Centro de Investigación Científica y de Educación Superior de Ensenada, Baja California

Programa de Posgrado en Ciencias en Ciencias de la Computación

---

Algoritmo de enrutamiento en redes ad hoc móviles

Tesis

para cubrir parcialmente los requisitos necesarios para obtener el grado de Maestro en Ciencias

Presenta:

Karime Jocelyn Esquivel Mendoza

Ensenada, Baja California, México

2016

Tesis defendida por

Karime Jocelyn Esquivel Mendoza

y aprobada por el siguiente Comité

Dr. Edgar Leonel Chávez González
Codirector del Comité

Dr. Ubaldo Ruiz López
Codirector del Comité

Dr. Jose Antonio García Macías

Dr. Roberto Conte Galván

Dr. Jesús Favela Vara
Coordinador del Programa de Posgrado en Ciencias de la Computación

Dra. Rufina Hernández Martínez
Directora de Estudios de Posgrado

Resumen de la tesis que presenta Karime Jocelyn Esquivel Mendoza como requisito parcial para la obtención del grado de Maestro en Ciencias en Ciencias de la Computación.

Algoritmo de enrutamiento en redes ad hoc móviles

Resumen aprobado por:

Dr. Edgar Leonel Chávez González	Dr. Ubaldo Ruiz López
Codirector de Tesis	Codirector de Tesis

Una red ad hoc inalámbrica es una red de comunicación descentralizada que no necesita de una infraestructura predefinida para realizar el proceso de reenvío de paquetes, ya que cada dispositivo perteneciente a la red participa en el proceso de entrega del paquete a su dispositivo destino. Las redes ad hoc móviles (MANETs por sus siglas en inglés) son un tipo de red ad hoc donde sus dispositivos se encuentran conectados inalámbricamente y poseen la libertad para desplazarse independientemente hacia cualquier dirección, provocando que la topología de la red y las rutas disponibles entre los dispositivos cambien sin previo aviso. Los principales problemas presentes en MANETs son: 1) El diseño adecuado de estrategias de enrutamiento que provean conectividad aún bajo una topología de red altamente dinámica, 2) La poca confiabilidad del medio inalámbrico y 3) Las capacidades limitadas de energía y procesamiento de los dispositivos.

Existen distintos tipos de protocolos de enrutamiento, uno de ellos es el enrutamiento geográfico, el cual ha emergido recientemente como una técnica eficiente para garantizar rutas de entrega sin la necesidad de inundar toda la red con mensajes de control. En este trabajo de investigación se propone el diseño de un nuevo protocolo de enrutamiento (VGHGR) implementando enrutamiento geográfico para la transmisión de paquetes sobre redes inalámbricas ad hoc móviles. El protocolo tiene como propósito reducir la cantidad de energía requerida por los dispositivos para el reenvío de paquetes y el consumo de memoria para el almacenamiento de información sobre la topología de la red, garantizando la entrega de los paquetes a su destino. Se realiza una evaluación con la ayuda de una herramienta de simulación (NS-3) para observar el desempeño del protocolo sobre una red ad hoc. En los resultados de la evaluación se observan las ventajas de nuestra propuesta en comparación con los protocolos líderes en la literatura.

Palabras Clave: **MANETs, Grafo Virtual, Tablas Hash Distribuidas, Enrutamiento Geográfico, Servicios de Localización**

Abstract of the thesis presented by Karime Jocelyn Esquivel Mendoza as a partial requirement to obtain the Master in Science degree in Computer Science.

Routing algorithm in mobile ad hoc networks

Abstract approved by:

Dr. Edgar Leonel Chávez González	Dr. Ubaldo Ruiz López
Thesis Co-Director	Thesis Co-Director

A wireless multihop ad hoc network is defined as wireless nodes that communicate to each other without using a fixed network infrastructure or centralized administration. In such a network, each node operates not only as a host, but also as a router, forwarding packets to other nodes. An ad hoc mobile network (MANET) is an autonomous system of mobile nodes connected by wireless links. These nodes are free to move at random and organize themselves arbitrarily; thus, the topology of the network can change rapidly and in an unpredictable manner. The main problems in MANETs are: 1) The design of routing strategies that provide connectivity even under a highly dynamic network topology, 2) The wireless medium is significantly less reliable than wired media and 3) The limited capacity of energy and computing power of nodes.

There are a lot of routing protocols that can be used on MANETs, one of them is the geographic routing which has emerged recently as a very efficient way to provide delivery routes without flooding the network with control messages. In this work, we propose the design of a new routing protocol (VGHGR) that implements geographic routing to forward packets in ad hoc mobile networks. The protocol has the purpose of reducing the required energy by the nodes to forward packets and save memory for storing the network topology, guaranteeing the delivery of packets to the destiny. We evaluate our proposal with the help of the simulation tool NS-3 which is used to observe the performance of the protocol in ad hoc networks. The simulation results prove the advantages of our proposal in comparison to state-of-the-art protocols in the literature.

Keywords: **MANETs, Virtual Graph, Distribution Hash Table, Geographic Routing, Location Services**

Dedicatoria

A todos aquellos que estuvieron presentes en esta travesía.

Agradecimientos

A mi familia por su apoyo incondicional, gracias por estar conmigo desde la distancia.

A mis directores de tesis, los Doctores Edgar Chávez y Ubaldo Ruiz. A ambos, gracias por aceptarme como su estudiante y por guiarme durante el desarrollo de esta tesis. Y sobre todo, por contar con tiempo disponible para resolver todas mis dudas.

A los miembros de mi comité de tesis, los Doctores Antonio García y Roberto Conte. Gracias por sus consejos, comentarios y críticas constructivas a este trabajo de tesis.

Al Dr. Rolando Menchaca por sus asesorías en el diseño de protocolos y su estudiante Luis Ricardo Gallego por guiarme en el peligroso mundo del simulador NS3.

A mis compañeros de maestría, gracias por su amistad y buenos recuerdos. Por su ayuda en programación, porque siempre es bueno tener más de una opinión en cuenta.

Al Centro de Investigación Científica y de Educación Superior de Ensenada.

Al Consejo Nacional de Ciencia y Tecnología (CONACyT) por brindarme el apoyo económico para realizar mis estudios de maestría.

Tabla de contenido

Página

Tabla de contenido (continuación)

Tabla de contenido (continuación)

Lista de figuras

Lista de figuras (continuación)

Lista de tablas

Capítulo 1. Introducción

Una red ad hoc inalámbrica es una red de comunicación descentralizada que no necesita de una infraestructura fija (*e.g.* Enrutadores, puntos de acceso) para realizar el proceso de reenvío de paquetes. Esto es posible ya que cada dispositivo perteneciente a la red participa en el proceso de entrega del paquete al destino. Las redes ad hoc móviles (MANETs por sus siglas en inglés) son un tipo de red ad hoc donde los dispositivos se mueven constantemente provocando que la topología de la red cambie frecuentemente.

En el paradigma de las MANETs (Conti y Giordano, 2007a) los dispositivos móviles se auto-organizan para crear una red haciendo uso de sus interfaces de red inalámbrica. Los dispositivos de la red deben cooperar para proveer la funcionalidad que en otros paradigmas es proporcionada por una infraestructura fija. La red ad hoc más simple es una red *peer-to-peer* (ver Figura 1); la cual está formada por un conjunto de dispositivos que actúan simultáneamente como cliente y servidor, permitiendo el intercambio directo de información.

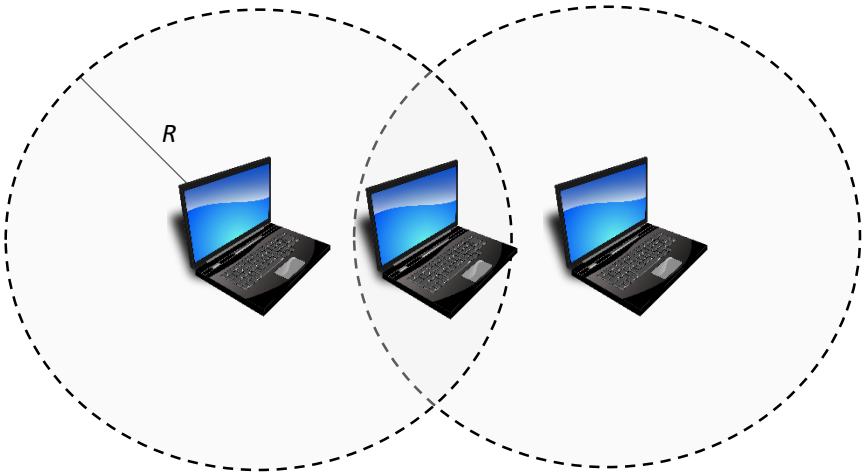


Figura 1: Una red ad hoc simple.

Los dispositivos que no se encuentren conectados directamente entre sí pueden comunicarse enviando mensajes a través de intermediarios (ver Figura 2), por lo que a este tipo de redes se les conoce como *redes inalámbricas ad hoc multi-salto*.

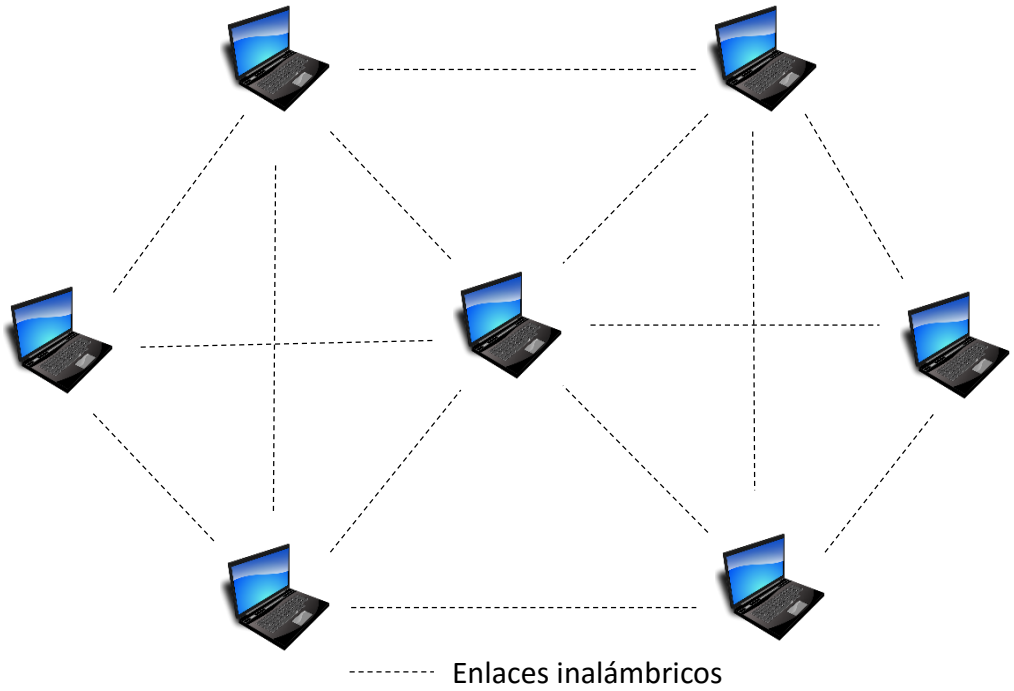


Figura 2: Una red inalámbrica ad hoc multi-salto.

1.1. Enrutamiento en redes ad hoc

Los protocolos de enrutamiento, cuyo objetivo es encontrar el mejor camino para que un dispositivo origen y un dispositivo destino puedan comunicarse, han sido ampliamente utilizados en redes multi-salto. Para descubrir el conjunto de posibles rutas para el reenvío de paquetes, los protocolos utilizan mecanismos de recolección de información que le permiten determinar los puntos intermedios por donde podrá viajar el paquete desde un dispositivo origen a un dispositivo destino (Rey *et al.*, 2014).

Existen distintos tipos de protocolos de enrutamiento, uno de ellos es el enrutamiento geográfico, el cual ha emergido recientemente como una técnica eficiente para garantizar rutas de entrega sin la necesidad de inundar toda la red con mensajes de control (Frey y Stojmenovic, 2006). Por garantizar la entrega nos referimos a la capacidad de transmitir un paquete del origen al destino de manera exitosa, siempre y cuando se cuente con al menos una ruta en la red que conecte a ambos dispositivos.

Los protocolos de enrutamiento geográfico resultan atractivos para las MANETs por las siguientes razones (Liu y Wu, 2006):

- Proporcionan menos sobrecarga para el descubrimiento de rutas comparados con otros tipos de protocolos, lo que se traduce en menos consumo de energía y ancho de banda.
- Son protocolos *sin estado*, en el sentido que los dispositivos no necesitan mantener información por cada destino, ya que toda la información necesaria para realizar las decisiones de reenvío se encuentra disponible en las cabeceras de los paquetes.

Para que los protocolos de enrutamiento puedan enviar un paquete desde un dispositivo origen a un destino, deben seguir las siguientes reglas (Kuhn *et al.*, 2003):

- Cada dispositivo conoce su posición geográfica y la de todo dispositivo que se encuentre conectado directamente a él.
- El dispositivo origen está informado sobre la posición geográfica del destino.
- Excepto por el almacenamiento temporal de los paquetes antes de su reenvío, el dispositivo no mantiene alguna otra información.
- El paquete no debe contener información de control de más de un dispositivo.

Algunos de estos protocolos (Karp y Kung, 2000; Kuhn *et al.*, 2003; Leong *et al.*, 2005) hacen uso de una estrategia voraz la cual consiste en enviar el mensaje al dispositivo vecino que se encuentre mejor localizado con respecto al destino. Para enviar un paquete desde el origen al destino, cada dispositivo intermediario selecciona localmente a su mejor vecino, haciendo uso de una métrica de enrutamiento para su elección (*e.g.* energía, cantidad de saltos, distancia, etc.). Es importante mencionar que las estrategias voraces pueden llegar a *mínimos locales*. Un ejemplo puede ser cuando un dispositivo no cuenta con un vecino que le permita acercarse al destino, esto ocasiona que sea necesario

un mecanismo de recuperación para encontrar un dispositivo que permita continuar con la estrategia voraz. El enrutamiento basado en caras conocido comúnmente como *Face Routing* por su nombre en inglés (Bose *et al.*, 2001) es la estrategia de recuperación más popular en la literatura.

Una red puede ser representada por un grafo no dirigido *G = (V, E)*, cuyos nodos corresponden a los dispositivos de la red, y las aristas a los enlaces entre pares de nodos que pueden comunicarse directamente. La idea principal en el método de Face Routing consiste en aplanar el grafo G, de tal manera que permita trasmitir el mensaje a lo largo de una secuencia de caras adyacentes que proporcionen progreso hacia el nodo destino. En un grafo plano, cualquier ciclo que rodee una región sin ninguna arista dentro de dicha región, se llamara cara. Si el nodo origen y el nodo destino están conectados, el método de Face Routing siempre encontrará una ruta al destino.

Un problema que presenta el método de Face Routing, es que el grafo G nunca es plano, por lo que es necesario el uso de un método local para la extracción de un sub-grafo plano, lo que deriva en la pérdida de muchas aristas de longitud larga (las cuales tienen mayor probabilidad de intersección), eliminando de esta manera la posibilidad de encontrar una ruta más corta al destino.

El uso de un *Grafo Virtual* (Tejeda et. al., 2006a), construido a partir de nodos virtuales y enlaces virtuales que representan a un conjunto de nodos reales y aristas reales, evita la perdida de las aristas del grafo original, por lo que todos los caminos están disponibles siempre. El enrutamiento basado en caras se ejecuta sobre este grafo virtual, pero son los nodos reales los que procesan los paquetes para su reenvío. Es decir, el grafo virtual sirve de guía para que los dispositivos decidan la dirección en la que se envía el mensaje.

1.1.1. Servicios de localización

Los dispositivos que hacen uso de un protocolo de enrutamiento geográfico necesitan conocer su posición geográfica, para esto, existen tres fuentes de las cuales puede obtener dicha información (Liu y Wu, 2006):

- Sistemas de posicionamiento (*e.g.* GPS): Cada dispositivo puede consultar su información geográfica a través de sistemas de posicionamiento. Estos sistemas no siempre ofrecen cobertura completa.
- Servicios de localización: Cada dispositivo reporta su posición periódicamente a servidores de localización que se encuentran en uno o un conjunto de dispositivos. La posición del destino se obtiene a través de estos servidores basados en los reportes proporcionados por los mismos dispositivos.
- Mecanismo de distribución de posición local: Los dispositivos distribuyen periódicamente su posición a sus vecinos, de esta manera cada dispositivo tiene conocimiento local de la topología.

Usualmente, los dispositivos utilizan la información geográfica obtenida a través de sistemas de posicionamiento como GPS. Cuando estos sistemas fallan en proporcionar una cobertura completa (Niculescu y Nath, 2003), es necesario diseñar sistemas que los reemplacen o proporcionen una extensión a sus servicios.

En la Figura 3, se muestra la taxonomía de los servicios de localización propuestos por (Das *et al.*, 2005), donde son divididos por la técnica que aplican para proporcionar la información geográfica a los dispositivos.

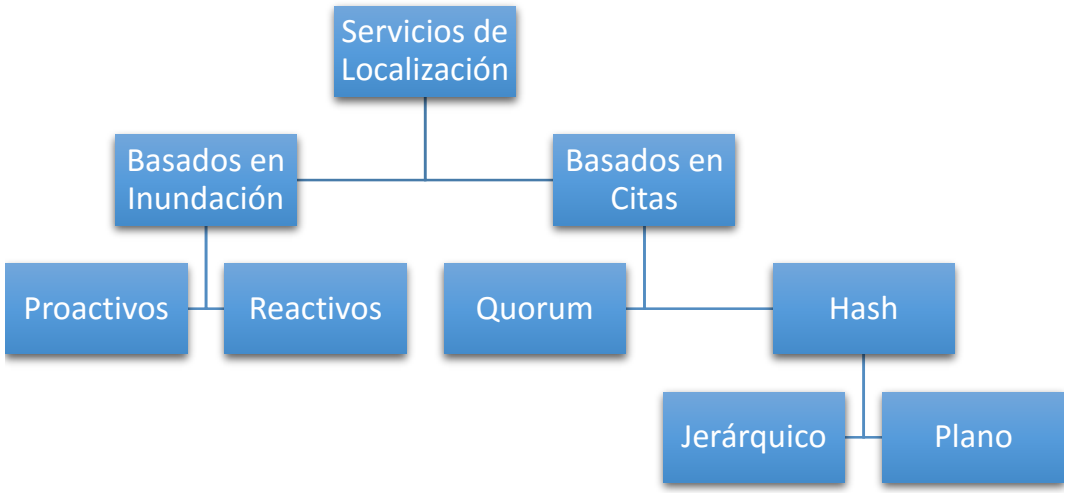


Figura 3: Taxonomía de los servicios de localización.

En los servicios de localización basados en inundación *proactiva* cada dispositivo en la red envía periódicamente su localización a otros dispositivos, y mantiene una tabla de localizaciones con la información más reciente de otros dispositivos. Mientras tanto en los *reactivos*, si un dispositivo no puede encontrar la posición reciente de un destino, inunda la red con una consulta en busca del destino.

En los servicios de localización basados en citas, todos los dispositivos en la red están de acuerdo en guardar los identificadores únicos de cada dispositivo en uno o más dispositivos. Los dispositivos encargados de almacenar los identificadores son llamados servidores de localización. Dos enfoques diferentes han sido propuestos para este tipo de servicios de localización: *quorum* y *hash*.

En el enfoque quorum, cada actualización y consulta de localización de un dispositivo es enviada a un subconjunto diferente de dispositivos disponibles. Los dos subconjuntos están diseñados de manera que su intersección no está vacía, y así la consulta será satisfecha por cualquier dispositivo del subconjunto de consulta que se encuentre en la intersección. En el enfoque hash, los servidores de localización son escogidos a través de funciones hash, ya sea en el espacio de identificadores del dispositivo o en el espacio de localizaciones. Este enfoque a su vez puede ser dividido en *hash jerárquico* y *hash plano*. En el hash jerárquico, el área donde residen los dispositivos es dividida recursiva-

mente de manera jerárquica en pequeñas cuadriculas. Para cada dispositivo, uno o más dispositivos en la cuadricula de cada nivel de la jerarquía son elegidos como su servidor de localización. En el hash plano, una función hash bien conocida es utilizada para almacenar cada uno de los identificadores únicos en una región *home*, la cual consiste en uno o más dispositivos con una localización fija en la red. Todos los dispositivos en la región *home* mantienen la información de localización recibida y pueden mandar una respuesta a cada una de las consultas recibidas. Cada dispositivo en la región *home* es considerado como un servidor de localización.

1.2. Planteamiento del problema

Las MANETs han sido un tema de investigación popular debido al enorme crecimiento de laptops, celulares y redes inalámbricas desde mediados de los 90's. La movilidad de los usuarios y sus dispositivos, afecta la capacidad de los protocolos de enrutamiento de garantizar una ruta de entrega ante los constantes cambios en la topología de la red. Para que los protocolos se adapten a estos cambios, cada cierto intervalo de tiempo los dispositivos deben notificar que se encuentran activos en la red, algo que resulta costoso en términos de energía.

Los protocolos de enrutamiento geográfico antes de reenviar un paquete, necesitan que el dispositivo origen conozca la localización del dispositivo destino. Como tal, un desafío central para este tipo de protocolos es diseñar un servicio de localización eficiente (Das *et al.*, 2005) que le permita seguir la localización de los dispositivos móviles en la red y responder a las consultas de localización recibidas, sin generar un consumo de energía excesivo comparado al uso de sistemas de posicionamiento (Liu *et al.*, 2012; Zhuang *et al.*, 2010). Este servicio debe operar idealmente usando enrutamiento geográfico. En general, un protocolo que implementa un servicio de localización debe ser eficiente, escalable, robusto, y con carga balanceada.

1.3. Objetivos de la investigación

El trabajo de investigación tiene dos grandes objetivos:

- Diseño e implementación de un protocolo de enrutamiento geográfico para el reenvío de paquetes en redes ad hoc móviles, que presente un bajo consumo de energía, porcentaje alto de garantía de entrega de paquetes y poco uso de memoria para almacenamiento de información.
- Diseño e implementación de un servicio de localización basado en tablas hash distribuidas como complemento del protocolo diseñado, para disminuir el consumo de energía por el constante uso del GPS.

1.4. Metodología de la investigación

La metodología seguida en este trabajo se describe a continuación:

- **Revisión de literatura**: Esta primera etapa consistió en una revisión exhaustiva de la literatura relacionada con el enrutamiento en redes ad hoc móviles, con un enfoque especial en los protocolos de enrutamiento geográfico. Para familiarizarse con la estructura en el simulador, se analizó el código de los protocolos del estado del arte elegidos para la parte de evaluación.
- **Diseño del protocolo de enrutamiento geográfico**: En la segunda etapa se procedió la construcción del protocolo propuesto. Se realizó la codificación del grafo virtual, el enrutamiento geográfico y la construcción e implementación de la tabla hash distribuida como servicio de localización.
- **Diseño del escenario**: Se estudiaron evaluaciones previas de los protocolos de el estado del arte para diseñar distintos escenarios de evaluación bajo las mismas condiciones.
- **Evaluación del protocolo propuesto**: Teniendo implementada la red en el simulador, se procedió a evaluar el protocolo propuesto y los protocolos del estado del arte elegidos para su comparación.

1.5. Organización de la tesis

En el Capítulo 2, se presentan detalles y características de las MANETs, además de algunos escenarios donde son desplegadas.

El Capítulo 3, contiene una descripción de los protocolos de enrutamiento tradicional en redes ad hoc móviles, incluyendo sus principales características.

El Capítulo 4, presenta diferentes propuestas de protocolos de enrutamiento geográficos y una explicación más detallada del método voraz y el enrutamiento basado en caras.

En el Capítulo 5, se describe el diseño del protocolo propuesto, el cual abarca la construcción del grafo virtual, el algoritmo del servicio de localización basado en tablas hash distribuidas, el funcionamiento del enrutamiento geográfico con estos nuevos componentes y elementos del protocolo para su correcto funcionamiento.

En el Capítulo 6, se muestran los resultados de la evaluación realizada para el protocolo propuesto.

Las conclusiones de este trabajo de investigación y algunas líneas de investigación a futuro se presentan en el Capítulo 7.

Finalmente, se enlistan las referencias bibliográficas utilizadas para dar sustento al presente trabajo.

Capítulo 2. Redes inalámbricas ad hoc móviles

2.1. Introducción

Una red inalámbrica ad hoc móvil (*MANET*) está formada por un conjunto de dispositivos móviles inalámbricos que conforman una red temporal ante la ausencia de una infraestructura fija y sin una administración centralizada. Los dispositivos son libres de moverse al azar hacia cualquier dirección y poseen la habilidad de auto-organizarse. Por lo tanto, la topología de la red puede cambiar rápidamente y de manera impredecible. Un ejemplo de una red ad hoc aparece cuando un grupo de personas se reúnen para realizar alguna actividad de colaboración computacional y sus dispositivos se comunican mutuamente de manera inalámbrica. Los dispositivos de los usuarios son los encargados de proporcionar la funcionalidad que generalmente es suministrada por la infraestructura de la red (*e.g.* enrutadores, conmutadores, servidores).

Una característica de las redes inalámbricas ad hoc, es que cada dispositivo es capaz de comunicarse directamente con cualquier otro dispositivo que se encuentre en su rango de transmisión. Esta comunicación depende de la potencia de transmisión y de la tasa de transmisión de cada dispositivo. Al incrementar la potencia de transmisión, también aumenta el número de enlaces de comunicación viables, sin embargo, esto produce que se incremente el consumo de energía y la interferencia con otros enlaces (Basagni *et al.*, 2004). Otra característica de los dispositivos móviles usados en redes ad hoc, es el uso de baterías como fuente de energía. Debido a que las baterías poseen duración limitada y existen requisitos adicionales para las operaciones de red dentro de cada dispositivo, la conservación de la energía se considera una prioridad en este tipo de redes (Chiasserini *et al.*, 2002).

Para que un dispositivo se comunique con otro dispositivo que se encuentre fuera de su rango de transmisión, es necesario utilizar dispositivos intermedios (*i.e.* sus vecinos) para retransmitir los mensajes salto a salto. Por lo tanto, es apropiado llamar a este tipo de redes, *redes inalámbricas ad hoc multisalto*. Para llevar a cabo un enrutamiento multisalto, es necesario el desarrollo de protocolos de enrutamiento que se adapten al comportamiento dinámico y limitaciones de una MANET.

En la Sección 2.2, se describen las características y limitaciones de las redes inalámbricas ad hoc móviles. En la Sección 2.3, se tratan distintos tipos de MANETs, por ejemplo, las redes vehiculares ad hoc. En la Sección 2.4, se describen las tecnologías de apoyo para la comunicación en MANET. En la Sección 2.5, se describen las funciones de red realizadas por los dispositivos. Y en la Sección 2.6, se especifica la clasificación de los protocolos de enrutamiento.

