ruta del movimiento de los nodos dentro de los límites del escenario que corresponden a 3000 x 3000 [m].

```
set topo              [new Topography]
$topo load_flatgrid 3000 3000
```

Creamos el objeto God (General operations director) que como su nombre lo indica, es el encargado de dirigir los eventos de nuestra simulación almacenando información global referente al estado del entorno, de la red y de los nodos como si se tratara de un observador omnisciente cuya información no puede ser conocida por cualquier participante de la red.

```
set god_ [create-god $val(nn)]
```

El objeto God almacena el número total de nodos móviles y la tabla con los menores números de saltos necesarios para ligar un nodo a otro. Esta información es obtenida del archivo escenario

antes de iniciar la simulación para evitar que sea cargado mientras se ejecuta la simulación, logrando agilizar el proceso.

A continuación se definen los nodos móviles escribiendo sus características en una interfaz de programación de aplicaciones o API (Application Programming Interface) de ns-2, entre los parámetros que se definen están el tipo de direccionamiento, de protocolo ad hoc de encaminamiento, capa de enlace, capa MAC, topología establecida, tipo de canal, tipo de antena, etc. En esta parte se deshabilita el movimiento aleatorio de los nodos pues más adelante bajo el archivo escenario, quedará establecido.

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

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

Cargamos los archivos escenario y tráfico al programa principal con las variables *sc* y *cp* de forma correspondiente.

```
source $val(sc)
source $val(cp)
```

Ahora corresponde definir el tamaño de la representación gráfica de cada nodo dentro del simulador nam. Seleccionamos el tamaño 20.

```
for {set i 0} {$i < $val(nn)} {incr i} {
    $ns_ initial_node_pos $node_($i) 20
}
```

Establecimos el tiempo final de la simulación para que ocurra al segundo 1000 donde los nodos paran su actividad, para que al segundo 1000.1 arranquen el proceso “stop” que deseche la información temporal de la simulación y se proceda a cerrar el apuntador del archivo *tracefile*.

```
for {set i 0} {$i < $val(nn)} {incr i} {
    $ns_ at 1000.0 "$node_($i) reset";
}
$ns_ at 1000.0 "stop"
$ns_ at 1000.1 "puts \"NS EXITING...\" ; $ns_ halt"
proc stop {} {
    global ns_ tracefd
    $ns_ flush-trace
    close $tracefd
}
```

Colocamos las cabeceras con los datos para el archivo *tracefile*, justo antes de que comience la simulación.

```
puts $tracefd "M 0.0 nn $val(nn) x $val(x) y $val(y) rp $val(rp)"
puts $tracefd "M 0.0 sc $val(sc) cp $val(cp) seed $val(seed)"
puts $tracefd "M 0.0 prop $val(prop) ant $val(ant)"
```

Finalmente arranca el proceso de simulación.

```
puts "Comenzando simulación..."

$ns_ run
```

La estructura del archivo tráfico *cbr-movilidad*, define una conexión UDP entre los nodos 52 y 53, un tamaño de paquete de 1 byte, 1000 es el número promedio de paquetes enviados a intervalos de 0.0001 segundo comenzando en el tiempo 0 segundos.

Nuestro estudio corresponde a un análisis de movilidad, de ahí que la transmisión de paquetes no es un punto a considerar, sin embargo nos ayuda en el momento de visualizar de manera gráfica a la red, pues las circunferencias concéntricas al nodo origen que establecen su área de vecindad, resultaron útiles para delimitar la zona de cobertura y para registrar el número de terminales iniciales vecinas que se mantenían en su interior a lo largo del tiempo. Con esto, simulamos tráfico cada 100 segundos en el escenario a 1 [m/s] y cada 25 segundos en el de 3 [m/s]. Mostramos sólo el inicio del archivo de tráfico del escenario a 1 [m/s] de nuestras simulaciones:

```
set udp_(0) [new Agent/UDP]
$ns_ attach-agent $node_(52) $udp_(0)
set null_(0) [new Agent/Null]
$ns_ attach-agent $node_(53) $null_(0)

#Envío de paquetes

set cbr_(0) [new Application/Traffic/CBR]
$cbr_(0) set packetSize_ 1
$cbr_(0) set interval_ 0.0001
$cbr_(0) set random_ 0
$cbr_(0) set minpkts_ 9999
$cbr_(0) set maxpkts_ 10001
$cbr_(0) attach-agent $udp_(0)
$ns_ connect $udp_(0) $null_(0)
$ns_ at 0 "$cbr_(0) start"

set cbr_(0) [new Application/Traffic/CBR]
$cbr_(0) set packetSize_ 1
$cbr_(0) set interval_ 0.0001
$cbr_(0) set random_ 0
$cbr_(0) set minpkts_ 9999
$cbr_(0) set maxpkts_ 10001
$cbr_(0) attach-agent $udp_(0)
$ns_ connect $udp_(0) $null_(0)
$ns_ at 100 "$cbr_(0) start"

...
```

Para concluir el proceso de simulación, generamos el archivo escenario con base en el programa *setdest* que pertenece a un módulo de ns-2 mismo que contiene el algoritmo de movilidad que utilizamos. La sintaxis para la creación de los escenarios posee las siguientes estructuras en sus dos versiones:

Usage:

```
<original 1999 CMU version (version 1)>
./setdest -v <1> -n <nodes> -p <pause time> -M <max speed>
-t <simulation time> -x <max X> -y <max Y>

OR
<modified 2003 U.Michigan version (version 2)>
./setdest -v <2> -n <nodes> -s <speed type> -m <min speed> -M <max speed>
-t <simulation time> -P <pause type> -p <pause time> -x <max X> -y <max Y>
(Refer to the script files make-scen.csh and make-scen-steadystate.csh for detail.)
```

En este análisis utilizamos la versión 2 modificada por la Universidad de Michigan con los siguientes parámetros como ejemplo:

```
./setdest -v 2 -n 300 -s 1 -m 0.99 -M 1.01 -t 1000.0 -P 1 -p 0 -x 3000 -y 3000 > FILENAME
```

Bajo el siguiente orden y significado de los parámetros:

- ✓ Versión = 2
- ✓ Número de nodos = 300
- ✓ Tipo de velocidad = 1
- ✓ Velocidad mínima [m/s]= 0.99
- ✓ Velocidad máxima [m/s]= 1.01
- ✓ Tiempo de simulación [s]= 1000
- ✓ Tipo de pausa = 1
- ✓ Tiempo de pausa = 0
- ✓ Longitud x del área de escenario [m]= 3000
- ✓ Longitud y del área de escenario [m]= 3000

El archivo escenario *FILENAME* resultante inicia con la siguiente información en donde se detallan las coordenadas de todos los nodos que se encuentran distribuidos de forma aleatoria en el escenario.

```
$node_(0) set X_ 632.235153695781
$node_(0) set Y_ 2152.073099424877
$node_(0) set Z_ 0.000000000000
$node_(1) set X_ 1162.366674927126
$node_(1) set Y_ 978.831258741417
$node_(1) set Z_ 0.000000000000
$node_(2) set X_ 2144.917251109795
$node_(2) set Y_ 404.096577561481
$node_(2) set Z_ 0.000000000000
$node_(3) set X_ 951.474289388082
$node_(3) set Y_ 1150.836408923840
$node_(3) set Z_ 0.000000000000
$node_(4) set X_ 2569.121668901599
$node_(4) set Y_ 1292.033439442272
$node_(4) set Z_ 0.000000000000
$node_(5) set X_ 476.253593587491
$node_(5) set Y_ 1394.986033010552
$node_(5) set Z_ 0.000000000000
$node_(6) set X_ 1791.992780923494
$node_(6) set Y_ 319.012807531903
$node_(6) set Z_ 0.000000000000
$node_(7) set X_ 2778.051528808362
$node_(7) set Y_ 1781.421653352609
$node_(7) set Z_ 0.000000000000
...
```

El tiempo efectivo de procesamiento necesario para elaborar una de las simulaciones realizadas en este estudio está determinado por dos procesos: el primero es la construcción del archivo escenario móvil y el segundo es la propia simulación en donde el programa OTcl carga los archivos tráfico y escenario móvil.

Después de realizar las primeras aproximaciones de escenarios nos percatamos que el tiempo total sólo depende del tamaño del escenario y del número total de nodos, pues el tráfico en nuestro trabajo no constituye un parámetro que afecta el tiempo de las simulaciones.

Referencias:

[1]

The Network Simulator - ns-2

<http://www.isi.edu/nsnam/ns/>

[2]

The Network Simulator ns-2: Validation Tests

<http://www.isi.edu/nsnam/ns/ns-tests.html>

[3]

Altaman Eitan, Jiménez Tania, *NS Simulator for beginners*, Univ. de Los Ángeles, Mérida, Venezuela and ESSI, 2003.

<http://www-sop.inria.fr/maestro/personnel/Eitan.Altman/COURS-NS/n3.pdf>

[4]

Chung Jae, Claypool Mark, *NS by Example*, Worcester Polytechnic Institute.

<http://nile.wpi.edu/NS/>

[5]

Greis Marc, *Tutorial for the Network Simulator*.

<http://www.isi.edu/nsnam/ns/tutorial/>

[6]

Hyytia Esa, Lassila Pasi, et al., *Spatial Node Distribution in the Random Waypoint Mobility Model*, Networking Laboratory, Helsinki University of Technology, Finland.

<http://www.netlab.tkk.fi/tutkimus/pannet/publ/rwp-td04-029.pdf>

[7]

Nain Philippe, Towsley Don, et al., *Properties of Random Direction Models*, University of Massachusetts, USA.

<http://www.cs.uml.edu/~bliu/pub/mobility.pdf>

[8]

Kevin Fall, Kannan Varadhan, *The ns Manual (formerly ns Notes and Documentation)*, VINT Project, UC Berkeley, LBL, USC/ISI and Xerox PARC.

<http://www.isi.edu/nsnam/ns/ns-documentation.html>

[9]

S. PalChaudhuri, J.-Y. Le Boudec, M. Vojnovic, *Perfect Simulation and Stationarity of a Class of Mobility Models*, IEEE Infocom 2005, Miami, FL, 2005 (Infocom 2005 Best Paper Award).

<http://icalwww.epfl.ch/RandomTrip/>

[10]

<http://www.netlab.tkk.fi/~esa/java/rwp/rwp-model.shtml>

4. Resultados de las simulaciones

En este capítulo presentamos los resultados de las simulaciones que fueron planteadas en el capítulo anterior y que se obtuvieron mediante la interfaz gráfica *nam* del simulador ns-2.

Algunas consideraciones respecto a forma en que desarrollamos los experimentos y en que obtuvimos los resultados son las siguientes:

- ❖ Seleccionamos un nodo al azar dentro del área central del escenario para ser considerado como nodo origen en cada uno de los experimentos.
- ❖ Rechazamos todos casos en los que el número inicial de nodos vecinos al nodo origen fue menor a siete y en el caso contrario, cuando era mayor, seleccionamos al azar sólo siete de ellos.
- ❖ La forma para discriminar si un nodo se encontraba dentro de la zona de cobertura fue mediante las coordenadas (x, y) de los nodos de interés que son proporcionadas por la herramienta *nam*. Delimitamos el área de cobertura simulando tráfico constante en intervalos muy pequeños de tiempo que es representado en *nam* con círculos concéntricos a partir del nodo origen. En cada muestra contamos el número de nodos vecinos del inicio de la simulación colocados dentro de su interior.
- ❖ Fue posible obtener la escala de la representación gráfica del escenario, pues nuestra referencia fueron los 3000 metros que medía cada uno de sus lados.
- ❖ La generación de tráfico establecida en el proceso de simulación ayudó a detener los escenarios en los tiempos de simulación, pues al ser ejecutados existe una breve pausa en su reproducción dentro de *nam* que permite parar la ejecución del escenario.

Estas son las condiciones de nuestros escenarios:

<i>Escenario A:</i>	<ul style="list-style-type: none"> ✓ Dimensiones: 3000 x 3000 [m] ✓ Número de nodos: 300 ✓ Velocidad promedio: 1 [m/s] ✓ Radio de la zona de cobertura: 250 [m] ✓ Tiempo de simulación: 1000 [s]
<i>Escenario B:</i>	<ul style="list-style-type: none"> ✓ Dimensiones: 3000 x 3000 [m] ✓ Número de nodos: 300 ✓ Velocidad promedio: 3 [m/s] ✓ Radio de la zona de cobertura: 250 [m] ✓ Tiempo de simulación: 300 [s]

A continuación mostramos en las figuras 4.1. a 4.4. una serie de imágenes tomadas en distintos tiempos de muestreo correspondientes a uno de los experimentos que realizamos. En las figuras se distingue el círculo de cobertura con radio de 250 [m], el nodo origen y sus nodos vecinos del inicio de la simulación.

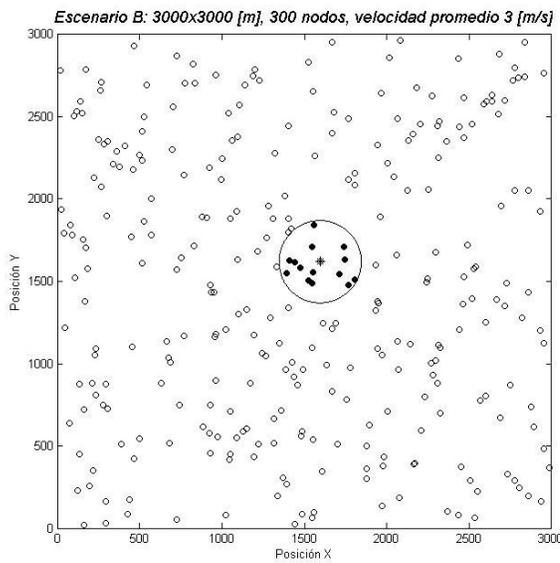


Figura 4.1.- Escenario a los 0 [s]

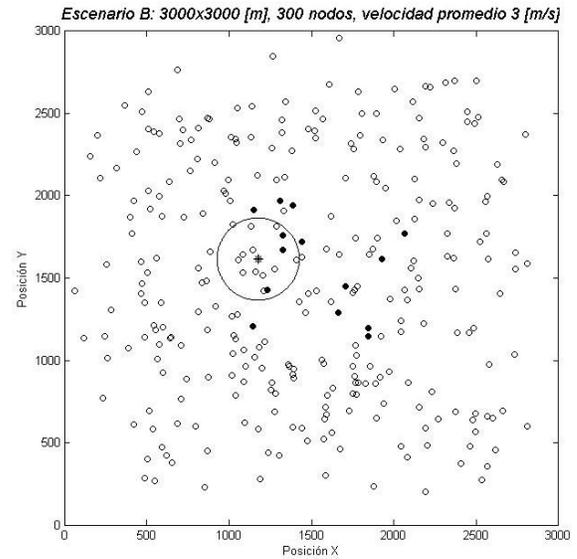


Figura 4.2.- Escenario a los 150 [s]

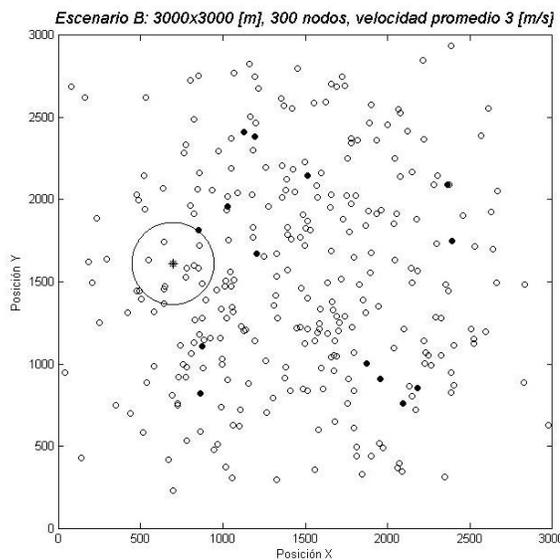


Figura 4.3.- Escenario a los 300 [s]

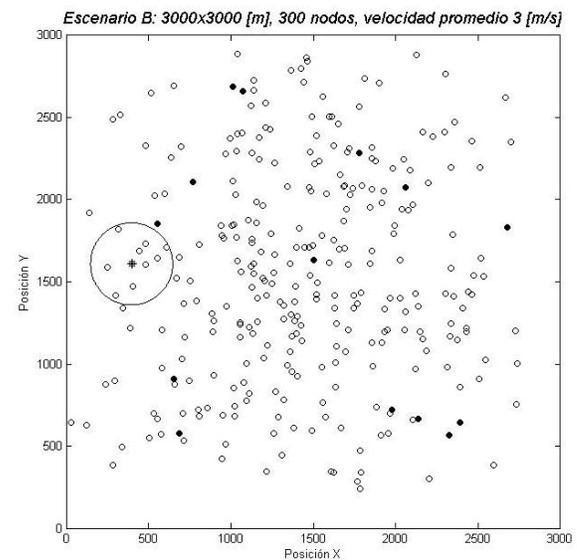


Figura 4.4.- Escenario a los 400 [s]

Contrasta la distribución aleatoria de las terminales al inicio de la simulación con la última imagen tomada, pues se perciben pocas terminales en los límites del escenario. Este efecto es producido por el algoritmo *random waypoint*.

