

INSTITUTO POLITÉCNICO NACIONAL
CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN



**“Enrutamiento Multicast Escalable para Redes Móviles
Ad-hoc utilizando regiones de interés”**

T E S I S

**QUE PARA OBTENER EL GRADO DE
MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN**

P R E S E N T A:

ÁNGEL IVÁN GUTIÉRREZ SÁNCHEZ

DIRECTORES DE TESIS:

M. en C. Sergio Sandoval Reyes
y
Dr. Miguel Jesús Torres Ruiz

MÉXICO D.F, JUNIO, 2015



INSTITUTO POLITÉCNICO NACIONAL SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REVISIÓN DE TESIS

En la Ciudad de México, D.F. siendo las 10:00 horas del día 31 del mes de octubre de 2014 se reunieron los miembros de la Comisión Revisora de la Tesis, designada por el Colegio de Profesores de Estudios de Posgrado e Investigación del:

Centro de Investigación en Computación

Enrutamiento Multicast Escalable para Redes Móviles Ad-hoc utilizando regiones de interés

Presentada por el alumno:

GUTIÉRREZ

Apellido paterno

SÁNCHEZ

Apellido materno

ÁNGEL IVÁN

Nombre(s)

Con registro:

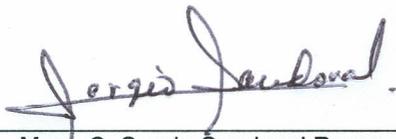
A	1	0	0	3	2	3
---	---	---	---	---	---	---

aspirante de: **MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN**

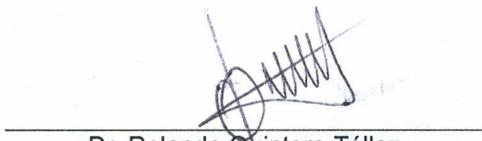
Después de intercambiar opiniones los miembros de la Comisión manifestaron **APROBAR LA TESIS**, en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias vigentes.

LA COMISIÓN REVISORA

Directores de Tesis


 M. en C. Sergio Sandoval Reyes


 Dr. Miguel Jesús Torres Ruiz


 Dr. Rolando Quintero Téllez


 Dr. Rolando Menchaca Méndez


 M. en C. Germán Téllez Castillo


 M. en C. Sandra Dinora Orantes Jiménez

PRESIDENTE DEL COLEGIO DE PROFESORES


 Dr. Luis Alfonso Villa Vargas

ESTADOS UNIDOS MEXICANOS
 INSTITUTO POLITÉCNICO NACIONAL
 CENTRO DE INVESTIGACIÓN
 EN COMPUTACIÓN
 DIRECCIÓN



INSTITUTO POLITÉCNICO NACIONAL
SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

CARTA CESIÓN DE DERECHOS

En la Ciudad de México el día 21 del mes Marzo del año 2015, el (la) que suscribe Ángel Iván Gutiérrez Sánchez alumno (a) del Programa de Mestría en Ciencias de la Computación con número de registro A100323, adscrito al Centro de Investigación en Computación, manifiesta que es autor (a) intelectual del presente trabajo de Tesis bajo la dirección de M en C. Sergio Sandoval Reyes y Dr. Miguel Jesús Torres Ruiz y cede los derechos del trabajo intitulado Enrutamiento Multicast Escalable para Redes Móviles Ad-hoc utilizando regiones de interés, al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del autor y/o director del trabajo. Este puede ser obtenido escribiendo a la siguiente dirección ivan.gutierrez.s@gmail.com. Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.

Nombre y firma

RESUMEN

Las redes móviles ad hoc, también conocidas por sus siglas en inglés como MANET (mobile ad hoc network) han tenido un crecimiento importante con el paso del tiempo. Este crecimiento se debe al mejoramiento de los dispositivos móviles. Existen varios protocolos en el ámbito de las MANETs, que inundan la red con mensajes de control para establecer sus rutas de la fuente al destino. Pero no son escalables.

En esta tesis utilizo el concepto de las regiones de interés. Una región de interés o enclave para un flujo multicast o unicast es un componente conectado de red que contiene aquellos nodos relevantes para la disseminación de información de un flujo, ya sean receptores, fuentes o nodos de reenvío que se encuentran localizados entre las fuentes y los receptores. El protocolo propuesto en esta tesis SSMRP, utiliza regiones de interés.

Se realizaron experimentos que prueban el desempeño de SSMRP y lo comparan con ODMRP, el protocolo estándar para enrutamiento multicast en MANET's. Los resultados muestran que el promedio de entrega de paquetes es similar o mayor al de ODMRP.

Estos resultados también muestran que el tráfico de control generado por SSMRP, el cual es necesario para establecer y mantener las regiones de interés, es menor que el tráfico que ODMRP genera para establecer y dar mantenimiento a la malla. También pudimos comprobar que SSMRP hace un uso más eficiente del ancho de banda, pues el total de bytes transmitidos por SSMRP para la entrega de sus paquetes es menor al de ODMRP.