2.2. Características y limitaciones

En las redes inalámbricas cada transmisión corresponde a una propagación espacio-temporal de una onda de energía que es recibida por todos los dispositivos que se encuentran en la proximidad, esto ocasiona interferencia entre los enlaces inalámbricos. Por lo tanto, las MANETs heredan los problemas tradicionales de la comunicación inalámbrica y las redes inalámbricas (Chlamtac *et al.*, 2003), los cuales se listan a continuación:

- El canal de comunicación no está protegido contra señales externas.
- El medio inalámbrico es significativamente menos confiable que el medio alámbrico.
- El canal de comunicación tiene propiedades de propagación asimétrica variables en el tiempo.
- Los fenómenos de terminal oculta [1] y terminal expuesta [2] pueden ocurrir.

Considerando los problemas del medio inalámbrico, la naturaleza multisalto y la falta de infraestructura fija, se añaden una serie de características, complejidades y limitaciones de diseño propias de una red ad hoc (Corson *et al.*, 1999; Chiasserini y Rao, 1999), las cuales se describen más adelante.

[1] El problema de terminal oculta se produce cuando dos o más terminales activas (i.e., dispositivos) no pueden comunicarse entre sí a pesar de que sus rangos de transmisión no son disjuntos.

[2] El problema de terminal expuesta ocurre cuando la transmisión entre dos terminales dadas tiene que ser retrasada debido a transmisiones irrelevantes entre otras terminales dentro del rango de transmisión.

Autonomía y falta de estructura fija

Para que una MANET funcione no es necesario establecer una infraestructura fija o administración centralizada. Esto es posible ya que cada dispositivo dentro de la red tiene la capacidad de realizar funciones de red y generar sus propios datos. La administración de la red recae en los dispositivos, lo cual incrementa la dificultad para la detección de fallas en la red.

Enrutamiento multisalto

No existen enrutadores dedicados que se encarguen de procesar los paquetes a reenviar, ya que cada dispositivo en la red actúa como un enrutador y comparte la información entre los dispositivos de la red. Un problema en redes multisalto es que a mayor número de saltos de un paquete la probabilidad de que la información se corrompa aumenta.

Cambios dinámicos en la topología de la red

Dado que los dispositivos pueden moverse arbitrariamente a cualquier parte, la topología de la red puede cambiar frecuentemente y sin previo aviso. Esto provoca que las rutas que se tenían guardadas cambien, se presenten desconexiones en la red y se produzcan perdidas de paquetes.

Variación en los enlaces y las capacidades de los dispositivos

Cada dispositivo está equipado con una o más interfaces de radio que tienen la capacidad de transmitir y recibir paquetes. Además cada dispositivo puede tener una configuración de software y hardware distinta a otros dispositivos, provocando que cada dispositivo tenga capacidades de procesamiento diferentes.

Operación limitada de energía

Los dispositivos móviles tienen una fuente de alimentación limitada ya que funcionan con baterías, las cuales pueden tardar en ser reemplazadas. Lo anterior resulta ser un gran problema para las MANETs, ya que los dispositivos deben actuar como un sistema final y como un enrutador al mismo tiempo, por lo que requieren de energía adicional para el envío de paquetes a otros dispositivos.

Escalabilidad de la red

Actualmente, la mayoría de los algoritmos para la administración de redes están diseñados para trabajar en redes cableadas o pequeñas redes inalámbricas. Varias aplicaciones en redes ad hoc móviles trabajan con miles de dispositivos, por ejemplo, las redes de sensores.

2.3. Tipos de MANETs

Las MANETs permiten construir una red donde no es necesario desplegar una infraestructura fija para implementar las funciones de la misma, además de que no requieren de una autoridad central para la administración y control de la red. Sin embargo, este enfoque carece de realismo, ya que construir una red ad hoc real para un escenario dado es típicamente muy costoso y generalmente difícil de repetir. Por lo tanto, actualmente las MANETs presentan baja explotación y poco interés (Conti y Giordano, 2007a) por parte de la industria y los usuarios. Las MANETs tendrán un valor en el mundo real cuando sean compatibles con un enfoque más pragmático. Una manera de lograrlo es diseñando redes especializadas que soporten un conjunto limitado de aplicaciones. A continuación, se describen redes ad hoc especializadas que han tenido buena aceptación tanto en el mundo real como en el académico.

2.3.1. Redes mesh

A partir de un escenario de aplicaciones bien conocido, es posible reducir la complejidad presente en redes MANET puras (*i.e.* redes sin una infraestructura fija y autoridad central). Las redes *mesh* están construidas sobre una mezcla de dispositivos fijos y dispositivos móviles interconectados a través de enlaces inalámbricos para formar una red inalámbrica ad hoc multisalto (Bruno *et al.*, 2005). Las redes mesh agregan dispositivos llamados *enrutadores mesh*, que se comunican inalámbricamente para formar un *backbone inalámbrico*, el cual cuenta con un número limitado de conexiones alámbricas a Internet.

Los dispositivos móviles pueden estar conectados directamente a cualquier otro dispositivo en la red y pueden obtener conectividad multisalto a Internet si se encuentran cerca de un *enrutador mesh*. En la Figura 4, podemos ver una estructura básica de un red mesh.

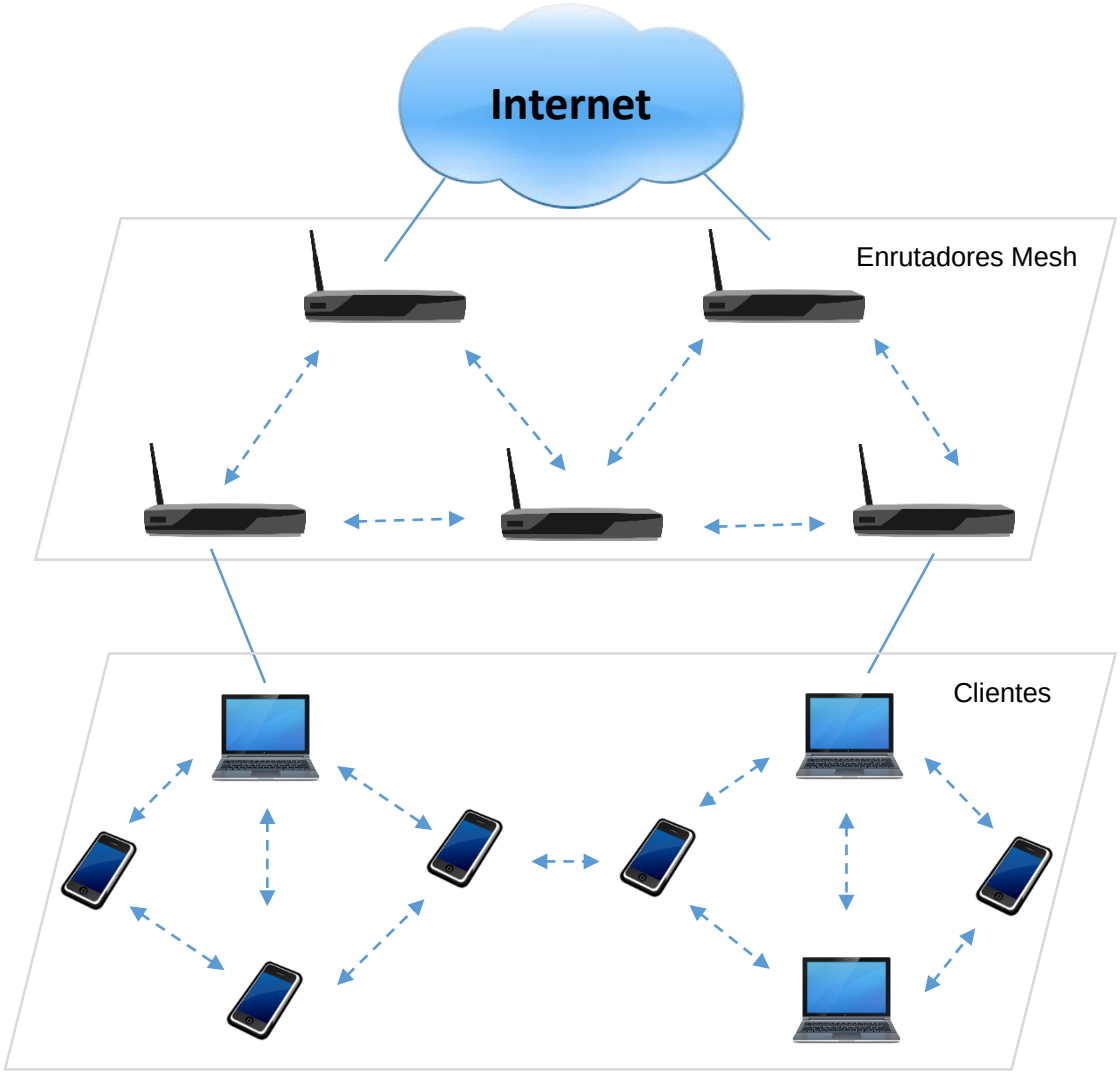


Figura 4: Una red inalámbrica mesh.

El bajo costo, la simplicidad y la confiabilidad que caracteriza a las redes mesh permiten el desarrollo de soluciones interesantes, principalmente en aplicaciones civiles, por ejemplo, acceso público a Internet (Conti y Giordano, 2007b).

2.3.2. Redes oportunistas

Las MANET presentan dificultades ante una baja densidad de dispositivos en la red ya que esto causa fallas en los enlaces de comunicación, provocando temporalmente una desconexión en la red. En este tipo de situaciones, las redes oportunistas (OppNet) pueden trabajar eficientemente con aplicaciones que no requieren la existencia de una ruta.

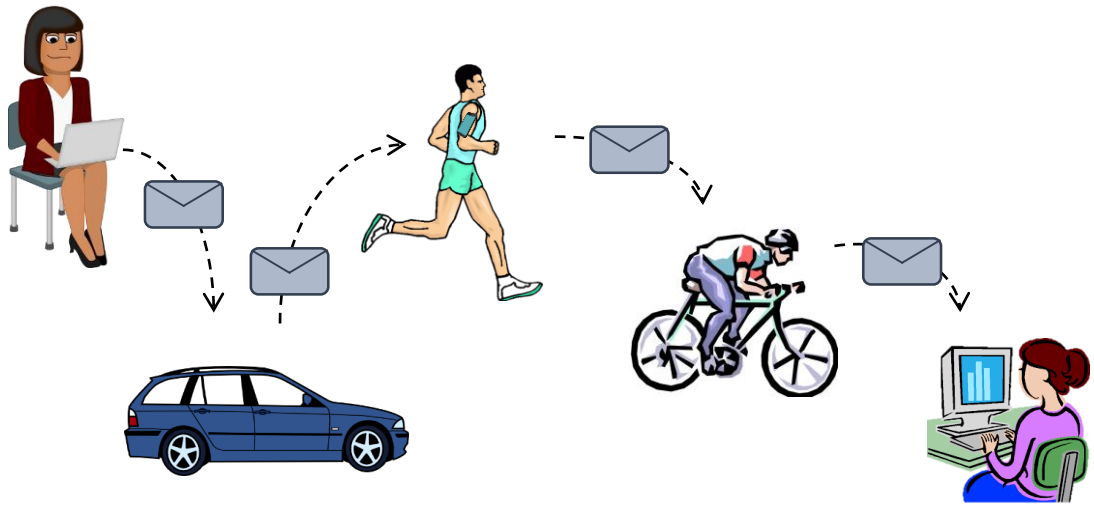


Figura 5: Una red oportunista.

La comunicación en las redes oportunistas es multisalto con dispositivos intermedios actuando como enrutadores para el reenvío de mensajes entre dispositivos. Los dispositivos guardan el mensaje hasta que encuentran una oportunidad de contacto con otro dispositivo móvil para reenviar la información, como se puede apreciar en la Figura 5. A diferencia de las MANETs donde la movilidad de los dispositivos afecta a la entrega de un paquete a su destino, en las OppNet la movilidad crea más oportunidades de comunicación con otros dispositivos permitiendo seguir reenviando el paquete (Conti y Giordano, 2007b).

2.3.3. Redes inalámbricas de sensores

Las redes inalámbricas de sensores (*WSN*) representan una clase especial de redes inalámbricas ad hoc multisalto, desarrolladas para el control y monitoreo a gran escala de eventos y fenómenos (*i.e.* es una red especializada creada a partir de una aplicación en mente) (Conti y Giordano, 2014).

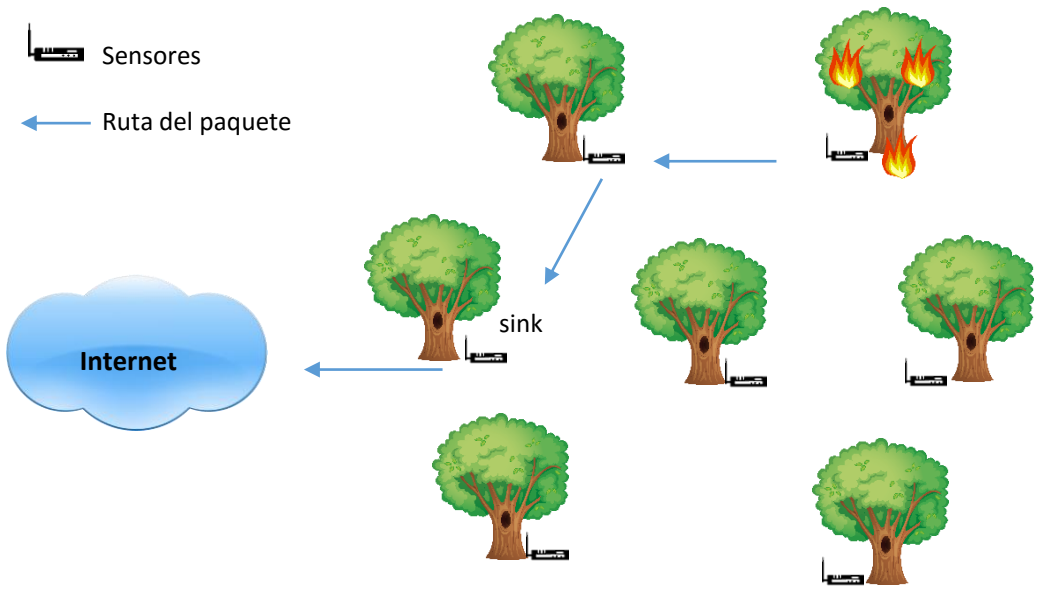


Figura 6: Escenario de aplicación para una red de sensores.

Los sensores inalámbricos pueden comunicarse con otros sensores siguiendo el paradigma multisalto, para transmitir la información recolectada a un sensor *sink* (*i.e.* el sensor encargado de enviar la información a Internet). En la Figura 6, se puede ver una aplicación de estas redes, donde se despliegan sensores en un bosque para detectar el inicio de un incendio rápidamente. Los sensores se encuentran sensando constantemente el ambiente para detectar cambios en la temperatura o presencia de humo, cuando uno detecta tal cambio manda la información a través de sus vecinos, al sensor *sink* que se encuentra en la red.

2.3.4. Redes ad hoc vehicular

Las redes ad hoc vehiculares (*VANET*) son un tipo red inalámbrica ad hoc multisalto donde los vehículos son los dispositivos que conforman la red, por lo que la comunicación entre ellos se da de forma esporádica. La comunicación incluye información del mismo camino y otros vehículos. Estas redes fueron inspiradas por los sistemas de transporte inteligentes (*ITS*) para la reducción de la congestión de tráfico, alto índice de accidentes, etc., (Conti y Giordano, 2014).

Las VANET tienen ventaja sobre las MANET puras, ya que los vehículos no tienen capacidades limitadas de espacio, procesamiento y energía. Desde sus inicios, la investigación de las VANET ha recibido un fuerte apoyo por instituciones gubernamentales, programas internacionales de investigación y de la misma industria automotriz (Conti y Giordano, 2007b). Un ejemplo de una VANET se muestra en la Figura 7.

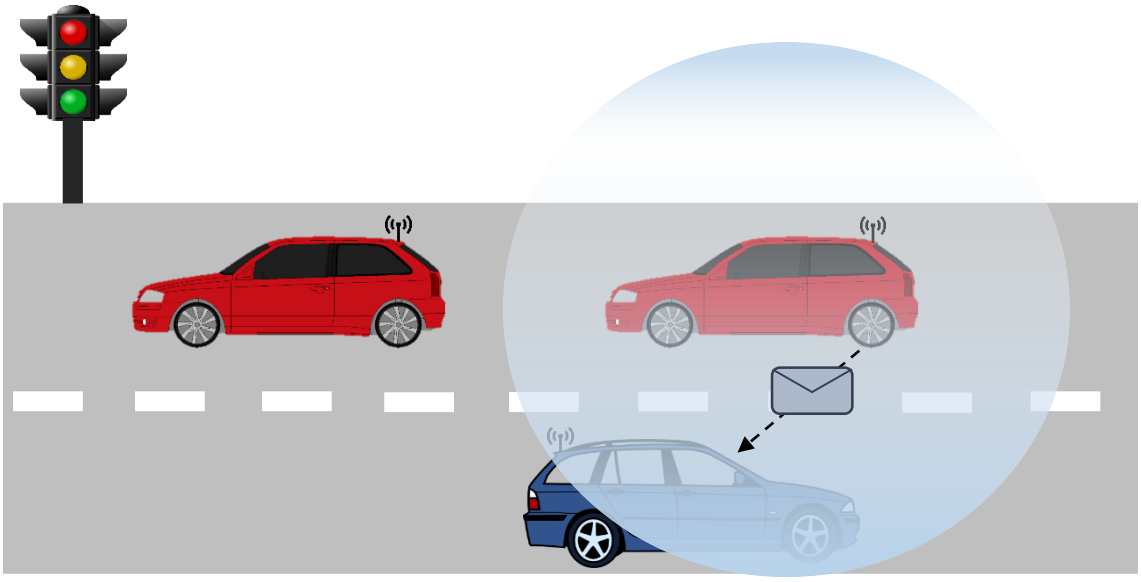


Figura 7: Escenario de aplicación para una red ad hoc vehicular.

Para que los dispositivos puedan comunicarse, es necesario contar con tecnologías de apoyo que garanticen la comunicación directa entre los dispositivos de los usuarios. Por lo tanto, es importante basarse en tecnologías estandarizadas que hacen factible la construcción generalizada de redes interoperables de una manera económica.

2.4. Tecnologías de apoyo para redes ad hoc móviles

Actualmente, han surgido dos estándares principales para las redes ad hoc inalámbricas: el estándar IEEE 802.11 para *WLAN* (Crow *et al.*, 1997) y el *Bluetooth* para comunicaciones inalámbricas de corto alcance (Bisdikian *et al.*, 2001).

Por su simplicidad, el estándar IEEE 802.11 es una plataforma adecuada para implementar una WLAN ad hoc de un solo salto. Las redes multisalto cubren áreas de varios kilómetros cuadrados que potencialmente se pueden construir explotando la tecnología de IEEE 802.11. En una escala menor, tecnologías como Bluetooth pueden ser usadas para construir redes BAN y PAN en modo ad hoc, las cuales son redes que conectan dispositivos de baja potencia utilizados en el cuerpo o son distribuidos en el entorno personal y local del usuario a un radio no mayor de 10 m.

2.4.1. Estándar IEEE 802.11

En 1977, la IEEE adopta el primer estándar para redes inalámbricas de área local, llamado IEEE 802.11, con una velocidad de datos de 2 Mbps. Desde entonces, varios grupos (designados con letras) han sido creados para extender el estándar IEEE 802.11. El 802.11b es creado para operar con una banda de frecuencia de 2.4GHz en WLAN, con una velocidad de transmisión de 11 Mbps y con compatibilidad con versiones anteriores. Y el 802.11a opera con una banda de frecuencia de 5GHz con una velocidad de 54 Mbps (Crow *et al.*, 1997).El estándar IEEE 802.11 tiene dos modos de operación para WLANs: el modo basado en infraestructura y el modo ad hoc. Las interfaces de red pueden trabajar con cualquiera de estos dos modos pero no con ambos simultáneamente (Chlamtac *et al.*, 2003). El modo basado en infraestructura generalmente se utiliza para construir Wi-Fi hotspots, *i.e.*, para proveer acceso inalámbrico a Internet. En el modo ad hoc, cualquier dispositivo que se encuentre en el radio de transmisión de un dispositivo dado, después de una fase de sincronización, pueden iniciar una comunicación con este sin la necesidad de infraestructura fija. Si uno de los dispositivos está conectado a Internet, esto permite que toda la red ad hoc tenga acceso inalámbrico a Internet.

2.4.2. Bluetooth

La tecnología Bluetooh es un estándar de bajo costo, con enlaces de radio a corto alcance entre dispositivos móviles como PC, teléfonos y otros dispositivos portables (Bisdikian *et al.*, 2001). La unidad de Bluetooh está integrada a un microchip, que permite la comunicación ad hoc inalámbrica, de voz y datos entre dispositivos electrónicos portables y/o estáticos.

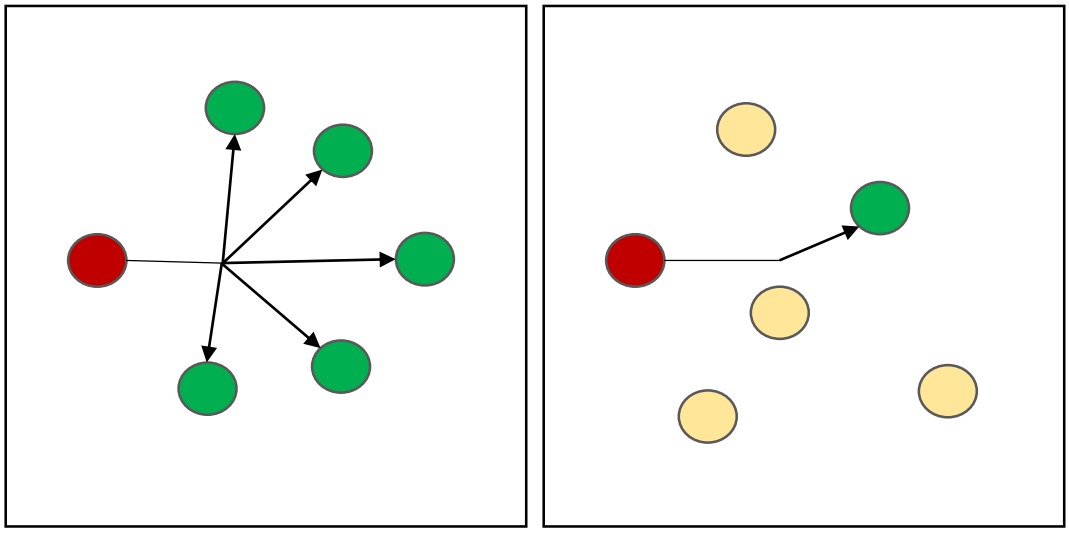
Con un bajo costo, soluciones con bajo consumo y con el soporte de la industria, la tecnología Bluetooh está lista para iniciar la revolución del mercado de la conectividad personal, la cual provee libertad de la conexión por cable.

La tecnología Bluetooh puede manejar un pequeño número de enlaces de comunicación punto a punto de bajo costo y enlaces de comunicación punto a multipunto, sobre una distancia no mayor a 10 metros, con una transmisión de menos de 1 mW. Opera en la banda de frecuencia ISM a 2.4 GHz y aplica el salto de frecuencia para transmitir datos de forma inalámbrica mediante una combinación de la conmutación de circuitos y de paquetes (Chlamtac *et al.*, 2003).

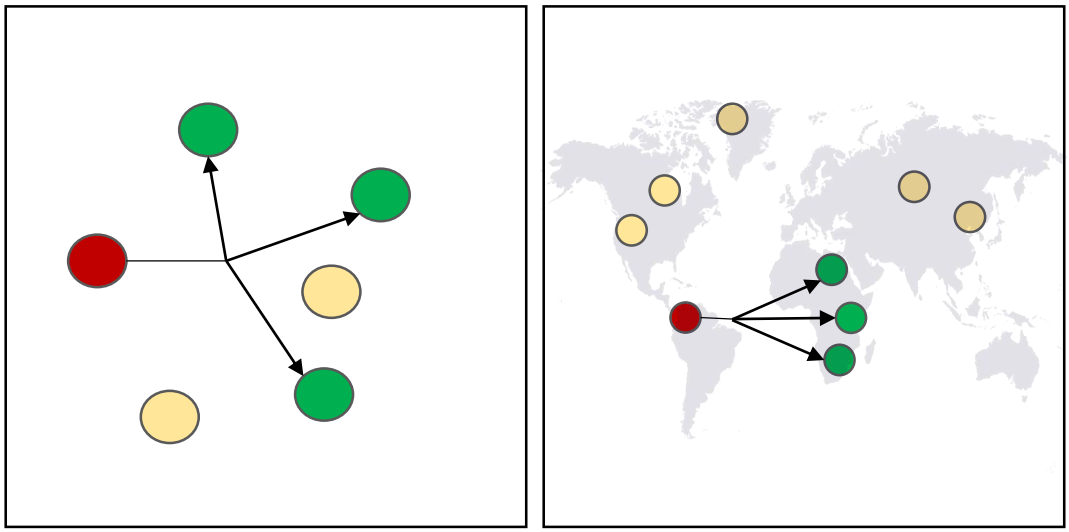
Los protocolos de red utilizan servicios de transmisión de un salto proporcionados por las tecnologías de apoyo para crear servicios de entrega fiables de extremo a extremo, que permitan enviar un paquete desde un emisor a uno o más receptores que se encuentren fuera del alcance de su rango de transmisión.

2.5. Enrutamiento y reenvío de paquetes

Los procesos de enrutamiento y reenvío de paquetes son servicios básicos de red, donde el proceso de enrutamiento consiste en identificar la ruta entre un dispositivo fuente y un dispositivo destino, y el proceso de reenvío tiene como objetivo entregar el paquete a través de la ruta (Conti y Giordano, 2007a). Para transmitir información en la red existen distintos métodos de transmisión los cuales son: broadcast, unicast y multicast, que son tratados a continuación.



(a) Broadcast. (b) Unicast.



(c) Multicast. (d) Geocast.

Figura 8: Modo de transmisión de datos.

2.5.1. Broadcast

Broadcast es el modo básico de operación sobre un canal de comunicación inalámbrico, el cual consiste en enviar un mensaje que será recibido por cada uno de los dispositivos que se encuentren a un salto de quien envió el broadcast (ver Figura 8.a).

La realización de broadcast es útil para dos tipos de escenarios:

- Cuando el dispositivo origen no conoce la dirección del dispositivo destino.
- Cuando un dispositivo necesita enviar la misma información a múltiples destinos.