Una modificación en la librería MAC del programa simulado en ns-2 nos proporcionó, en los tiempos de muestreo las coordenadas de los nodos en esos instantes; ésta modificación se presenta dentro de la sección de apéndices. Una vez que obtuvimos las coordenadas, programamos una rutina en Matlab (*fotos.m*) que graficó los puntos que muestran las figuras, pues la herramienta *nam* no muestra imágenes tan detalladas como las que aquí presentamos; el código de esta rutina también está contenida en la sección de apéndices. Enseguida mostramos en las tablas 4.1. y 4.2., los números de vecinos iniciales que permanecieron en el área de cobertura del nodo origen junto con las gráficas de valores promedio, figuras 4.5. y 4.6.

	Tiempo de simulación [s]										
	0	100	200	300	400	500	600	700	800	900	1000
Número de experimento:	Número de nodos vecinos del inicio										
1	7	3	2	1	1	0	0	0	0	0	0
2	7	4	4	1	1	1	1	1	0	0	0
3	7	7	4	1	1	0	0	0	0	0	0
4	7	6	2	0	0	0	0	0	0	0	0
5	7	6	2	2	1	1	1	0	0	0	0
6	7	3	3	2	2	2	2	2	1	1	1
7	7	6	3	3	2	2	1	1	0	0	0
8	7	6	4	2	0	0	0	0	0	0	0
9	7	5	4	1	1	0	0	0	0	0	0
10	7	7	5	4	4	3	1	1	0	0	0
Valores promedio	7	5.30	3.30	1.70	1.30	0.90	0.60	0.50	0.10	0.10	0.10

Tabla 4.1.- Resultados del escenario A.

	Tiempo de simulación [s]										
	0	100	200	300	400	500	600	700	800	900	1000
Número de experimento:	Número de nodos vecinos del inicio										
1	7	7	6	3	1	1	1	1	1	0	0
2	7	7	6	2	1	1	1	1	1	0	0
3	7	6	4	4	3	2	1	1	1	0	0
4	7	5	3	2	1	1	1	1	0	0	0
5	7	6	4	3	0	0	0	0	0	0	0
6	7	6	4	3	2	2	1	0	0	0	0
7	7	7	5	3	2	2	2	2	2	2	2
8	7	6	3	3	2	2	1	1	0	0	0
9	7	7	5	4	3	3	2	2	0	0	0
10	7	7	5	4	2	2	2	2	1	1	1
Valores promedio	7	6.40	4.50	3.10	1.70	1.60	1.20	1.10	0.60	0.30	0.30

Tabla 4.2.- Resultados del escenario B.

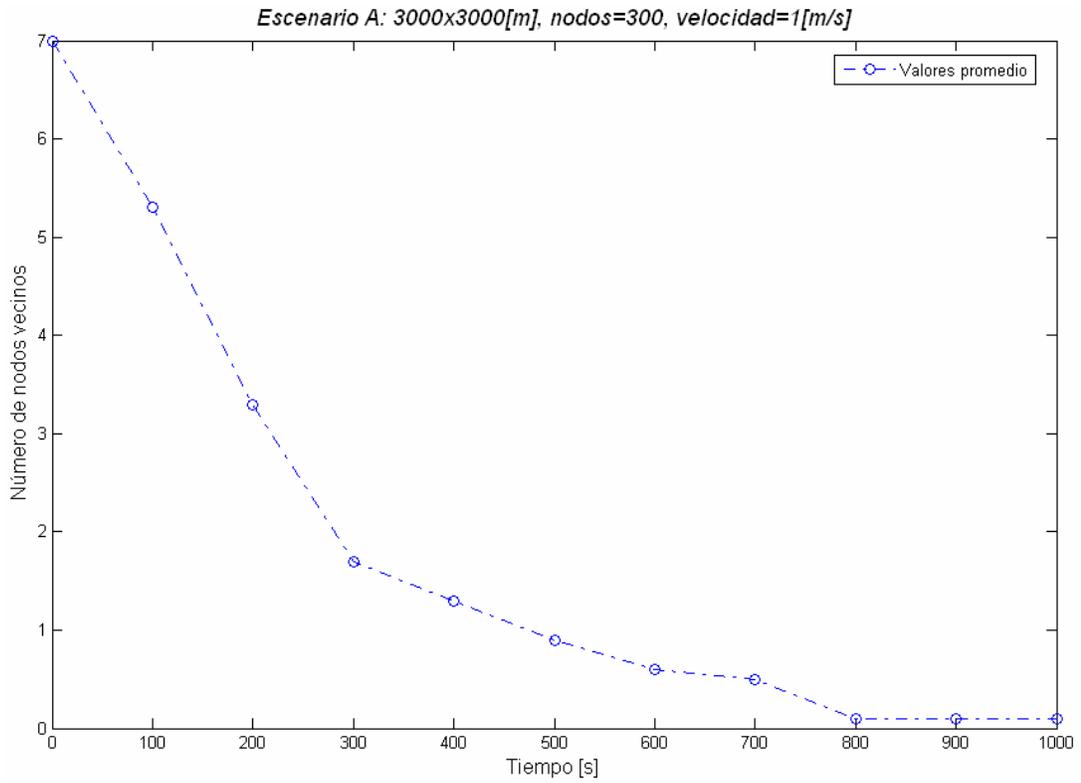


Figura 4.5.- Resultados del escenario A.

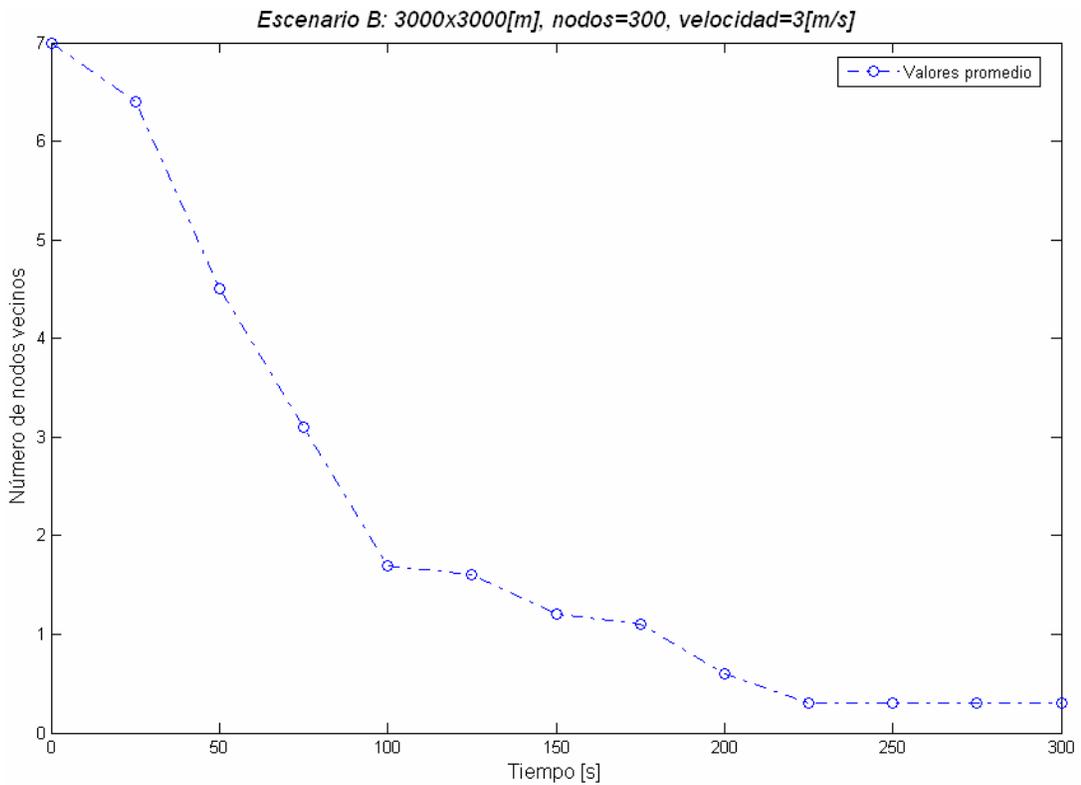


Figura 4.6.- Resultados del escenario B.

5. Análisis matemático del tiempo de permanencia

El movimiento de las terminales en una red inalámbrica responde al tipo de implementación o situación que se analice. Por ejemplo, el movimiento mostrado por los dispositivos con conexión inalámbrica en una zona de cobertura celular urbana, en un supermercado, en una zona de desastre natural y en un campo de batalla, son distintos entre sí; pues la distribución de las terminales móviles, los tiempos de reposo y movimiento, las trayectorias de desplazamiento, la aceleración y las velocidades de los equipos siguen patrones diferentes. Es entonces importante la existencia de modelos generales de movilidad que permitan estudiar a las redes inalámbricas móviles y en particular durante la evaluación del desempeño de los protocolos de comunicación.

En este capítulo desarrollaremos un modelo matemático basado en los resultados obtenidos en las simulaciones del capítulo anterior, basado en una aproximación matemática por regresión lineal, obtendremos los intervalos de confianza para las muestras recabadas en los experimentos y por último propondremos un modelo general de movimiento.

5.1. Planteamiento del problema

El problema consiste en establecer un modelo matemático que determine la distribución de probabilidad del tiempo de permanencia en el que un nodo vecino se encuentra dentro del área de cobertura de otro nodo que es tomado como referencia.

Los nodos vecinos corresponden a las terminales localizadas a un salto de comunicación de otra tomada como referencia y que se encuentra dentro de su rango de transmisión o vecindad, misma que es delimitada por un radio de 250 metros con centro en ella.

Una vez establecida esta relación, encontraremos una función que describa el número de nodos vecinos en el inicio de la simulación, dentro del rango de conexión del nodo origen, durante el tiempo de vida del escenario.

Con base en las gráficas 4.5 y 4.6 proponemos una variable aleatoria τ que se asume sigue una distribución de probabilidad exponencial, de tal modo que:

$$(1) \quad P(\tau > t) = e^{-\lambda t}$$

O bien:

$$(2) \quad P(\tau \leq t) = 1 - e^{-\lambda t}$$

La variable aleatoria τ mostrada en la figura 5.1., representa el tiempo que tarda un nodo vecino inicial en permanecer dentro de la zona de cobertura de otro que también se encuentra en movimiento.

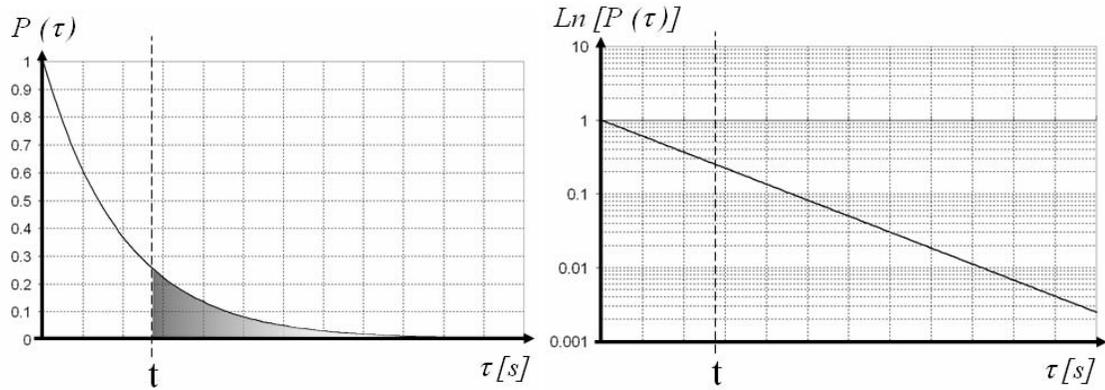


Figura 5.1.- Definición de la variable aleatoria con distribución exponencial.

Para obtener una aproximación del número de nodos que permanecen dentro de una zona de cobertura en movimiento $F(t)$, basta con multiplicar la densidad de nodos en el escenario con el valor del área de cobertura y con la función de probabilidad del tiempo de permanencia, esto es:

$$(3) \quad F(t) = \delta A_c P(t)$$

$$(4) \quad F(t) = \delta A_c e^{-\lambda t}$$

Donde:

δ = Densidad de nodos

A_c = Área de la zona de cobertura

$P(t)$ = Probabilidad de tiempo de permanencia

Nótese que el valor δA_c corresponde al número de vecinos en el tiempo inicial del escenario dentro del área de cobertura móvil.

Por otro lado los parámetros δA_c y λ pueden aproximarse con los datos obtenidos en nuestras simulaciones mediante el método de regresión lineal denominado método por mínimos cuadrados; para ello realizamos la siguiente transformación a la ecuación (4):

$$Ln[F(t)] = Ln[\delta A_c e^{-\lambda t}]$$

Separamos los miembros de la ecuación para obtener (5):

$$Ln[F(t)] = G(t)$$

$$Ln[\delta A_c e^{-\lambda t}] = Ln[\delta A_c] - \lambda t$$

$$(5) \quad G(t) = Ln[\delta A_c] - \lambda t$$

La ecuación (5) presenta la forma de la ecuación de una recta del tipo:

$$(6) \quad Y = A + Bx$$

Entonces la ordenada al origen A de la recta y su pendiente B son los parámetros por calcular en nuestra aproximación por mínimos cuadrados y están representados por:

$$(7) \quad A = \text{Ln}[\delta A_c]$$

$$(8) \quad B = -\lambda$$

5.2. Aproximación por mínimos cuadrados

El objetivo consiste en aproximar una curva mediante el método de mínimos cuadrados [1], a la serie de datos que representan el valor promedio del número de vecinos en cada tiempo de muestreo, resultado de nuestras simulaciones.

Con el fin de facilitar el cálculo de la aproximación de la curva exponencial realizamos la siguiente transformación, pues algunos valores medios obtenidos son menores a la unidad y su logaritmo natural es menor a cero, siendo imposible su representación en el plano semilogarítmico propuesto en donde la función exponencial queda representada por una recta con pendiente negativa.

La transformación es:

$$(9) \quad F'(t) = 10F_{\text{muestras}}$$

Considerando: $y = \text{Ln}(F'(t))$

Presentamos la tabla 5.1., con los valores promedio del número de vecinos pertenecientes a los dos escenarios A y B descritos en el capítulo anterior para aproximarlos a una recta:

<i>Escenario A</i>				<i>Escenario B</i>			
t	F_{muestras}	$F'(t)$	y	t	F_{muestras}	$F'(t)$	y
0	7	70	4.25	0	7	70	4.25
100	5.3	53	3.97	25	6.4	64	4.16
200	3.3	33	3.50	50	4.5	45	3.81
300	1.7	17	2.83	75	3.1	31	3.43
400	1.3	13	2.56	100	1.7	17	2.83
500	0.9	9	2.20	125	1.6	16	2.77
600	0.6	6	1.79	150	1.2	12	2.48
700	0.5	5	1.61	175	1.1	11	2.39
800	0.1	1	0	200	0.6	6	1.79
900	0.1	1	0	225	0.3	3	1.10
1000	0.1	1	0	250	0.3	3	1.10
				275	0.3	3	1.10
				300	0.3	3	1.10

Tabla 5.1.- Valores base en la aproximación por el método de mínimos cuadrados.

Las relaciones para obtener los coeficientes de la aproximación son:

$$(11) \quad B = \frac{\sum t \cdot y - n \cdot \bar{t} \cdot \bar{y}}{\sum t^2 - n \cdot \bar{t}^2}$$

$$(12) \quad A = \bar{y} - B \cdot \bar{t}$$

$$(13) \quad r = \sqrt{\frac{\sum [y' - \bar{y}]^2}{\sum [y - \bar{y}]^2}}$$

Donde:

\bar{t} = Valor promedio de la variable t

\bar{y} = Valor promedio de la variable y

n = Número de muestras

y = De la ecuación (10)

y' = Valor calculado con los coeficientes A y B

r = Coeficiente de correlación

El coeficiente de correlación establece el grado de aproximación de los datos a un modelo exponencial; entre más cercano es este valor a la unidad, mejor se ajustan los datos al modelo propuesto.

Para ambos escenarios tenemos los resultados en la tabla 5.2.:

<i>Escenario A</i>	<i>Escenario B</i>
$\sum t \cdot y = 6272.55$	$\sum t \cdot y = 3486.01$
$n = 11$	$n = 13$
$\bar{t} = 500$	$\bar{t} = 150$
$\bar{y} = 2.06$	$\bar{y} = 2.49$
$\sum t^2 = 3.85 \times 10^6$	$\sum t^2 = 4.0625 \times 10^5$
$\sum [G(t) - \bar{y}]^2 = 23.49$	$\sum [G(t) - \bar{y}]^2 = 16.32$
$\sum [y - \bar{y}]^2 = 24.38$	$\sum [y - \bar{y}]^2 = 16.94$
$\Rightarrow B = -0.0046$	$\Rightarrow B = -0.0119$
$\Rightarrow A = 4.3753$	$\Rightarrow A = 4.2829$
$\Rightarrow r = 0.9816$	$\Rightarrow r = 0.9815$

Tabla 5.2.- Valor de coeficientes bajo la aproximación por el método de mínimos cuadrados.