ABSTRACT

The ad hoc mobile networks, also known like MANET (mobile ad hoc network) have grown significantly over time. This growth is due to the improvement of mobile devices. There are several protocols in the field of MANETs, flooding the network with control messages to establish routes from source to destination. But they are not scalable.

In this thesis the concept of the regions of interest is used. A region of interest or enclave for a multicast or unicast stream is a connected network component, that contains those relevant nodes for the dissemination of information flow, whether receptors, sources or forwarding nodes that are located between the sources and receptors. SSMRP, the protocol proposed in this thesis uses regions of interest.

Several experiments were performed to test the performance of SSMRP and compare it with ODMRP, the standard protocol for multicast routing in MANETs. The results showed that the average packet delivery ratio is similar to or greater than ODMRP.

These results also showed that the control traffic generated by SSMRP, which is necessary for establishing and maintaining the regions of interest, is less than the traffic that ODMRP generates to establish and maintain the mesh. We could also verify that SSMRP makes more efficient use of the bandwidth because the total bytes transmitted by SSMRP for the delivery of their packages, were lower than ODMRP.

AGRADECIMIENTOS

A Dios y la Virgen de Guadalupe.

Al Consejo Nacional de Ciencia y Tecnología (CONACYT) por la beca otorgada durante mis estudios de maestría en el Centro de Investigación en Computación del Instituto Politécnico Nacional.

Al Instituto Politécnico Nacional.

Al Centro de Investigación en Computación del Instituto Politécnico Nacional, por proporcionar las bases y guía para la realización de este trabajo.

A mi asesor el M. en C. Sergio Sandoval por su guía y apoyo durante el desarrollo de esta tesis.

Al Dr. Miguel Torres por su incomparable ayuda.

Al Dr. Rolando Menchaca Méndez por toda su dirección en el desarrollo del protocolo.

A mis amigos y compañeros con los que compartí esta experiencia.

A mi familia y mi novia por todo su apoyo incondicional durante toda la maestría, este logro es para ustedes.

Índice

RESUMEN	i
ABSTRACT	ii
AGRADECIMIENTOS.....	iii
Lista de Figuras.....	vii
Lista de Tablas	viii
Capítulo 1	1
REDES MANET Y PROTOCOLOS DE ENRUTAMIENTO MULTICAST	1
1.1 Introducción a las redes MANET	1
1.2 Protocolos de enrutamiento multicast	2
1.2.1 El protocolo ODMRP.....	3
1.3 Planteamiento del problema.	4
1.3.1 Extensiones de ODMRP	5
1.4 Justificación.	5
1.5 Objetivos	5
1.5.1 Objetivo general.....	5
1.5.2 Objetivos específicos.....	6
1.6 Organización de la tesis.....	6
Capítulo 2	7
ESTADO DEL ARTE: TRABAJOS RELACIONADOS CON PROTOCOLOS DE ENRUTAMIENTO MULTICAST	7
2.1 Protocolos basados en mallas	7
2.1.1 NSMP. Neighbor-Supporting Ad Hoc Multicast Routing Protocol.	7
2.1.1.1 Creación de la malla	7
2.1.1.2 Descubrimiento de rutas locales.....	8
2.1.1.3 Descubrimiento de rutas por inundación	8
2.1.2 DCMP. Dynamic Core Based Multicast Routing Protocol	9
2.1.2.1 Establecer la ruta.....	9
2.1.2.2 Ejemplo.....	10
2.1.3 PUMA. Protocol for Unified Multicasting through Announcements	11
2.2 Protocolos basados en árboles	13

2.2.1 AMRoute. Ad hoc Multicast Routing.....	13
2.2.2 AMRIS. Ad hoc Multicast Routing protocol utilizing Increasing id-numberS	13
2.3 Protocolos Híbridos	13
2.3.1 MMARP. Multicast MANet Routing Protocol.	13
2.4 Propuesta de solución	15
Capítulo 3	17
DISEÑO DEL ALGORITMO SSMRP	17
3.1 Descripción del algoritmo	17
3.2 Región de interés	18
3.3 Reenvío de datos.....	19
3.4 Flexibilidad	19
3.5 Selección de los valores del temporizador.....	20
3.6 Enrutamiento Unicast	20
3.7 Estructura de datos	20
3.7.1 El protocolo SSMRP contiene las siguientes estructuras de datos	20
3.8 Criterio de selección de ruta	21
3.9 Operación	21
3.9.1 Reenvío de grupo	21
3.9.1.1 Generando un <i>JoinQuery</i>	21
3.9.1.2 Procesando un <i>JoinQuery</i>	22
3.9.1.3 Generando un <i>Join Reply</i>	22
3.9.1.4 Cuando se recibe un <i>Join Reply</i>	22
Capítulo 4	24
IMPLEMENTACIÓN DEL ALGORITMO SSMRP.....	24
4.1 El simulador de red NS2	24
4.2 El protocolo SSMRP	24
4.3 Corridas del simulador.	26
4.4 Implementación de algoritmo Enclave.....	27
Capítulo 5	30
PRUEBAS Y RESULTADOS.....	30
5.1 Pruebas.....	30
5.2 Desempeño con tráfico multicast	32