2.5.2. Unicast

Una transmisión unicast es aquella donde se da una comunicación uno a uno, i.e., un dispositivo origen transmite un paquete de datos a un solo destino. Muchos protocolos de enrutamiento en redes ad hoc aplican este tipo de transmisión de información (ver Figura 8.b)

2.5.3. Multicast

La transmisión multicast es llevada a cabo cuando un dispositivo necesita enviar un mismo mensaje a múltiples destinos simultáneamente, ver Figura 8.c. Un tipo de multicast es el *geocast*, el cual se usa para entregar paquetes a un grupo de dispositivo que se encuentran en un área geográfica específica, ver Figura 8.d. Se puede decir que es un tipo de broadcast restringido, ya que el mensaje es recibido por un dispositivo en la región geográfica y este aplica una transmisión broadcast dentro de la región.

2.6. Clasificación de los protocolos de enrutamiento

Los protocolos de enrutamiento en MANET son típicamente subdivididos en dos categorías principales: enrutamiento proactivo y enrutamiento reactivo (Royer y Toh, 1999). Los protocolos de enrutamiento proactivo intentan mantener de manera consistente y actualizada la información de ruteo para cada dispositivo en la red, propagando de manera proactiva actualizaciones de ruta cada cierto intervalo de tiempo. Por otra parte, los protocolos de enrutamiento reactivo, establecen rutas a un destino solamente cuando es necesario establecer una comunicación. El dispositivo origen mediante un proceso de descubrimiento de ruta, usualmente inicia una petición de ruta a los dispositivos vecinos.

2.6.1. Proactivos

La principal característica de estos protocolos, es que mantienen una ruta para cada uno de los dispositivos pertenecientes a la red, esta información se encuentra almacenada en tablas de ruteo, por tal motivo, a los protocolos proactivos se les suele llamar protocolos dirigidos por tablas. Periódicamente los protocolos proactivos envían mensajes a través de la red para descubrir nuevos dispositivos y las rutas hacia ellos.

La ventaja principal que ofrecen estos protocolos, es que los dispositivos pueden obtener información de enrutamiento fácilmente y establecer una sesión de manera sencilla y rápida.

2.6.2. Reactivos

Para reducir la sobrecarga que se genera en la red ante el constante envío de mensajes de control, las rutas entre dos dispositivos son descubiertas solamente cuando son necesarias, por tal motivo, estos protocolos son llamados protocolos bajo demanda. Cada dispositivo mantiene en caché las rutas aprendidas por el dispositivo. El proceso de descubrimiento de ruta es iniciado únicamente cuando un dispositivo origen no tiene una ruta válida al dispositivo destino en su caché de rutas.

Las rutas en caché se mantienen actualizadas hasta que el destino se vuelva inaccesible o la ruta ya no es requerida. De esta manera se reduce el tamaño de las tablas de enrutamiento al solo mantener un número limitado de destinos en las tablas.

2.6.2.1. Híbridos

Ademas de los protocolos proactivos y reactivos, existe otra clase de protocolos que se pueden identificar como protocolos híbridos, los cuales combinan elementos proactivos y reactivos. Estos protocolos dividen la red en una zona interna y una zona externa, donde de manera simultánea, se lleva a cabo un enrutamiento proactivo en el interior de la zona y un enrutamiento reactivo en el exterior de la zona.

2.7. Resumen

Los problemas y limitaciones descritos en la Sección 2.2 representan grandes retos para el desarrollo de un red inalambrica ad hoc móvil. Los protocolos de enrutamiento diseñados para este tipo de redes deben tomar en cuenta aspectos como la movilidad, los cambios dinámicos en la topología, la poca confiabilidad del medio y las capacidades limitadas de energía y procesamiento de los dispositivos. Por lo tanto, muchas de las funcionalidades presentes en los protocolos de enrutamiento deben ser rediseñadas. En los capítulos 3 y 4 se describen protocolos de enrutamiento diseñados para trabajar en redes ad hoc móviles.

Capítulo 3. Enrutamiento tradicional en redes ad hoc móviles

3.1. Introducción

En este capítulo se revisarán los protocolos tradicionales desarrollados para resolver el problema de enrutamiento en redes inalámbricas ad hoc móviles. Estos protocolos se dividen en dos grupos (Royer y Toh, 1999): proactivos o dirigidos por tablas y reactivos o bajo demanda. También se agrega un tercer grupo el cual es una combinación de los anteriores, conocidos como híbridos.

Los protocolos proactivos son aquellos que guardan información correspondiente a toda la topología de la red y las rutas a cada uno de los destinos posibles. Por lo que periódicamente emiten paquetes para descubrir nuevos dispositivos en la red y rutas hacia ellos, con la finalidad de que en un futuro puedan ser utilizadas. Para guardar esta información y mantenerla actualizada hacen uso de tablas de ruteo. La principal ventaja de estos protocolos es que si un dispositivo desea enviar un paquete a otro dispositivo, puede obtener la información de enrutamiento de manera facíl y rápida. Entre sus desventajas se encuentra que para almacenar dicha información, la cual es proporcional al tamaño de la red, requiere de mucha memoria en cada uno de los dispositivos. Además es necesario enviar una gran cantidad de mensajes de control por toda la red para mantener actualizadas las tablas.

Estos protocolos suelen ser utilizados en redes de tamaño pequeño y en redes de alta densidad de tráfico, debido a la ventaja que representa el intercambio continuo de información realizada sobre la topología de la red.

En la Sección 3.2 se describe el protocolo OLSR (*Optimized Link State Routing*) y en la Sección 3.3 el protocolo DSDV (*Destination Sequenced Distance Vector*), ambos proactivos.

Los protocolos reactivos se caracterizan por que realizan el descubrimiento de rutas solamente cuando un dispositivo desea comunicarse con un dispositivo destino, haciendo uso del proceso de inundación de mensajes de solicitud de ruta, reduciendo el número de mensajes de control en la red. Cuando una ruta es descubierta se mantiene hasta que el destino sea inaccesible o la ruta ya no sea requerida, reduciendo el tamaño de las tablas de ruteo.

Una desventaja presente en los protocolos reactivos es que si los dispositivos no tienen una ruta disponible para comunicarse con un dispositivo destino esta deberá ser descubierta, agregando un retardo significativo en el envío del primer paquete, además de aumentar la posibilidad de saturación de la red debido al proceso de inundación.

Los protocolos reactivos AODV (*Ad-hoc On Demand Distance Vector*) y DSR (*Dynamic Source Routing*) serán revisados en las Secciones 3.4 y 3.5, respectivamente.

El último tipo de protocolos a analizar, son los protocolos híbridos los cuales son preferidos en redes de gran tamaño donde hay una gran cantidad de dispositivos. La mayoría de estos protocolos aplican el concepto de zonas, donde el enrutamiento dentro de la zona se realiza de manera proactiva y de manera reactiva fuera de ella. En la Sección 3.6 se describe el protocolo ZRP (*Zone Routing Protocol*).

3.2. OLSR

El protocolo *Optimized Link State Routing (OLSR)* propuesto por (Jacquet *et al.*, 2001) es un protocolo de enrutamiento proactivo basado en el estado de los enlaces, diseñado para trabajar de forma completamente distribuida y que no requiere de una entrega ordenada de sus paquetes. Al ser un protocolo proactivo, OLSR mantiene las rutas para todos los destinos de la red por lo que periódicamente envía mensajes de control que le permiten aprender la topología de la red. Para reducir la cantidad de mensajes de control, enviados por broadcast o inundación, introduce el concepto de *Multipoint Relays (MPR)*. Estos son un conjunto de dispositivos seleccionados entre el mismo vecindario a un salto de cada dispositivo, capaces de cubrir a dispositivos que se encuentren a una distancia de un salto como se muestra en la Figura 9.

Los MPRs son los únicos dispositivos capaces de retransmitir los mensajes de control que difunde un dispositivo normal, reduciendo significativamente el número de retransmisiones de mensajes.

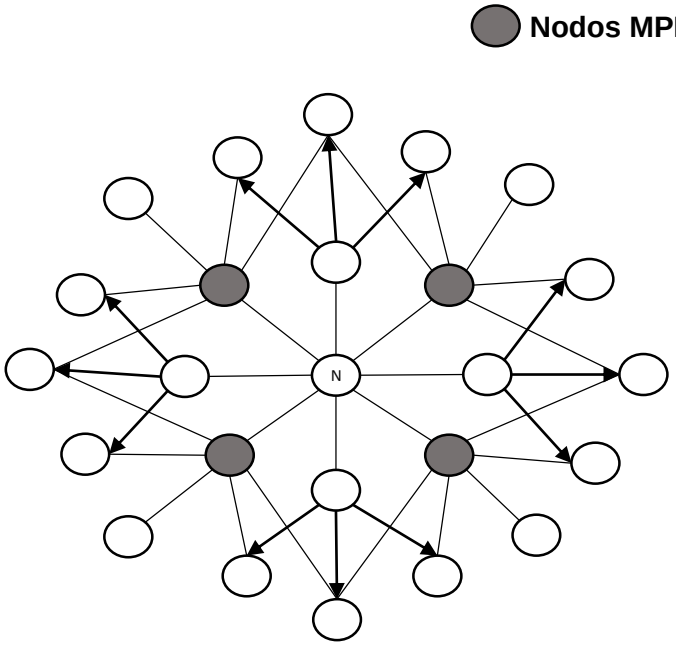


Figura 9: Multipoint relay en OLSR.

Además del periodo de transmisión de mensajes de control, el protocolo no genera tráfico extra ante la falla de un enlace o una nueva adición.

El protocolo OLSR maneja tres tipos de tablas: la tabla de vecinos, la tabla de la topología y la tabla de ruteo. La *tabla de vecinos* es creada a partir de los mensajes HELLO que se transmiten mediante broadcast, estos mensajes contienen la información acerca de los vecinos a un salto, el estado de los enlaces y la información de su vecindario a dos saltos. Mientras que la *tabla de la topología* es alimentada a través de mensajes de *Control de Topología*, el cual contiene la dirección de todos los dispositivos marcados como MPR. Finalmente, la *tabla de ruteo* es creada a partir de la información en las dos tablas anteriores, donde cada entrada contiene la dirección del destino, dirección del siguiente salto y la distancia estimada al destino.

3.3. DSDV

El protocolo *Destination Sequenced Distance Vector (DSDV)* es un protocolo vectorial de distancia salto a salto propuesto por (Perkins y Bhagwat, 1994), que permite el enrutamiento multisalto entre los dispositivos de la red.

Cada dispositivo en la red mantiene una tabla de enrutamiento en donde se listan todos los destinos posibles y el número de saltos necesarios para llegar a ellos. Cada una de estas entradas está ligada a un numero de secuencia originado por el dispositivo destino el cual indica que tan antigua es la ruta, entre mayor sea el número de secuencia más actualizada está la información de la ruta. El uso de este número de secuencia logra que DSDV sea un protocolo libre de ciclos. DSDV debe estar transmitiendo periódicamente información correspondiente a la topología de la red y en caso de presentarse un cambio en la misma, notificarlo inmediatamente a cada uno de los dispositivos para que mantengan actualizadas sus tablas.

Los paquetes de control que DSDV transmite mediante broadcast contienen la información presentada a continuación:

- La dirección del destino.
- El número de saltos requeridos para alcanzar el destino.
- El número de secuencia de la información recibida, originado por el dispositivo destino.

Cuando un dispositivo descubre a través de estos paquetes de control que una ruta en su tabla ya no está disponible, el dispositivo crea un nuevo paquete con un número de secuencia mayor y el número de salto lo marca como ∞, inmediatamente después avisa a sus vecinos del cambio en la topología.

DSDV requiere de muchos paquetes de control para mantener actualizada la información de las rutas. Para reducir la cantidad de información que contienen los paquetes, se definen dos tipos de paquetes listados a continuación:

- **Full Dump**. Este paquete transporta toda la información disponible sobre el enrutamiento en la red.
- **Incremental Dump**. Este paquete solo contiene información que ha cambiado desde el último *full dump* recibido.

Debido al cambio constante en la topología de una red inalámbrica ad hoc móvil, el protocolo DSDV requiere que todos los dispositivos envíen periódicamente toda la información de enrutamiento que hayan aprendido, o que lo hagan inmediatamente cuando se presenta un cambio en la información. La principal desventaja de este protocolo es que la información deberá ser recibida por cada dispositivo de la red, por lo que tomará un tiempo para que la información converja, produciendo un bajo desempeño en redes de alta densidad.

3.4. AODV

Ad-hoc Demand Distance Vector(AODV) propuesto por (Perkins y Royer, 1999) es un protocolo creado sobre la base del protocolo DSDV, a diferencia de este último, AODV es meramente reactivo, es decir, un dispositivo no necesita descubrir o mantener una ruta con otro dispositivo hasta que los dos necesiten comunicarse, en otras palabras, la adquisición de rutas es puramente bajo demanda.

Los objetivos principales de este protocolo son:

- Transmitir los paquetes de descubrimiento sólo cuando sea necesario.
- Distinguir entre el mantenimiento de la conectividad local (detección de vecindario) y el mantenimiento general de la topología.
- Difundir información sobre los cambios en la conectividad local a los dispositivos vecinos que son propensos a necesitar la información.

Una característica fundamental del protocolo AODV es que los dispositivos destinos, antes de proporcionar información de enrutamiento, crean un número de secuencia de destino (concepto tomado del protocolo DSDV), que constituye un instrumento para evaluar cuando se ha actualizado determinada ruta evitando formación de lazos. De esta manera, si un dispositivo debe elegir entre varios caminos hacia un destino, elegirá aquel que cuente con el número de secuencia mayor, el cual corresponde a la información de enrutamiento más reciente.

La principal ventaja de AODV es que no crea tráfico adicional para la comunicación pero cuenta con la limitante que necesita tiempo para establecer una conexión y la comunicación inicial para establecer una ruta es más complicada que otros protocolos.

3.4.1. Descubrimiento de rutas en AODV

Cuando un dispositivo desea enviar un paquete a otro dispositivo, primeramente busca una ruta al destino en su tabla de enrutamiento, si esta existe envía el paquete de datos, de lo contrario, inicia el proceso de descubrimiento de ruta. En este caso, el dispositivo enviará en modo broadcast un mensaje *RREQ* (petición de ruta). El mensaje *RREQ* contiene la dirección de origen, la dirección de destino, el número de secuencia de origen, el identificador de difusión y el número de secuencia más reciente entre el dispositivo origen y el dispositivo destino.

El dispositivo estará en espera de que se reciba un mensaje *RREP* (mensaje de respuesta de ruta). Si no se recibe un mensaje *RREP* después de un tiempo fijo, el dispositivo intentará nuevamente una cantidad finita de veces. Si vuelve a fallar, el dispositivo concluirá que no existe ruta a ese destino.

Cuando un dispositivo recibe un mensaje *RREQ* para un destino *t*, una de las siguientes acciones podrá ocurrir:

- El dispositivo que recibió el mensaje *RREQ* no cuenta con información sobre el destino, por lo que reenvía el mensaje a sus dispositivos vecinos a través de broadcast. También guarda de manera temporal la ruta de regreso al dispositivo destino.
- El dispositivo que recibe el mensaje es el dispositivo destino o tiene información de la ruta hacia el destino *t*. En cualquier caso, manda un mensaje *RREP* de regreso al dispositivo origen a través de la ruta por la cual el dispositivo recibio el primer mensaje *RREQ*. Cuando el origen recibe el mensaje *RREP*, una ruta al destino *t* se crea y podrá ser usada.

Todos los dispositivos mantienen una tabla de enrutamiento con la información de cada destino de interés, creada a partir de los mensajes *RREQ* recibidos y se mantiene activa si es usada por cualquier dispositivo vecino activo. Cada entrada en dicha tabla contiene la siguiente información:

- Dirección del dispositivo destino.
- Dirección del siguiente salto.
- Número de saltos para llegar al destino.
- Número de secuencia del destino.
- Vecinos activos para esa ruta.
- Tiempo de expiración para la ruta.

3.4.2. Mantenimiento de rutas en AODV

El movimiento de los dispositivos no afecta a la ruta activa de enrutamiento a un dispositivo destino. Si el que se mueve es el dispositivo origen de una ruta activa, se puede reiniciar el proceso de descubrimiento de ruta para establecer una nueva ruta al destino. Cuando es el dispositivo destino o alguno de los dispositivos intermedios se mueve, se envía un mensaje *RREP* especial.

Periódicamente, los dispositivos de la red que forman parte de una ruta activa envían mensajes *HELLO (i.e. un mensaje RREP especial)* a sus dispositivos vecinos. La no recepción de estos mensajes se interpretará como pérdida de la conexión con el dispositivo que dejó de enviar estos mensajes. El dispositivo que detectó esta desconexión manda un mensaje *RERR* (mensaje de error) en broadcast, todo dispositivo que reciba este mensaje cancelará las rutas que pasan por el dispositivo que se ha vuelto inaccesible.

Cuando un dispositivo origen recibe un mensaje *RERR* puede reiniciar un proceso de descubrimiento de ruta, si la ruta aún es necesaria. Para determinarlo, el dispositivo debe revisar si la ruta ha sido usada recientemente. Si decide reconstruir la ruta al dispositivo destino, entonces envía un mensaje *RREQ* con un número de secuencia de destino mayor al previamente conocido para asegurar que se construirá una nueva ruta viable.

3.5. DSR

El protocolo *Dynamic Source Routing (DSR)* de (Johnson y Maltz, 1996) es otro protocolo de enrutamiento reactivo para redes inalámbricas ad hoc móviles. La diferencia que presenta con respecto al protocolo AODV, es que el dispositivo origen conoce la ruta salto a salto que deberá ser tomada durante el proceso de transmisión del paquete desde el origen al destino. En la memoria caché de cada dispositivo, almacena información de enrutamiento, es decir, es una caché de rutas la cual contiene conocimiento sobre la topología actual de la red.

Cuando un dispositivo desea enviar un paquete, consulta su caché de rutas. Si encuentra una ruta, la guarda en la cabecera del paquete a enviar y los dispositivos intermediarios a lo largo del proceso de enrutamiento solo tienen que consultar la ruta en la cabecera para enviar el paquete al siguiente vecino indicado en la ruta. Si no cuenta con una ruta, inicia un proceso de descubrimiento de ruta.

3.5.1. Descubrimiento de rutas en DSDV

Durante este proceso, el dispositivo origen consulta su caché de rutas en busca de un trayecto al dispositivo destino; si existe, envía el paquete de datos al dispositivo destino, de lo contrario, inicia un proceso de descubrimiento de ruta enviando en broadcast un mensaje de *petición de ruta*. Este paquete contiene un identificador único escogido por el dispositivo origen con la finalidad de detectar peticiones de ruta duplicadas, además, contiene un *registro de ruta* el cual va almacenando los saltos que le toma al paquete llegar al destino.

Cuando un dispositivo recibe un paquete de petición de ruta, es procesado de acuerdo a los siguientes pasos:

1. Si el identificador de petición de ruta se encuentra en la lista de paquetes recientemente recibidos, se descarta y no se procesa.
2. Si la dirección del dispositivo actual aparece ya en la ruta de la cabecera del paquete, se descarta el paquete y no se procesa.
3. Si el dispositivo a quien va dirigido el paquete es el dispositivo actual, manda un paquete de *respuesta de ruta* a quien inició la petición
4. De lo contrario, añade su propia dirección al *registro de ruta* en el paquete y lo reenvía.

3.5.2. Mantenimiento de rutas en DSDV

El mantenimiento de rutas se utiliza solo cuando el dispositivo envía realmente paquetes al destino, es decir, mientras los dispositivos están mandando paquetes a través de la ruta encontrada. Un dispositivo es capaz de detectar (mientras lee la ruta en la cabecera de un paquete recibido) si la topología de la red ha cambiado. Si detecta que un enlace está roto, crea un paquete de error para notificar a sus vecinos que la ruta ha fallado, entonces los dispositivos que reciban este paquete eliminarán esta ruta de su memoria. Si el dispositivo que detectó la falla aún desea enviar un paquete al dispositivo destino, puede buscar otra ruta en su caché. Si no cuenta con una ruta, puede iniciar un nuevo proceso de descubrimiento de ruta.

La desventaja que presenta el proceso de mantenimiento de ruta, es que no repara localmente un enlace dañado, esto se debe a que el protocolo DSR no envía periódicamente paquetes para actualizar la información sobre los dispositivos vecinos.

3.6. ZRP

Zone Routing Protocol (ZRP) de (Haas *et al.*, 2002) está basado en el concepto de zonas. Una zona de enrutamiento está definida individualmente por cada dispositivo, donde las zonas de los dispositivos vecinos pueden traslaparse y no ocasionar problemas. Las zonas tienen un radio p el cual es expresado en saltos. Un dispositivo se encuentra distanciado en p saltos y no por una distancia física como se muestra en la Figura 10.

Los dispositivos de una zona están divididos en dispositivos periféricos y dispositivos interiores, donde los primeros son aquellos que se encuentran a una distancia mínima exactamente igual al radio de la zona respecto al dispositivo central. Los dispositivos interiores son aquellos donde la distancia mínima al dispositivo central es menor que el radio de la zona.

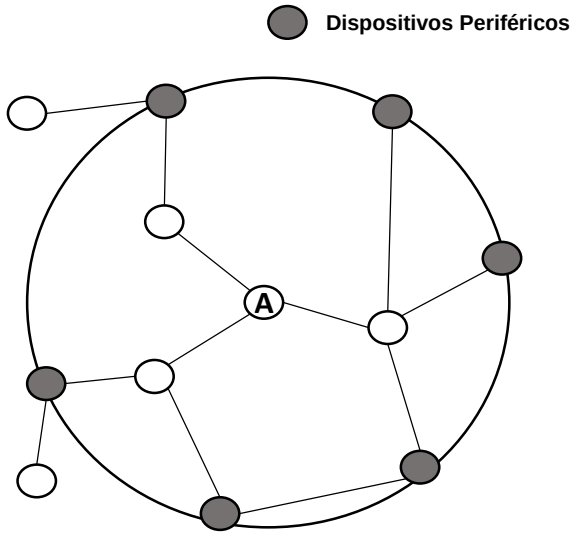


Figura 10: Zona de enrutamiento del dispositivo A con p = 2.

ZRP usa dos mecanismos para el enrutamiento, *Intra-zone Routing Protocol (IARP)* es un algoritmo proactivo el cual mantiene la información de enrutamiento para los dispositivos que están dentro de la zona de un dispositivo. *Inter-Zone Routing Protocol (IERP)* es un algoritmo reactivo el cual ofrece los servicios de descubrimiento y mantenimiento de ruta basado en el monitoreo realizado por IARP. Para detectar si hay un nuevo dispositivo vecino o un enlace fallido, hace uso de mensajes *HELLO* enviados cada cierto intervalo de tiempo para que los dispositivos mantengan actualizado su vecindario.

3.7. Resumen

En este capítulo se presentaron las características principales, además de las ventajas y desventajas de los protocolos proactivos y reactivos para el enrutamiento en redes inalámbricas ad hoc móviles. De igual forma, se describe como los protocolos híbridos buscan mitigar los problemas de los protocolos proactivos y reactivos mediante una combinación de ambos. En el próximo capítulo se hablará de los protocolos basados en posición o de enrutamiento geográfico, los cuales no requieren mantener tablas de enrutamiento o inundar la red con mensajes de control.

Capítulo 4. Enrutamiento geográfico en redes ad hoc móviles

4.1. Introducción

Los protocolos basados en posición o de enrutamiento geográfico hacen uso de la información geográfica (*i.e.* coordenadas) de los dispositivos en la red para la elección del siguiente salto en la ruta hacia el destino. Esta clase de protocolos requieren que cada dispositivo en la red determine sus propias coordenadas y la del dispositivo con el cual desea comunicarse, esta información puede obtenerse a través de cualquier servicio de localización (*e.g. GPS*). Las decisiones para la elección del siguiente salto pueden hacerse utilizando la posición geográfica del dispositivo actual y del dispositivo destino.

La ventaja principal que presentan los protocolos basados en posición es que no requieren mandar mensajes de control para mantener tablas de enrutamiento o inundar la red para descubrir rutas. Los dispositivos en la red envían periódicamente una señal la cual contiene el identificador del dispositivo y sus coordenadas geográficas. Los dispositivos que reciben esta señal almacenan la información adjunta en su tabla de conectividad. Utilizando esta información los dispositivos pueden realizar localmente una elección óptima del siguiente salto.

Algunos de estos protocolos hacen uso de una estrategia voraz y una estrategia de recuperación para llevar acabo el reenvío de paquetes. Ambos métodos son explicados a continuación.

4.2. Método voraz

El método voraz consiste en elegir localmente, para cada dispositivo que interviene en la transmisión del paquete, el dispositivo vecino que presente un mayor avance con respecto a la posición geográfica del dispositivo destino. Una vez que se realiza dicha elección, el paquete es enviado al dispositivo elegido.

En la Figura 11, se muestra un ejemplo de este método. El dispositivo origen *s* desea enviar un paquete al dispositivo destino *t* pero este se encuentra fuera de su rango de transmisión. El dispositivo *s* hace uso de sus dispositivos vecinos para poder enviar el paquete al dispositivo *t*. El dispositivo *v* se encuentra más cerca a la posición del dispositivo *t* que cualquier otro vecino del dispositivo *s*, por lo tanto, el mensaje es enviado al dispositivo *v*.

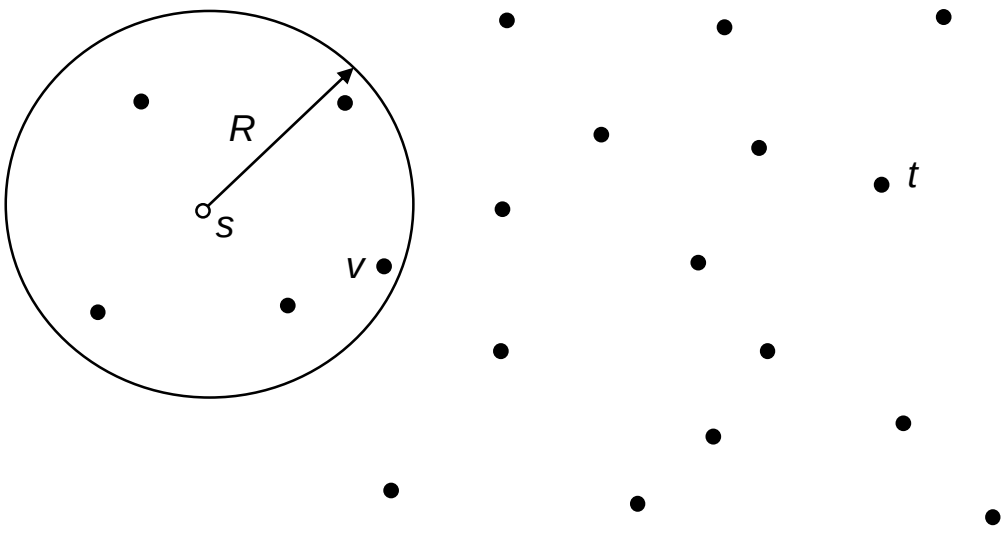


Figura 11: Elección del dispositivo *v* con el método voraz.