Finalmente, despejando de las ecuaciones (7) y (8):

$$(14) \quad \delta A_c = \frac{1}{10} e^A$$

$$(15) \quad \lambda = -B$$

Escenario A		Escenario B	
	$\delta A_c = 7.9468$		$\delta A_c = 7.2455$
	$\lambda = 0.0046$		$\lambda = 0.0119$
	$r = 0.9816$		$r = 0.9815$
(I)	$F(t) = 7.9468e^{-0.0046t}$	(II)	$F(t) = 7.2455e^{-0.0119t}$

Tabla 5.3.- Funciones aproximadas por el método de mínimos cuadrados.

El cálculo de los parámetros de la aproximación junto con las gráficas que se muestran en la tabla 5.3., las realizamos mediante un programa en Matlab que forma parte de la sección de apéndices de este trabajo.

Enseguida mostramos las figuras 5.2. y 5.3., con las gráficas en escala semilogarítmica del número promedio de nodos vecinos respecto al tiempo, con el fin de visualizar su aproximación a una recta por el método de mínimos cuadrados.

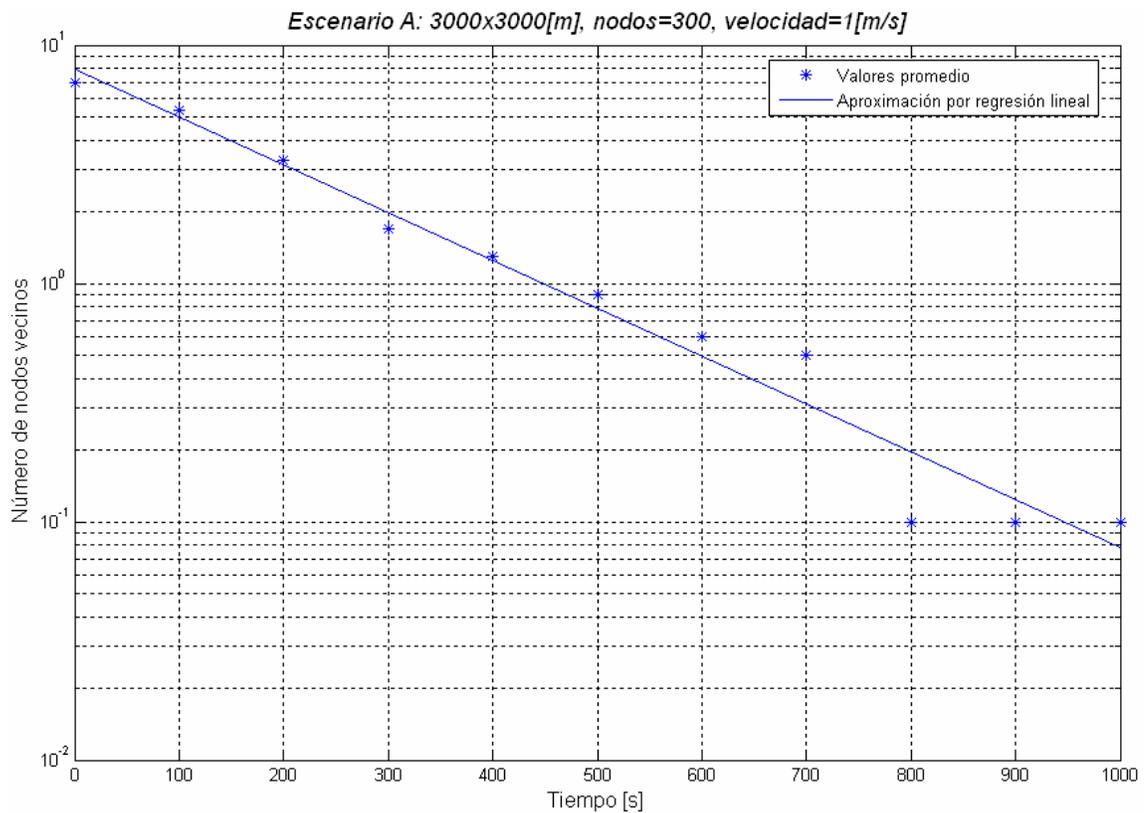


Figura 5.2.- Valores promedio con su curva de aproximación para el escenario A.

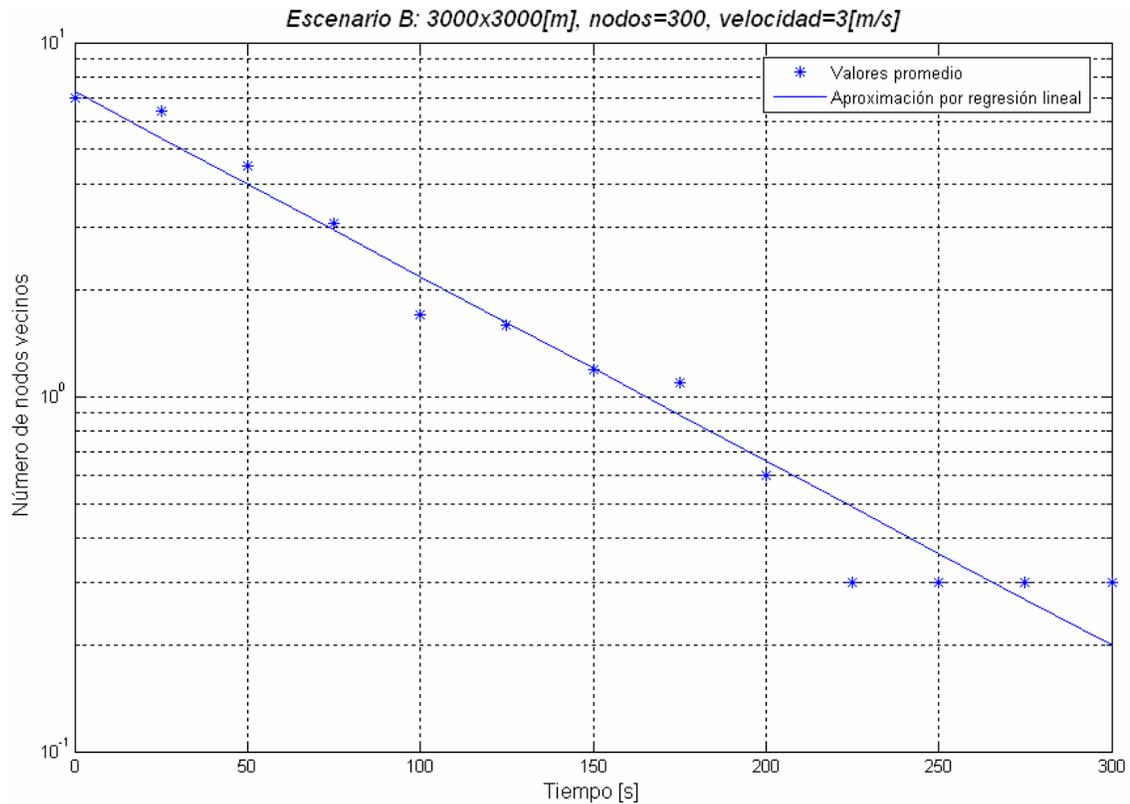


Figura 5.3.- Valores promedio con su curva de aproximación para el escenario B.

5.3. Intervalos de confianza

Ahora corresponde determinar los intervalos de confianza [2] de los datos recabados en las simulaciones con el fin de fundamentar nuestro modelo matemático más general en nuestro análisis de movilidad.

Suponemos que los datos recabados en cada instante de muestreo son independientes y poseen una distribución normal con media μ y varianza σ^2 , entonces, consideramos a la variable aleatoria T que tiene distribución *t-student* con $n-1$ grados de libertad:

$$(16) \quad T = \frac{Z}{\sqrt{\frac{Y}{n-1}}}$$

Con:

$$(17) \quad Z = \sqrt{n} \frac{\bar{X} - \mu}{\sigma}$$

$$(18) \quad Y = \frac{1}{\sigma^2} \sum_{i=1}^n (X_i - \bar{X})^2$$

Sustituyendo tenemos 17 y 18 en 16:

$$(19) \quad T = \frac{\sqrt{n} \frac{\bar{X} - \mu}{\sigma}}{\sqrt{\frac{\frac{1}{\sigma^2} \sum_{i=1}^n (X_i - \bar{X})^2}{n-1}}} = \sqrt{n} \frac{\bar{X} - \mu}{\sqrt{\frac{\sum_{i=1}^n (X_i - \bar{X})^2}{n-1}}} = \sqrt{n} \frac{\bar{X} - \mu}{S}$$

Con:

$$(20) \quad S^2 = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2$$

Consideramos al valor promedio de la muestra \bar{x} como un valor observado de \bar{X} y de la misma forma a la varianza de la muestra s^2 como un valor observado de S^2 .

Queremos encontrar dos variables aleatorias Θ_1 y Θ_2 tales que:

$$(21) \quad P(\Theta_1 \leq \mu \leq \Theta_2) = \gamma$$

De manera que se elegimos γ , para que la muestra genere los valores observados de θ_1 y θ_2 de Θ_1 y Θ_2 respectivamente que proporcionan un intervalo al $\gamma\%$ de confianza.

$$(22) \quad CONF\{\theta_1 \leq \mu \leq \theta_2\}$$

Bajo las suposiciones realizadas T tiene una distribución *t-student* con $n-1$ grados de libertad, entonces utilizamos la función inversa de esta distribución con $n-1$ grados de libertad para determinar el valor de c tal que:

$$(23) \quad P(-c \leq T \leq c) = F(c) - F(-c) = \gamma$$

Como la distribución *t-student* es simétrica: $F(-c) = 1 - F(c)$, entonces la ecuación (23) toma la forma:

$$(24) \quad P(-c \leq T \leq c) = 2F(c) - 1 = \gamma \Rightarrow F(c) = \frac{1}{2}(1 + \gamma).$$

Sustituyendo la ecuación 19 en la desigualdad $-c \leq T \leq c$ tenemos:

$$(25) \quad -c \leq \sqrt{n} \frac{\bar{X} - \mu}{S} \leq c$$

Convirtiendo la desigualdad (25) en términos de μ :

$$(26) \quad \bar{X} - \frac{Sc}{\sqrt{n}} \leq \mu \leq \bar{X} + \frac{Sc}{\sqrt{n}}$$

Sustituyendo los valores observados en la desigualdad 26, finalmente tenemos que el intervalo de confianza al $\gamma\%$ para μ es:

$$(27) \quad CONF \left\{ \bar{x} - \frac{sc}{\sqrt{n}} \leq \mu \leq \bar{x} + \frac{sc}{\sqrt{n}} \right\}$$

Elegimos un valor de $\gamma = 95$ para mostrar en ambos escenarios los intervalos de confianza a cada tiempo de muestreo. En la tabla 5.4., mostramos los valores calculados.

Escenario A				Escenario B			
$n = 11$				$n = 13$			
$c = 2.5275$				$c = 2.4607$			
t	\bar{x}	$\bar{x} - \frac{sc}{\sqrt{n}}$	$\bar{x} + \frac{sc}{\sqrt{n}}$	t	\bar{x}	$\bar{x} - \frac{sc}{\sqrt{n}}$	$\bar{x} + \frac{sc}{\sqrt{n}}$
0	7	7	7	0	7	7	7
100	5.3	4.2196	6.3804	25	6.4	5.9867	6.8133
200	3.3	2.5341	4.0659	50	4.5	3.8616	5.1384
300	1.7	0.8617	2.5383	75	3.1	2.6639	3.5361
400	1.3	0.4617	2.1383	100	1.7	1.1393	2.2607
500	0.9	0.1044	1.6956	125	1.6	1.1016	2.0984
600	0.6	0.0945	1.1055	150	1.2	0.8262	1.5738
700	0.5	0.0112	1.0112	175	1.1	0.6639	1.5361
800	0.1	0*	0.3286	200	0.6	0.1867	1.0133
900	0.1	0*	0.3286	225	0.3	0*	0.6989
1000	0.1	0*	0.3286	250	0.3	0*	0.6989
				275	0.3	0*	0.6989
				300	0.3	0*	0.6989

* Notar que cuando algún límite inferior fue negativo, nosotros lo consideramos con valor de cero pues no tiene caso hablar un número negativo de vecinos.

Tabla 5.4.- Intervalos de confianza al 95 % para los valores promedio del número de vecinos.

Como en la aproximación por mínimos cuadrados, los intervalos de confianza y las gráficas que se muestran las obtuvimos con ayuda de un programa en Matlab. Este programa también lo incluimos en la sección de apéndices.

Mostramos en las figuras 5.4. y 5.5., las gráficas de los valores promedio del número de vecinos en cada tiempo de muestreo con sus respectivos intervalos de confianza al 95%.

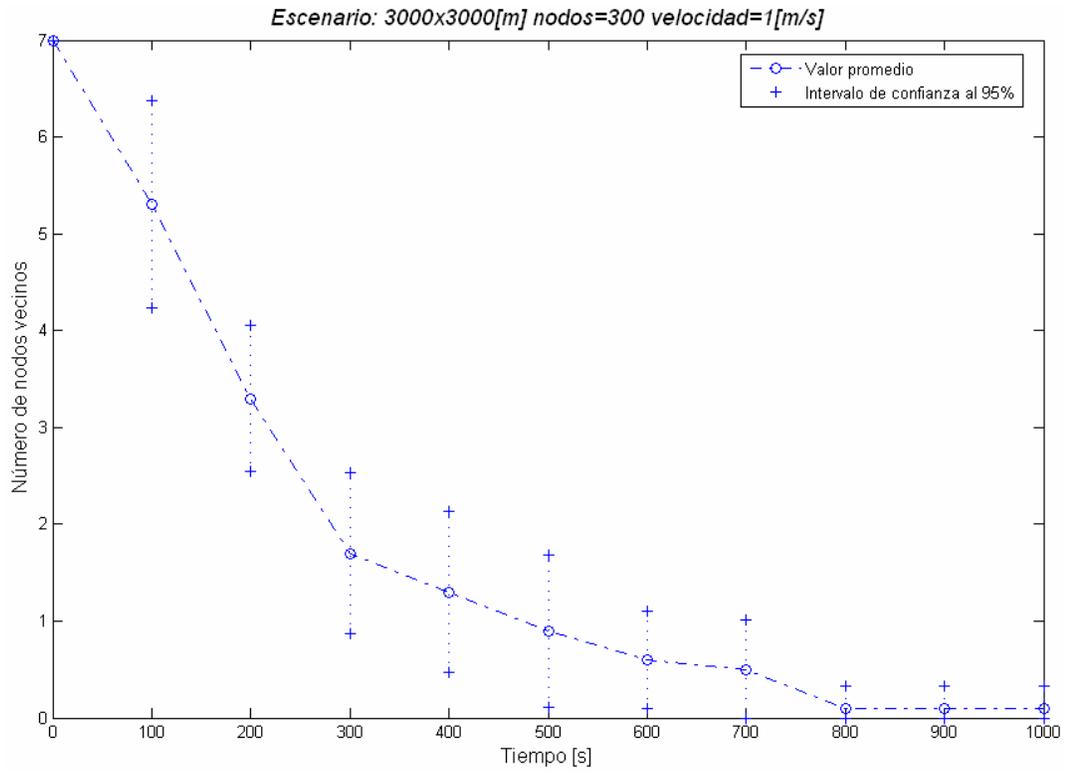


Figura 5.4.- Intervalos de confianza al 95 % de los valores promedio en el escenario A.

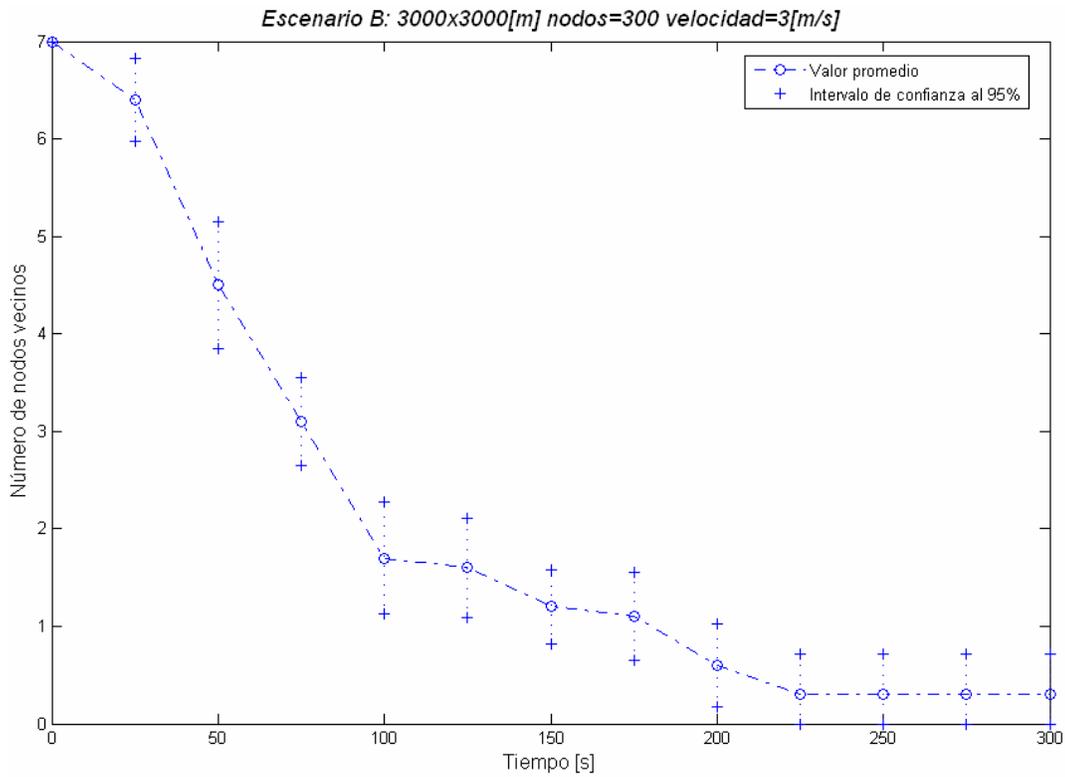


Figura 5.5.- Intervalos de confianza al 95 % de los valores promedio en el escenario B.