5.2.1 Incrementando el número de fuentes	32
5.3 Discusión	35
Capítulo 6	36
CONCLUSIONES	36
6.1 Logros alcanzados	36
6.2 Trabajo a futuro.....	37
6.3 Aportaciones	38
Bibliografía	39
ANEXO : Glosario de términos	41

Lista de Figuras

Figura 1.1 MANET	1
Figura 1.2 Malla de una MANET.....	2
Figura 1.3 Árbol de una MANET.....	3
Figura 1.4 Procedimiento bajo demanda.....	4
Figura 2.1 Creación de la malla multicast	8
Figura 2.2 Topología de Malla	10
Figura 2.3 Diseminación de anuncios multicast	12
Figura 2.4 Red Híbrida.....	14
Figura 3.1 Algoritmo Región de Interés	19
Figura 4.1 Clase ssmrp.cc	25
Figura 4.2 Simulación 1 de NS2.....	26
Figura 4.3 Simulación 2 de NS2.....	27
Figura 5.1 Desempeño al incrementar el número de fuentes. PacketDelivery Ratio	32
Figura 5.2 Desempeño al incrementar el número de fuentes. EndtoEndDelay	33
Figura 5.3 Desempeño al incrementar el número de fuentes. Control Overhead	34
Figura 5.4 Desempeño al incrementar el número de fuentes. Total Overhead	34

Lista de Tablas

Tabla 4.1 Entorno de simulación.....	24
Tabla 5.1 Escenario de simulación.....	32

Capítulo 1

REDES MANET Y PROTOCOLOS DE ENRUTAMIENTO MULTICAST

1.1 Introducción a las redes MANET

Las redes móviles ad hoc, también conocidas por sus siglas en inglés como MANET (mobile ad hoc network) han tenido un crecimiento importante con el paso del tiempo. Este crecimiento se debe al mejoramiento de los dispositivos móviles tales como laptops y celulares. Asimismo la duración de la batería es más prolongada y como resultado los usuarios pueden moverse con sus dispositivos móviles a lo largo de un área por más tiempo, y sin la necesidad de estar conectados a la corriente y utilizar cables para tener acceso a la red. La Figura 1.1 muestra lo que es una MANET. No existen cables para conectarse entre los nodos (los puntos negros en el centro de los círculos), la conexión se hace de forma inalámbrica y los caminos entre un nodo y otro están dados por saltos (*hop by hop*) entre cada uno de ellos.

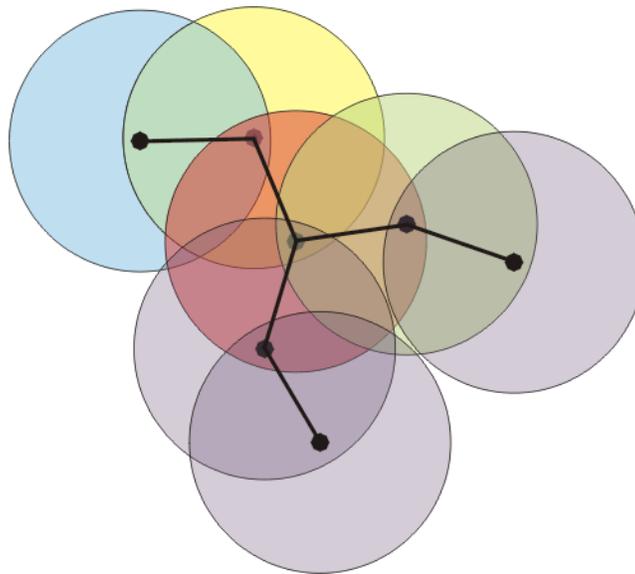


Figura 1.1 Una red MANET

Al tratarse de redes que no requieren una infraestructura fija (como Ethernet), las redes MANET son de vital importancia, ya que son fácilmente desplegables. Son útiles en entornos donde sea difícil instalar una red fija o el costo de implementación sea muy alto. Algunas de las aplicaciones en las que se emplean las redes MANET son las siguientes [1]:

Entornos militares: Las redes MANET permiten establecer comunicación entre distintas unidades, personas, tanques, y centros de mando sin necesidad de establecer una infraestructura fija, lo cual podría ser imposible.

Situaciones de emergencia: En momentos de desastres naturales, los equipos de emergencia, rescate o salvamento necesitan establecer comunicación entre ellos para una buena coordinación. Desplegar una red MANET es una solución rápida y eficiente para lograr este propósito.

Entornos civiles: En este tipo de entornos las aplicaciones para las MANET es muy amplia. Se pueden crear redes MANET para compartir información entre los participantes de un congreso, una conferencia, una clase. Otros lugares pueden ser estadios, aeropuertos, restaurantes, museos, centros comerciales, etc.