Dado que el método voraz tiene la característica de que solo requiere conocer la información de sus dispositivos vecinos inmediatos, se puede ver que por sí mismo no es capaz garantizar la entrega de paquetes.

Existen topologías de red, tal como se muestra en la Figura 12, donde un paquete puede ser enviado a un dispositivo a partir del cual no se puede hacer progreso hacia la meta, esto se conoce como *mínimo local*. En estos casos, se debe aplicar una estrategia de recuperación, siendo el enrutamiento basado en caras el más utilizado.

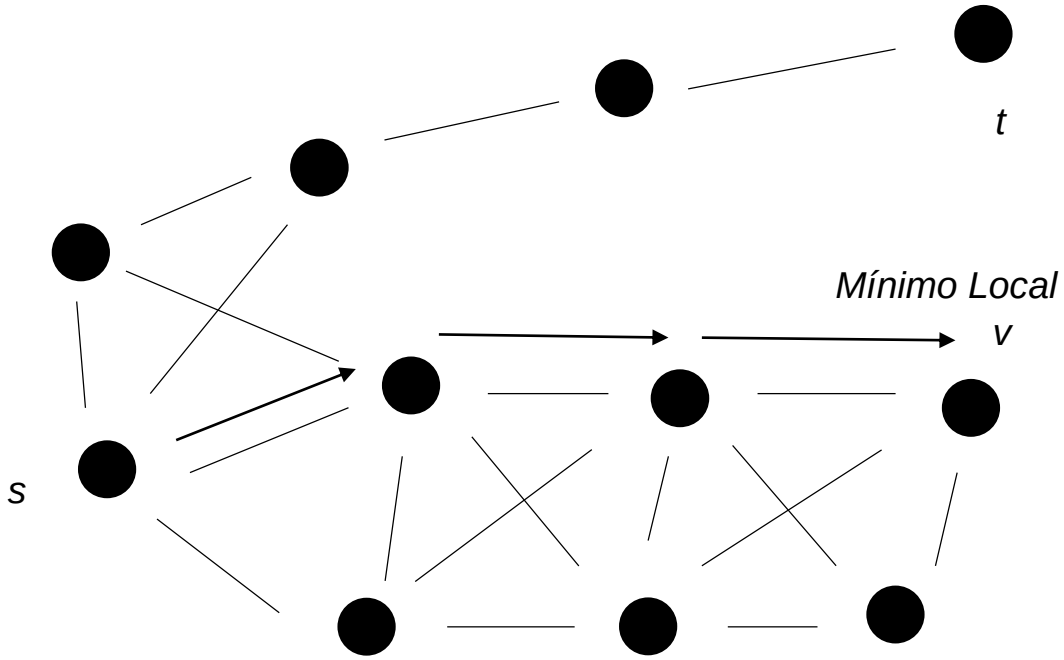


Figura 12: Falla del método voraz para encontrar una ruta de *s* a *t* al llegar al dispositivo *v*.

4.3. Enrutamiento basado en caras

El enrutamiento basado en caras es el método de recuperación más utilizado entre los protocolos geográficos que usan un método voraz. El primer paso consiste en la extracción de un subgrafo plano (el cual no contiene aristas de cruce entre sí) a partir de un *Grafo de Unidad de Disco* (UDG por sus siglas en inglés), el cual es utilizado para representar la topología de la red. Un UDG es un grafo donde una arista [u,v] entre dos dispositivos u y v existe sí y solo sí la $dist(u,v) \leq r$, donde r es el radio de transmisión de los dispositivos.

En un grafo plano, cualquier ciclo que rodee una región que no contenga una arista dentro de ella, se llamara cara. Durante el proceso de enrutamiento basado en caras, los paquetes son enviados progresivamente a través de dispositivos intermedios que conforman un conjunto de caras adyacentes hacia el dispositivo destino haciendo uso de la conocida *regla de la mano derecha* (Bondy y Murty, 1976). Esta regla hace analogía a que es posible recorrer toda pared de un laberinto si se mantiene la mano derecha sobre las paredes mientras se camina hacia delante.

A continuación, se describen dos algoritmos de enrutamiento basado en caras, el primero propuestos por (Kranakis *et al.*, 1999) y el segundo por (Bose *et al.*, 2001).

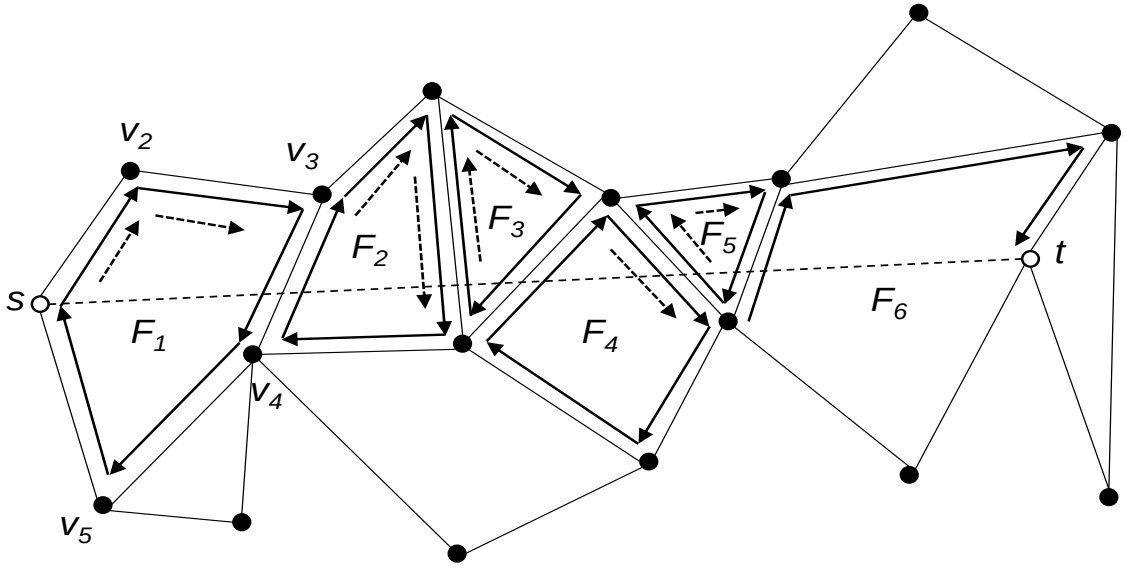


Figura 13: El algoritmo de caras-1.

El funcionamiento del algoritmo propuesto por (Kranakis *et al.*, 1999), denotado como *caras-1*, se muestra en la Figura 13. El primer paso consiste en trazar una línea entre los dispositivos origen y destino, la cual indica las caras adyacentes a visitar. Sean *s* el dispositivo origen y *t* el dispositivo destino del paquete. Sea F_1 una cara que contiene al dispositivo *s* como parte de sus vértices. El algoritmo recorre todas las aristas de la cara F_1, pasando por los dispositivos v_2,v_3, v_4 y v_5 hasta regresar al dispositivo *s*.

Una vez terminado el recorrido, se selecciona de entre los dispositivos visitados el más cercano al destino t, y se reenvía el paquete al dispositivo seleccionado, en este caso suponemos que es v_3.

Se pasa a la siguiente cara adyacente de acuerdo a la línea trazada, en este caso F_2 y se repite el proceso anterior iniciando en el dispositivo v_3. En este ejemplo, se vistan de manera sucesiva las caras F_2, F_3, F_4, F_5 y F_6, siendo esta última la que contiene al destino *t*, poniendo fin al proceso. Se puede observar que en cada repetición el paquete es enviado alrededor del perímetro de la cara y una vez que se ha recorrido completamente deberá ser enviado al dispositivo elegido.

(Bose *et al.*, 2001) proponen una mejora al primer algoritmo, denominado *caras-2*. Si durante el recorrido de una cara detecta que en un vértice ha dejado de acercarse al destino, toma el último vértice visitado como el siguiente salto y lo utiliza para enviar el mensaje. El algoritmo termina en una cantidad menor de saltos que *caras-1* al no ser necesario recorrer toda la cara. En la Figura 14, se puede observar el funcionamiento del algoritmo *caras-2*.

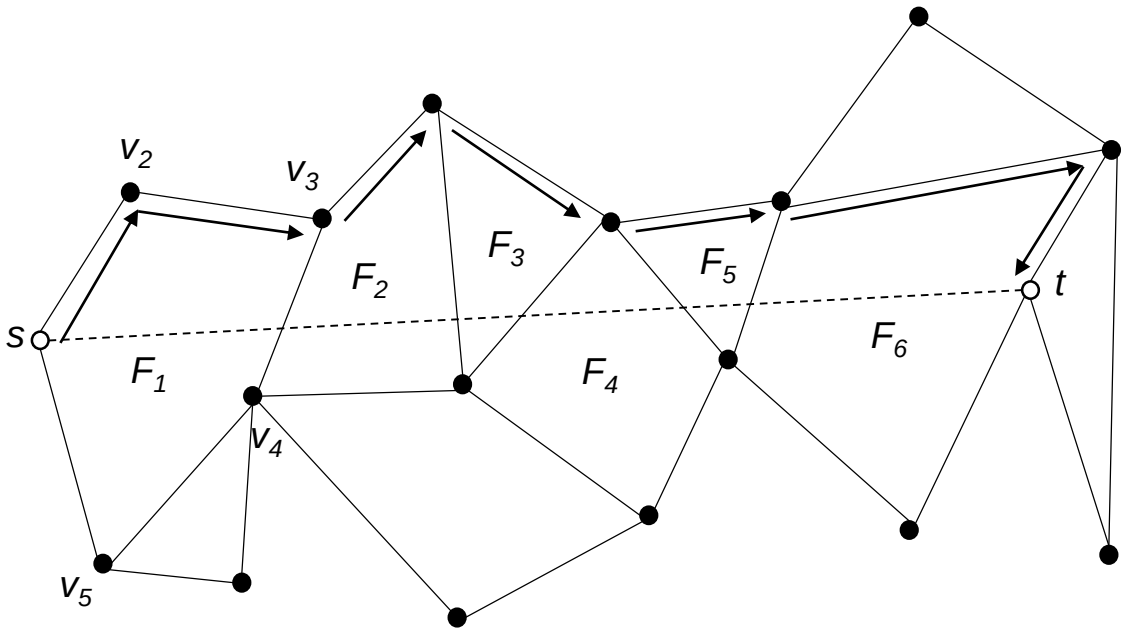


Figura 14: El algoritmo de caras-2.

4.4. GFG

El primer algoritmo que propone una combinación del método voraz y el enrutamiento basado en caras es *Greedy-Face-Greedy (GFG)* de (Bose *et al.*, 2001). Este algoritmo comienza aplicando el método voraz para reenviar el paquete, buscando en cada salto el dispositivo vecino que le proporcione mayor avance hacia el dispositivo destino. Al detectar un mínimo local, el proceso de enrutamiento cambia y hace uso del enrutamiento basado en caras. Se realiza el proceso de recorrido de caras en busca de un dispositivo que le permita acercase al destino. Una vez encontrado tal dispositivo, se reanuda el proceso con el método voraz.

4.5. GPSR

El protocolo *Greedy Perimeter Stateless Routing(GPSR)* de (Karp y Kung, 2000), utiliza una combinación similar al algoritmo GFG de (Bose *et al.*, 2001). Hace uso del método voraz, sin embargo, para el método de recuperación usa una estrategia diferente al enrutamiento basado en caras que será explicada más adelante.

Para actualizar las posiciones de los dispositivos en la red, GPSR hace uso de *beacons (i.e. señales únicas que permiten localizar y detectar otros dispositivos)* cada intervalo de tiempo para mantener actualizas las tablas de posiciones de cada uno de los dispositivos en la red. Además, hace uso de un *piggyback* el cual consiste en enviar la posición del dispositivo en cada uno de los paquetes que reenvía. Cuando un dispositivo recibe un paquete puede retrasar el próximo beacon a enviar. Esta característica es una de las razones por las cuales sigue siendo uno de los protocolos más representativos y utilizados en la literatura.

4.5.1. Enrutamiento perímetro progresivo

El método utilizado por GPSR para recuperarse ante un mínimo local es el enrutamiento de perímetro progresivo, el cual se muestra en la Figura 15. El dispositivo *s* se encuentra más cerca a *t* que sus vecinos *a* y *d*. A pesar de que existen dos rutas al destino *t*, (s $\rightarrow$ a $\rightarrow$ b $\rightarrow$ t) y (s $\rightarrow$ d $\rightarrow$ c $\rightarrow$ t), *s* no elegirá a ninguno de sus vecinos para reenviar el paquete de manera voraz. Por lo que hace uso del enrutamiento de perímetro progresivo para salir de esta situación.

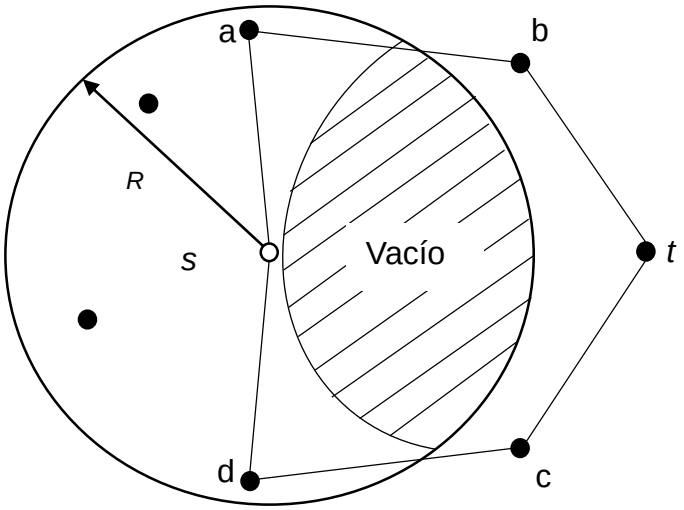


Figura 15: GPSR: Enrutamiento alrededor del vacío.

La región sombreada que no cuenta con ningún dispositivo es nombrada *vacío*. El dispositivo *s* intentará reenviar el paquete alrededor del vacío presentado. El algoritmo hará uso de la regla de la mano derecha para ayudar a reenviar el paquete alrededor del vacío. La regla es aplicada de la siguiente manera: cuando el paquete llegue al dispositivo s, el siguiente salto será el próximo dispositivo que aparece en sentido contrario a las manecillas del reloj, a diferencia del enrutamiento basado en caras donde la regla de la mano derecha recorre la cara en el sentido de las manecillas del reloj.

Aplicando esta regla para reenviar el paquete se obtiene la ruta (s $\rightarrow$ a $\rightarrow$ b $\rightarrow$ t $\rightarrow$ c $\rightarrow$ d $\rightarrow$ s) la cual rodea el vacío. Este recorrido es llamado perímetro y está formado por los enlaces que conectan a los dispositivos de la red que se encuentran en los extremos del área vacía, incluyendo el dispositivo destino.

4.6. GOAFR

Greedy Other Adaptative Face Routing (GOAFR) de (Kuhn *et al.*, 2003) agrega el método voraz al algoritmo *Other Adaptative Face Routing (OAFR)* del mismo autor, el cual es una variación del enrutamiento basado en caras donde la exploración está restringida por una área de búsqueda en forma de elipse como se muestra en la Figura 16.

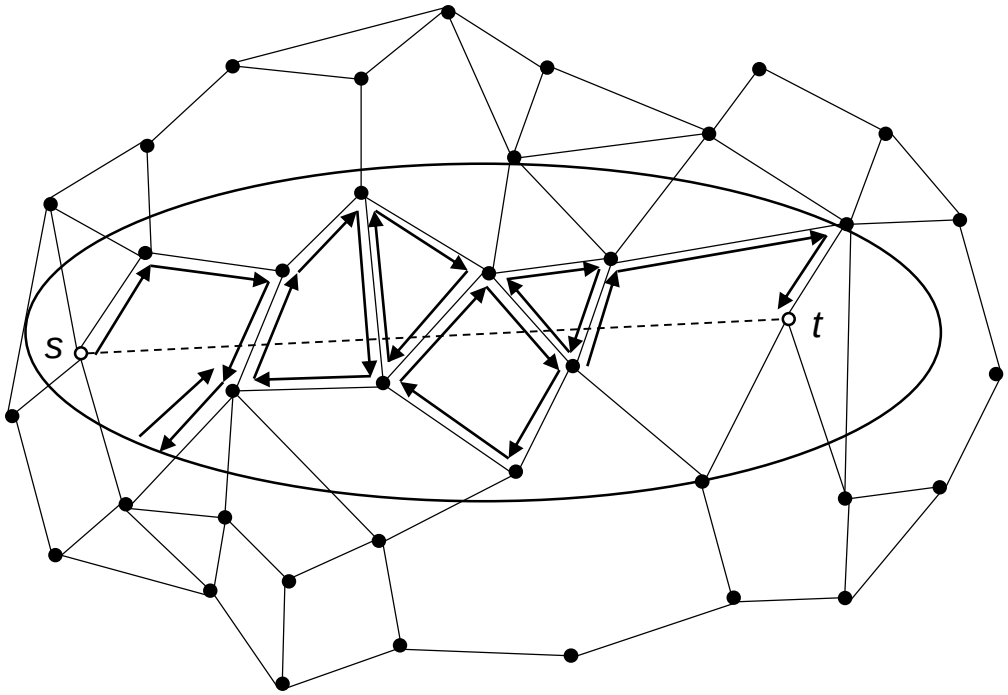


Figura 16: Área de restricción para el enrutamiento basada en caras en GOAFR.

El tamaño de la elipse ε se asigna de acuerdo a una estimación de la longitud de la ruta óptima, de tal manera que se asume que el área de la elipse contiene la ruta óptima del dispositivo origen al dispositivo destino.

El algoritmo GOAFR se basa en dos algoritmos, OBFR que es utilizado para recorrer las caras y OAFR que controla el tamaño de la elipse. Ambos algoritmos son descritos a continuación:

El algoritmo OBFR consiste en los siguientes pasos:

1. A partir de s, se inicia la exploración de la primera cara *F*.
2. Se recorren todos los bordes de la cara, si se alcanza el borde de la elipse se continúa el recorrido hacia el lado opuesto, hasta que se alcance por segunda vez el borde.
3. Terminada la exploración de la cara *F*, se avanza hacia el dispositivo más cercano al destino. Se sigue el proceso con la siguiente cara partiendo del paso 2.

El algoritmo OAFR tiene los siguientes pasos:

1. Inicializar ε a $\varepsilon(2\times \mid \overline{st} \mid)$, donde $\mid \overline{st} \mid$ es la longitud del segmento que une a *s* y *t*.
2. Iniciar OBFR con ε
3. Si el destino no es alcanzable, incrementar el tamaño de ε y regresar al paso 2.

Por lo tanto, el algoritmo implementado por GOAFR queda de la siguiente manera:

1. Inicializar ε a $\varepsilon(2\times \mid \overline{st} \mid)$ y comenzar en *s*
2. Ejecutar el método voraz hasta que se alcance *t* ó se presente un mínimo local n_m. Si el siguiente paso está más allá de ε , incrementar el tamaño de ε . Si se presenta un mínimo local, proceder con el paso 3 iniciando en n_m.
3. Ejecutar OAFR solamente en la primera cara. Incrementar el tamaño de ε si es necesario.
4. Terminar si OAFR encontró a *t*. Si OAFR detecta una desconexión, debe reportarlo a *s* (usando GOAFR). De lo contrario, continuar con el paso 2 sobre el dispositivo más cercano a *t* encontrado por OAFR.

4.7. GPVFR

En el protocolo *Greedy Path Vector Face routing (GPVFR)* de (Leong *et al.*, 2005) para el recorrido de caras se traza la línea st. En la primera cara, se busca el punto q de intersección entre la cara y la línea st. Una vez que se conoce q, se busca el vértice más cercano a él en dicha cara. Una vez que el paquete es enviado a ese vértice, cambias de cara y trazas una nueva línea entre ese vértice y el destino. Se busca nuevamente el punto q de intersección entre la nueva línea trazada, repitiendo el proceso de moverse al vértice más cercano y cambiando de cara.

En la Figura 17, se muestra un ejemplo de este protocolo. Siguiendo la regla de la mano izquierda, el paquete encuentra una intersección en q_1 y reinicia el enrutamiento basado en caras en el dispositivo v_1 considerando la línea v_1t. Siguiendo la regla de la mano izquierda el paquete es enviado al dispositivo v_2 ya que la siguiente intersección es detectada en q_2. Aplicando sucesivamente el algoritmo, el paquete recorre las caras F_1 ,F_2 ,F_3 , F_4 y F_6, hasta llegar al dispositivo destino *t*.

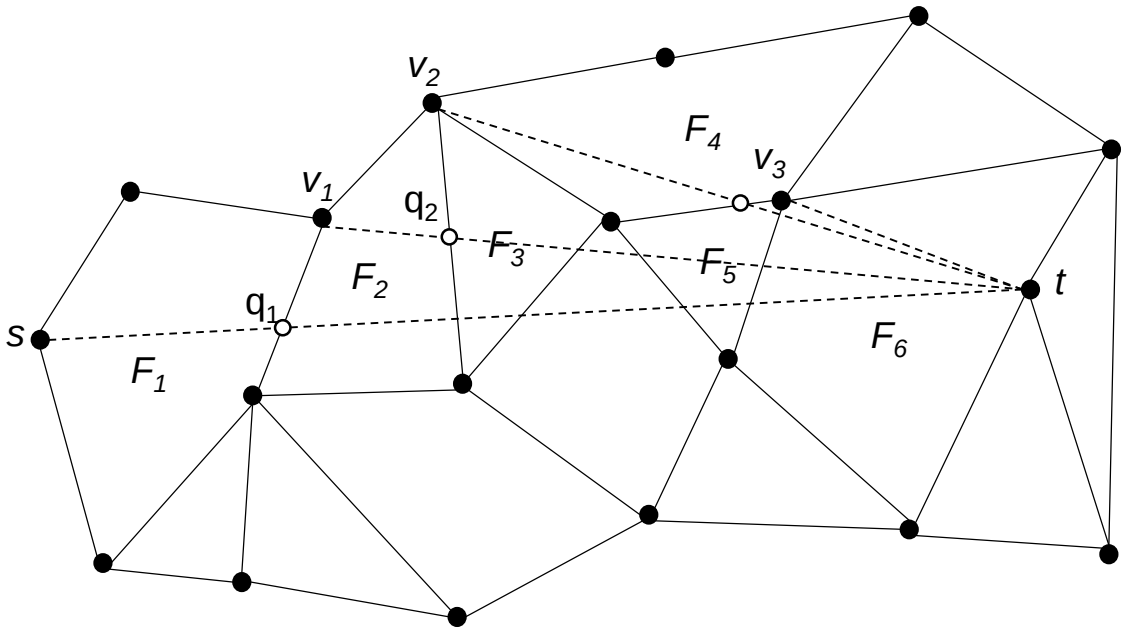


Figura 17: Enrutamiento basado en caras en GPVFR.

4.8. Resumen

En este capítulo, se revisaron los protocolos de enrutamiento geográfico que hacen uso de una estrategia voraz para el reenvío de paquetes y ante presencia de un mínimo local cambian a una estrategia de recuperación basado en caras. Los protocolos de enrutamiento geográfico no requieren establecer rutas y mantenerlas. La decisión del siguiente salto se realiza de manera local y solo requiere la posición del dispositivo destino y la de sus vecinos, por lo que no es necesario almacenar información de enrutamiento. El uso de información geográfica evita búsquedas amplias en la red, ya que tanto los paquetes de control y de datos se envían hacia las coordenadas geográficas del destino. Esta característica hace de los protocolos de enrutamiento geográfico se adapten rápidamente a los cambios de la topología, haciéndolos más escalables que los protocolos mencionados en el capítulo anterior.

Capítulo 5. Diseño de un nuevo protocolo de enrutamiento geográfico para redes ad hoc móviles

5.1. Introducción

En este capítulo, se describe el protocolo de enrutamiento geográfico propuesto sobre redes ad hoc móviles. A dicho protocolo se decidió llamarle *Virtual Graph Hashing for Geographic Routing*, (VGHGR) por sus siglas en inglés.

El protocolo VGHGR fue diseñado específicamente para redes ad hoc móviles puras. Tres de las consideraciones claves para su diseño son:

1) Disminuir el uso del GPS implementando un servicio de localización administrado por los mismos dispositivos que conforman la red. De esta manera, se reduce el consumo de energía por un uso constante del GPS.

2) El servicio de localización no debe consumir más espacio en memoria que las tablas de enrutamiento presentes en los protocolos tratados en el Capítulo 3.

3) VGHGR no es orientado a conexión. De esta manera, se disminuye la información de control necesaria para rearmar los paquetes en cada extremo. Mejorando el aprovechamiento de las capacidades de cada salto de la red sin sobrecargarla.

Para el servicio de localización se hace uso de tablas hash distribuidas (DHT por sus siglas en inglés). Frecuentemente las DHT son utilizadas bajo el paradigma de almacenamiento centrado en datos, tal como Chord (Stoica *et al.*, 2001), CAN (Ratnasamy *et al.*, 2001), Pastry (Rowstron y Druschel, 2001) y Tapestry (Zhao *et al.*, 2001). En estos sistemas, un objeto dato es asociado a una clave y cada dispositivo en el sistema es responsable de almacenar cierto rango de claves. En redes ad hoc móviles, una DHT puede ser utilizada para proveer servicios de localización o almacenamiento de información de los dispositivos. Estas proveen una estructura de localización distribuida para mantener dicha información. Las DHT cuentan con dos métodos principales, llamados *inserción (k,v)* y *consulta(k)*, donde k y v representan la clave y su valor, respectivamente.

5.2. Diseño e implementación de la DHT como servicio de localización

Uno de los aspectos fundamentales de nuestra arquitectura es que utilizamos celdas para agrupar los valores de la DHT, donde todos los dispositivos que se encuentran en la celda almacenan una copia de la información asociada a ella. Cuando un dispositivo se desplaza este cambia constantemente sus coordenadas geográficas, por lo que es necesario estar realizando actualizaciones periódicamente, las que se traducen en una sobrecarga de la red. Para evitar este problema, en nuestra propuesta las actualizaciones tienen lugar solo cuando un dispositivo cambia de celda lo que reduce considerablemente la cantidad de mensajes de control.