5.4. Modelo propuesto

Con base en la función exponencial expresada en términos de los valores aproximados por la regresión lineal, en particular en el coeficiente de correlación r muy cercanos a uno que justifican la forma exponencial de la curva y considerando los intervalos de confianza al 95% obtenidos, basamos nuestra siguiente propuesta:

Encontramos que en nuestros escenarios el parámetro λ de la función (4) se aproxima a la relación v/R , de ahí que planteamos el siguiente modelo como representación general del número de nodos que permanecen dentro de una zona de cobertura circular con radio en movimiento:

$$(28) \quad \lambda \approx \frac{v}{R}$$

$$(29) \quad \Phi(t) = \delta A_c e^{-\frac{v}{R}t}$$

Donde los parámetros v y R que corresponden a:

v = velocidad promedio de los nodos del escenario.

R = radio de la zona de cobertura.

Al considerar las características de ambos escenarios y sustituirlas en este modelo, el número de nodos que permanecen dentro del círculo de cobertura está expresado por $\Phi(t)$ mostrado en la tabla 5.5.:

<i>Escenario A</i>	<i>Escenario B</i>
$\delta = \frac{300}{(3000)^2} \left[\frac{\text{nodos}}{m^2} \right]$ $R = 250[m]$ $A_c = \pi(250)^2 [m^2]$ $v = 1 \left[\frac{m}{s} \right]$	$\delta = \frac{300}{(3000)^2} \left[\frac{\text{nodos}}{m^2} \right]$ $R = 250[m]$ $A_c = \pi(250)^2 [m^2]$ $v = 3 \left[\frac{m}{s} \right]$
(III) $\Phi(t) = 6.545e^{-0.004t}$	(IV) $\Phi(t) = 6.545e^{-0.012t}$

Tabla 5.5.- Funciones basadas en el modelo propuesto de movilidad.

Presentamos en las figuras 5.6. y 5.7., las gráficas que verifican la cercanía de los valores de la función aproximada por mínimos cuadrados y la curva basada en el modelo general propuesto; a demás mostramos los valores promedio del número de vecinos con sus respectivos intervalos de confianza. Estas gráficas fueron obtenidas con el mismo programa en Matlab que hemos mencionado dentro de la sección de apéndices. En las figuras 5.8. y 5.9., mostramos la diferencia del número de nodos entre el modelo propuesto y el valor promedio en los experimentos.

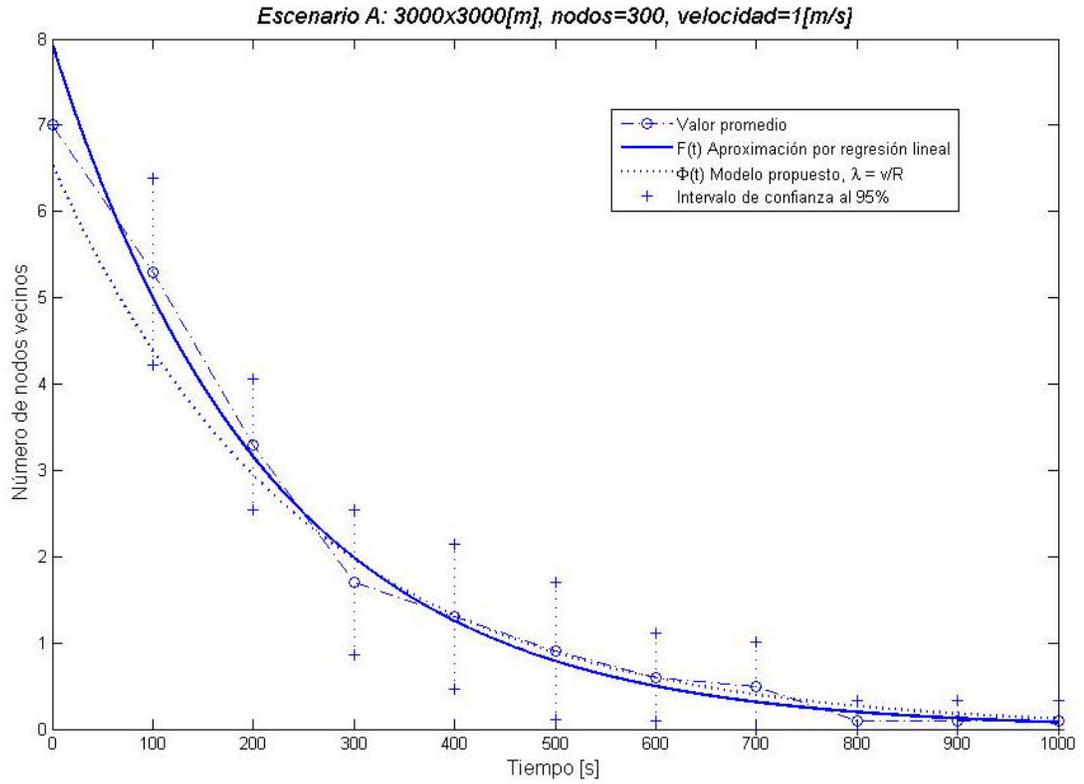


Figura 5.6.- Curvas resultantes para el escenario A.

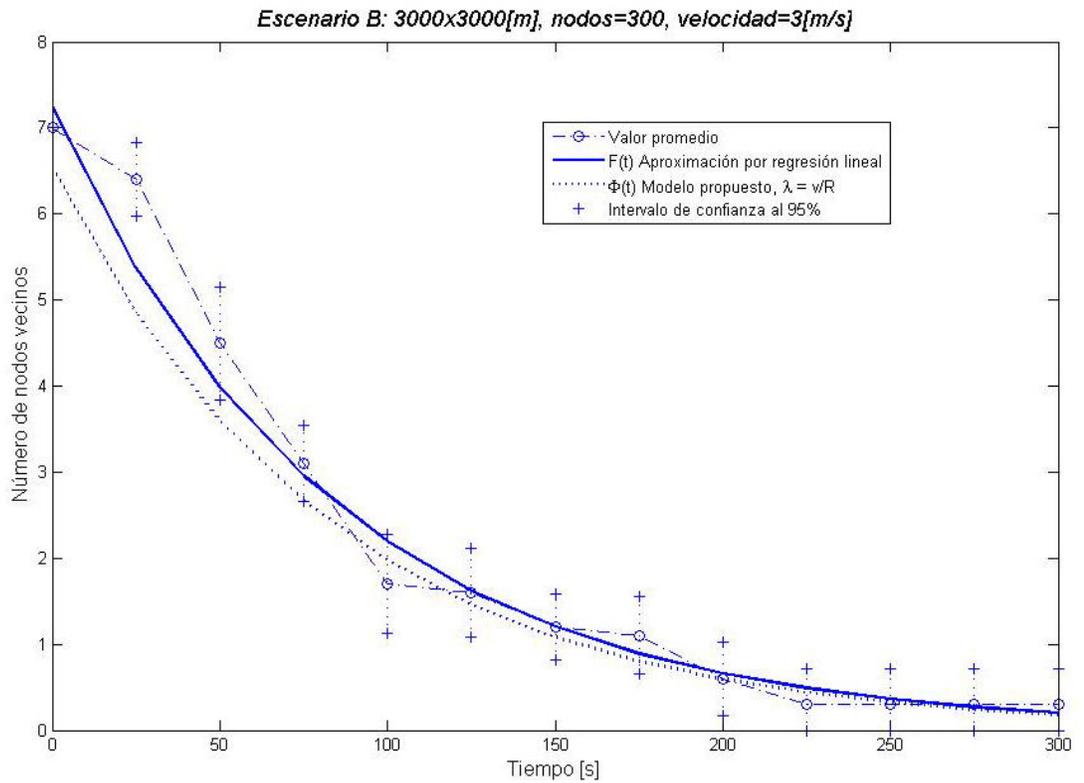


Figura 5.7.- Curvas resultantes para el escenario B.

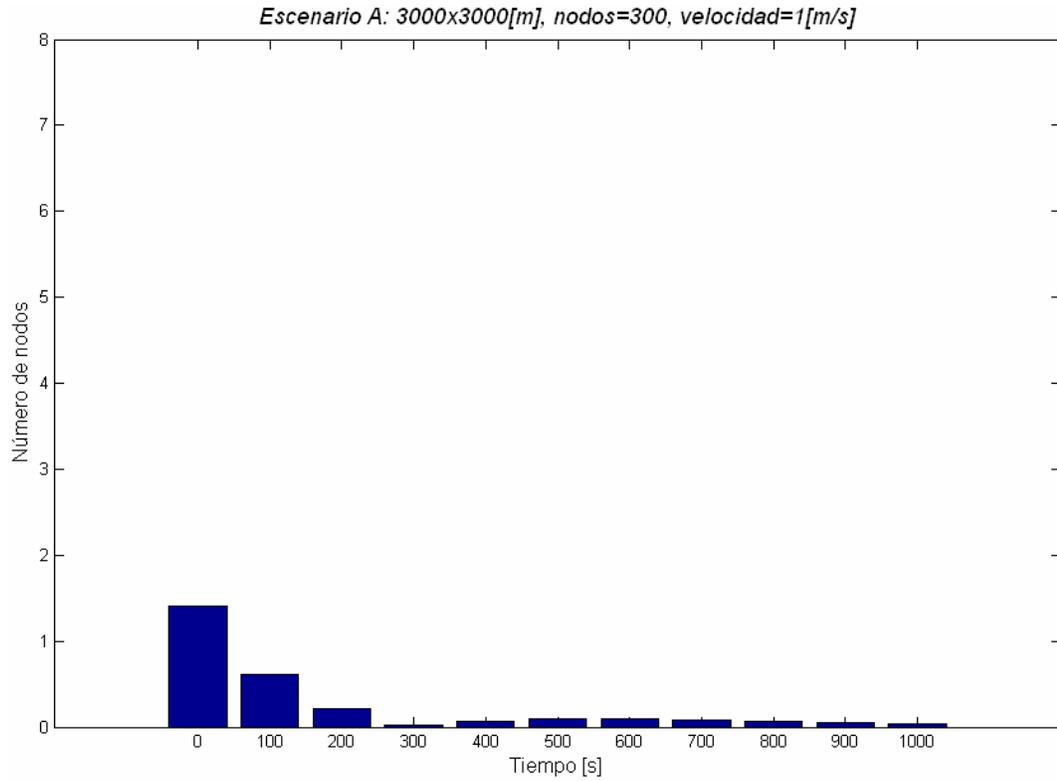


Figura 5.8.- Diferencia del número de nodos entre el modelo propuesto y el valor promedio en los experimentos del escenario A.

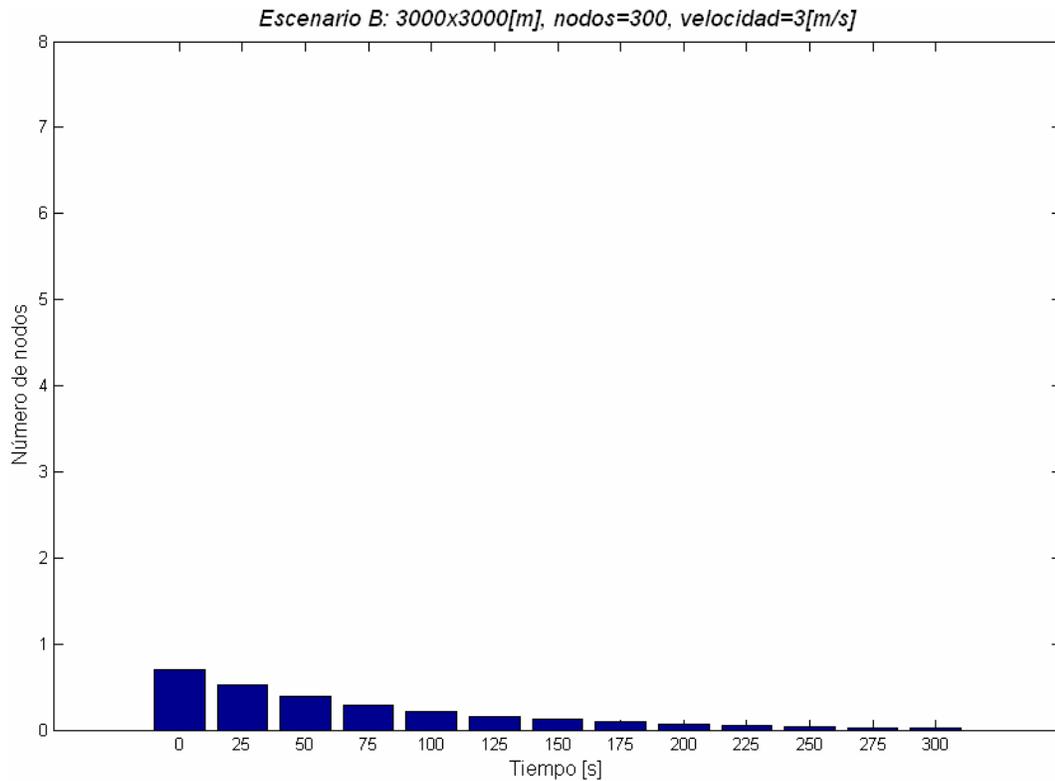


Figura 5.9.- Diferencia del número de nodos entre el modelo propuesto y el valor promedio en los experimentos del escenario B.

5.5. Porcentaje de error relativo

Al comparar las expresiones I con III para el escenario A y II con IV para el B, obtuvimos la diferencia de valores que enunciamos como porcentaje de error relativo ($\%Er$) del modelo general $\Phi(t)$ en (29) respecto al modelo aproximado $F(t)$ en (4):

$$(30) \quad \%Er = \left| \frac{F(t) - \Phi(t)}{F(t)} \right| \times 100$$

Mostramos en las figuras 5.10. y 5.11., las gráficas de porcentaje de error relativo en ambos escenarios.

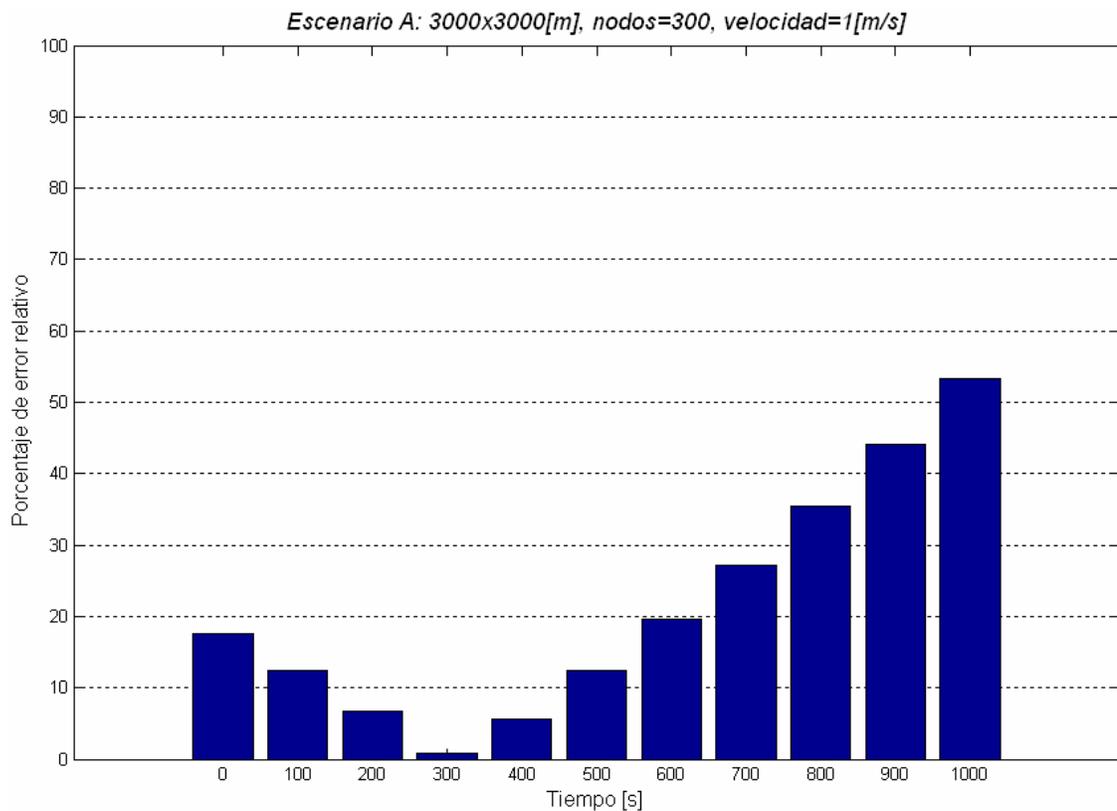


Figura 5.10.- Tabulación de los errores relativos para el escenario A.