Redes de área local. Las redes LAN las cuales conectan dispositivos por medio de un cableado estructurado pueden ahora conectarse a través de una red MANET.

1.2 Protocolos de enrutamiento multicast

En particular, los protocolos de enrutamiento multicast para MANETs tienen como objetivo establecer comunicación desde un nodo origen a un grupo específico de nodos destino. Estos protocolos están basados en mallas (Figura 1.2) o en árboles (Figura 1.3), en donde el origen es el nodo "h" y el grupo destino está formado por los nodos sombreados "d, z, b, g, f, j, l y p". Nótese que los caminos de multidifusión son diferentes en ambas estructuras.

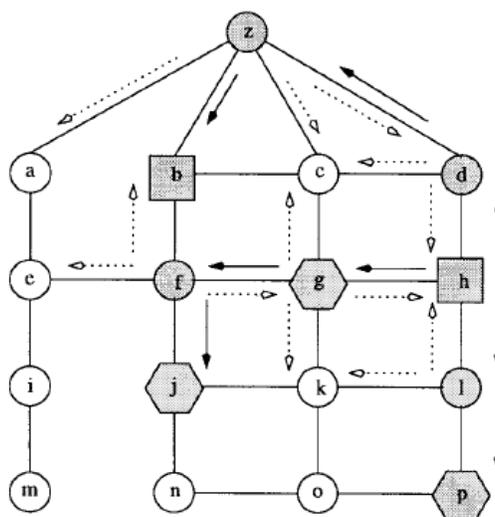


Figura 1.2 Malla de una MANET

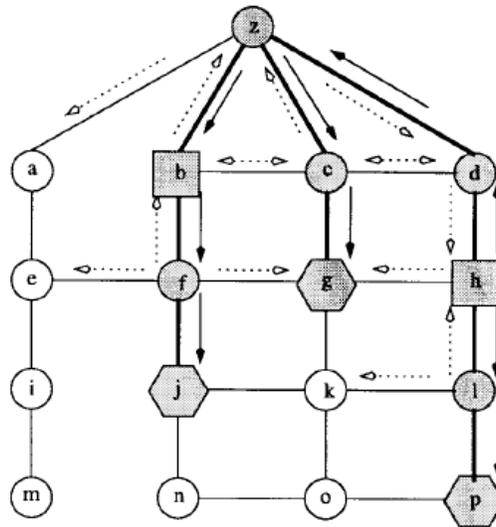


Figura 1.3 Árbol de una MANET

Los que están basados en árboles (Figura 1.3), pueden generar un impacto negativo en la red, ya que en redes móviles los nodos cambian de posición continuamente, lo que genera que la topología de la red sea altamente dinámica. En este contexto las estructuras de árbol tienden a ser demasiado frágiles por lo que los algoritmos de enrutamiento entran continuamente en fases de reparación de los árboles, lo que genera un intercambio de mensajes de control que puede ser excesivo, limitando el ancho de banda y excediendo el uso de batería. Las mallas, en cambio, tienen una topología más resistente a la movilidad de los nodos conectados en la red. Una malla proporciona una mejor conectividad entre miembros de un grupo multicast que un árbol, además de generar caminos más cortos. La Figura 1.2 por ejemplo, muestra el envío de paquetes del nodo **h** al grafo entero en una malla. Si éste quisiera enviar paquetes a **g**, lo haría en un salto, pues **h** es vecino de **g**. En cambio en un árbol el envío de paquetes desde **h** tiene que enviarse primero a **z** el nodo raíz, para que éste a su vez distribuya los paquetes al grafo entero. Es evidente que en el árbol, el camino de **h** a **g** es más largo (cuatro saltos) [11]. Tres ejemplos de protocolos basados en malla son: CAMP, ODMRP y PUMA. El enrutamiento multicast escalable para las MANETs es un tema de investigación, y éste es el propósito de la tesis.

1.2.1 El protocolo ODMRP

En ODMRP (On Demand Multicast Routing Protocol, por sus siglas en inglés) [10], los miembros de un grupo y rutas multicast son establecidas y actualizadas por la fuente bajo demanda. Esto quiere decir que mientras una fuente tenga paquetes que enviar (nodo "S" de la Figura 1.4), ésta emite periódicamente por la red entera paquetes al grupo multicast destino (nodos receptores "R" de la Figura 1.4).