Una celda se considera activa cuando cuenta con al menos un dispositivo que almacene la información asociada a ella. Desafortunadamente, debido a la autonomía de los dispositivos no es posible asegurar que todas las celdas se encuentren activas. Para evitar que la información sobre los dispositivos se pierda al ser asociada a una celda inactiva, replicamos la información en diferentes celdas.

5.2.1. Mecanismo básico

En la configuración más básica, la función hash determina la celda donde será almacenada el par(clave, valor) correspondiente a un dispositivo. Dado un par(clave A, valor A), la celda cuyo identificador sea igual al resultado del hash (clave A) será la responsable de almacenar el valor A. Por ejemplo, consideramos el par (10.1.0.1, 20), donde la clave está ligada a la celda 56. En este caso, el valor 20 será almacenado en la celda 56 con la clave 10.1.0.1.

5.2.2. Función hash

Para minimizar el riesgo de pérdida de información por celda inactiva, el protocolo cuenta con k funciones hash. Cada función apunta a una celda responsable, de esta manera si una celda está vacía no afecta al rendimiento del protocolo ya que hay otras celdas que pueden proporcionar/almacenar la información. Cada una de las funciones hash recibe el identificador único del dispositivo (e.g. dirección ip) como clave y tienen el siguiente formato:

$$h(id) = (id \times M_n) \mod C \tag{1}$$

donde C es el conjunto de celdas y M_n un numero primo de Mersenne (Robinson, 1954) aleatorio.

5.2.3. Inicialización y actualización

El proceso de inicialización de un dispositivo en la red consiste en enviar su información geográfica a las celdas asignadas por las k funciones hash. A estas celdas les llamaremos celdas responsables del dispositivo. Cada dispositivo que se encuentre dentro de las celdas responsables almacena una copia del nuevo registro.

El algoritmo del proceso de inicialización se lista a continuación:

1. El dispositivo aplica k funciones hash para saber quiénes son sus celdas responsables.
2. Por cada celda responsable, el dispositivo crea un paquete de registro adjuntando su información geográfica a la cabecera del paquete.
3. Los paquetes son enviados a través de la red a cada una de las celdas responsables aplicando enrutamiento geográfico.

El proceso de actualización se lleva a cabo cuando un dispositivo detecta que ha cambiado de celda. Como la información está ligada a las celdas y no a los dispositivos, cada que un dispositivo cambia de celda debe eliminar todo registro asociado a la celda anterior. El dispositivo procede con la ejecución del siguiente algoritmo:

1. El dispositivo manda un paquete de solicitud a un dispositivo vecino que se encuentre en la celda nueva. Este paquete le permitirá obtener información asociada a su nueva celda.
2. El dispositivo elimina toda información asociada a su celda anterior.
3. El dispositivo notifica a cada una de sus celdas responsables de su nueva posición, aplicando el algoritmo del proceso de inicialización descrito previamente.

5.2.3.1. Propagación de la información

Cualquier dispositivo en cada una de las celdas responsables puede recibir los paquetes de registro enviados a la celda. Cuando esto sucede el dispositivo procede de dos maneras: Primeramente el dispositivo verifica entre sus registros si cuenta con la clave recibida, si encuentra una coincidencia el dispositivo procede a actualizar el registro con la información recibida. Si la clave de la información recibida es nueva, el dispositivo almacena la nueva información. Después de procesar el paquete de registro, el dispositivo envía una copia a cada uno de los dispositivos vecinos que se encuentren en la misma celda responsable. De esta manera, todos los dispositivos en la celda responsable mantienen la misma información y pueden responder a una consulta dirigida a la celda.

5.2.4. Consulta al servicio de localización

Los dispositivos en la red solo conocen el identificador del dispositivo a quien desean enviarle un paquete de datos, por lo que necesitan convertir ese identificador a una posición geográfica para realizar el enrutamiento.

Cuando un dispositivo *s* desea enviarle un paquete a un dispositivo *t* y no son vecinos, el dispositivo *s* debe realizar una consulta al servicio de localización.

Para realizar esta consulta, el dispositivo *s* debe aplicar las k funciones hash que le corresponden al dispositivo *t*. Ingresando el identificador del dispositivo *t*, cada una de las funciones le regresará una celda responsable a donde debe enviar un paquete de consulta.

Cuando un dispositivo recibe un paquete de consulta y se encuentra en una de las k celdas responsables, el dispositivo verifica en sus registros si cuenta con una clave que coincida con el identificador del dispositivo al que se hace referencia en la cabecera del paquete. Si la clave coincide, el dispositivo crea un paquete de respuesta con la información solicitada y lo envía de regreso al dispositivo que originó la consulta.

Una vez que el dispositivo que originó la consulta recibe un paquete de respuesta, este envía el paquete de datos al dispositivo destino. El dispositivo descarta cualquier otro paquete de respuesta una vez que haya enviado el paquete de datos.

Al ser la celda la entidad relevante de la red, los paquetes son enviados aplicando geodifusión (*i.e.* geocast). Cuando un dispositivo envía un paquete a otro dispositivo, aplica enrutamiento geográfico usando las coordenadas de la celda para la elección del siguiente salto. El paquete es enviado a la celda donde cualquier dispositivo en ella puede recibirlo, si no es el dispositivo destino busca entre sus vecinos quién lo es. Para llevar a cabo este enrutamiento es necesario dividir el plano donde se encuentra la red, en regiones o particiones regulares (*i.e.* celdas). Dicho proceso se explica a continuación.

5.2.5. Particiones regulares del plano

La idea consiste en dividir el plano en regiones formadas por polígonos regulares (*e.g.* cuadrados, hexágonos, triángulos). Donde cada región o celda está representada por un solo punto virtual, el centroide del polígono regular. Dependiendo de la topología de la red, algunas de estas regiones estarán activas debido a que contienen al menos un dispositivo en ellas, mientras otras regiones estarán inactivas al encontrarse vacías. En este trabajo se utiliza una partición basada en cuadrados. Para la construcción con otros polígonos se puede consultar el trabajo doctoral de (Tejeda, 2005).

Un aspecto importante a considerar en la división del plano, es el tamaño de los polígonos regulares. Para la selección del tamaño se pueden tomar en cuenta los siguientes criterios:

a) El tamaño del polígono es mayor que el radio de transmisión R de los dispositivos, por lo que no se garantiza que un dispositivo de la red alcance cualquier punto que se encuentra dentro del polígono, teniendo entonces que realizar tareas adicionales para garantizar la conectividad en la celda. Ver Figura 18.a.

b) El tamaño del polígono contiene al radio de transmisión R, es decir, cualquier par de dispositivos que se encuentren en la frontera del polígono tiene una distancia menor o igual al radio de transmisión R. A diferencia del caso anterior, cualquier dispositivo dentro del polígono se puede comunicar con cualquier otro dentro del polígono. Ver Figura 18.b.

c) El radio de transmisión es mayor respecto al par de dispositivos más alejados en la frontera del polígono, con lo anterior, se tiene que un dispositivo dentro de alguna celda alcance a dispositivos que se encuentren en las celdas vecinas. Cuando el radio de transmisión es igual a la distancia entre dos dispositivos vecinos que se encuentren en la frontera de sus respectivas celdas, los dispositivos podrán ver a todos los dispositivos que se encuentran en celdas vecinas. Ver Figura 18.c.

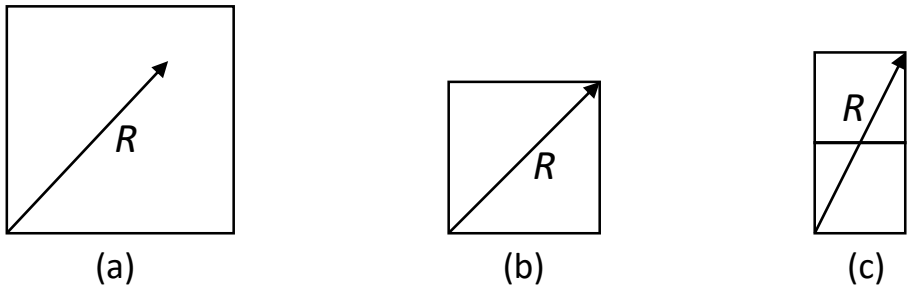


Figura 18: Variación del tamaño del polígono dependiendo de la cobertura deseada de los dispositivos dentro de la celda.

En este trabajo nos interesa que cada dispositivo en una celda pueda comunicarse directamente con cada uno de los dispositivos que se encuentren en las celdas vecinas, en este caso son las celdas que comparten un mismo lado con la celda donde se encuentra. El tamaño seleccionado es el mostrado en la Figura 18.c, obteniendo una partición del plano similar a la mostrada en la Figura 19.

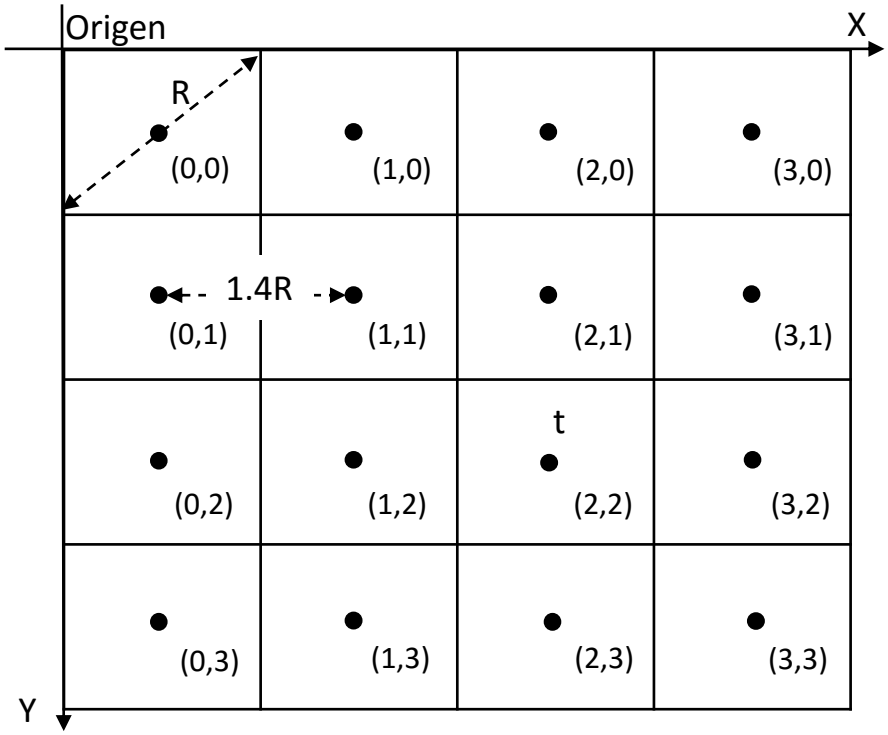


Figura 19: Partición del plano basada en cuadrados.

Cada celda es identificada por un par de coordenadas como se muestra en la Figura 19. Los dispositivos solo requieren conocer el tipo de polígono utilizado para crear la partición y el radio de transmisión *R* implementado para determinar la celda en que se encuentran. Utilizando las coordenadas (x_n, y_n) del dispositivo, se procede a realizar las siguientes operaciones para obtener las coordenadas virtuales de la celda:

$$x \leftarrow truncar\left(\frac{\sqrt{2x_n}}{R}\right) \tag{2}$$

$$y \leftarrow truncar\left(\frac{\sqrt{2y_n}}{R}\right) \tag{3}$$

Cada dispositivo puede realizar los cálculos de manera independiente sin la necesidad de una autorización central. Conociendo las coordenadas virtuales (*x,y*) de las celdas, se puede calcular las coordenadas (x_c, y_c) del centroide de la celda con las siguientes operaciones:

$$x_c \leftarrow \frac{(x+0.5)R}{\sqrt{2}} \tag{4}$$

$$y_c \leftarrow \frac{(y+0.5)R}{\sqrt{2}} \tag{5}$$

5.3. Algoritmo de enrutamiento geográfico implementando la partición del plano

El algoritmo diseñado consta de dos métodos para el reenvío de paquetes:

1. El método voraz, el cual consiste en reenviar un paquete de datos a un destino usando la posición geográfica de los dispositivos vecinos para la elección del siguiente salto, utilizándolo siempre que sea posible.
2. El enrutamiento basado en caras, el cual consiste en reenviar un paquete de datos a través de un conjunto de caras adyacentes hacia un dispositivo destino. Se utiliza en las regiones donde el método voraz falla.

Para detectar a sus vecinos, cada dispositivo hace transmisión de un paquete mediante broadcast el cual contiene el identificador único del dispositivo, sus coordenadas geográficas (x,y) y el identificador de la celda donde se encuentra actualmente. Para evitar la sincronización con otras transmisiones de dispositivos vecinos, se agrega un jitter (i.e. retraso) de 1 a 100 milisegundos. Cuando un dispositivo recibe uno de estos paquetes, agrega o actualiza un registro en su tabla de conectividad.

A continuación se describe el funcionamiento del método voraz y el enrutamiento basado en caras, haciendo uso de la partición del plano.

5.3.1. Método voraz

En VGHGR los paquetes contienen las coordenadas de la celda destino, como resultado, los dispositivos encargados de reenviar el paquete pueden tomar una decisión voraz localmente óptima, en la elección del siguiente salto. Específicamente, si un dispositivo conoce la posición de sus vecinos, puede enviar el paquete a su vecino más cercano con respecto al destino del paquete. Este proceso se aplica de manera sucesiva hasta que el destino es alcanzado.

Considerando que se tiene una partición regular del espacio, el dispositivo envía el paquete a una región D, guiado por las coordenadas del centroide que representa a dicha región. El algoritmo del método voraz usando los centroides de las regiones queda de la siguiente manera:

Algorithm 1 Elección del siguiente salto usando el método voraz

1: **procedure** REENVIOVORAZ(paquete)
2: $minimoLocal \leftarrow verdadero$
3: $distanciaActual \leftarrow Distancia(nodoActual, celdaDestino)$
4: **for all** $vecino$ in V **do** : ▷ Donde V es el conjunto de vecinos
5: $distanciaVecino \leftarrow Distancia(vecino, celdaDestino)$
6: **if** $distanciaVecino < distanciaActual$ **then**:
7: $distanciaActual \leftarrow distanciaVecino$
8: $siguienteSalto \leftarrow vecino$
9: $minimoLocal \leftarrow falso$
10: **if** $minimoLocal == verdadero$ **then**:
11: $RuteoPorCaras(paquete)$
12: **else**
13: $ReenviarPaquete(siguienteSalto, paquete)$

Cuando un dispositivo dentro de la celda destino recibe un paquete, puede proceder de dos maneras distintas:

- Si el dispositivo que recibió el paquete no es el destino, busca entre sus vecinos al dispositivo destino para reenviarle el paquete.
- Si el dispositivo es el destinatario del paquete, el dispositivo procesa el paquete.

Cuando un dispositivo no cuenta con un vecino que le permita seguir avanzando de manera voraz al destino, es necesario que cambie al enrutamiento basado en caras. Para contar con todas las rutas disponibles para el reenvío del paquete, se crea un grafo virtual creado a partir de las conexiones entre los centroides de la celda. Donde el enrutamiento basado en caras se lleva acabo sobre este grafo que también es plano.

A continuación se explica como se construye este grafo virtual y como funciona el enrutamiento basado en caras sobre dicho grafo.

5.3.2. Grafo virtual

El grafo virtual de (Tejeda *et al.*, 2006) se encuentra formado por un conjunto de nodos y enlaces virtuales, donde un nodo virtual representa a uno o más dispositivos, y una arista virtual representa a uno o varios enlaces. El grafo virtual puede ser visto como una representación de la red, en donde el enrutamiento se hace usando el conjunto de dispositivos y enlaces reales teniendo como referencia al grafo virtual. De esta manera, una vez elegida la dirección del siguiente salto, se tiene la posibilidad de escoger el enlace real que más convenga para seguir avanzando.

Para obtener el grafo virtual es necesario conectar algunos de los nodos virtuales, estas conexiones dependerán de la conexión que tengan los dispositivos en el grafo original que representa a la red. Dos nodos virtuales compartirán una arista si en sus respectivas celdas existen dos dispositivos que son mutuamente alcanzables. Para decidir la existencia de aristas virtuales entre las celdas y que a su vez nos garantice obtener un subgrafo plano, es necesario aplicar una prueba local, la cual es explicada en el siguiente apartado.

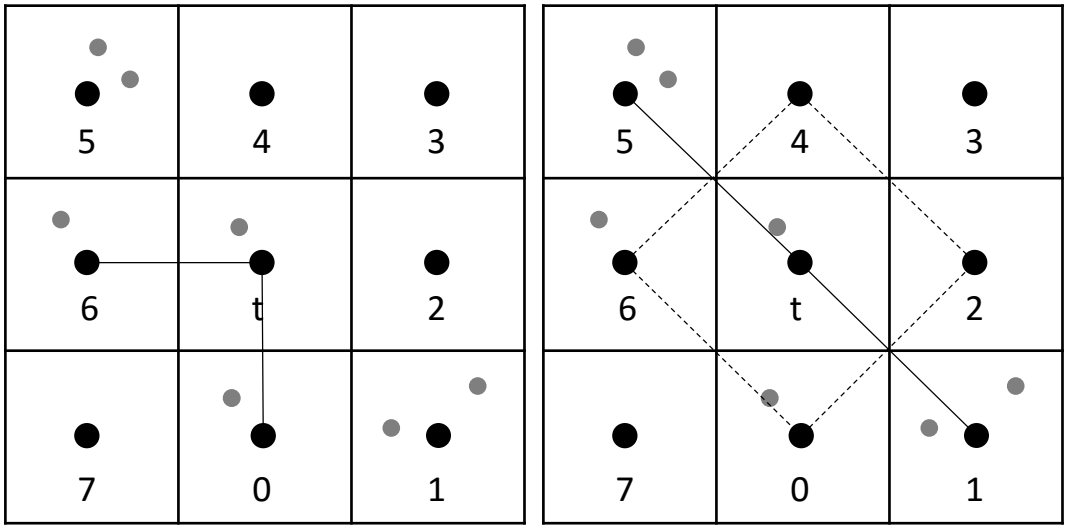
5.3.2.1. Prueba local

El grafo virtual de (Tejeda *et al.*, 2006) aplica dos pruebas locales para su construcción. La primera consiste en validar y agregar aristas a aquellas celdas vecinas adyacentes y la segunda permite validar y agregar aristas a las celdas que se encuentran a dos saltos.

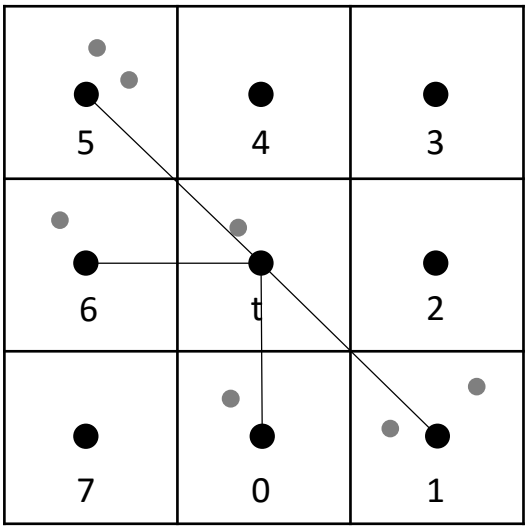
Existen algunos casos donde la segunda prueba genera un grafo con aristas de cruce, por lo que en este trabajo aplicamos únicamente la primera para obtener un grafo plano y conexo tomando en cuenta las celdas vecinas inmediatas.

El Algoritmo 2 recibe la posición de la celda *t* que será revisada, así como las posiciones de las celdas vecinas *I* respecto a la celda *t*. Las celdas vecinas están etiquetas del 0 al 7 como se muestra en la Figura 20. La función *ValidarArista(i,j)* devuelve *verdadero* si hay un par de dispositivos mutuamente alcanzables, donde uno de ellos se encuentra en la celda *i* y el otro en la celda *j*. La función *AgregarArista(i,j)* coloca una arista entre la celda *i* y la celda *j*. El Algoritmo 2 se explica a continuación:

- En las líneas 3-6 del Algoritmo 2 se revisan las celdas 0, 2, 4 y 6 como se muestra en la Figura 20.a. Si existe una arista valida (*i.e.* hay un dispositivo en cada celda donde son mutuamente alcanzables) entre la celda t y una alguna de estas celdas, se agrega una arista al grafo.
- En las líneas 17-21 del Algoritmo 2 se revisan las celdas 1, 3, 5 y 7. Para agregar una arista entre t y alguna de estas celdas, debe verificarse que no se presente un cruce localmente. Por ejemplo, para poner una arista entre t y la celda 1, no debe existir el enlace entre las celdas 0 y 2, tal como se muestra en la Figura 20.b.
- Los enlaces entre las celdas 0-2, 2-4, 4-6 y 6-0 se verifican en las líneas 8-15 del Algoritmo 2. El criterio es que no existan aristas virtuales que podrian sean intersectadas por la arista que se desea agregar. La vista local del grafo virtual que corresponde a la celda t se muestra en la Figura 20.c.



(a) Revisión de las celdas 0, 2, 4 y 6. (b) Revisión de las celdas 1, 2, 3 y 5.



(c) Vista local del grafo virtual para la celda t.

Figura 20: Aplicación de la prueba local para la construcción del grafo virtual.

El enrutamiento basado en caras se realiza sobre el grafo virtual obtenido. Cuando se inicie un recorrido de cara, el dispositivo encargado de enviar el paquete deberá aplicar el Algoritmo 2 para obtener una vista local del grafo virtual. Con esta nueva información, el dispositivo elige quien es el siguiente salto aplicando la regla de la mano derecha. A continuación, se muestra el algoritmo de la prueba local para la construcción del grafo virtual.

Algorithm 2 Prueba local para la construcción del grafo virtual

1: **procedure** PRUEBAGRAFO ▷ sea t la celda actual y I el conjunto de celdas vecinas
2: $k \leftarrow 0$
3: **while** $k < 8$ **do**
4: **if** $ValidarArista(t, I_k)$ **then**
5: $AgregarArista(t, I_k)$
6: $k \leftarrow k + 2$
7: $k \leftarrow 0$
8: **while** $k < 4$ **do**
9: $d \leftarrow 2k$
10: $e \leftarrow (2k + 2) \mod 8$
11: **if** $ValidarArista(I_d, I_e)$ **then**
12: $bVer_k = verdadero$
13: **else**
14: $bVer_k = falso$
15: $k \leftarrow k + 1$
16: $k \leftarrow 0$
17: **while** $k < 4$ **do**
18: $a \leftarrow 2k + 1$
19: **if** $!bVer_k \ \& \ ValidarArista(t, I_a)$ **then**
20: $AgregarArista(t, I_a)$
21: $k \leftarrow k + 1$

En la Figura 21, se puede apreciar que aplicando la prueba local se obtiene un grafo plano donde el enrutamiento basado en caras puede operar sin ningún problema, siempre que el grafo resultante sea conexo.

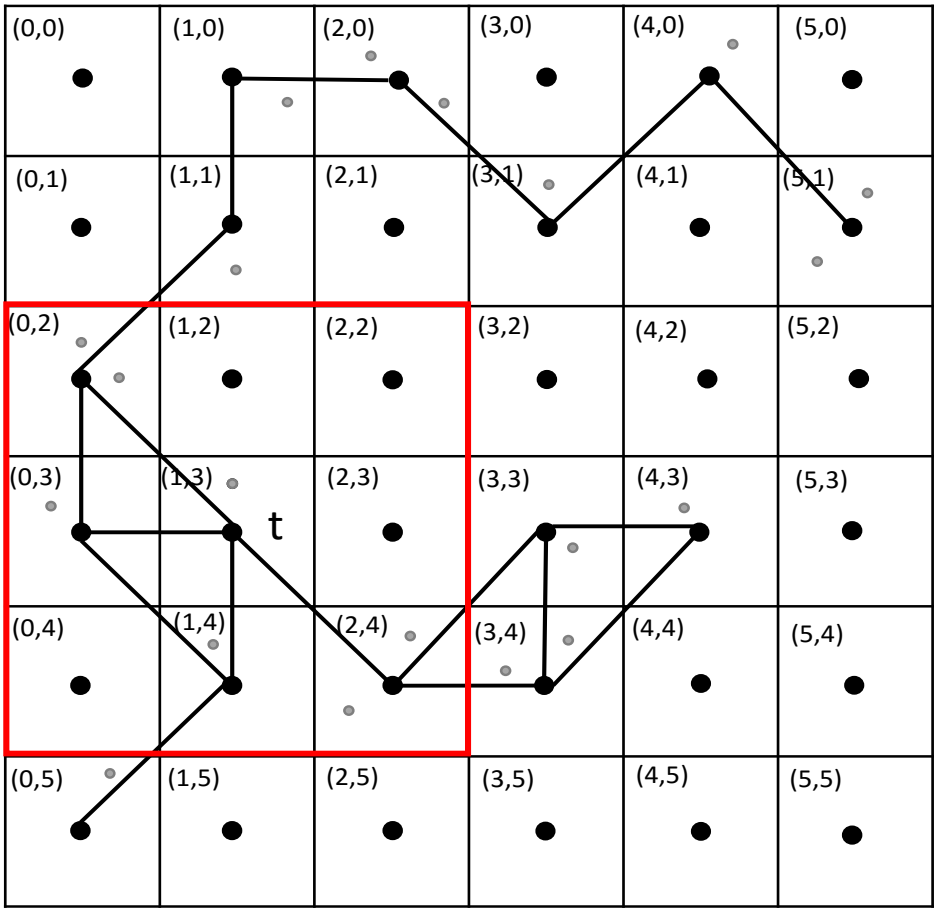


Figura 21: Construcción del grafo virtual usando la prueba local sobre una partición basada en cuadrados.

5.3.3. Enrutamiento basado en caras

En esta propuesta, el enrutamiento basado en caras consta de dos etapas. En la primera etapa, se efectúa el recorrido por caras sobre el grafo virtual para determinar la celda a la que debe ser enviado el mensaje. Dado que los nodos virtuales no cuentan con la capacidad para transmitir mensajes, en la segunda etapa el dispositivo responsable de transmitir el mensaje realiza el envío a un dispositivo en la celda seleccionada. Una descripción del algoritmo del enrutamiento basado en caras se presenta a continuación.

1. Utilizando sus coordenadas, el dispositivo determina la celda virtual a la que pertenece y el nodo virtual que la representa.

2. Utilizando la información geográfica de sus vecinos, el dispositivo calcula las aristas virtuales correspondientes de acuerdo a la prueba local explicada anteriormente.

3. En el modo enrutamiento basado en caras, el dispositivo hace uso del grafo virtual para seleccionar la arista virtual a seguir. Una vez seleccionada, se define la celda vecina que debe ser alcanzada. El dispositivo envía el paquete a cualquier dispositivo que se encuentre en la celda vecina seleccionada.