Esta gráfica resalta que el error relativo aumenta conforme se incrementa el tiempo a partir de 300 [s], sin embargo notemos en la figura 5.6., que el número de nodos vecinos del inicio de la simulación ya es pequeño, siendo no mayor a 2.

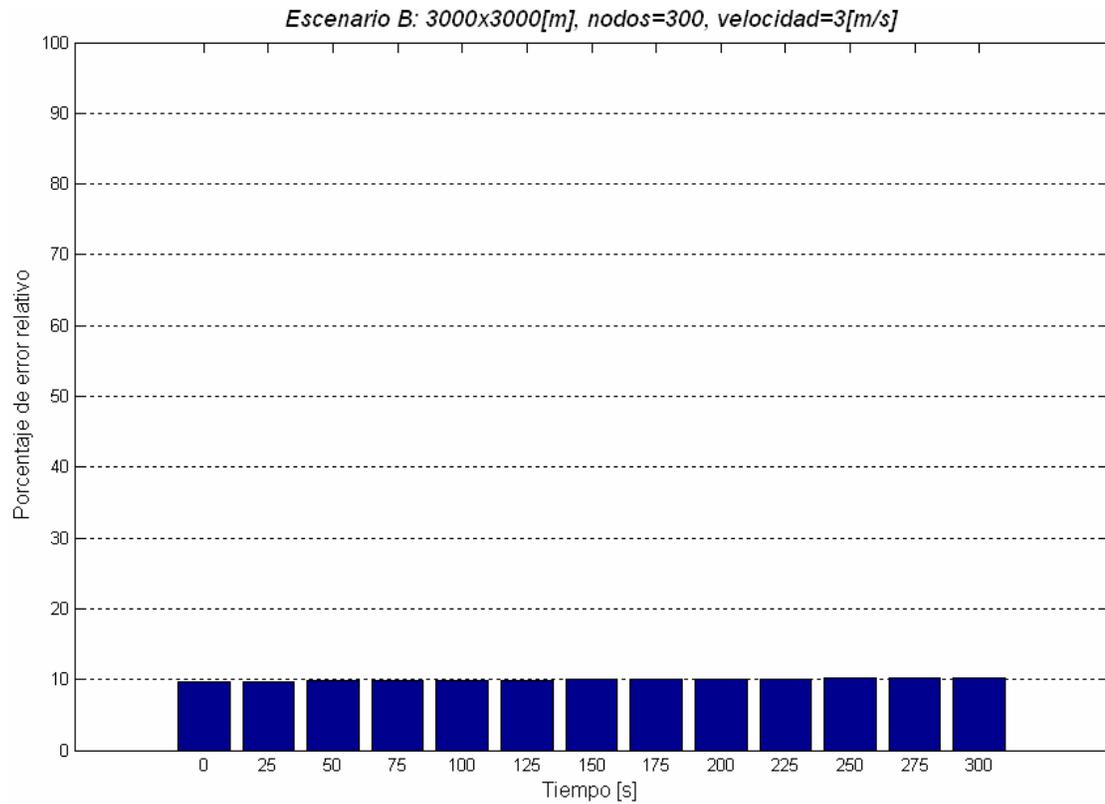


Figura 5.11.- Tabulación de los errores relativos para el escenario B.

Referencias:

- [1]
Grossman Stanley I., *Aplicaciones de Álgebra Lineal*, Grupo Editorial Iberoamericana, México, 1988.
- [2]
Kreyszig Erwin, *Estadística matemática*, Editorial Limusa, México, 1980.

6. Conclusiones

En este trabajo realizamos un análisis de la movilidad de terminales en una red de tipo ad hoc simulada dentro del programa ns-2. Los resultados que obtuvimos se muestran en el capítulo 4 y su correspondiente tratamiento en el capítulo 5. Como parte de la interpretación de los resultados y apoyados en las gráficas que presentamos en el capítulo anterior, proponemos un modelo analítico que describe la dinámica de este tipo de redes dentro del simulador. Presentamos las siguientes conclusiones de este trabajo:

⇒ **Sobre los protocolos de encaminamiento y nuestro análisis de movilidad.-**

Los protocolos de encaminamiento que describimos en el capítulo 2 son sólo algunos de los utilizados en las redes de tipo ad hoc, pues nuevas propuestas surgen día con día de acuerdo a las características de los escenarios por resolver.

Para un escenario con alta movilidad y un nivel reducido de nodos, lo más recomendable es la utilización del protocolo AOVD debido a que la información contenida en la tabla de encaminamiento en los nodos se obtiene diseminando de manera amplia los mensajes, inundando a la red con paquetes que podrían afectar su desempeño en caso de contar con una alta densidad de terminales.

En un escenario en donde las conexiones sean esporádicas pero que esté presente una alta dinámica en su movimiento, la mejor opción es la implementación del protocolo TORA ya que su alto carácter reactivo utiliza de forma eficiente los enlaces de la red ad hoc.

Por el contrario, el algoritmo de encaminamiento DSR es óptimo para redes con velocidades promedio bajas y con una densidad baja de terminales, pues el tamaño de los paquetes con toda la información de control necesaria se incrementa notablemente con el número de nodos que pertenecen a la red.

Como punto importante respecto a los protocolos de encaminamiento que revisamos, está que la pérdida de los enlaces responde de forma directa a la dinámica de la topología de la red ad hoc, es por ello que resulta útil conocer la forma en cómo se comporta la posición de un cierto número de nodos vecinos dentro de un área de cobertura, tal y como lo desarrollamos en este análisis.

⇒ **Sobre las condiciones controladas de simulación en ns-2 y de nuestros experimentos.-**

Aunque en la bibliografía encontramos que el simulador ns-2 es flexible para elegir distintos algoritmos de movimiento, a menos que se posean los algoritmos para ser implementados como los publicados en algunos artículos recientes, no es posible cambiar las condiciones del simulador de redes ns-2; en particular nos referimos al algoritmo de movimiento *random waypoint* en los escenarios móviles. Sin embargo la constitución del simulador en módulos, vistos como diferentes archivos para diferentes condiciones del escenario, hace posible resolver de forma aislada los parámetros en la simulación.

En los experimentos que realizamos, el tamaño y el número de nodos asignados respondieron bien ante las consideraciones que los justifican, pues mantuvimos a la región de cobertura

equiprobable a la permanencia de nodos que fueron vecinos al inicio de la simulación durante la ejecución del escenario.

⇒ **Sobre las curvas de aproximación matemática a partir de los valores promedio del número de nodos vecinos.-**

Una vez calculados los coeficientes del método de regresión lineal, el valor que obtuvimos en el coeficiente de correlación para ambos casos, es muy cercano a uno. Con esto se concluye que los valores promedio del número de vecinos iniciales que permanecen el área de cobertura del nodo origen están correctamente ajustados a una distribución exponencial negativa con parámetro λ en su argumento.

Todos nuestros experimentos comenzaron con un número de nodos igual a siete, de ahí que las curvas tienen esa convergencia en este valor inicial que es muy cercano al esperado de acuerdo a la densidad de nodos del escenario y al área del círculo de cobertura del nodo origen.

⇒ **Sobre el modelo general propuesto.-**

El modelo exponencial negativo que proponemos tiene como parámetros la velocidad promedio de los nodos y el radio del círculo de vecindad en donde nos interesa conocer el número de terminales que se mantienen dentro. Después de comparar el modelo general con la aproximación que obtuvimos por mínimos cuadrados encontramos que el error relativo:

- En el caso del escenario A o escenario lento, después de los 600 [s] tenemos un error mayor al 20% y hasta llegar a poco más del 50%. Nótese que después de los 600 [s] el número de nodos es menor a un séptimo del número inicial de nodos.
- En el escenario B o rápido, durante toda la simulación se presenta un error relativo cercano al 10% que es poco significativo.

Observamos que ambas funciones: la aproximada y la propuesta, se alejan considerablemente de los valores promedio experimentales en los primeros segundos de simulación y dicho efecto se agudiza conforme más rápida sea la velocidad promedio de los nodos en el escenario. Este efecto ocurre en el primer cuarto del tiempo de simulación.

⇒ **Sobre las propuestas para futuros análisis.-**

Con el fin de eliminar el efecto señalado al inicio de las simulaciones, proponemos para trabajos posteriores evaluar varias alternativas:

- ✓ Verificar si este fenómeno sigue ocurriendo para escenarios con velocidades promedio mucho mayores y de ser el caso:
 - Incrementar el número de escenarios iguales a los que proponemos.
 - Incrementar la cantidad de nodos en cada escenario.

Por otra parte, este ha sido un análisis considerando el algoritmo de movimiento *random waypoint*; resultaría interesante conocer el comportamiento de estos escenarios de redes de tipo ad hoc bajo otros algoritmos de movimiento.

Glosario

Ad hoc: Es un método de conexión de una red en la mayoría de los casos, de dispositivos inalámbricos. Una conexión en una red ad hoc se establece en el tiempo que existe la sesión y no requiere estación base alguna para su control, en lugar de ello, los dispositivos recurren a otros dentro su de rango de comunicación para constituir la red. En un transmisión, las terminales ad hoc buscan a sus destinos fuera de su rango inundando la red con mensajes que al ser respondidos, forman una o varias rutas hacia el destino. Las conexiones se realizan por medio de múltiples saltos entre las terminales.

Backbone: La red backbone es un elemento importante de las redes pues 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 con diferentes protocolos. Generalmente la capacidad de la red backbone es mayor que las conectadas a ella.

Bit error rate (BER): Número de bits erróneos divididos por el número total de bits transmitidos, recibidos o procesados en un cierto intervalo de tiempo.

Bluetooth: Nombre asignado a una especificación de redes inalámbricas de área personal., que es desarrollada por el grupo Bluetooth SIG. Permite que dispositivos constituyan por sí mismos una red inalámbrica de área pequeña.

Broadcast: Tipo de transmisión de datos utilizada en redes con canal compartido, como el caso de Ethernet. Se resume como un método de transmisión uno a muchos en donde el mensaje llega a todos los dispositivos asociados a la red.

Dwell time: Variable aleatoria que representa el tiempo total en el que una terminal móvil mantiene comunicación satisfactoria con una estación base. En términos generales es el tiempo en el cual un proceso dinámico permanece detenido con el fin de que ocurra otro proceso.

Encaminamiento: Proceso para determinar o establecer una trayectoria hacia la terminal destino. Este proceso llega a ser complejo en una red ad hoc por la cantidad de terminales intermedias en movimiento que un mensaje debe atravesar para llegar al destino.

Flooding: Mecanismo de descubrimiento de un camino o ruta hacia el destino, que se utiliza de forma general en redes cableadas e inalámbricas Su propagación se realiza en forma de broadcast y se propaga hacia toda la red.

Handoff: En los sistemas celulares es el proceso para transferir una llamada telefónica que se desarrolla en una célula controlada por una estación base a ciertas frecuencias, a otra célula controlada por otra estación base bajo un rango diferente de frecuencias, sin interrumpir la llamada.

Home RF: Es una designación de redes inalámbricas de área local para dispositivos domésticos que comparten información.

Overhead: Información digital transmitida por una interfase funcional entre un usuario y un sistema de telecomunicaciones o entre unidades funcionales dentro del sistema de

telecomunicaciones; con el propósito de encaminar o controlar la transferencia de la información del usuario o bien, detectar y corregir errores. La información overhead es sólo aquella que es generada por el sistema de telecomunicaciones sin ser entregada al usuario.

Paquete Hello: Paquete de control que se envía a periodos constantes de tiempo entre nodos para establecer vecindad entre ellos.

Request for comments: Conjunto de notas técnicas y organizativas donde se describen los estándares o recomendaciones de Internet.

Roaming: Concepto utilizado en comunicaciones inalámbricas relacionado con la capacidad de un dispositivo para moverse de una zona de cobertura a otra. El concepto de roaming se utiliza en las redes Wi-Fi y en este contexto significa que el dispositivo Wi-Fi cliente puede desplazarse e ir registrándose en diferentes bases o puntos de acceso. En telefonía móvil el roaming es la capacidad de hacer y recibir llamadas en redes móviles fuera del área de servicio local.

Throughput: Número de bits, caracteres o paquetes que viajan en un sistema de comunicación de datos. El throughput se expresa en unidades de datos sobre unidades de tiempo. En ocasiones así es como se le denomina a la máxima capacidad de un canal o sistema de comunicaciones. Es también una medida de la cantidad de trabajo realizado por un sistema en un periodo de tiempo.

Siglas y acrónimos

AHN	Ad Hoc Network
AODV	Ad-hoc On Demand Distance Vector
ARPANET	Advanced Research Projects Agency Network
ATM	Asynchronous Transfer Mode
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
DARPA	Defense Advanced Research Projects Agency
DSDV	Destination-Sequenced Distance Vector
DSN	Distributed Sensor Networks
DSR	Dynamic Source Routing
DSSS	Direct Sequence Spread Spectrum
ETSI	European Telecommunication Standard Institute
FCC	Federal Communications Commission
FHSS	Frequency Hopping Spread Spectrum
HIPERLAN	High Performance Radio Local Area Network
IEEE	Institute of Electric and Electronic Engineers
IETF	Internet Engineering Task Force
IP	Internet Protocol
ISM Frequencies	Industrial, Scientific and Medical frequencies
LOS	Low Orbit Satellite
MANET	Mobile Ad hoc NETWORKs
OFDM	Orthogonal Frequency Division Multiplexing
OLSR	Optimized Link State Routing protocol
PAN	Personal Area Network
PWRP	Predictive Wireless Routing Protocol
RFC	Request for comments
SOSUS	Sound Surveillance System
TORA	Temporally-Ordered Routing Algorithm
UMTS	Universal Mobile Telecommunications System
WLAN	Wireless Local Area Network

Apéndices

1.- Código *gráfica.m*: programa en Matlab que calcula y grafica los resultados de las simulaciones realizadas.

```

1 ***** Escenario A: 3000x3000 n=300 v=1[m/s] t=1000[s] *****
2
3 - clc
4 - clear
5 - close all
6
7 - n=11; ***** Número de muestras
8 - Fil=fopen('imps.txt','r'); ***** Abre archivo con los datos experimentales
9 - Info=fscanf(Fil,'%f',[n 10]); ***** Coloca el contenido del archivo imps.txt en la matriz Info
10 - fclose(Fil); ***** Cierra apuntador de archivo
11
12 - Time=[0:100:1000]; ***** General vector de tiempos de muestreo
13 - Time=Time'; ***** Transpone la matriz Time
14
15 - for j=1:n ***** Son 11 muestras
16 -     for i=1:10 ***** Son 10 experimentos
17 -         X(i)=Info(j,i); ***** Coloca en vector X los datos de cada muestreo
18 -     end
19
20 ***** Cálculo del intervalo de confianza al 95 % *****
21
22 - c=2.527485776; ***** Valor de la función inversa t-student con n-1
23 ***** grados de libertad al 95% de confianza
24
25 - Vmedio=mean(X); ***** Calculo de valor promedio de las muestras
26 - SumaXXmed=0; ***** Inicia suma (Xk-Xmed)^2 en cero
27 - for k=1:10
28 -     SumaXXmed=SumaXXmed+(X(k)-Vmedio)^2; ***** Suma (Xk-Xmed)^2
29 - end
30 - S=sqrt(1/(n-1)*SumaXXmed); ***** Calculo de desviación estándar de las muestras
31
32 - Inferior=Vmedio-S*c/sqrt(n); ***** Limite inferior de intervalo
33 - Superior=Vmedio+S*c/sqrt(n); ***** Limite superior de intervalo
34
35 - if Inferior<0
36 -     Inferior=0; ***** En caso de ser limite menor a cero
37 - end
38 - if Superior>7
39 -     Superior=7; ***** En caso de ser limite mayor al máximo
40 - end
41
42 *****
43
44 - CI(j,1)=Inferior; ***** Vector con limites inferiores
45 - CI(j,2)=Superior; ***** Vector con limites superiores
46 - Media(j)=Vmedio; ***** Vector con valores promedios
47 - end
48
49 - figure(1) ***** Inicia figura 1
50 - box on; ***** Activa borde de la gráfica
51 - hold
52
53 - title('\it{Escenario A: 3000x3000[m] nodos=300 velocidad=1[m/s]}',
54 - 'FontSize',14); ***** Título de figura 1
55
56 - xlabel('Tiempo [s]','FontSize',12); ***** Etiqueta en eje de abscisas
57 - ylabel('Número de nodos vecinos','FontSize',12); ***** Etiqueta en eje de ordenadas
58 - stem(Time,Info,'b*','MarkerSize',8); ***** Gráfica de valores experimentales
59 - plot(Time,Media,'-bo','MarkerSize',6,'LineWidth',1); ***** Grafica valores medios
60
61 - figure(2) ***** Inicia figura 2
62 - hold
63 - box on; ***** Activa recuadro de figura 2
64 - plot(Time,Media,'-bo','MarkerSize',6,'LineWidth',1); ***** Grafica valores promedio
65
66 ***** Aproximación por regresión lineal a una recta: método de mínimos cuadrados *****
67
68 - T=0:100:1000; ***** Valor de abscisas T
69 - Y=log(10*Media); ***** Valor de ordenadas Y escaladas a 10
70 - SumaTY=sum(T.*Y); ***** Suma de productos T Y