En este enrutamiento primero se debe establecer la ruta al grupo multicast destino. Esto se hace por medio de paquetes JOIN REQUEST y JOIN REPLY. Inicialmente el protocolo ODMRP va inundando la red con paquetes JOIN REQUEST, y cuando un nodo recibe este paquete (que no sea duplicado) guarda el ID en dirección a la fuente ("next host") y retransmite el paquete. Cuando un paquete JOIN REQUEST finalmente alcanza a un receptor del grupo multicast, el receptor crea o actualiza la entrada de la fuente en su tabla de miembros. Mientras existan entradas válidas en la tabla de miembros, los nodos "R" emiten mensajes JOIN TABLES periódicamente a los vecinos (Ver Figura 1.4). Cuando un nodo recibe un mensaje JOIN TABLE, éste verifica el ID del próximo nodo para saber si concuerda con su propio ID (de "next host"). Si es así, el nodo se da cuenta que está en el camino a la fuente y forma parte del grupo de reenvío. Así la JOIN TABLE es propagada por cada miembro del grupo de reenvío hasta que finalmente alcanzan la fuente multicast vía el camino más corto. De esta forma se establecen las rutas multicast bajo demanda.

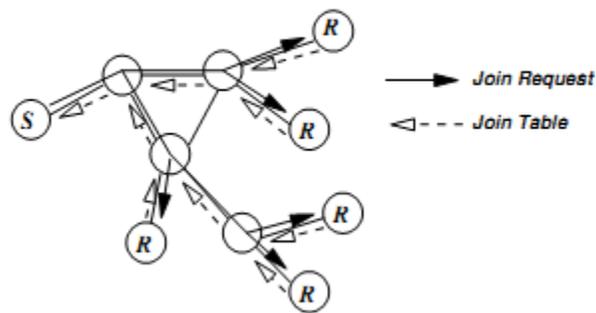


Figura 1.4 Establecimiento de rutas multicast bajo demanda

1.3 Planteamiento del problema.

Una de los principales inconvenientes de ODMRP es el número de veces que tiene que inundar la red con mensajes JOIN REQUEST. Estas inundaciones son altamente costosas y hacen que ODMRP no sea un protocolo escalable en términos del número de flujos multicast que soporta. Se han presentado múltiples propuestas para resolver este problema (por ejemplo, los protocolos DCMP, NSMP y MMARP que se describen a continuación), sin embargo todas ellas son soluciones parciales que no eliminan o reducen de manera importante el número de operaciones de inundación que se emplean para establecer las estructuras de enrutamiento.

1.3.1 Extensiones de ODMRP

Varias extensiones de ODMRP han sido propuestas. DCMP (Dynamic Core Multicast Protocol) [7] designa ciertos transmisores como núcleos y reduce el número de transmisores que realizan inundación. NSMP (Neighbor Supporting ad hoc Multicast routing Protocol) [4] reduce la inundación de paquetes de control a un subconjunto de la red entera. Sin embargo estos dos protocolos no eliminan completamente que múltiples nodos realicen inundación de paquetes de control para cada grupo.

MMARP (Mesh-based Multicast Ad Hoc Routing Protocol) [5] que es otra extensión de ODMRP construye mallas multicast como la unión de un conjunto de árboles que se aproximan a árboles de Steiner con raíces en cada fuente. Los autores reportan una mejora del 40-50% en eficiencia de reenvío con respecto a ODMRP; sin embargo, la saturación de paquetes de control es del mismo orden que ODMRP [10].

Dado los problemas mencionados anteriormente nos proponemos lograr un protocolo como se especifica en el objetivo general.

1.4 Justificación.

Como se indicó anteriormente el enrutamiento multicast juega un papel importante en las redes móviles ad hoc. Además de tener en cuenta la importancia de la escalabilidad, el crecimiento del tráfico en la red puede perjudicar el rango de entrega si el protocolo de enrutamiento no es escalable. Por eso es sobresaliente el concepto de *Región de interés*.

Una región de interés o enclave para un flujo multicast o unicast es un componente conectado de red que contiene aquellos nodos relevantes para la diseminación de información de un flujo, ya sean receptores, fuentes o nodos de reenvío que se encuentran localizados entre las fuentes y los receptores.

Un protocolo con estas características puede aplicarse en dispositivos móviles ya sean teléfonos celulares o laptops para que exista comunicación sin necesidad de una red centralizada. Puede utilizarse en aeropuertos, salas de conferencias o en campus universitarios o también en desastres naturales.

1.5 Objetivos

1.5.1 Objetivo general

Diseñar un protocolo de enrutamiento multicast escalable, que sea iniciado por el transmisor (sender initiated) y que utilice regiones de interés.

1.5.2 Objetivos específicos

- 1.- Analizar protocolos de enrutamiento para MANETs.
- 2.- Implementar el protocolo propuesto en un simulador de redes.
- 3.- Diseñar una serie de experimentos para evaluar el desempeño del protocolo propuesto.
- 4.- Analizar e interpretar los resultados.