4. Una vez que el dispositivo en la celda seleccionada reciba el paquete, este envía el paquete, repitiendo el proceso hasta que se encuentre un dispositivo que permita regresar al método voraz.

En la Figura 22, se muestra el funcionamiento del algoritmo de enrutamiento basado en caras. El primer paso consiste en determinar la posición de las celdas donde se encuentran los nodos 20 y 4, en este caso, se localizan en las celdas (3,4) y (5,1), respectivamente. En este ejemplo, suponemos que el envío del paquete tiene lugar utilizando únicamente el método de enrutamiento por caras. Inicialmente, se visita la cara formada por las celdas (3,4), (4,3) y (3,3), en las que no existe un nodo que permita regresar al método voraz. El enrutamiento basado en caras se aplica en la siguiente cara a la que pertenece el nodo 20, resultando ser la cara externa. Cuando se explora la cara externa se descubre la celda destino. La ruta virtual descubierta está conformada por las celdas (4,3), (3,3), (2,4), (1,3), (0,2), (1,1), (1,0), (2,0), (3,1), (4,0) y (5,1).

Como se había mencionado anteriormente, el grafo virtual es tan solo una referencia. El enrutamiento real se hace a través de los dispositivos de la red, por lo que la ruta corresponde a los dispositivos 12, 6, 14, 19, 8, 11, 15, 18, 7, 10 y finalmente 4.

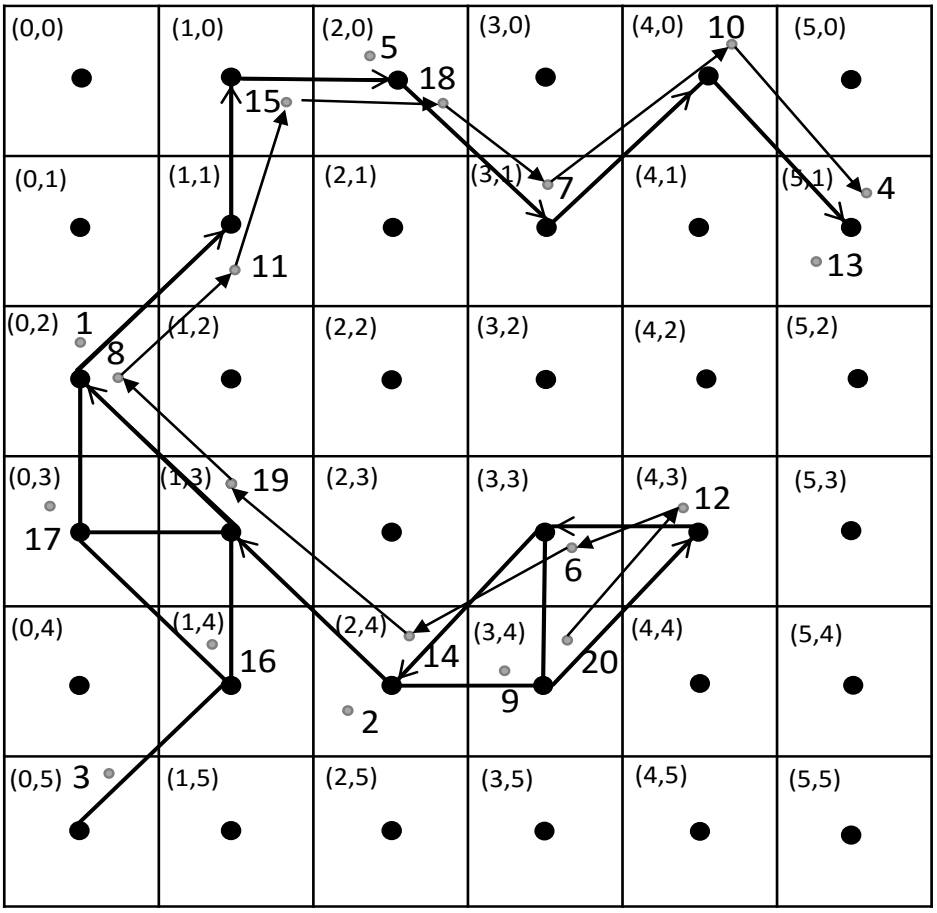


Figura 22: Enrutamiento basado en caras usando el grafo virtual.

5.4. Algoritmo de predicción de posición

Los dispositivos en la red no requieren estar notificando su posición constantemente a sus vecinos. La actualización se realiza cuando el dispositivo sabe que cambiará de celda en un tiempo t. Conociendo la dirección y velocidad del dispositivo, el algoritmo calcula la posición futura del dispositivo en los próximos 2 segundos. Una descripción del algoritmo de predicción de posición se presenta a continuación.

1. Se calcula la posición futura del dispositivo x en el próximo segundo. Si hay un cambio de celda, el dispositivo envía un mensaje a sus vecinos con su posición futura. De lo contrario, se calcula la posición futura de x en el segundo dos.

- El dispositivo vecino que reciba el mensaje debe verificar si el dispositivo x seguirá en una celda vecina. Si es así, se actualiza la posición de x en la tabla de vecinos. De lo contrario, se elimina la información que le corresponde.

2. Al presentarse un cambio de celda, el dispositivo realiza lo siguiente:

   - Envía un mensaje de actualización a su conjunto de celdas responsables.
   - Solicita la información asociada de la nueva celda a uno de los dispositivos que se encuentre en ella.
   - Elimina toda información almacenada sobre la celda anterior.

5.5. Implementación del protocolo VGHGR

En esta sección de describe los componentes que conforman al protocolo propuesto para su funcionamiento en una red ad hoc móvil.

5.5.1. Cabeceras y tipos de paquetes

El diseño de los tipos de paquetes y sus cabeceras para el nivel correspondiente al enrutamiento geográfico se realizó siguiendo el estudio de los protocolos presentados en el Capítulo 3. De estos protocolos, se han seleccionado aquellos tipos de paquetes y campos de sus cabeceras que eran más adecuados para llevar a cabo la implementación del protocolo propuesto. Además se han creado nuevos tipos de paquetes para mantener la información del servicio de localización.

Los paquetes siguen el esquema mostrado en la Figura 23. En esa figura se puede observar que la cabecera para el nivel de enrutamiento geográfico está compuesta por una cabecera común para todos los tipos de paquetes que necesitan ser reenviados a través de la red y por otra parte dependiente del tipo de paquete. En los siguientes apartados se muestran los tipos de paquetes y sus cabeceras resultantes.

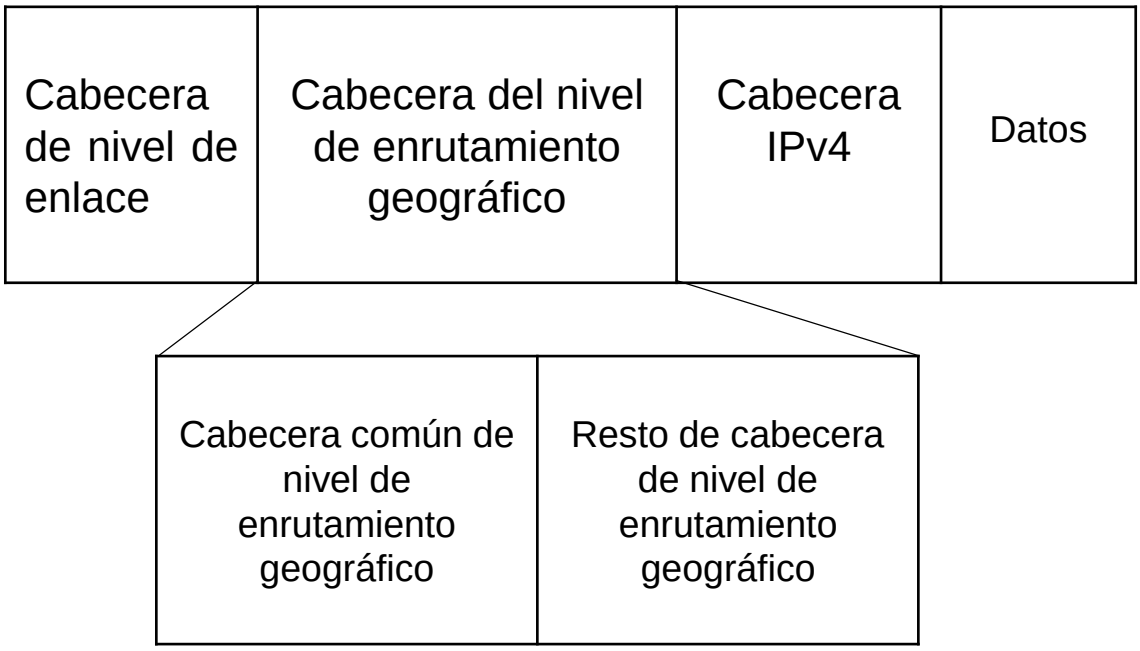


Figura 23: Estructura de un paquete.

5.5.1.1. Definición de tipos de paquetes

Cada paquete que necesita ser reenviado a través de la red, tiene una cabecera en común, el cual proporciona toda la información necesaria para realizar un enrutamiento geográfico aplicando el método voraz y el método por caras.

En la Tabla 1 se indican los tipos de paquetes a nivel enrutamiento geográfico que se crearon para este protocolo. El identificador (TM) le permite saber a los dispositivo el tipo de paquete recibido, por lo que siempre va incluido en la cabecera de cada paquete.

Tabla 1: Tipo de paquete.

Paquete	Identificador	Descripcion
HELLO	4	Compartir posición geográfica del dispositivo
REQUEST	6	Petición de tabla de conectividad y registros de tabla hash
REGISTER	1	Registro de ubicación para celdas responsables
QUERY	2	Consulta de ubicación de destino a celdas responsables
RESPONSE	3	Respuesta a una consulta de ubicación

5.5.1.2. Descripción del formato de los paquetes

En este apartado se presenta el formato de los paquetes y se describe la función de cada campo en las cabeceras.

Cabecera común del nivel de enrutamiento geográfico

La cabecera común se encuentra presente en cada uno de los paquetes que necesitan ser reenviados a través de la red, ya que cuenta con la información necesaria para poder realizar un enrutamiento geográfico. Está formado por los siguientes campos:

- *SC (Celda Origen)*: Es el identificador de la celda origen, el cual está compuesto por las coordenadas del centroide de la celda.
- *SD (Celda Destino)*: Es el identificador de la celda destino, el cual está compuesto por las coordenadas del centroide de la celda.
- *TTL (Tiempo de vida del paquete)*: Es el tiempo de vida del paquete en saltos, cuando se llega al máximo del número permitido de saltos, el paquete es descartado automáticamente sin ser procesado.
- *FM (Método por Cara)*: Es un valor booleano que indica si un paquete será reenviado usando el enrutamiento basado en caras.
- *SF (Cara Origen)*: Es el identificador del vértice que representa el inicio de la cara, es usado para evitar ciclos en el enrutamiento basado en caras. Este campo es necesario cuando el campo FM es verdadero.
- *NF (Cara Actual)*: Identificador de la cara actual, requerida para el funcionamiento del método de enrutamiento basado en caras. Este campo es necesario cuando el campo FM es verdadero.
- *SF (Paso)*: Indica en que paso del método de enrutamiento basado en caras se encuentra. Este campo es necesario cuando el campo FM es verdadero.

- *ADV (avance)*: Es un valor booleano que indica si ha existido un avance anteriormente, se requiere para validar si el dispositivo actual permite regresar al método voraz. Este campo es necesario cuando el campo FM es verdadero.
- *FD (Distancia final)*: Es un campo que complementa al campo ADV. Para validar si el dispositivo actual permite regresar al método voraz, es necesario saber la distancia del último dispositivo que permitió un progreso hacia el destino. Este campo es necesario cuando el campo FM es verdadero.

Hay que señalar que la cabecera del nivel de enrutamiento geográfico se modifica en cada salto. Por lo tanto los valores presentes se refieren al último dispositivo que reenvió el paquete y no al dispositivo fuente que lo transmitió.

Paquete HELLO

El paquete HELLO se utiliza para informar a todos los dispositivos vecinos a un salto (i.e. aquellos que se encuentren dentro del rango de transmisión) de la posición del dispositivo emisor. Su formato se muestra en la Figura 24 y se encuentra dividido en 4 bloques de 8 bits.

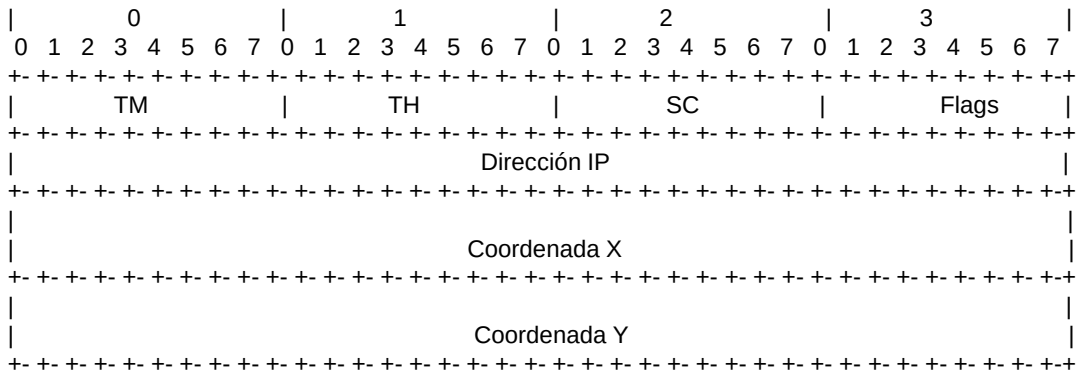


Figura 24: Representación ASCII de un paquete HELLO.

Se encuentra formado por los siguientes campos:

- *TH (Tipo de paquete hello)*: Es un número entero sin símbolo que representa el tipo de paquete HELLO que se envía. Los tipos de mensajes son los siguientes:
    1. Es un paquete HELLO normal, envía la posición del dispositivo vía broadcast.
    2. Es un paquete HELLO que se encarga de notificar la futura posición del dispositivo. Se envía cuando un dispositivo cambia de celda y solo notifica a los dispositivos que tiene en su tabla de conectividad.
    3. Es un paquete HELLO normal, con excepción de que este paquete espera una respuesta. La respuesta es la ubicación de todos los dispositivos vecinos que recibieron el paquete.
- *SC (Celda Origen)*: Es la ubicación del dispositivo emisor a nivel celda. Este valor es necesario cuando se aplica un enrutamiento basado en caras.
- *Dirección IP*: Este campo es la dirección IP del dispositivo emisor. Es necesario cuando un dispositivo vecino decide enviarle un paquete.
- *Coordenada X*: Es la posición geográfica en X del dispositivo emisor.
- *Coordenada Y*: Es la posición geográfica en Y del dispositivo emisor.
- *Flags*: Es una variable necesearia para serializar las coordenadas en el paquete.

Paquete REQUEST

El paquete REQUEST se utiliza para realizar una petición de la tabla de conectividad y tabla de registros asociados a la celda actual. Este paquete se envía cuando un dispositivo cambia de celda, solicitando a un dispositivo vecino que se encuentre en la misma celda la información mencionada anteriormente. Para disminuir el tamaño del paquete la respuesta es enviada en dos partes. Su formato aparece en la Figura 25.

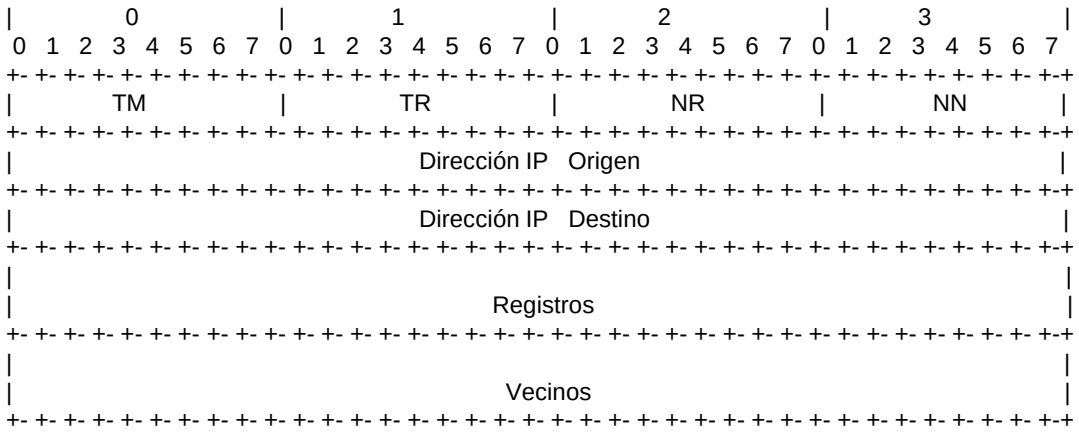


Figura 25: Representación ASCII de un paquete REQUEST.

Este paquete está formado por los siguientes campos:

- *TR (Tipo de paquete REQUEST)*: Es un número entero sin símbolo que representa el tipo de paquete REQUEST que se envía. Los tipos son los siguientes:

  1. Es un paquete REQUEST normal, donde se hace una petición de información de manera unicast a un dispositivo vecino.
  2. Es la respuesta a un paquete REQUEST tipo 1. Este paquete contiene los registros de la tabla hash asociados a la celda actual.
  3. Es la segunda parte de la respuesta a un paquete REQUEST tipo 1. Este paquete contiene la tabla de conectividad asociada a la celda actual.

- *Dirección IP origen*: Es la dirección IP del dispositivo emisor, este campo es necesario cuando se envía una respuesta al paquete.

- *Dirección IP Destino*: Es la dirección IP del dispositivo a quien va dirigido el paquete.

- *NR (Número de registros)*: Es la cantidad de registros de la tabla hash asociada a la celda que lleva el paquete. Este campo es requerido cuando es un paquete REQUEST tipo 2.

- *NN (Número de vecinos)*: Es la cantidad de registros de la tabla de conectividad asociada a la celda que lleva el paquete. Este campo es requerido cuando es un paquete REQUEST tipo 3.
- *Registros*: Es un vector que contiene la información de los elementos en la tabla de registros asociados a la celda actual. Este campo es requerido cuando se tiene un paquete REQUEST tipo 2.
- *Vecinos*: Es un vector que contiene la información de la tabla de conectividad asociada a la tabla. Este campo es requerido cuando es un paquete REQUEST tipo 3.

Paquete REGISTER

El paquete REGISTER es utilizado para enviar la información de la ubicación geográfica a nivel celda de cada uno de los dispositivos que conforman la red, a cada una de sus celdas responsables. Cuando un dispositivo entra a la red lo primero que hace es enviar un paquete REGISTER a cada una de las celdas que están a cargo de él. Cuando un dispositivo cambia de celda, manda un paquete REGISTER a cada una de sus celdas responsables, para mantener actualizada la información. Cuando un dispositivo es el primero en recibir el paquete REGISTER en la celda destino, debe propagar la información a sus vecinos en la misma celda. Su formato aparece en la Figura 26.

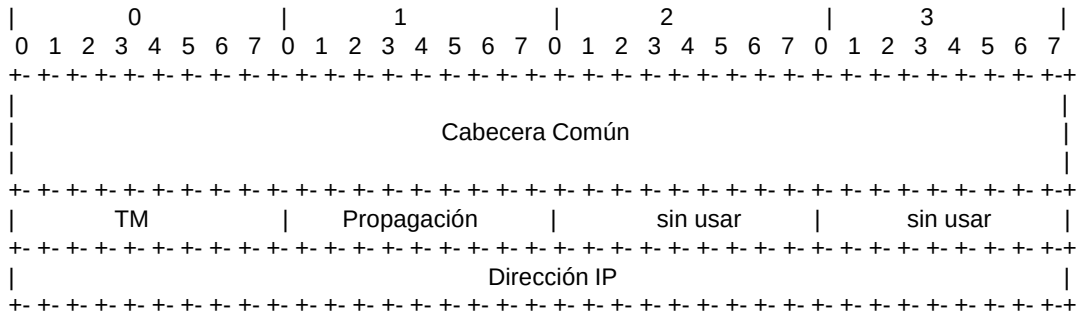


Figura 26: Representación ASCII de un paquete REGISTER.

Este paquete está formado por los campos siguientes:

- *Cabecera común*: Es la cabecera común para todos los tipos de paquetes requieren ser reenviados a través de la red. Está formada, entre otras cosas, por el identificador de celda del dispositivo emisor, el cual es requerido para el registro del mismo. Sus campos han sido descritos previamente.
- *Dirección IP*: Es la dirección ip del dispositivo, el cual sirve como identificador del registro en la tabla de registros de las celdas responsables.
- *Propagación*: Es un valor booleano que le dice al dispositivo si debe o no propagar el paquete a sus vecinos en la misma celda.

Paquete QUERY

El paquete QUERY es utilizado cuando un dispositivo desea enviar un paquete de datos y no conoce la ubicación del dispositivo destino. Antes de transmitir el paquete de datos, envía un paquete QUERY a las celdas responsables del dispositivo destino. Su formato aparece en la Figura 27.

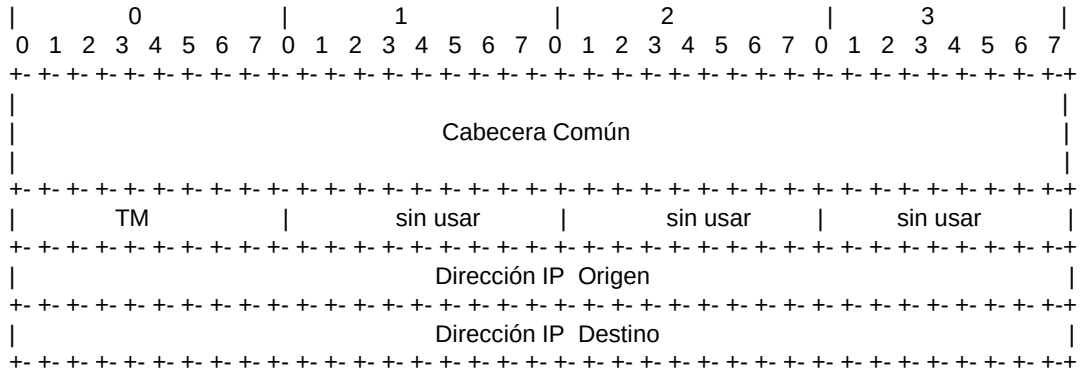


Figura 27: Representación ASCII de un paquete QUERY.

Los campos que conforman este paquete son los siguientes:

- *Cabecera Común*: Es la cabecera común para todos los tipos de paquetes que ocupan ser reenviados a través de la red. Sus campos han sido descritos previamente.
- *Dirección IP Origen*: Es la dirección IP del dispositivo que envía la consulta. Este campo es necesario para saber a quién se le mandara la respuesta.
- *Dirección IP Destino*: Es la dirección IP del dispositivo destino, sirve como clave para consultar la tabla de registros en las celdas responsables del dispositivo destino.

Paquete RESPONSE

El paquete RESPONSE es enviado como respuesta a un paquete QUERY. Este paquete contiene la ubicación a nivel celda del dispositivo buscado. Cualquier dispositivo en una celda responsable, puede recibir un paquete QUERY y enviar una respuesta al dispositivo que originó la consulta. Si el dispositivo no tiene la información solicitada, no manda ninguna respuesta. El formato de un paquete RESPONSE aparece en la Figura 28.

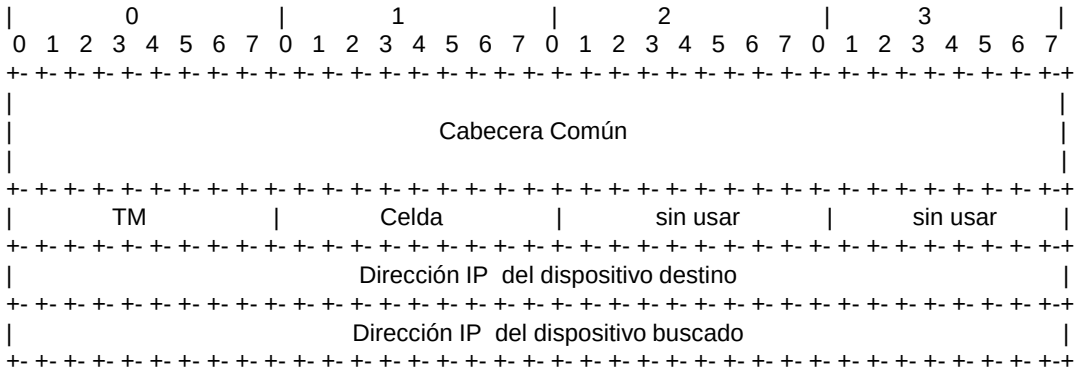


Figura 28: Representación ASCII de un paquete RESPONSE.

Los campos que conforman este paquete son los siguientes:

- *Cabecera Común*: Es la cabecera común para todos los tipos de paquetes requieren ser reenviados a través de la red. De la cabecera recibida en el paquete QUERY, se toma el valor del campo de la Celda origen del paquete. Sus campos han sido descritos previamente.
- *Celda*: Es el identificador de la celda donde se encuentra el nodo buscado.
- *Dirección IP del dispositivo buscado*: Es la dirección IP del dispositivo buscado. Este campo se mantiene para que el dispositivo que hizo la consulta verifique a quien debe enviar el paquete de datos.
- *Dirección IP del dispositivo destino*: Es la dirección IP del dispositivo que envió el paquete QUERY. Este campo es necesario para saber a quién va dirigido el paquete una vez se llegue a la celda destino.

5.5.2. Tabla de conectividad

Se trata de una tabla para almacenar los datos de la posición geográfica de los dispositivos vecinos, es decir, aquellos que se encuentran dentro del rango de transmisión. Cada registro en la tabla contiene la dirección IP, el identificador de la celda y las coordenadas geográficas del dispositivo vecino, tomando un espacio en memoria de 13 Bytes por cada registro. La Tabla 2 muestra el tamaño de cada campo.

Tabla 2: Tabla de conectividad.

Dirección IP	Celda	Coordenada X	Coordenada Y
4 bytes	1 byte	4 bytes	4 bytes

5.5.3. Tabla de registros

La tabla de registros permite, a cada uno de los dispositivos en una celda, guardar una copia de los registros asociados a la celda. Cada registro es un par (clave, valor) en la tabla donde la clave es la dirección IP del dispositivo y el valor es el identificador de la celda donde se encuentra el dispositivo, tomando un espacio en memoria de 5 bytes por cada registro. La Tabla 3 muestra el tamaño por cada campo mencionado.

Tabla 3: Tabla de registros.

Dirección IP	Celda
4 bytes	1 byte

5.5.4. Buffer de paquetes

Es una tabla donde se almacenan cada uno de los paquetes que son originados por el dispositivo, para disminuir la congestión en la red. El buffer tiene un límite de paquetes a almacenar, si el buffer se encuentra lleno y se quiere agregar otro paquete, se elimina el paquete de mayor antigüedad en el buffer. Los paquetes son eliminados automáticamente si exceden un tiempo de caducidad o ya fueron enviados. También se maneja un campo de prioridad, siempre que se tenga un paquete de datos en el buffer este será el primero en enviarse.