```

```

71 - MediaT=mean(T);
72 - MediaY=mean(Y);
73 - SumaT2=sum(T.*T);
74
75 - B=(SumaY-n*MediaT*MediaY)/(SumaT2-n*MediaT^2);
76 - A=MediaY-B*MediaT;
77 - a=exp(A)/10;
78 - b=B;
79
80 - YNueva=A+B.*T;
81 - SumYMedTN=sum((MediaY-YNueva).^2);
82 - SumYTN=sum((MediaY-Y).^2);
83 - r=sqrt(SumYMedTN/SumYTN);
84
85
86
87 - t=0:0.1:1000;
88 - yreg=a*exp(b*t);
89 - plot(t,yreg,'-b','LineWidth',2);
90
91 - y=6.545*exp(-1/250*t);
92 - plot(t,y,'.b','LineWidth',2);
93
94
95
96 - plot(Time,CI,'+b','MarkerSize',6);
97 - j=1;
98 - for i=0:100:1000
99 -     min=CI(j,1);
100 -    max=CI(j,2);
101 -    A=[min:0.1:max];
102 -    plot(A,A,'.b','MarkerSize',1);
103 -    clear A;
104 -    j=j+1;
105 - end
106
107 - title('\tit(Escenario A: 3000x3000[m] nodos=300 velocidad=1[m/s]'),'
108 - 'FontSize',14)
109
110 - legend('Valor promedio','Aproximación por regresión lineal','F(t) = 6.545 exp(-\lambda t)');\lambda = v/R '
111 - 'Intervalo de confianza al 95% ',1);
112
113 - xlabel('Tiempo [s]','FontSize',12);
114 - ylabel('(Número de nodos vecinos)','FontSize',12);
115
116
117
118 - figure(3)
119 - box on;
120 - grid on;
121 - semilogy(Time,Media,'b','MarkerSize',7);
122 - ylim([0,10]);
123 - hold
124
125 - semilogy(t,yreg,'-b','LineWidth',1);
126 - semilogy(t,y,'.b','LineWidth',1);
127
128 - title('\tit(Escenario A: 3000x3000[m] nodos=300 velocidad=1[m/s]'),'
129 - 'FontSize',14)
130
131 - legend('Valores promedio','Aproximación por regresión lineal',
132 - 'f(t) = 6.545 exp(-\lambda t); \lambda = v/R ',1);
133
134 - xlabel('Tiempo [s]','FontSize',12);
135 - ylabel('(Número de nodos vecinos)','FontSize',12);
136
137 - T=0:100:1000;
138 - yreg=a*exp(b*T);
139 - y=6.545*exp(-1/250*T);
140
141 - Error=abs(y-yreg)./yreg*100;
142
143 - figure(4)
144 - box on;
145 - bar(T,Error);
146 - ylim([0,100]);
147
148 - title('\tit(Escenario A: 3000x3000[m] nodos=300 velocidad=1[m/s]'),'
149 - 'FontSize',14)
150
151 - xlabel('Tiempo [s]','FontSize',12);
152 - ylabel('(Porcentaje de error relativo)','FontSize',12);
153 -

```

2.- Código *fotos.m*: programa en Matlab que grafica a los nodos en los instantes de muestreo.

```

1 ***** Este programa muestra imágenes a tiempos de muestreo *****
2 - clc
3 - clear
4 - close all
5
6 - Fil=fopen('t400.txt','r');          *** Abre archivo con las coordenadas de todos los nodos
7 - Datos=fscanf(Fil,'%f',[3 300]);    *** Coloca el contenido del archivo t400.txt en la matriz Datos
8 - fclose(Fil);                       *** Cierra apuntador de archivo
9 - Datos=Datos';                       *** Transpone matriz Datos
10
11 - figure(1)                           *** Inicia figura 1
12 - hold;
13 - plot(3000,3000,'.k','MarkerSize',1); *** Gráfica de un punto en la esquina superior derecha
14 - box on;                              *** Activa borde de la gráfica
15 - xlabel('Posición X');                *** Etiqueta en eje de abscisas
16 - ylabel('Posición Y');                *** Etiqueta en eje de ordenadas
17
18 - Vecinos=[238 214 277 261 22 3 259 173 142 67 176 221 271 129]; *** Identificador de los nodos vecinos
19 - D=size(Vecinos);                    *** Número de nodos vecinos
20
21 ***** Gráfica de los vecinos *****
22
23 - for i=1:D(2)
24 -     X=Datos(Vecinos(i)+1,2);          *** Coordenada X del vecino
25 -     Y=Datos(Vecinos(i)+1,3);          *** Coordenada Y del vecino
26 -     plot(X,Y,'+k','MarkerSize',7);    *** Gráfica del vecino
27 - end
28
29 ***** Gráfica del resto de los nodos *****
30
31 - for i=0:299
32 -     if find(Vecinos==i)
33 -     else
34 -         X=Datos(i+1,2);                *** Coordenada X del nodo
35 -         Y=Datos(i+1,3);                *** Coordenada Y del nodo
36 -         plot(X,Y,'ok','MarkerSize',4); *** Gráfica del nodo
37 -     end
38 - end
39
40 ***** Gráfica del nodo origen *****
41
42 - Origen=108;                            *** Identificador del nodo origen
43 - XO=Datos(Origen+1,2);                  *** Coordenada X del nodo origen
44 - YO=Datos(Origen+1,3);                  *** Coordenada Y del nodo origen
45 - plot(XO,YO,'+k','MarkerSize',8);      *** Gráfica del nodo origen
46
47 ***** Gráfica de la circunferencia límite de la zona de cobertura *****
48
49 - R=250;                                  *** Radio de la circunferencia
50 - X=XO-R:1:XO+R;                          *** Definición del dominio de la función
51 - Yp=sqrt(250^2-(X-XO).^2)+YO;             *** Cálculo de sección superior de la circunferencia
52 - Yn=-sqrt(250^2-(X-XO).^2)+YO;           *** Cálculo de sección inferior de la circunferencia
53 - plot(X,Yp,'-k');                        *** Gráfica de la sección superior de la circunferencia
54 - plot(X,Yn,'-k');                        *** Gráfica de la sección inferior de la circunferencia
55
56 - title('\it(Escenario B: 3000x3000 [m], 300 nodos, velocidad promedio 3 [m/s]'); *** Título de la gráfica
57 - 'FontSize',14);
58 - pbaspect([1 1 1]);                      *** Activa el aspecto cuadrado de la gráfica
59 ***** Fin de programa *****

```


Apéndices

1.- Código gráfica.m: programa en Matlab que calcula y grafica los resultados de las simulaciones realizadas.

```

1 ***** Escenario A: 3000x3000 n=300 v=1[m/s] t=1000[s] *****
2
3 - c1c
4 - clear
5 - close all
6
7 - n=11;
8 - Fil=fopen('lmps.txt','r');
9 - Info=fscanf(Fil,'%f',[n 10]);
10 - fclose(Fil);
11
12 - Time=[0:100:1000];
13 - Time=Time';
14
15 - for j=1:n
16 -     for i=1:10
17 -         X(i)=Info(j,i);
18 -     end
19
20 - ***** Cálculo del intervalo de confianza al 95 % *****
21
22 - c=2.527485776;
23
24
25 - Vmedio=mean(X);
26 - SumaXXmed=0;
27 - for k=1:10
28 -     SumaXXmed=SumaXXmed+(X(k)-Vmedio)^2;
29 - end
30 - S=sqrt(1/(n-1)*SumaXXmed);
31
32 - Inferior=Vmedio-S*c/sqrt(n);
33 - Superior=Vmedio+S*c/sqrt(n);
34
35 - if Inferior<0
36 -     Inferior=0;
37 - end
38 - if Superior>7
39 -     Superior=7;
40 - end
41
42 - *****
43
44 - CI(j,1)=Inferior;
45 - CI(j,2)=Superior;
46 - Media(j)=Vmedio;
47 - end
48
49 - figure(1)
50 - box on;
51 - hold
52
53 - title(['it(Escenario A: 3000x3000[m] nodos=300 velocidad=1[m/s]')',
54 - 'FontSize',14]);
55
56 - xlabel('Tiempo [s]','FontSize',12);
57 - ylabel('(Número de nodos vecinos)','FontSize',12);
58 - stem(Time,Info,'b*','MarkerSize',8);
59 - plot(Time,Media,'-bo','MarkerSize',6,'LineWidth',1);
60
61 - figure(2)
62 - hold
63 - box on;
64 - plot(Time,Media,'-bo','MarkerSize',6,'LineWidth',1);
65 - ***** Aproximación por regresión lineal a una recta: método de mínimos cuadrados *****
66
67
68 - T=0:100:1000;
69 - Y=log(10*Media);
70 - SumaTY=sum(T.*Y);

```

```

71 - MediaT=mean(T);
72 - MediaY=mean(Y);
73 - SumaT2=sum(T.*T);
74
75 - B=(SumaTY-n*MediaT*MediaY)/(SumaT2-n*MediaT^2);
76 - A=MediaY-B*MediaT;
77 - a=exp(A)/10;
78 - b=B;
79
80 - YNuevas=A+B.*T;
81 - SumYMedTN=sum((MediaY-YNuevas).^2);
82 - SumYTN=sum((MediaY-Y).^2);
83 - r=sqrt(SumYMedTN/SumYTN);
84
85
86
87 - t=0:0.1:1000;
88 - yreg=a*exp(b*t);
89 - plot(t,yreg,'-b','LineWidth',2);
90
91 - y=6.545*exp(-1/250*t);
92 - plot(t,y,'b','LineWidth',2);
93
94
95
96 - plot(Time,CI,'b','MarkerSize',6);
97 - j=1;
98 - for i=0:100:1000
99 -     min=CI(j,1);
100 -     max=CI(j,2);
101 -     A=[min:0.1:max];
102 -     plot(i,A,'b','MarkerSize',1);
103 -     clear A;
104 -     j=j+1;
105 - end
106
107 - title('\it{Escenario A: 3000x3000[m] nodos=300 velocidad=1[m/s]}',
108 - 'FontSize',14)
109
110 - legend('Valor promedio','Aproximación por regresión lineal','F(-) = 6.545 exp(-\lambda t);\ lambda = v/R ',
111 - 'Intervalo de confianza al 95%',1);
112
113 - xlabel('Tiempo [s]','FontSize',12);
114 - ylabel('(Número de nodos vecinos) ','FontSize',12);
115
116
117
118 - figure(3)
119 - box on;
120 - grid on;
121 - semilogy(Time,Media,'b*','MarkerSize',7);
122 - ylim([0.01,10]);
123 - hold
124
125 - semilogy(t,yreg,'-b','LineWidth',1);
126 - semilogy(t,y,'b','LineWidth',1);
127
128 - title('\it{Escenario A: 3000x3000[m] nodos=300 velocidad=1[m/s]}',
129 - 'FontSize',14)
130
131 - legend('Valores promedio','Aproximación por regresión lineal',
132 - 'f(t) = 6.545 exp(-\lambda t);\ lambda = v/R ',1);
133
134 - xlabel('Tiempo [s]','FontSize',12);
135 - ylabel('(Número de nodos vecinos) ','FontSize',12);
136
137 - T=0:100:1000;
138 - yreg=a*exp(b*T);
139 - y=6.545*exp(-1/250*T);
140
141 - Error=abs(y-yreg)./yreg*100;
142
143 - figure(4)
144 - box on;
145 - bar(T,Error);
146 - ylim([0,100]);
147
148 - title('\it{Escenario A: 3000x3000[m] nodos=300 velocidad=1[m/s]}',
149 - 'FontSize',14)
150
151 - xlabel('Tiempo [s]','FontSize',12);
152 - ylabel('(Porcentaje de error relativo) ','FontSize',12);
153

```

2.- Código *fotos.m*: programa en Matlab que grafica a los nodos en los instantes de muestreo.

```

1  ***** Este programa muestra imágenes a tiempos de muestreo *****
2  -  clc
3  -  clear
4  -  close all
5
6  -  Fil=fopen('t400.txt','r');          *** Abre archivo con las coordenadas de todos los nodos
7  -  Datos=fscanf(Fil,'%f',[3 300]);   *** Coloca el contenido del archivo t400.txt en la matriz Datos
8  -  fclose(Fil);                      *** Cierra apuntador de archivo
9  -  Datos=Datos';                      *** Transpone matriz Datos
10
11 -  figure(1)                           *** Inicia figura 1
12 -  hold;
13 -  plot(3000,3000,'.k','MarkerSize',1); *** Gráfica de un punto en la esquina superior derecha
14 -  box on;                             *** Activa borde de la gráfica
15 -  xlabel('Posición X');                *** Etiqueta en eje de abscisas
16 -  ylabel('Posición Y');               *** Etiqueta en eje de ordenadas
17
18 -  Vecinos=[238 214 277 261 22 3 259 173 142 67 176 221 271 129]; *** Identificador de los nodos vecinos
19 -  D=size(Vecinos);                    *** Número de nodos vecinos
20
21 ***** Gráfica de los vecinos *****
22
23 -  for i=1:D(2)
24 -      X=Datos(Vecinos(i)+1,2);        *** Coordenada X del vecino
25 -      Y=Datos(Vecinos(i)+1,3);        *** Coordenada Y del vecino
26 -      plot(X,Y,'+k','MarkerSize',7);  *** Gráfica del vecino
27 -  end
28
29 ***** Gráfica del resto de los nodos *****
30
31 -  for i=0:299
32 -      if find(Vecinos==i)
33 -          else
34 -              X=Datos(i+1,2);          *** Coordenada X del nodo
35 -              Y=Datos(i+1,3);          *** Coordenada Y del nodo
36 -              plot(X,Y,'ok','MarkerSize',4); *** Gráfica del nodo
37 -          end
38 -      end
39
40 ***** Gráfica del nodo origen *****
41
42 -  Origen=108;                           *** Identificador del nodo origen
43 -  XO=Datos(Origen+1,2);                 *** Coordenada X del nodo origen
44 -  YO=Datos(Origen+1,3);                 *** Coordenada Y del nodo origen
45 -  plot(XO,YO,'+k','MarkerSize',8);     *** Gráfica del nodo origen
46
47 ***** Gráfica de la circunferencia límite de la zona de cobertura *****
48
49 -  R=250;                                *** Radio de la circunferencia
50 -  X=XO-R:1:XO+R;                        *** Definición del dominio de la función
51 -  Yp=sqrt(250^2-(X-XO).^2)+YO;          *** Cálculo de sección superior de la circunferencia
52 -  Yn=-sqrt(250^2-(X-XO).^2)+YO;        *** Cálculo de sección inferior de la circunferencia
53 -  plot(X,Yp,'-k');                      *** Gráfica de la sección superior de la circunferencia
54 -  plot(X,Yn,'-k');                      *** Gráfica de la sección inferior de la circunferencia
55
56 -  title('\it(Escenario B: 3000x3000 [m], 300 nodos, velocidad promedio 3 [m/s])',
57 -  'FontSize',14);                       *** Título de la gráfica
58 -  pbaspect([1 1 1]);                    *** Activa el aspecto cuadrado de la gráfica
59 ***** Fin de programa *****

```


Fireworks: A Novel Route-discovery Protocol for Mobile Ad-Hoc Networks

Javier Gomez, Juan M. Cervantes, Victor Rangel and Rodrigo Atahualpa
Department of Electrical Engineering
National University of Mexico, UNAM
Mexico, Mexico City
emails...

Abstract— Routing protocols in mobile ad hoc networks usually discover routes by flooding the entire network with signaling packets, this technique is known as blind-flooding. Flooding the network with control packets, however, puts a heavy burden on network performance (i.e., congestion) especially in large dense networks. This document presents *Fireworks*, an efficient route-discovery protocol for wireless ad-hoc networks. In *Fireworks*, a source node floods a limited portion of the network looking not only for the target node but also for other nodes (called neighbors) that were known to be close to the target node in the past. Neighbor nodes can be used as anchor points where another limited flooding can take place in search for the target node. Because only a limited portion of the network is flooded by control packets around the source and target nodes, fireworks can significantly reduce the signaling overhead compared with blind flooding. A key feature of fireworks is that the signaling overhead is independent of network size, and therefore, it can scale to large networks.