1.6 Organización de la tesis.

En el capítulo 2 se describen trabajos relacionados con protocolos de enrutamiento multicast basados en mallas y basados en árboles. También se indica nuestra propuesta de solución. En el capítulo 3 se presenta el diseño del algoritmo de nuestro protocolo de enrutamiento multicast escalable en regiones de interés. En el capítulo 4 se describe la implementación del protocolo en el simulador de redes NS2 y los recursos en hardware utilizados. En el capítulo 5 se describen las pruebas realizadas a nuestro protocolo y se reportan los resultados obtenidos. En el capítulo 6 se presentan nuestras conclusiones, aportaciones y mejoras a futuro. En la Bibliografía se presenta una lista de las referencias recopiladas durante la elaboración de la tesis. Por último, en el Anexo se incluye un Glosario de términos.

Resumen

En este capítulo se formuló el planteamiento del problema, los objetivos y la justificación que sustentan esta tesis.

En el siguiente capítulo se describen los protocolos relacionados con el estado del arte, el estándar de los protocolos multicast: ODMRP, y nuestra propuesta de solución.

Capítulo 2

ESTADO DEL ARTE: TRABAJOS RELACIONADOS CON PROTOCOLOS DE ENRUTAMIENTO MULTICAST

En este capítulo se describen protocolos relacionados con el protocolo multicast ODMRP, y nuestra propuesta de solución.

2.1 Protocolos basados en mallas

2.1.1 NSMP. Neighbor-Supporting Ad Hoc Multicast Routing Protocol.

El protocolo NSMP utiliza la estructura basada en mallas y realiza enrutamiento local para reducir la saturación de mensajes de control y mantenimiento del grupo multicast. Este protocolo tiene dos formas de descubrir rutas: descubrimiento local y descubrimiento por inundación. El enrutamiento local es restringido a un grupo pequeño de nodos que están directamente relacionados con el grupo multicast. El enrutamiento por inundación se realiza para establecer una ruta inicial o reparar un enlace, difundiendo mensajes de control a toda la red [4].

NSMP utiliza la eficiencia de la estructura de árboles para la transmisión de paquetes y la estructura de mallas para mejorar la resiliencia.

2.1.1.1 Creación de la malla

Para crear la malla multicast una fuente inicialmente envía un paquete FLOOD_REQ. Este paquete contiene un campo llamado *upstream*, que es el punto previo en la cadena de transmisiones. Cuando un nodo intermedio recibe un paquete FLOOD_REQ, éste almacena la dirección del nodo anterior (up stream node), que es el nodo que está en dirección a la fuente y actualiza el campo con su propia dirección antes de reenviar el paquete al siguiente nodo. Cuando el paquete llega al receptor, éste envía un paquete respuesta REP al nodo del cual recibió el paquete. El nodo que recibe el paquete REP se suma al grupo y actualiza su tabla de enrutamiento.

Éste entonces reenvía el paquete REP al nodo que está hacia la fuente, y así el paquete REP eventualmente alcanza la fuente. Los nodos intermedios que retransmiten el paquete REP son llamados nodos de reenvío (forwarding nodes). La malla multicast de un grupo consiste de fuentes, receptores, nodos de reenvío y los enlaces que los conectan. Los nodos en una malla multicast son llamados nodos malla. La elección del líder del grupo, se escoge de entre todos los nodos fuente del mismo grupo multicast y es el nodo fuente que tiene la dirección más chica. Solo el líder del grupo puede hacer inundación. En la Figura 2.1 se muestra lo descrito.

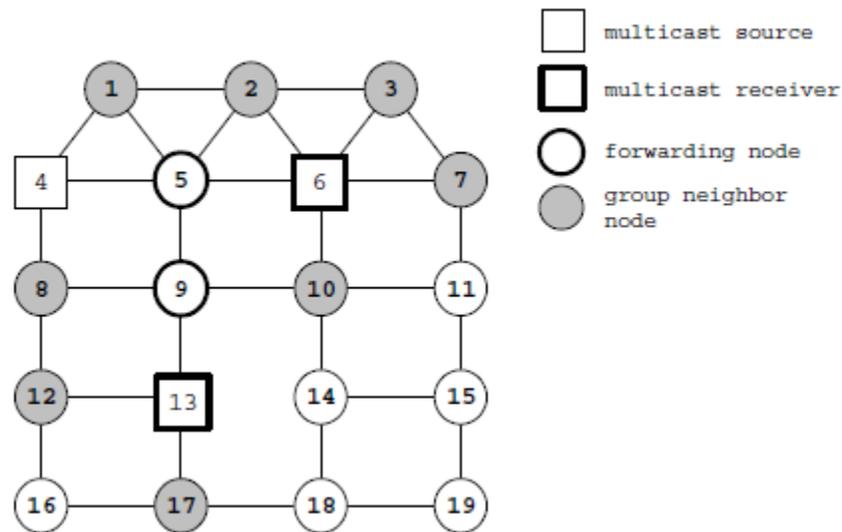


Figura 2.1 Creación de la malla multicast

2.1.1.2 Descubrimiento de rutas locales

Cada fuente transmite periódicamente un paquete LOCAL_REQ, y solamente nodos malla y nodos vecinos pueden retransmitir el paquete. Así, todos los nodos a dos saltos de distancia de nodos malla reciben el paquete LOCAL_REQ. Este mecanismo reduce la saturación de mensajes de control. Nodos de reenvío y nodos vecinos pertenecientes al grupo son actualizados a través de paquetes REP que son retransmitidos a la fuente.