5.6. Resumen

El uso de una tabla hash distribuida como servicio de localización introduce autonomía y descentralización en la red, permitiendo a los dispositivos comunicarse con cualquier otro sin ninguna coordinación central, ya que son ellos mismos quienes mantienen la tabla. Este enfoque mejora la tolerancia a fallas de la red si continuamente se tienen dispositivos que entran o salen de la red. A diferencia de los protocolos proactivos (e.g. OLSR), la ruta al dispositivo destino no está disponible inmediatamente. En este protocolo el dispositivo origen primero obtiene la posición del dispositivo destino de una celda responsable, introduciendo un retardo al envío del paquete de datos inicial.

Capítulo 6. Evaluación del desempeño de VGHGR

6.1. Introducción

En este capítulo, se muestra la evaluación del protocolo VGHGR sobre redes ad hoc móviles. Para validar los resultados obtenidos, el desempeño de VGHGR es comparado con dos protocolos proactivos (OLSR y DSDV) y un protocolo reactivo (AODV), los cuales son comúnmente utilizados en la literatura. Para realizar dicha evaluación se propusieron diferentes escenarios en el simulador de redes de eventos discretos NS -3. Las métricas de desempeño evaluadas en este trabajo son la tasa de entrega de paquetes, throughput, energía consumida para la transmisión de los paquetes, memoria requerida, escalabilidad y tiempo de entrega de los paquetes.

6.2. Descripción del simulador NS-3

NS-3 es un simulador de redes de eventos discretos desarrollado para proporcionar una plataforma abierta y extensible para la simulación de redes, dirigido principalmente para la investigación y uso educativo [1]. NS-3 proporciona modelos para el funcionamiento de las redes de paquetes de datos y provee un motor de simulación para que los usuarios lleven a cabo sus experimentos. Algunas de las razones para utilizar NS-3 incluyen la realización de estudios que son difíciles o imposibles de efectuar en sistemas reales, para estudiar el comportamiento del sistema en un entorno altamente controlable y reproducible, y para aprender sobre el funcionamiento de las redes.

Existen muchas herramientas para el estudio de simulaciones de redes, algunas de las características que distinguen a NS-3 sobre ellas son las siguientes:

- NS-3 está diseñado como un conjunto de librerías que pueden ser combinadas entre si y también con otras librerías externas, esto se debe que está escrito en C++ con la opción de incluir módulos de Python. Varios analizadores de datos y herramientas de visualización pueden ser utilizados con NS-3, siempre que se encuentren construidos en C++ o Python.

[1] Pagina Web de NS-3. https://www.nsnam.org/

- NS-3 es usado principalmente sobre sistemas Linux, aunque existe soporte para FreeBSD, Cygwin (para Windows), y el soporte nativo de Visual Studio de Windows está en proceso de desarrollo.
- NS-3 no es un producto de software con apoyo oficial de alguna empresa. NS-3 es open-source, por lo que la mayor contribución de su estructura es gracias a sus usuarios.

Por su fácil aprendizaje para la implementación de un nuevo módulo, el soporte para redes ad hoc y la encapsulación de toda la pila de red de Linux en un nodo ns-3, se eligió este simulador para realizar las pruebas del protocolo VGHGR.

6.3. Metodología de evaluación

Para evaluar el desempeño de la implementación de nuestro protocolo de enrutamiento VGHGR, las simulaciones se llevaron a cabo en la versión 3.25 del simulador NS-3 (última versión estable). Configuramos los parámetros de la simulación lo más cercanos posibles a estudios previos (Jha y Kharga, 2015; Narra *et al.*, 2011; Singla y Jain, 2014) con el fin de obtener resultados comparables. Las simulaciones se realizaron en un área de 1000 x 1000 m^2 y se promedian los resultados de 5 ejecuciones en cada caso. Estudiamos los efectos de la variación en la densidad de dispositivos en la red, la cantidad de paquetes enviados y tamaño de los mismos.

La capa de enlace utilizada es 802.11b MAC sobre el modelo de pérdida de propagación Friis para limitar el radio de transmisión de los nodos. El radio de transmisión de los nodos para esta evaluación es de 250 m. Para alcanzar este radio de transmisión, la potencia de transmisión es de 8.9048 dBm. Para los parámetros únicos de cada protocolo se dejaron los valores por defecto que cada uno presenta en sus respectivos módulos en el simulador NS-3. Algo importante a mencionar, es que todas las pruebas se realizaron sobre una red estática, dejando la evaluación en un ambiente móvil para trabajo a futuro. Los parámetros completos de las simulaciones se muestran en la tabla 4.

Tabla 4: Parámetros de las simulaciones

Parametro	Valor
Simulador	NS-3.25
Tiempo de simulación	100-200-300-400-500 seg
Inicio para el envió de datos	50 seg
Nodos transmisores	1 a 30
Nodos receptores	Depende la cantidad de paquetes enviados
Paquetes enviados por nodo	10-20-30-40-50
Tiempo entre paquete enviado	2 seg
Tamaño de paquetes	64-256-512-1024-2048-4096 bytes
Velocidad DSSS	11 Mbps
Rango de Transmisión	250 m
Modelo de Propagación	Modelo Friis
Area	1000 x 1000 m^2

6.3.1. Métricas de evaluación

Las métricas de desempeño para la evaluación del protocolo VGHGR consideradas en este trabajo de tesis son:

- **Garantía de entrega de paquete**. Es el número de paquetes recibidos sobre el número de paquetes enviados por la aplicación.
- **Tiempo de entrega**. El tiempo que le tomó al paquete alcanzar al dispositivo destino.
- **Energía**. Consumo total de energía para la transmisión de los paquetes.
- **Memoria**. Cantidad de memoria requerida para el almacenamiento de información sobre la vecindad de los dispositivos y la topología de la red.
- **Throughput total**. Cantidad de paquetes recibidos con éxito ante el aumento de tráfico en la red por dispositivos transmitiendo simultáneamente.
- **Escalabillidad**. Es la habilidad del protocolo para manejar el crecimiento continuo de trabajo de manera fluida, aumentando la cantidad de dispositivos en la red.

6.4. Pruebas realizadas y resultados

Los resultados de las simulaciones son presentados en 6 apartados mostrados a continuación. El comportamiento de cada uno de los protocolos es comparado con base en las métricas previamente mencionadas. En cada una de las simulaciones los nodos son distribuidos uniformemente.

6.4.1. Garantía de entrega de paquete

En este primer escenario, se crea una red de 100 nodos distribuidos uniformemente por todo el espacio. Cada uno de los nodos transmisores envía un paquete cada 2 segundos. Se varía la cantidad de paquetes enviados para analizar el efecto del incremento del tráfico en la red sobre la cantidad de paquetes entregados satisfactoriamente. Además, se experimenta con distintos tamaños de paquetes para simular el tipo de mensaje enviado: mensajes de texto, audio, video y fotografías.

A continuación, se muestran los resultados obtenidos en el escenario descrito previamente. Los resultados son divididos en 5 conjuntos, donde cada uno representa el tamaño de los paquetes transmitidos.

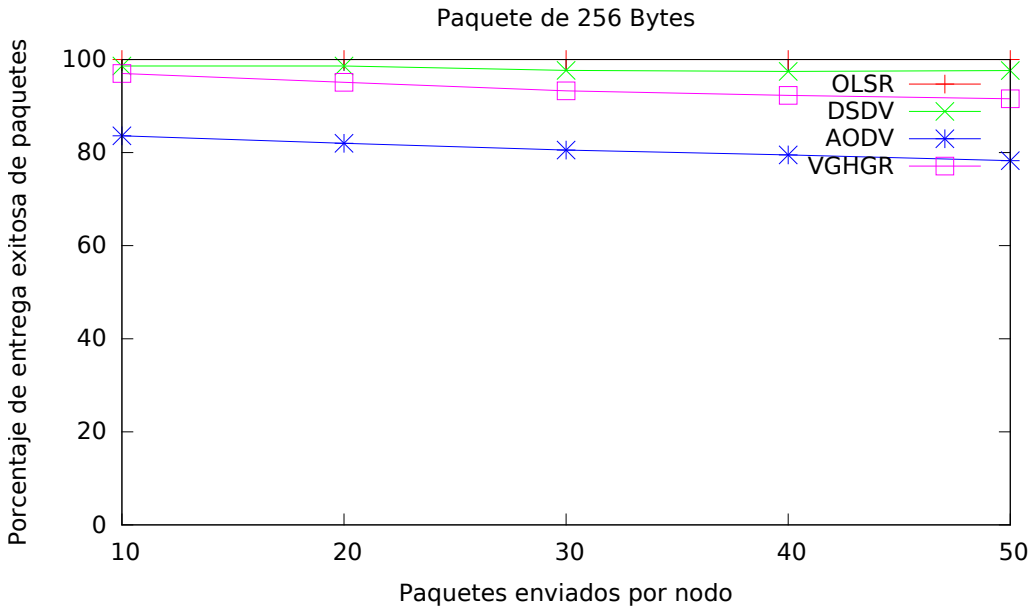
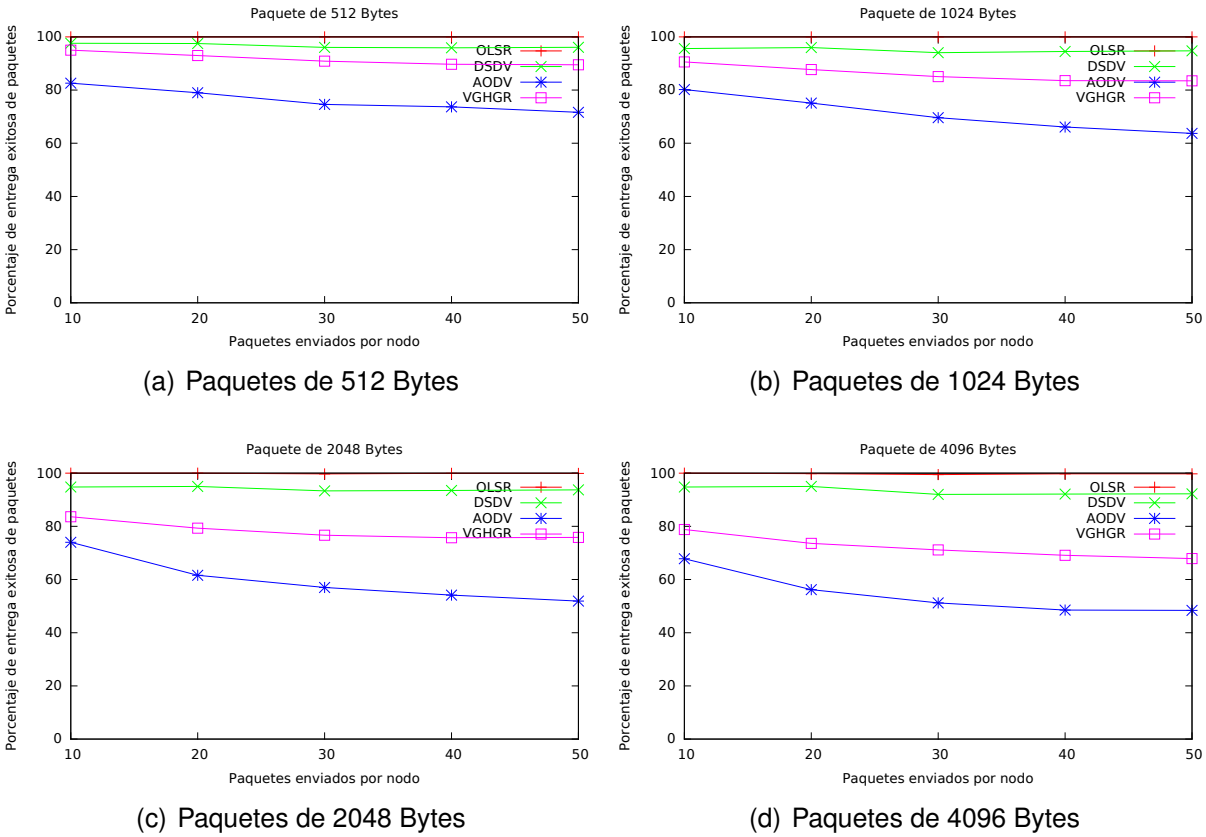


Figura 29: Porcentaje de entrega exitosa de paquetes de 256 Bytes.

La Figura 29, muestra el porcentaje de entrega de paquetes con un tamaño de 256 bytes. En todos los casos, el protocolo OLSR muestra un mejor desempeño seguido por el protocolo DSDV. Lo anterior muestra que los protocolos proactivos presentan una ventaja al mantener tablas de enrutamiento. El protocolo reactivo AODV obtuvo el peor desempeño. El protocolo VGHGR mantiene un porcentaje de entrega del 90 % o superior para la pruebas efectuadas.



(a) Paquetes de 512 Bytes

(b) Paquetes de 1024 Bytes

(c) Paquetes de 2048 Bytes

(d) Paquetes de 4096 Bytes

Figura 30: Porcentaje de entrega de paquetes a medida que se incrementa el tamaño de los paquetes enviados.

La Figura 30, muestra los resultados con distintos tamaños de paquete. A excepción del protocolo OLSR, todos los protocolos se ven afectados. Esto se debe a que el canal de comunicación está transportando más bytes, por lo tanto, la probabilidad de colisiones aumenta. Por otro lado, cada bit que se transmite tiene cierta probabilidad de ser recibido con error por lo que si el paquete es más largo también aumenta la probabilidad de perderlo. El protocolo VGHGR muestra su peor resultado (68 %) cuando el tráfico es constante y los paquetes enviados son de 4096 Bytes.

6.4.2. Throughput total

La evaluación del thoughput para el protocolo VGHGR se realizó bajo el siguiente escenario. Se crea una red de 100 nodos distribuidos aleatoriamente en el espacio, donde se varía el número de nodos transmisores. Se analiza la cantidad de paquetes entregados satisfactoriamente ante el incremento del tráfico en la red por nodos transmitiendo simultáneamente. El tiempo de espera entre el envío de cada paquete es de 2 segundos. En estas simulaciones se transmitieron paquetes de 64 bytes, para disminuir la probabilidad de colisiones o errores durante la transmisión.

En la Figura 31 se muestran los resultados obtenidos en el escenario descrito previamente. Los protocolos OLSR, DSDV y VGHGR mantienen un porcentaje de entrega del 95 % o superior para las pruebas realizadas. Una vez más, el protocolo OLSR muestra un mejor desempeño.

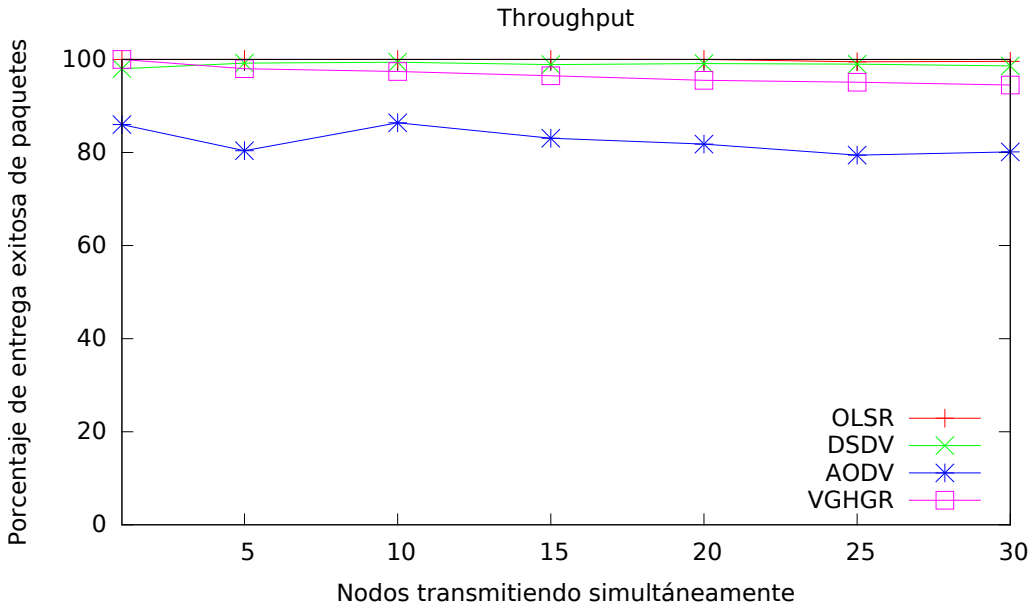


Figura 31: Porcentaje de entrega exitosa de paquetes variando el número de nodos transmitiendo simultáneamente.

6.4.3. Escalabilidad

Para probar la escalabilidad del protocolo VGHGR ante en el incremento de nodos en la red, se hizo una modificación al escenario presentado en la evaluación de la garantia de entrega de paquetes. En este caso, los nodos transmiten durante toda la simulación 10 paquetes cada uno de los cuales está dirigido a un destino distinto. El tiempo de espera entre el envío de cada paquete es de 2 segundos. Para disminuir la probabilidad de que se presenten colisiones por el tamaño del paquete o errores durante la transmisión, en estas simulaciones se transmiten paquetes de 64 bytes.

En la Figura 32, se muestran el porcentaje de entrega en redes de 100 a 400 nodos. A medida que se incrementa el número de nodos en la red, salvo en el protocolo OLSR, disminuye el porcentaje de paquetes entregados correctamente. El protocolo VGHGR tiene porcentaje de entrega del 76 % en una red de 400 nodos, sufriendo una caída del 21 % con respecto al desempeño presentado en redes de 100 nodos. Los protocolos DSDV y AODV muestran un desempeño pobre en redes muy densas, alcanzando un porcentaje de entrega de 26 % y 7.5 %, respectivamente.

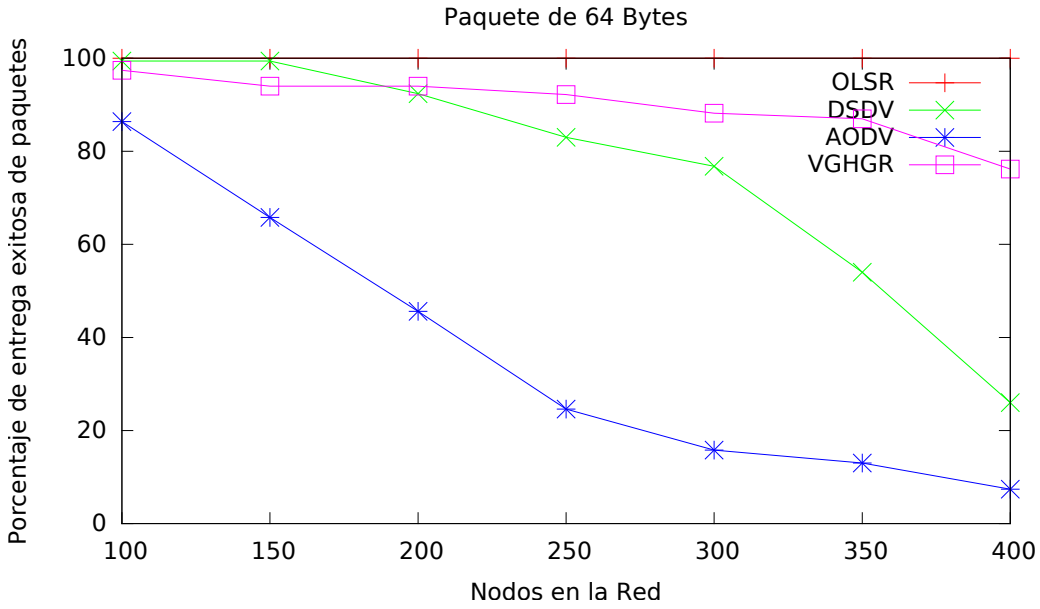
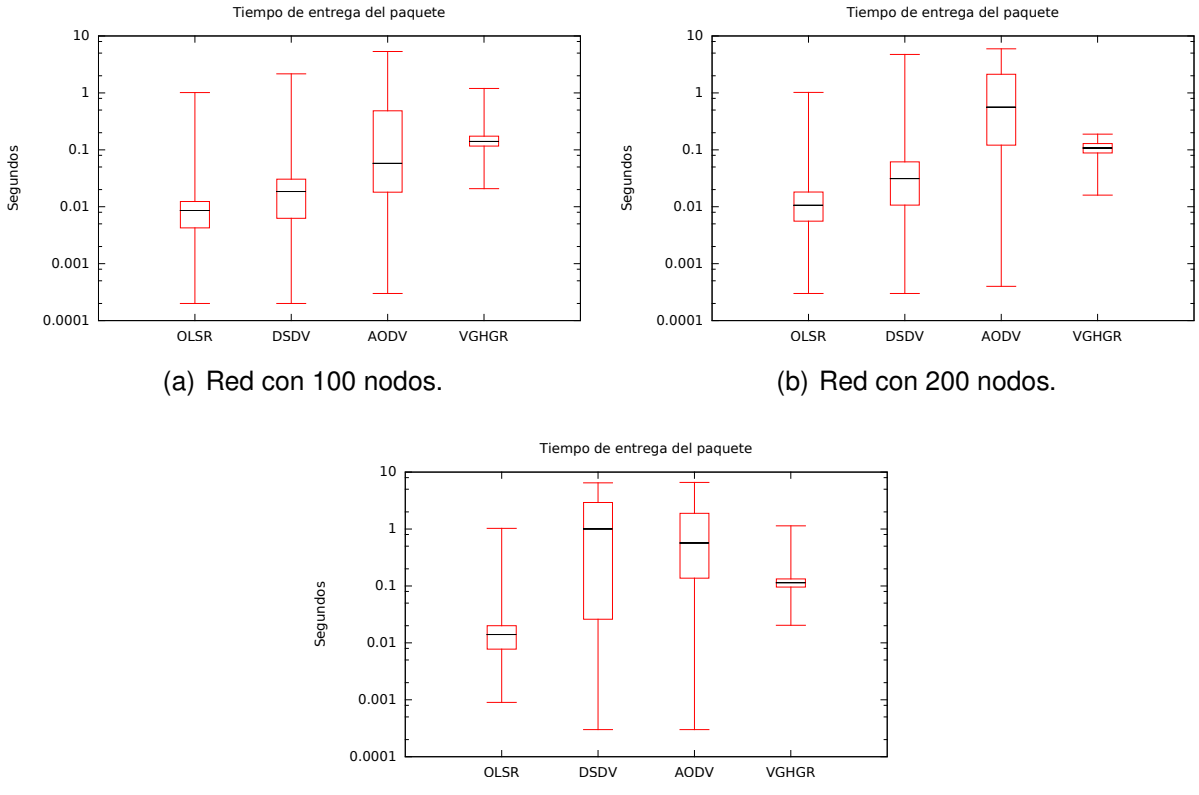


Figura 32: Porcentaje de entrega de paquetes de 64 Bytes variando el número de nodos presentes en la red.

6.4.4. Tiempo de entrega

La evaluación del tiempo de entrega de los paquetes se obtuvo simultáneamente con la evaluación de la escalabilidad. En la Figura 33, se muestra el tiempo de entrega para 100 paquetes en redes de 100, 200 y 300 nodos, para cada uno de los protocolos evaluados. En el eje Y se muestra los tiempo en segundos utilizando una escala logarítmica. La figura muestra los cuartiles, la mediana, el mínimo y máximo de los tiempos registrados. En la figura se puede observar que el protocolo OLSR y DSDV muestran un mejor tiempo de entrega ya que mantienen tablas de enrutamiento y los dispositivos cuentan con la información de ruteo de manera inmediata a diferencia de los protocolo AODV y VGHGR cuyo proceso de búsqueda del dispositivo destino incrementan el tiempo de envío del paquete de datos. Una ventaja del protocolo VGHGR, es que los tiempos de entrega son más predecibles con respecto al resto de los protocolos, lo cual se concluye observando el valor máximo y mínimo de cada uno de ellos.



(a) Red con 100 nodos.

(b) Red con 200 nodos.

(c) Red con 300 nodos.

Figura 33: Tiempo de entrega de paquetes.

6.4.5. Memoria

Al igual que en el caso anterior, durante la evaluación de la escalabilidad del algoritmo se hizo un registro de la memoria utilizada para almacenar la información requerida para el funcionamiento de los protocolos. La información por cada protocolo se muestra a continuación:

- Para el protocolo OLSR se consideraron la tabla de vecinos, la tabla de la topología y la tabla de enrutamiento .
- Para el protocolo DSDV se consideraron la tabla de vecinos y la tabla de enrutamiento.
- Para el protocolo AODV se consideraron la tabla de enrutamiento y la tabla de vecinos.
- Para el protocolo VGHGR se consideraron la tabla de registros y la tabla de conectividad.

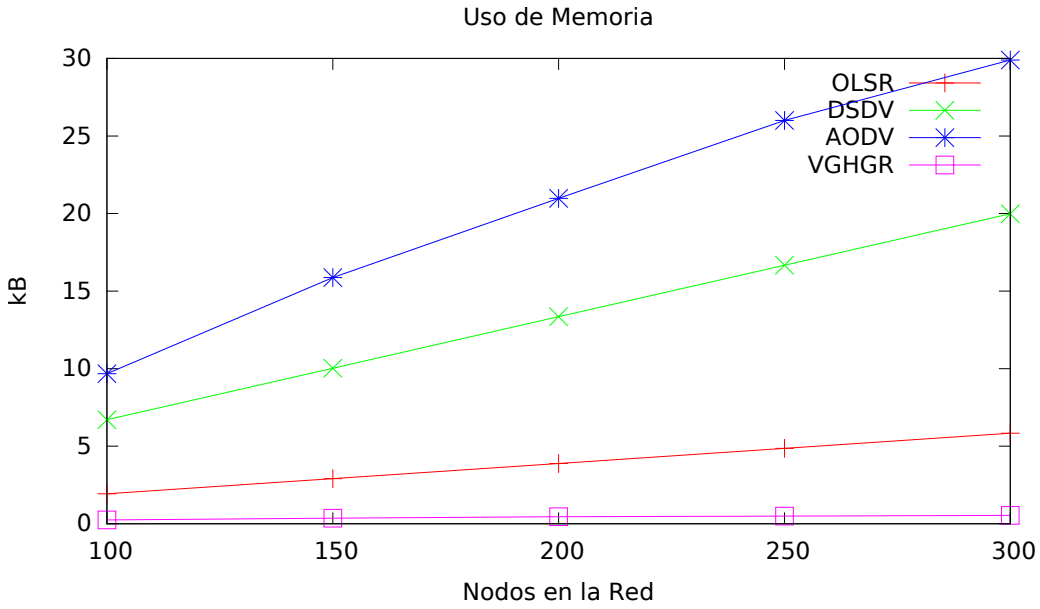


Figura 34: Memoria requerida para almacenamiento de información.