I. INTRODUCTION

An ad hoc network is a collection of nodes equipped with a common wireless interface in an area where there is not a significant fixed infrastructure in place. Because nodes have a limited range usually, it is necessary that nodes participate in the routing process relaying packets between source-destinations pairs. This type of routing is also known as multi-hop routing. Finding a route among nodes has been, since the origins of ad hoc networks in the 70's, the main challenge and even today there is a significant amount of research going on in this area showing that routing is still far away from being solved.

Books typically divide routing protocols for ad hoc networks as either proactive, reactive or hybrid (i.e., a mix of both). Proactive protocols discover routes from any node to all the other nodes in the network in advance, and these routes are periodically updated as route changes occur. In this category we find the wireless variations of the well known link-state and distance vector protocols. Protocols in this category have the main advantage that whenever a node needs to send a packet to any other node in the network, there is already a route available so there is no delay involved. Reactive protocols, on the other hand, discover routes only when they are needed. This operation adds a delay involved in finding a route before data packets can be sent to the destination. Because reactive protocols generate far less signaling overhead compared with

proactive protocols they are preferred in most ad hoc scenarios that are not for military use.

Most of the routing protocols standardized by the MANET working group of the IETF fall are reactive. Routing protocols such as DSR [1], AODV [2], TORA [3] and others discover routes (the route-discovery phase) only after there is a need for a route. Once a route is found, data packets can then move from source to destination. In static networks routes are always available, and they only disrupted if some nodes are turned off or run out of power. In mobile networks, a node that is part of a route in a given time, it may no longer be in the same route a time later because of mobility. Whatever the reason, any time a route is no longer valid, it must be repaired in order to continue the flow of packets (the route maintenance phase). unicast, multicast, anycast...

In this paper we focus on the route-discovery phase of unicast routing protocol for ad hoc networks. Although there are several improved proposals for route-discovery (proposals that are discussed later in detail in this paper), the blind-flooding technique remains as the most widely used protocol in practice. Blind flooding operated as follow. A source node transmits a broadcast control packet to announce its intention to communicate with a certain destination node. Nodes overhearing this request retransmit the control packet, thus increasing the coverage area of the search. This simple mechanism has the effect of flooding the network with control packets from the source node to the entire network. Because a node may overhear the same request but from different nodes, each control packet has a unique ID number so nodes retransmit a request once only.

Blind flooding is adequate for small, low density and slow mobility networks. When this is not the case, blind flooding generates a large number of control packets every time there is a need for a route. These control packets compete with data packets for the limited bandwidth found in ad hoc networks.

For static ad-hoc networks, route discovery has no extra difficulties except for congestion in the network and the inherited forwarding latency. Besides, with lack of mobility, routes lifetime is long and a new discovery is needed only when a node stops functioning or when traffic congestion is present. On the other hand when nodes are mobile, route lifetime is short, being necessary several attempts to find a new route to

reach the destination when the route is not longer available, flooding the network any continuously, wasting bandwidth and increasing network latency.

II. RELATED WORK

Most existing ad-hoc protocols distribute or collect topology information as queries/responses throughout the network on a proactive or reactive manner. Reactive protocols such as DSR [1], AODV [2], TORA [3], create routes to destinations every time it needed. This topology information (routes known by each node) is saved on a route cache so that nodes receiving a new query can answer quickly to topology changes, shortening routes. This collected information is not always used to improve the performance of flooding process.

Geographic positions can be used also to find routes to nodes as it is shown by Finns Cartesian Routing [7], which addresses each node with a unique identifier and geographic location using apparently a GPS system (no defined by the author) to send the packets through the closest neighbor to the destination node.

Another approach similar to Cartesian Routing is GLS [4], which gets the location information by means of a service such as GPS in terms of latitude and longitude. This information is then broadcasted periodically by each node in HELLO packets so that all nodes can form a table with identities and geographic position of each neighbor. When a node needs to forward a packet towards the intended destination, it places the destination identity in the header of the packet. Each node receiving the packet consults its neighbor table and chooses the closest neighbor to the destination until being reached.

GPSR[5] is a geographic routing system that uses a planar sub graph of the wireless network graph to route around the holes. Both, GLS and GPSR are designed for large metropolitan area networks, but they need a high node density and expensive location devices.

Finally, FRESH [6] is an algorithm for efficient route discovery in mobile ad hoc networks. Nodes keep a record of their most recent encounter times with all others nodes. Instead of searching for the destination, the source node searches for any intermediate node that has reached the destination more recently than the source node did itself. The intermediate node then searches for a node that had encountered the destination yet more recently, and the procedure iterates until the destination is finally reached.

III. FIREWORKS

Opposite to blind-flooding where the entire network is flooded with control packets, Fireworks floods only two smaller regions in the network. One region is centered at the source node while the other region is located in the vicinity of the target node. In this way Fireworks is capable of reducing significantly the number of control packets in the network use for route-discovery, thus freeing precious bandwidth for data packets.

The main innovation of our approach is that the source node performs a restricted search looking not only for the

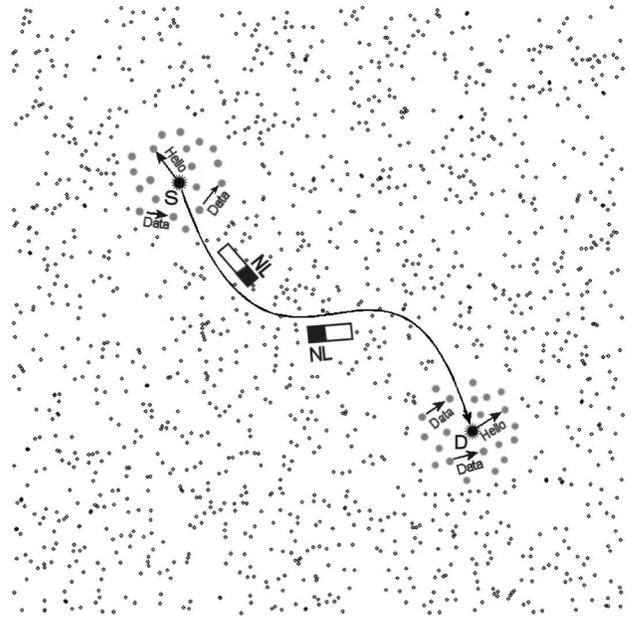


Fig. 1. Nodes transmit data or HELLO packets which are overheard by all nodes within the transmission range of the transmitter node. Nodes collect and keep this information in a neighbor list (NL). Any two nodes participating in a connection exchange their respective neighbor lists.

target node, but also for routing information about nodes that were neighbors of the target node recently. Upon neighbor information is collected, the source node sends unicast packet to the neighbors found which then perform an additional localized search. This additional search reduces to the flooded area by limiting it to a certain number of hops (k hops). See Figure 4

Fireworks is composed of three main mechanisms call neighbor discovery, network search and fireworks that are explained below:

Neighbor discovery: In Fireworks, all nodes collect neighbor information by overhearing packets from other nodes. A node can overhear packets by other nodes by putting its transceiver into "promiscuous mode". These packets can data or control packets. When a node has not transmitted any packet for a while, it may transmit a explicit control packet called "hello packets" so its one-hop-away neighbors learn about its presence (see Figure 1). These hello packets are not retransmitted in order to limit the signaling overhead. Every time a connection is established, source and destination nodes exchange their neighbor lists, along with the node's current routing table.

Route search : When a node receives a packet from upper layers or from another node, it checks weather it has a fresh route to the destination node. In case a route is available, the packet is relayed to the next hop immediately. In case no routing information is available, it transmit a Route Request packet (RREQ) including source and destination node IP addresses in it. This initial search is limited to only n hops away from the source node. In this RREQ the source node looks not only

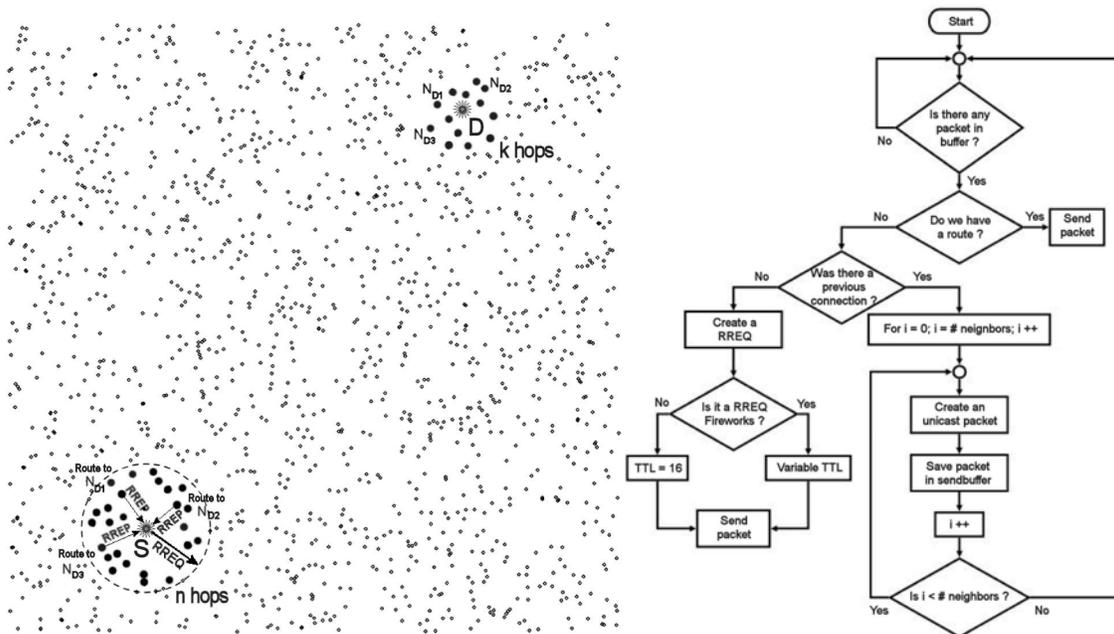


Fig. 2. Neighbor search phase: A source node floods an area limited to "n" hops and receives routing information of neighbors of the target node.

for the target node, but also for information about the location of neighbors of the target node. Upon receiving a RREQ, a node checks whether it has fresh information about either the target node or about neighbors of the target node. All nodes receiving a Fireworks RREQ, will open the neighbors' list table, checking for each one if it has already a route (for the destination node too);

In case a node has a route either to the target node or about neighbors of the target node, it answers via a RREP unicast packet to the source node. When this packet reaches the source node, data packets can then flow toward the target node. The process is shown in figure 5.

The parameter n plays a similar role in Fireworks as in other limited search flooding mechanisms such as the one described in . When n is small, only a limited region of the network is flooded with control packets, however there is also a higher probability that no routing information is collected and a second, larger-area search, becomes necessary. An opposite tradeoff applies when n is large. In Section XXX we give some insides about choosing the value of n .

Fireworks : Upon collecting routing information about neighbors of the target node, the source node sends unicast packets to each neighbor found in the neighbor search phase, we call these control packets "fireworks" packets. Upon receiving a firework packet, a neighbor node checks whether it has information about the target node, otherwise it performs a second search in a region limited to k hops looking for the target node. The number of hops of the new request varies according to the size of the network and mobility. This picture of a node sending a unicast packet that suddenly "explodes" into several packets resembles the well-known fireworks, giving the name to our protocol. Under this scheme,

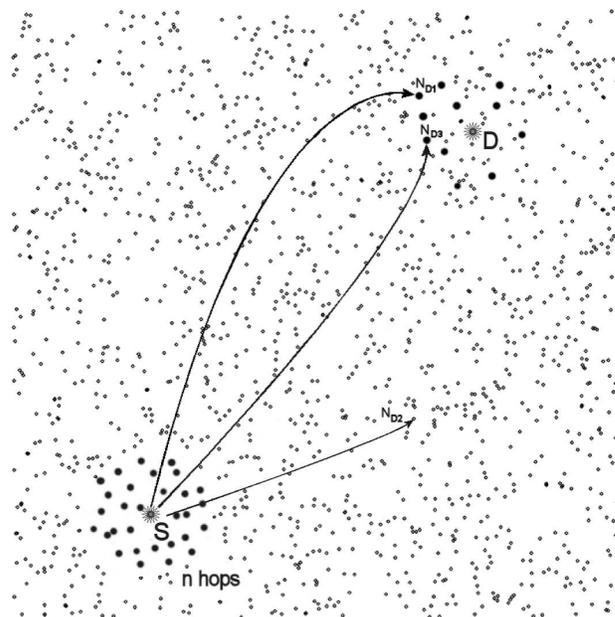


Fig. 3. Destination node (D) and its neighbors have moved during the time. Source node (S) has recovered routes to neighbors ND1, ND2 y ND3, sending the Fireworks packet (unicast) with the node (D) IP address included in the header.

only a small portion of the network near the target node will be flooded by control packets as it is illustrated in figure ?? . In case the target node is reached in this phase, it sends a route Reply (RREP) towards the source node which, as soon as it receives the RREP, it can transmit data packets to the target node.

When a neighbor node receives the Fireworks unicast

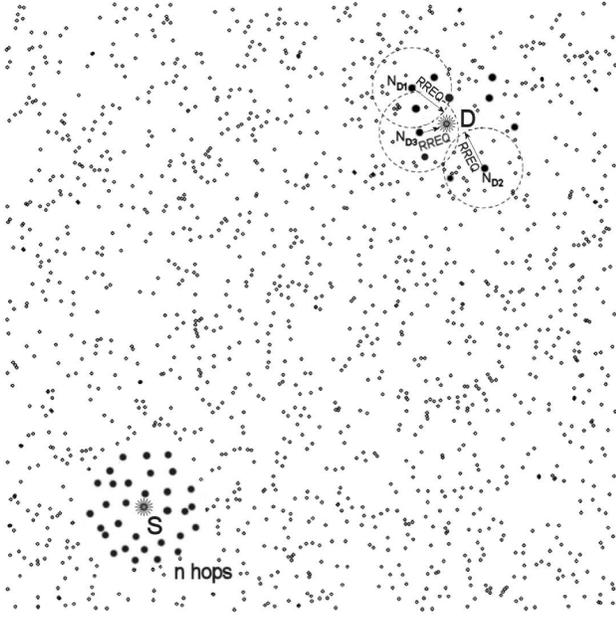


Fig. 4. Nodes ND1, ND2 and ND3, receive Fireworks packet and perform a new search for destination node (D) k hops away.



Fig. 5. Handling a Fireworks RREQ.

packet, it constructs a new RREQ packet with the IP address of the target node in the packet header. This packet is flooded to an area limited by k hops. The target node will then answer with a RREP to the source node so communication can be started.

When a source node has not a fresh route to send the packet With Fireworks in place, the source node may not have a route to the target node, but may have information about routes to target nodes neighbors. In this case the source node creates and sends few unicast Fireworks packets to each neighbor known; otherwise, it must perform a limited route-discovery search limited to n hops including in the header the neighbors nodes' IP addresses along with the destination address. The process

is shown in figure ??.

Using Fireworks, the route found between source and destination nodes may not be the shortest one. Routes known by the source node's neighbors and destination node neighbor in Fireworks where generated separately and when concatenated, may present more hops than necessary. In this case, there are already route shortening algorithms [1] that can be used to remove some links that are no longer necessary. Even without considering route shortening protocols, the great reduction of control packets achieved by Fireworks justifies the increased hop count of routes found by Fireworks.

IV. PERFORMANCE

Fireworks was implemented using the NS2 network simulator [8]. We replace the route-discovery mechanism of DSR and put Fireworks instead. However we kept the route maintenance component of DSR to take advantage of route information generated by all requests, replies and packet relying of DSR. As we mention before, Fireworks is not tied to any routing protocol in particular, and it can be used with any protocol replacing the route-discovery mechanism.

In order to control the number of hops that a route request RREQ is propagated in the network, we modified the time-to-live value of the request (16 hops is the default in DSR), so that we can control its value to a fixed small number.

In what follows, we present the performance of Fireworks under different network conditions to stress its advantages and disadvantages. In all cases ad hoc nodes use the standard IEEE 802.11 MAC protocol running at 2 mbps, DSR is used for route maintenance. Table X shows the parameter used in the experiments.

Data rate	1pkt/s
Packet size	100 bytes
Node speed	0, 2 and 10 m/s
Scenario size	2200x1200m a
number of connections	10, 30, 60, 80, 100

TABLE I
SIMULATION PARAMETERS

On each simulation, a number random connections are created prior a route-discovery search is performed with Fireworks (scenarios for 10, 30, 60 and 80 random connections were evaluated). The results obtained are compared to those obtained by standard DSR under similar conditions. All connections are of type UDP/CBR transmitting a 100 byte long packet every 1 second. The results shown in Figures X and X are the result of averaging 10 different scenarios.

Figure XX shows (a) number of different neighbors found in the route-search phase, (b) number of those neighbors that were reached by the source node with unicast packets and (c) number of times that reached neighbors found the target node after performing a second localized search.