2.1.1.3 Descubrimiento de rutas por inundación

NSMP realiza inundación en varios casos. Cuando un nodo se convierte en nodo fuente, éste envía un paquete FLOOD_REQ para crear una red inicial. Para NSMP, un nodo dentro de dos saltos de distancia de un nodo malla puede unirse al grupo como receptor respondiendo con un paquete LOCAL_REQ. Sin embargo, un nodo a más de dos saltos de distancia de un nodo malla debe hacer inundación a través de un paquete MEM_REQ. Adicionalmente, cuando ocurren particiones en una red solo puede ser recuperada con paquetes FLOOD_REQ [4].

2.1.2 DCMP. Dynamic Core Based Multicast Routing Protocol

El protocolo DCMP construye y mantiene una malla compartida, la cual es formada por un grupo de núcleos basados en árboles [7]. DCMP utiliza las ventajas de los núcleos basados en árboles para mejorar la escalabilidad del protocolo. CBT (Core-Based Trees, por sus siglas en inglés) es una propuesta para la construcción de árboles de enrutadores para grupos multicast escalables [6]. La diferencia de otros esquemas multicast es que el enrutamiento de árbol comprende varios núcleos (también llamados centros). Los enrutadores se pueden ir sumando conforme el crecimiento de las “*ramas*” del árbol, comprendiendo una cadena de enrutadores, desde los enrutadores núcleo hacia los enrutadores directamente conectados a miembros de grupos multicast [6].

En DCMP las fuentes son clasificadas en tres categorías.

- Fuentes activas
- Fuentes núcleo activas
- Fuentes pasivas

Las **fuentes activas** hacen inundación con paquetes de control *JoinReq* a intervalos regulares. Las **fuentes núcleo activas** son fuentes activas que funcionan como núcleo para una o más fuentes pasivas. DCMP utiliza también nodos núcleo, éstos son responsables de crear una malla a favor de las fuentes pasivas, asociadas a estos nodos. Una **fuentes pasiva** no transmite paquetes de control *JoinReq* para la creación de mallas multicast. La fuente pasiva depende de lo cerca que esté de la fuente activa para enviar paquetes. El número máximo de **fuentes pasivas** que puede soportar una **fuentes núcleo activa** está limitado por el parámetro MaxPassSize. La distancia entre estos dos tipos de fuentes se delimita por el parámetro MaxHop. Estos parámetros previenen que un gran número de **fuentes activas** se conviertan en **fuentes pasivas**. Esto ayuda a que la malla sea más robusta. La robustez de la malla depende básicamente del número de **fuentes activas** en el grupo multicast. Conforme este número se incrementa, la malla se vuelve más robusta, debido a la transmisión periódica de paquetes *JoinReq* por cada fuente activa.

2.1.2.1 Establecer la ruta

Cuando un nodo recibe un paquete de control no duplicado *JoinReq*, éste reenvía el paquete después de haber guardado el número de identificación del nodo upstream (ID). Un identificador único es asignado a cada nodo.

Cuando un paquete de control *JoinReq* llega al receptor, éste responde enviando un paquete *Reply* por el camino inverso a la fuente. Cuando nodos intermedios a través del camino inverso a la fuente reciben el paquete *Reply*, revisan el campo *NextNode ID* del paquete. Si el ID del nodo coincide con cualquiera de las entradas del campo, entonces enciende la bandera de reenvío (*FgFlag*) y se convierte en un nodo de reenvío para un grupo multicast particular. Este nodo luego hace un paquete *Reply* y lo difunde. De esta forma se establece la ruta por la transmisión de paquetes *JoinReq* y *Reply*.

Cuando una *Fuente Activa* en un grupo multicast recibe un paquete *JoinReq*, ésta cambia su estado a Fuente Pasiva si cumple las siguientes tres condiciones:

1. Está encendida la bandera *CoreAcceptance*.
2. La distancia recorrida por el paquete *JoinReq* es menor o igual que *MaxHop*.
3. El ID del nodo de la fuente que recibe el paquete *JoinReq* (para ser nombrada después como *Fuente Pasiva*) es menor que el ID del nodo de la fuente que lo envía (para ser nombrada después como *Fuente Núcleo*).

2.1.2.2 Ejemplo

En la figura 2.2, hay cuatro fuentes (nodos S1, S2, S3, y S4) y tres receptores (nodos R), en el grupo multicast. Los parámetros *MaxPassSize* y *MaxHop* tienen valor 1 y 2 respectivamente.