En la Figura 34, se muestran las cantidades de memoria en KB que requiere cada uno de los protocolos, variando la cantidad de dispositivos en la red.

El protocolo VGHGR muestra los mejores resultados en esta evaluación, demostrando que el mantener una tabla hash distribuida no requiere de mucha memoria por parte de los dispositivos. Le siguen los protocolos OLSR, DSDV y finalmente AODV. A pesar de ser un protocolo reactivo, AODV requiere de más memoria. Esto se debe a que almacena la ruta completa hacia cada destino.

6.4.6. Consumo de energía

Para evaluar el consumo de energía del protocolo VGHGR, se propuso el escenario descrito a continuación. Los nodos consumen mayor energía cuando transmiten una mayor cantidad de bytes. Para evaluar esta suposición se aumenta la cantidad de paquetes a enviar por cada nodo transmisor, enviando hasta 50 paquetes por nodo transmisor con un tiempo de espera de 2 segundos entre cada paquete enviado. La evaluación se realiza considerando el tamaño de los paquetes.

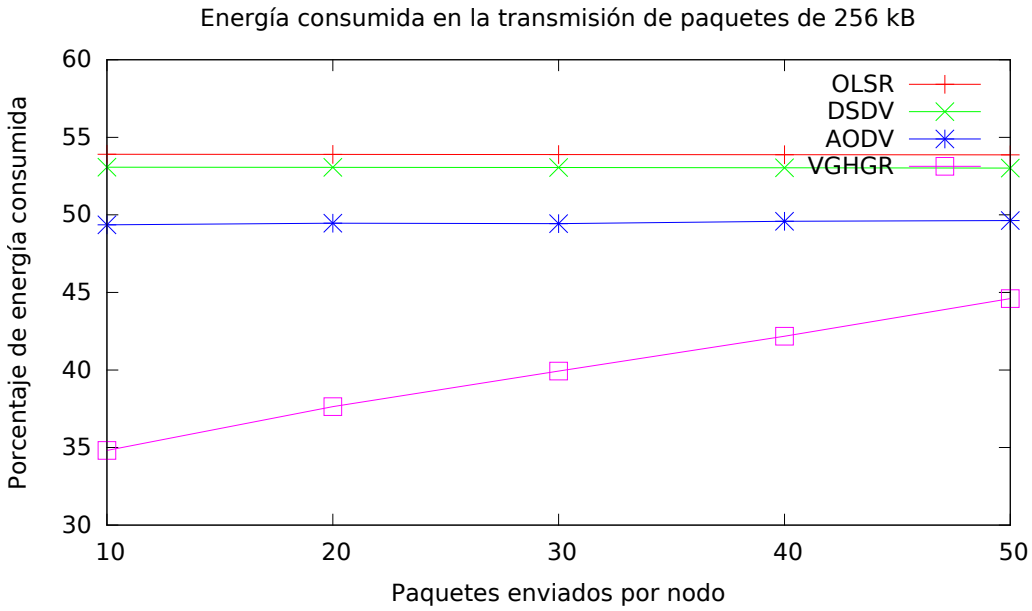
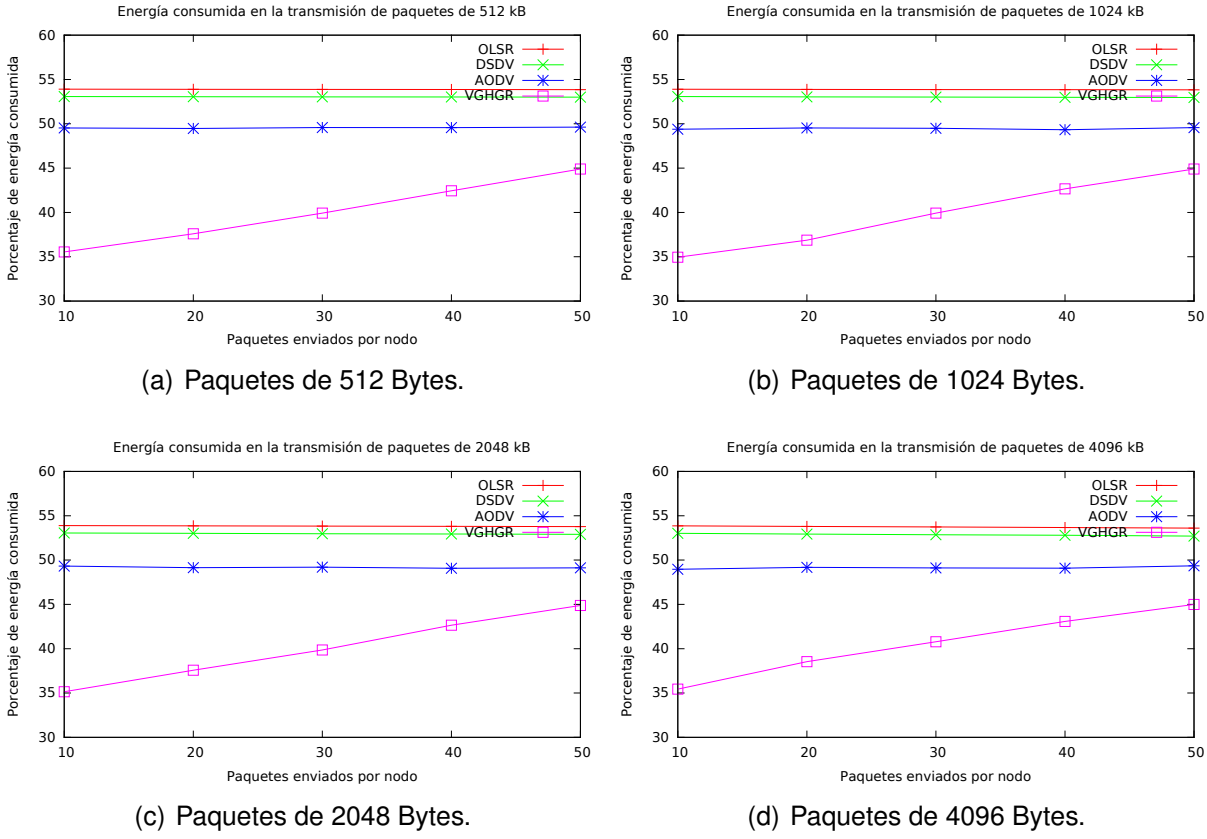


Figura 35: Energía consumida en la transmisión de paquetes de 256 Bytes.

En la Figura 35, se muestran los resultados de las simulaciones. En esta ocasión, un menor porcentaje indica un mejor desempeño, lo cual quiere decir que el protocolo consume menos energía. De acuerdo a los resultados obtenidos, VGHGR muestra mejores resultados para todos los tamaños de paquetes evaluados.

El peor resultado de VGHGR es un consumo del 44.6 % de energía en el envío de 50 paquetes por dispositivo transmisor, mientras el resto de los protocolos tiene un consumo superior al 49 %.

En la Figura 36, se muestran los resultados de consumo de energía variando el tamaño de los paquetes. Se puede observar que los protocolos proactivos OLSR y DSDV son los que tienen un mayor consumo, esto es una consecuencia de un mayor envío de paquetes de control para mantener las tablas de ruteo.



(a) Paquetes de 512 Bytes.

(b) Paquetes de 1024 Bytes.

(c) Paquetes de 2048 Bytes.

(d) Paquetes de 4096 Bytes.

Figura 36: Consumo de energía.

Un segundo escenario para la evaluación del consumo de energía es descrito a continuación. En las redes inalámbricas ad hoc móviles se desea extender el tiempo de vida de la red. Para evaluar esta suposición, se aumenta el tiempo de duración de las simulaciones. En la Figura 37 se muestran los resultados obtenidos. Al igual que los resultados anteriores, los protocolos DSDV y OLSR son los que presentan un mayor consumo de energía. El protocolo VGHGR muestra los mejores resultados nuevamente. La batería es completamente consumida en las simulaciones de 400 y 500 segundos.

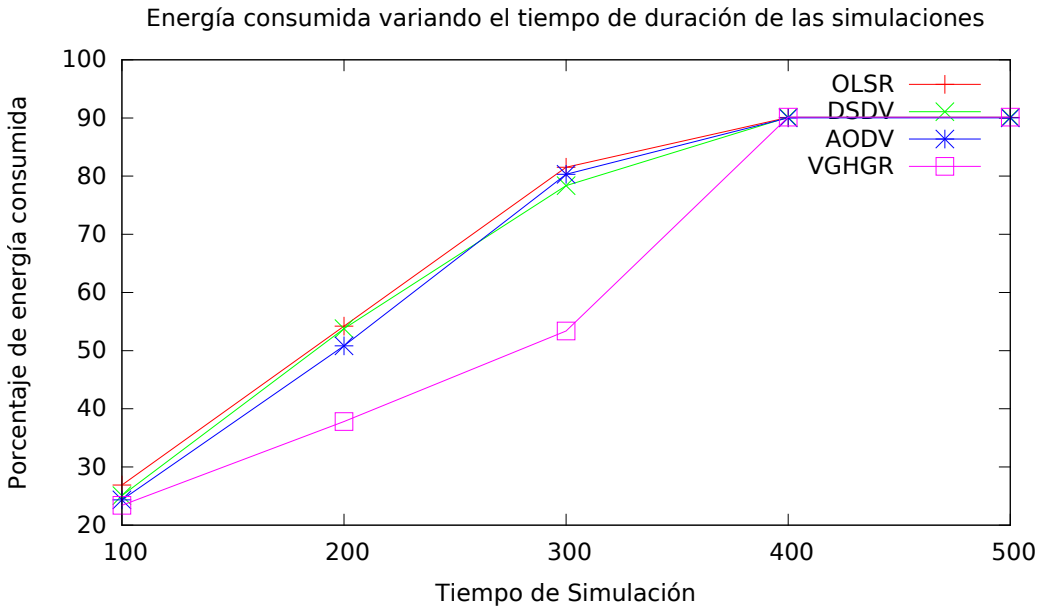


Figura 37: Energía consumida variando el tiempo de duración de las simulaciones.

6.5. Resumen

En este capítulo, se evaluó el protocolo VGHGR para enrutamiento geográfico en redes ad hoc móviles. El desempeño del algoritmo propuesto fue comparado con el mostrado por los protocolos OSLR, DSDV y AODV.

De acuerdo a los resultados obtenidos, nuestra propuesta tuvo un mejor desempeño en cuanto a consumo de energía y memoria requerida para almacenamiento. Los resultados en el consumo de energía se deben a que necesita de una mínima cantidad de paquetes extras para mantener los registros asociados al servicio de localización. Por otro lado, los resultados en memoria se deben a que los protocolos de enrutamiento geográfico no requieren de mucha información para la elección del siguiente salto en la ruta de entrega, por lo que la información almacenada en los nodos es mínima. Respecto a los tiempos de entrega, nuestra propuesta tiene la ventaja de que los tiempos registrados son más predecibles con respecto al resto de los protocolos. Sobre el porcentaje de entrega de paquetes, mostró un porcentaje de entrega superior al 80 % en paquetes con un peso máximo de 2048 bytes, alcanzando el 97 % en paquetes inferiores a 512 bytes. En la evaluación del throughput, nuestra propuesta mostró un porcentaje de entrega superior al 95 % y en la métrica de escalabilidad, se encuentra arriba del 78 % en cada uno de los escenarios presentados.

Capítulo 7. Conclusiones y trabajo a futuro

7.1. Introducción

Este capítulo concluye la investigación realizada y resume las contribuciones más importantes del trabajo. De igual forma, se incluyen algunas líneas de investigación que pueden explorarse en el futuro.

La principal aportación de este trabajo de tesis recae en el diseño, desarrollo y evaluación de un nuevo protocolo de enrutamiento geográfico sobre redes ad hoc móviles.

7.2. Conclusiones

En esta sección, se presentan las principales conclusiones y aportaciones desarrolladas durante el proceso de investigación para el diseño, desarrollo y evaluación del nuevo protocolo. Las conclusiones están divididas en dos partes, en el apartado 7.2.1 se discuten aquellas relacionadas con el diseño del protocolo, mientras en el apartado 7.2.2 se presentan las conclusiones de acuerdo a los resultados obtenidos en el proceso de evaluación.

7.2.1. Sobre el diseño del protocolo

A continuación, se presentan las características de diseño más sobresalientes del protocolo propuesto.

- Para disminuir la probabilidad de colisiones, los paquetes diseñados para el funcionamiento de este protocolo fueron optimizados para transportar la mínima información necesaria, reduciendo el tamaño de los mismos.
- El hecho de trabajar sobre tecnologías inalámbricas le agrega un mayor grado de dificultad al proceso de reenvío. Esto se debe a que los canales inalámbricos no son estables, por lo que es necesario estar enviando periódicamente beacons para mantener actualizadas las tablas de conectividad. Dos dispositivos vecinos no siempre se ven, esto se debe a la interferencia generada por el radio de transmisión de otros dispositivos.

- Para el reenvío de los paquetes a través de la red, en la medida de lo posible se dío preferencia al método voraz. La desventaja de realizar esto, es que se suelen sobrecargar ciertos dispositivos, en particular, los más alejados al dispositivo origen, por lo que aumenta la probabilidad de colisiones en estos dispositivos. Por lo tanto, se debe desarrollar un mecanismo que balancee la carga entre ellos.

- La partición del espacio permite crear un grafo virtual para el enrutamiento basado en caras y la implementación de un servicio de localización, por lo que se considera puedan existir otras formas de aprovechar la partición para el proceso de enrutamiento.

- Utilizar un grafo virtual en lugar de obtener un subgrafo plano directamente del grafo de la topología de la red, permite hacer uso de todos los caminos posibles para el envío de los paquetes.

- Sobre el servicio de localización basado en tablas hash distribuidas, la conectividad de los dispositivos es el mínimo requerimiento para el funcionamiento del servicio. En ambientes de alta movilidad, los cambios en la topología de la red se vuelven más frecuentes provocando un mayor grado de mantenimiento y consumo de recursos de los dispositivos. Mientras mayor sea el número de cambios en la red, mayor será la frecuencia con la que los dispositivos deben ejecutar los procedimientos para asegurar que la información en las tablas se encuentre actualizada.

- Los dispositivos deben enviar mensajes de actualización para mantener el servicio de localización con la información más reciente. Estos mensajes se generan siempre que hay un cambio en su posición geográfica. Para evitar el envío constante de estos mensajes los dispositivos solo notifican cuando detectan que han cambiado de celda. El algoritmo de predicción de posición desarrollado permite un mayor control del tráfico de la red, ya que los dispositivos pueden conocer su posición futura en un tiempo t por lo que se puede controlar el instante en que los mensajes de actualización son enviados.

7.2.2. Sobre la evaluación de desempeño

Teniendo definida la propuesta del nuevo protocolo de enrutamiento (VGHGR) y los diferentes escenarios, se procedió a realizar la evaluación de cada uno de los escenarios en el simulador NS-3.

- A pesar de contar con simuladores como NS-3, los cuales tienen una gran variedad de capacidades, realizar la evaluación de un sistema de comunicaciones es complicado, debido a las diferentes tecnologías que lo conforman. La cantidad de parámetros a considerar para las evaluaciones tuvieron que ser reducidos considerablemente. Aun así, esto da inicio a una serie de evaluaciones futuras, considerando ya la movilidad de los dispositivos en la red.

- Las características en el diseño del protocolo VGHGR permiten obtener mejoras visibles en los resultados presentes en cada uno de los escenarios de evaluación. Por ejemplo, en la evaluación del consumo de energía, el peor resultado de nuestra propuesta es comparable con el mejor resultado de cada uno de los protocolos con los que se le comparó. Asímismo, el consumo de memoria para almacenamiento fue mucho mejor que los protocolos proactivos y reactivos.

- Suponíamos que a mayor número de dispositivos en la red menor sería la cantidad de celdas vacías, por lo que las consultas al servicio de localización siempre serían respondidas. Sin embargo, los resultados de evaluación de escalabilidad muestran una caída del 21 % en la tasa de entrega de paquetes en la red con un mayor número de dispositivos. Aún así se mantiene como el segundo mejor protocolo evaluado en cuestión de escalabilidad.

- En la garantía de entrega de paquetes, nuestra propuesta se mostró como un protocolo competitivo con un porcentaje superior al 80 %, el cual se ve afectado por el tamaño de los paquetes. La evaluación del throughput muestra un porcentaje superior al 95 %, por lo que, el protocolo no presenta problemas al aumentar los dispositivos origen transmitiendo simultáneamente.

- Los tiempos de entrega registrados por nuestra propuesta nos permite realizar un análisis rápido. Para calcular la velocidad de entrega de los paquetes contamos con la siguiente información: la distancia máxima entre dos dispositivos, que de acuerdo al espacio utilizado en las simulaciones es de 1000 metros y el tiempo de entrega máximo registrado por nuestro protocolo fue de un segundo. Si un dispositivo envía un paquete de extremo a extremo, el paquete viajará a una velocidad de 1000 m/s hacia el destino. Esta información nos permite concluir que siempre que los dispositivos se muevan a una velocidad menor a la calculada, no deberá afectar a la entrega de los paquetes. Esta velocidad no considera el tráfico extra generado por la movilidad de los dispositivos o lo poco confiable del medio inalámbrico. La información sobre velocidad presentada es muy importante para el diseño de pruebas para la evaluación futura de nuestra propuesta en un ambiente de alta movilidad. Ya que nos dice que es necesario desarrollar un mecanismo que controle la sobrecarga de la red generada por la movilidad de los dispositivos, para que no se vea afectada la tasa de entrega de los paquetes.

7.3. Trabajo a futuro

Una vez cumplidos los objetivos planteados inicialmente, se considera que aún existen oportunidades de desarrollo en el área de protocolos de enrutamiento para redes ad hoc móviles. A continuación, se describen algunas recomendaciones relacionadas con este trabajo de investigación que pueden llevarse a cabo en el futuro.

- Evaluar el consumo de energía variando el intervalo de tiempo que los protocolos de enrutamiento geográfico utilizan para enviar periódicamente señales beacons.
- Para la implementación en un ambiente de alta movilidad, se puede desarrollar un mecanismo que disminuya la cantidad de recursos necesarios para mantener el servicio de localización.
- Crear un mecanismo para el balanceo de carga entre dispositivos, esto con la finalidad de disminuir la probabilidad de colisiones por la elección del siguiente salto aplicando el método voraz.

- Desarrollar una plataforma de simulaciones con múltiples modelos de movilidad y variación de pausa (tiempo de no movilidad por parte de los dispositivos) y velocidad.
- Implementar los mismos escenarios aplicados en este trabajo pero con múltiples modelos de propagación, para tener en cuenta los efectos del canal de comunicación inalámbrico.

Lista de referencias bibliográficas

Basagni, S., Conti, M., Giordano, S., y Stojmenovic, I. (2004). *Mobile ad hoc networking*. John Wiley & Sons.

Bisdikian, C. *et al.* (2001). An overview of the bluetooth wireless technology. *IEEE Commun Mag*, **39**(12): 86–94.

Bondy, J. A. y Murty, U. S. R. (1976). *Graph theory with applications*, Vol. 290. Citeseer.

Bose, P., Morin, P., Stojmenović, I., y Urrutia, J. (2001). Routing with guaranteed delivery in ad hoc wireless networks. *Wireless Networks*, **7**(6): 609–616.

Bruno, R., Conti, M., y Gregori, E. (2005). Mesh networks: commodity multihop ad hoc networks. *IEEE Communications Magazine*, **43**(3): 123–131.

Carroll, A. y Heiser, G. (2010). An analysis of power consumption in a smartphone. En: *USENIX annual technical conference*. Boston, MA, Vol. 14.

Chiasserini, C., Chlamtac, I., Monti, P., y Nucci, A. (2002). Optimal energy design of wireless ad hoc networks. *Lecture Notes in Computer Science*, **2345**.

Chiasserini, C.-F. y Rao, R. R. (1999). Pulsed battery discharge in communication devices. En: *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*. ACM, pp. 88–95.

Chlamtac, I., Conti, M., y Liu, J. J.-N. (2003). Mobile ad hoc networking: imperatives and challenges. *Ad Hoc Networks*, **1**(1): 13 – 64.

Conti, M. y Giordano, S. (2007a). Multihop ad hoc networking: The theory. *IEEE Communications Magazine*, **45**(4): 78–86.

Conti, M. y Giordano, S. (2007b). Multihop ad hoc networking: The reality. *IEEE Communications Magazine*, **45**(4): 88–95.

Conti, M. y Giordano, S. (2014). Mobile ad hoc networking: milestones, challenges, and new research directions. *IEEE Communications Magazine*, **52**(1): 85–96.

Corson, M. S., Macker, J. P., y Cirincione, G. H. (1999). Internet-based mobile ad hoc networking. *IEEE internet computing*, **3**(4): 63–70.

Crow, B. P., Widjaja, I., Kim, L. G., y Sakai, P. T. (1997). Ieee 802.11 wireless local area networks. *IEEE Communications Magazine*, **35**(9): 116–126.

Das, S. M., Pucha, H., y Hu, Y. C. (2005). Performance comparison of scalable location services for geographic ad hoc routing. En: *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, March. Vol. 2, pp. 1228–1239 vol. 2.

Frey, H. y Stojmenovic, I. (2006). On delivery guarantees of face and combined greedy-face routing in ad hoc and sensor networks. En: *Proceedings of the 12th annual international conference on Mobile computing and networking*. ACM, pp. 390–401.

Haas, Z. J., Pearlman, M. R., y Samar, P. (2002). The zone routing protocol (zrp) for ad hoc networks.

Jacquet, P., Muhlethaler, P., Clausen, T., Laouiti, A., Qayyum, A., y Viennot, L. (2001). Optimized link state routing protocol for ad hoc networks. En: *Multi Topic Conference, 2001. IEEE INMIC 2001. Technology for the 21st Century. Proceedings. IEEE International*. pp. 62–68.

Jha, R. K. y Kharga, P. (2015). A comparative performance analysis of routing protocols in manet using ns3 simulator. *International Journal of Computer Network and Information Security*, **7**(4): 62.

Johnson, D. B. y Maltz, D. A. (1996). *Dynamic Source Routing in Ad Hoc Wireless Networks*, pp. 153–181. Springer US, Boston, MA.

Karp, B. y Kung, H. T. (2000). Gpsr: Greedy perimeter stateless routing for wireless networks. En: *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, New York, NY, USA. ACM, MobiCom '00, pp. 243–254.

Kranakis, E., Singh, H., y Urrutia, J. (1999). Compass routing on geometric networks. En: *IN PROC. 11 TH CANADIAN CONFERENCE ON COMPUTATIONAL GEOMETRY*. pp. 51–54.

Kuhn, F., Wattenhofer, R., y Zollinger, A. (2003). Worst-case optimal and average-case efficient geometric ad-hoc routing. En: *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking &Amp; Computing*, New York, NY, USA. ACM, MobiHoc '03, pp. 267–278.

Leong, B., Mitra, S., y Liskov, B. (2005). Path vector face routing: geographic routing with local face information. En: *13TH IEEE International Conference on Network Protocols (ICNP'05)*, Nov. pp. 12 pp.–.

Liu, C. y Wu, J. (2006). Swing: Small world iterative navigation greedy routing protocol in manets. En: *Proceedings of 15th International Conference on Computer Communications and Networks*, Oct. pp. 339–350.

Liu, J., Priyantha, B., Hart, T., Ramos, H. S., Loureiro, A. A. F., y Wang, Q. (2012). Energy efficient gps sensing with cloud offloading. En: *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, New York, NY, USA. ACM, SenSys '12, pp. 85–98.

Narra, H., Cheng, Y., Cetinkaya, E. K., Rohrer, J. P., y Sterbenz, J. P. (2011). Destination-sequenced distance vector (dsdv) routing protocol implementation in ns-3. En: *Proceedings of the 4th International ICST Conference on Simulation Tools and Techniques*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), pp. 439–446.

Niculescu, D. y Nath, B. (2003). Localized positioning in ad hoc networks. En: *Sensor Network Protocols and Applications, 2003. Proceedings of the First IEEE. 2003 IEEE International Workshop on*, May. pp. 42–50.

Perkins, C. E. y Bhagwat, P. (1994). Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers. *SIGCOMM Comput. Commun. Rev.*, **24**(4): 234–244.

Perkins, C. E. y Royer, E. M. (1999). Ad-hoc on-demand distance vector routing. En: *Proceedings of the Second IEEE Workshop on Mobile Computer Systems and Applications*, Washington, DC, USA. IEEE Computer Society, WMCSA '99, pp. 90–.

Ratnasamy, S., Francis, P., Handley, M., Karp, R., y Shenker, S. (2001). *A scalable content-addressable network*, Vol. 31. ACM.

Rey, L. C., Quiñones, T. O. L., y García, W. B. (2014). Protocolos de enrutamiento aplicables a redes manet. *Revista Telem@ tica*, **13**(3): 59–74.

Robinson, R. M. (1954). Mersenne and fermat numbers. *Proceedings of the American Mathematical Society*, **5**(5): 842–846.

Rowstron, A. y Druschel, P. (2001). Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. En: *IFIP/ACM International Conference on Distributed Systems Platforms and Open Distributed Processing*. Springer, pp. 329–350.

Royer, E. M. y Toh, C.-K. (1999). A review of current routing protocols for ad hoc mobile wireless networks. *IEEE personal communications*, **6**(2): 46–55.

Singla, S. y Jain, S. (2014). Comparison of routing protocols of manet in real world scenario using ns3. En: *Control, Instrumentation, Communication and Computational Technologies (ICCICCT), 2014 International Conference on*. IEEE, pp. 543–549.

Stoica, I., Morris, R., Karger, D., Kaashoek, M. F., y Balakrishnan, H. (2001). Chord: A scalable peer-to-peer lookup service for internet applications. *ACM SIGCOMM Computer Communication Review*, **31**(4): 149–160.

Tejeda, H. (2005). *El Grafo Virtual para Ruteo Geométrico en una Red Inalámbrica Ad-Hoc*. Tesis de doctorado, Universidad Michoacana de San Nicolás de Hidalgo.

Tejeda, H., Chávez, E., Sanchez, J. A., y Ruiz, P. M. (2006). *A Virtual Spanner for Efficient Face Routing in Multihop Wireless Networks*, pp. 459–470. Springer Berlin Heidelberg, Berlin, Heidelberg.

Zhao, B. Y., Kubiatowicz, J., Joseph, A. D., *et al.* (2001). Tapestry: An infrastructure for fault-tolerant wide-area location and routing.

Zhuang, Z., Kim, K.-H., y Singh, J. P. (2010). Improving energy efficiency of location sensing on smartphones. En: *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*, New York, NY, USA. ACM, MobiSys '10, pp. 315–330.