Figure XXX shows the performance of Fireworks with no mobility. As expected, we observe that the number of neighbors found in the neighbor search phase increases as

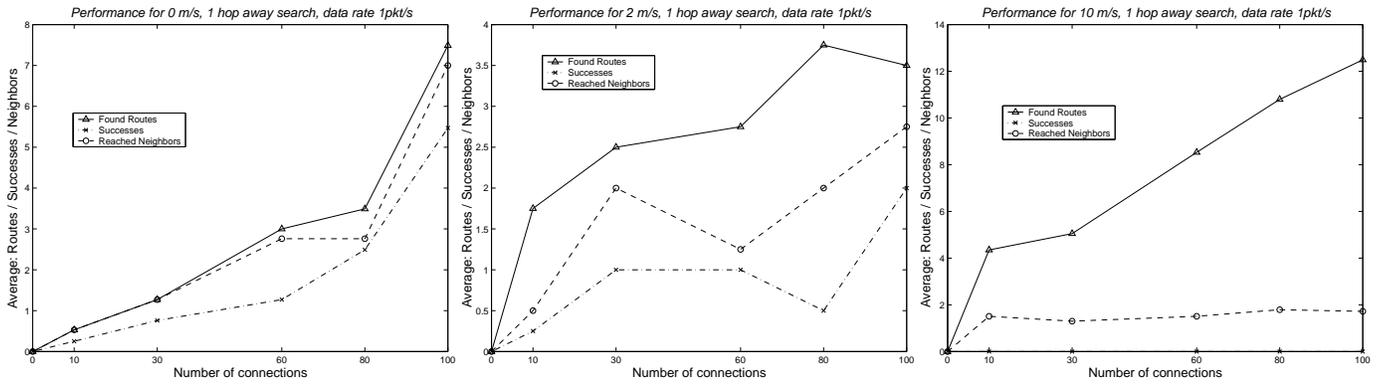


Fig. 6. Fireworks Performance at 0 m/s, 2 m/s and 10m/s

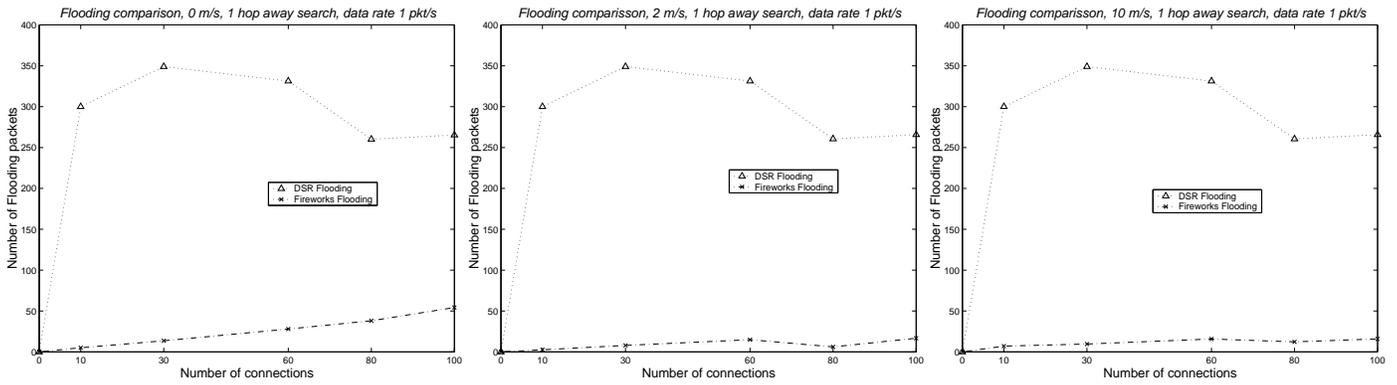


Fig. 7. Flooding Performance 0m/s, 2 m/s and 10 m/s

the number of connections in the network increases. As we mention before the main drawback of Fireworks is that it requires a minimum amount of background traffic to work properly. The good news is that even for few connections some information about neighbors can be found in the first search. In this figure we also observe that most fireworks packets sent by the source node reached their intended receiver except for high traffic conditions. The packets that did not reach the other end were dropped by the network because of traffic congestion. Once the neighbor got the fireworks packet, it always reached the target node except for high traffic loads again due to congestion and dropped packets.

In Figures X and Y we show the Fireworks performance when nodes move at 2 m/s and 10 m/s, respectively. Mobility brings both good and bad news to the performance of Fireworks. When nodes move, they move around with their corresponding knowledge about routing information to other nodes in the network. Since mobility makes it easier for a node to meet with other nodes, a node increases its overall knowledge of routing information as time passes. This is in contrast with the static case where a source node can not retrieve routing information from nodes outside the area covered by the first search. This phenomenon can be seen in Figure X where the number of neighbors found in the first search increases faster with respect to the number of connections compared with figure X. On the other hand, mobility makes routing information

less reliable as time passes. This can be observed in Figure X and Y where some fireworks packets did not reach the intended neighbor due to broken links. An extreme case of the impact of mobility can be seen in Figure Y where only few fireworks packets reached the intended neighbors, and even if some firework packets reached some neighbors, none found the target node during the second search.

Figures 7 a-c show the amount of signaling packets generated by Fireworks for the $N + 1$ connection in the network only, where N is the background traffic determined by the number of existing flows in the ad hoc network. For comparison purposes we also plot the signaling overhead in case DSR is used instead of Fireworks. As we can see in Figures 7 a and b, Fireworks outperforms DSR by a great factor. This is again due to the fact that Fireworks floods only two small regions of the network as opposed to blind flooding where the entire network is flooded with control packets. Because no valid routes are found in Figure 7c we cannot compare the flooding performance for 10 m/s. However it is worth mentioning that both Fireworks and DSR failed to find useful routes for 10 m/s. This highlights that the performance of Fireworks is restricted by the same rules governing the behavior of proactive MANET protocols.

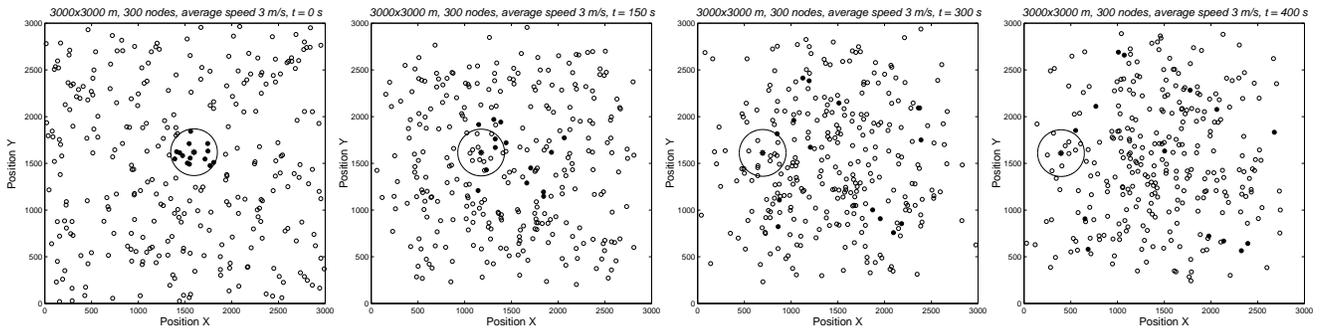


Fig. 8. Snapshot at 400 s

V. MODELING FIREWORKS ACCURACY

The performance of Fireworks, as many other routing protocols used in ad hoc networks, is influenced by the node dynamics [9], [12] therefore, mobility models should be detailed when showing simulation results. Moreover, there are studies related to node mobility in wireless networks [10], [11] to make the node distribution along the time a perfect random movement, generating better interpretations about the protocol proposals. The effectiveness of these movement algorithms is linked with the simulation scenario characteristics. As a matter of fact, node speed response during simulation time with a random waypoint scenario generates an average speed decay over the time [11].

The term known as dwell time or residence time [14], [15] is an independent variable that represents the total time of a base station that keeps satisfactory communication with a mobile node. Dwell time depends on many factors such as node speeds, paths, transmission power, signal propagation and interferences. This dwell time has an important role in the network evaluation and designing, furthermore this parameter is useful when calculating signaling algorithms and the channel occupation time.

In this section we analyzed the dwell time in a circular, mobile and full-connected area as our ideal scenario, considering an isotropic propagation model free of interferences.

Fireworks analysis showed dependence on simulated mobility scenarios specially the one with node speed equal to 10 m/s thus; the next report is focused on showing the mobility performance on a random waypoint scenario similar to the developed in the performance analysis detailed. Using ns-2 random walkpoint mobility algorithm, we create the next scenario:

- Size: 3000x3000 m
- Node number: 300
- Node average speed: 1 m/s

The objective is to record the quantity of neighbor nodes along the time surrounding a selected source node within the area established by a one-hop radius during all time needed to stop finding the initial neighbor nodes in this area. We prepared several independent events to be simulated and based on MAC 802.11, we use a 250 m radius. Figures ?? to 8 show snapshots

of an example recorded simulation with an average speed of 3 m/s.

Scenario size was limited by the previous Fireworks simulations in order keep the whole one-hop area with the same probability of neighbor nodes existence. Additionally, we selected randomly the source node within the scenario central area and in all cases, in which the initial number of nodes was bigger than seven, we discard nodes to select just seven from all the possible ones.

The first problem is to find a model that represents the time probability distribution for a neighbor node to be located within the source node's communication range. Then, we will find an analytical expression meaning the number of neighbor nodes within the source node's communication range at the beginning of the simulation time.

We define a random variable t assumed to be exponential distributed:

$$P(\tau > x) = e^{-\lambda t} \quad (1)$$

Or,

$$P(\tau \leq x) = 1 - e^{-\lambda t} \quad (2)$$

The variable t is the time in which a node is located within the source node connection area.

To have an estimated value about the number of nodes at the beginning of the simulation time, we found the node density and the connection area having assumed a uniform distribution of the nodes. So we have:

$$F(t) = \delta A_c P(t) \quad (3)$$

$$F(t) = \delta A_c e^{-\lambda t} \quad (4)$$

With:

- δ = Node density
- A_c = Circular connection area
- $P(t)$ = Permanence time probability

On the other hand, parameters δA_c and λ can be approximated with our simulation results using the linear regression method.

$$G(t) = A + Bt$$

$$G(t) = Ln(\delta A_c) - \lambda t$$

$$\implies A = Ln(\delta A_c) \implies \delta A_c = e^A$$

$$\implies B = -\lambda \implies \lambda = -B$$

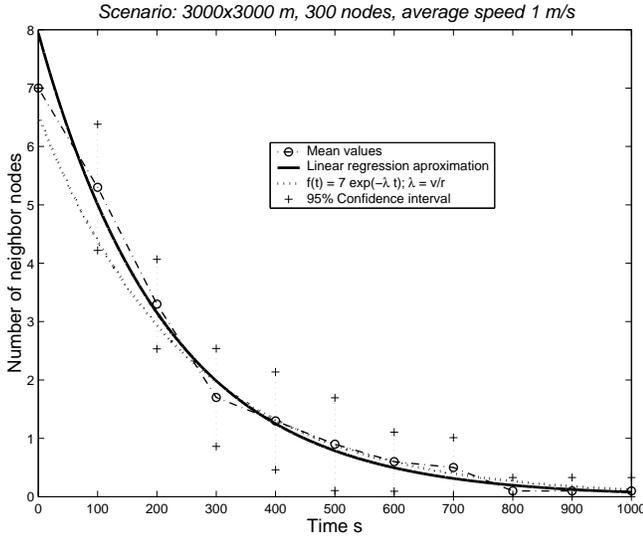


Fig. 9. Mobility results

Thus, the procedure is to approximate the data series to a line function making minimal the distances between the data point and the proposal line. Additionally there is a parameter called correlation coefficient that shows how close our approximation analysis is to the exponential model.

Calculating the linear regression coefficients (See figure 9), we have:

- $\delta A_c = 7.9468$
- $\lambda = 0.0046$
- $r = 0.9816$

$$F(t) = 7.9468e^{-0.0046t} \quad (5)$$

Based on the negative exponential function supported by the correlation coefficient r close to 1 justifying the exponential model and considering confidence intervals at 95%; we can formulate the next proposition:

$$\Phi(t) = \delta A_c e^{-\frac{v}{R}t} \quad (6)$$

Where the coefficients are:

- v = Node mean speed
- R = Connection area ratio

Substituting our scenario values, the number of nodes within the circular mobile connection area is expressed by :

$$\Phi(t) = 6.545e^{-\frac{1}{250}t} \quad (7)$$

A. Mobility results

Graph ?? has the samples of ten simulations with their each mean point. Moreover, it is shown the 95% confidence interval calculated assuming all samples with a normal probability distribution.

As a final analysis, we compared both functions by getting the values difference expressed as relative error percentage

from the general model $\Phi(t)$ to the linear regression approximated model $F(t)$. See figure ??.

$$\%Er = \frac{|F(t) - \Phi(t)|}{F(t)} 100 \quad (8)$$

VI. CONCLUSIONS

For this work a novel routing protocol for efficient route discovery was designed. Many scenarios were analyzed varying values such as network area, number of nodes, data rate, number of random connections and node mobility. From the performance observed above, we can conclude:

to neighbors' found routes, the factors which define the maximum number reached are traffic and mobility. When a node moves faster, it participates in the formation and update of more routes, and has the opportunity of listening to many others communications occurring in different places in the network. In the same way, when packets are send faster, changes in topology are detected more rapidly, updating routing table more frequently. On the other side, mobility impacts route life time significantly. When node moves faster, they change their position rapidly, causing ruptures on fresh links. Another important factor is traffic which causes contentions for channel use and collisions. It was observed that the number of reached neighbors rises when traffic and mobility increases. However, due to factors like network size and mobility, only a small portion of the found routes are useful for Fireworks unicast packet relaying. Related to the number of successes, we conclude that it is limited by the increasing of traffic and mobility. For small, medium and large static networks Fireworks performance is excellent. Besides, for small and medium networks, with mobility around 2 and 10m/s is pretty good. Related to flooding, Fireworks demonstrated to be more efficient than DSR, needing less than the half of control packets to reach the destination node successfully in all cases. Related to the scenarios mobility, we used random walkpoint algorithms to test Fireworks and we found that the distribution of an initial state neighbor nodes follows closely a negative exponential function over time $\Phi(t)$. This stable behavior could be implemented in further proposals.

REFERENCES

- [1] David B. Jhonson, David A. Maltz, *The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks*, IETF MANET Working Group, November 2000.
- [2] C. Perkins, E. Belding-Royer, *Ad Hoc On-Demand Distance Vector (AODV) Routing*, Nokia Research Center, University of Santa Barbara, IETF Internet Draft, July 2003.
- [3] V. Park, S. Corson, *Temporally-Ordered Routing Algorithm (TORA) Version 1*, Flarion Technologies Inc, IETF Internet Draft, July 2001.
- [4] Jinyang Li, John Jannotti, *A Scalable Location Service for Geographic Ad Hoc Routing*, M.I.T. Laboratory for Computer Science, MOBICOM 2000, Boston MA, USA.
- [5] Brad Karp, H.T. Kung, *GPSR: Greedy perimeter stateless routing for wireless networks*, ACM/IEEE MobiComm, August 2000.
- [6] Henri Dubois-Ferrie, Matthias Grossglauser, Martin Vetterli, *Efficient Route Discovery in Mobile Ad Hoc Networks Using Encounter Ages*, School of Computer and Communications Sciences, EPFL, Laussane Switzerland.
- [7] Gregory G. Finn, *Routing and Addressing problems in large metropolitan-scale internetworks*, ISI/RR-87, ISI, March 1987.

- [8] Kevin Fall, Kannan Varadhan, *The NS Manual*, The VINT project, UC Berkeley, LBL, USC/ISI and Xerox PARC, July 16, 2003.
- [9] Xiaolong Li, Dharma P. Agrawal, Qing-An Zeng, *Impact of Mobility on the Performance of Mobile Ad Hoc Networks*, Department of ECECS, University of Cincinnati, 2004.
- [10] Bettstetter Christian, *Mobility Modeling in Wireless Networks: Categorization, Smooth Movement and Border Effects*, Technische Universität München, Institute of Communication Networks, Munich, Germany, Mobile Computing and Communications Review 2003, Volume 5, Number 3, 2003.
- [11] Boudec J.-Y. Le, Vojnovic M., *Perfect Simulation and Stationarity of a Class of Mobility Models*, IEEE Infocom 2005, Miami, FL, 2005.
- [12] Guolong Lin, Guevara Noubir, Rajmohan Rajaraman, *Mobility Models for Ad hoc Network Simulation*, College of Computer & Information Science, Northeastern University, Boston, MA, InfoCom IEEE 2004.
- [13] Yoon Jungkeun, Liu Mingyan, Noble Brian, *Random Waypoint Considered Harmful*, University of Michigan, Michigan IEEE Infocom 2003.
- [14] Orlik Philip, Rappaport Stephen S., *Traffic performance and mobility modeling of cellular communications with mixed platforms and highly variable mobilities*, State University of New York, Stony Brook, New York, IEEE 19997.
- [15] Jabbari Bijan, Zhou Yong, Hillier Frederick, *Random Walk Modeling of Mobility in Wireless Networks*, George Mason University, Fairfax, VA, USA, IEEE 1998.