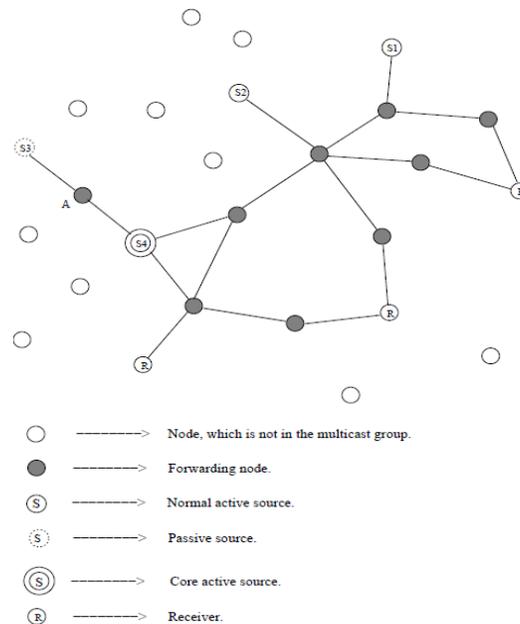


Figura 2.2 Topología de Malla

Inicialmente, para descubrir nodos receptores, cada fuente inunda con paquetes *JoinReq*. Éstos llegarán a los receptores y a las otras fuentes. Cuando los paquetes llegan a los receptores, se realiza el procedimiento descrito anteriormente de establecimiento de ruta.

Cuando la fuente S3 recibe el paquete *JoinReq* de S4, ésta verifica la bandera *CoreAcceptance*. Esta bandera sirve para saber si S4 soporta o no más *Fuentes Pasivas*. Tomando en cuenta que, a) la bandera *CoreAcceptance* está activada en el paquete *JoinReq* (enviado por S4), b) los saltos de distancia hechos por el paquete *JoinReq* son 2, y c) el ID de S3 es menor que el ID de S4, esto satisface las tres condiciones mencionadas anteriormente para el cambio de estado. De esta forma, la fuente S3 envía un paquete *PassReq* a S4. Y así S3 cambia su estado de Fuente Activa a Fuente Pasiva (*ToBePassiveSource*) y S4 comienza a ser el nodo núcleo (*ToBeCoreNode*) para S3. Los nodos núcleos transmiten los paquetes de las Fuentes Pasivas [7].

2.1.3 PUMA. Protocol for Unified Multicasting through Announcements

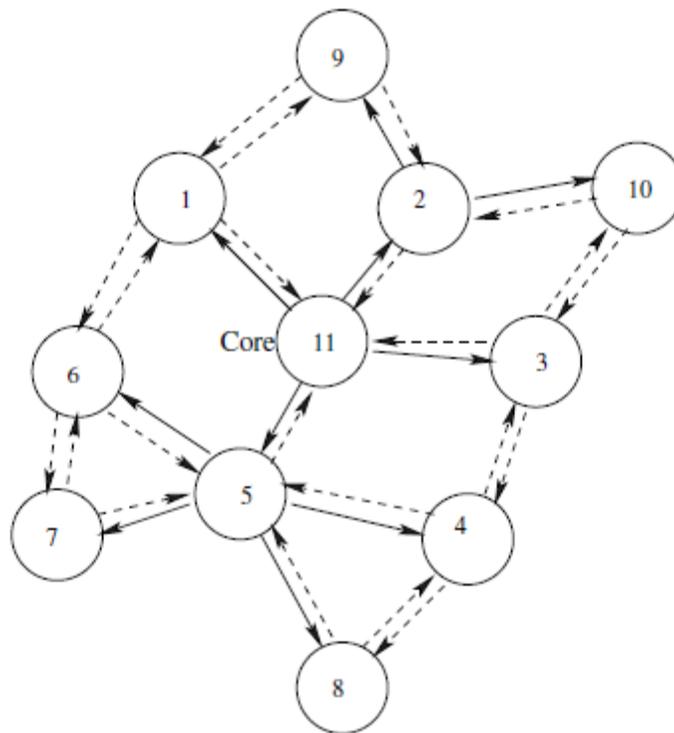
Puma soporta el modelo IP multicast permitiendo a cualquier fuente enviar paquetes multicast direccionado a un grupo dado sin tener que saber cómo está constituido el grupo. Además, las fuentes no necesitan unirse al grupo multicast con el propósito de enviar datos al grupo [8].

Como CAMP [11] y MAODV [12], PUMA usa el modelo *iniciado por el receptor* (receiver-initiated) en el cual el receptor se une al grupo multicast usando la dirección de un nodo especial (núcleo en CAMP o líder del grupo en MAODV), sin la necesidad de hacer multidifusión (flooding) de paquetes de control a la red entera. Como MAODV, PUMA elimina la necesidad de un protocolo de enrutamiento unicast y la pre-asignación de núcleos para grupos multicast.

PUMA implementa un algoritmo distribuido para seleccionar uno de los receptores del grupo como el núcleo del grupo. La elección del algoritmo usado en PUMA es esencialmente el mismo que el algoritmo de árbol de expansión (spanning tree) introducido por Perlman. Dentro de un tiempo finito proporcional al tiempo necesitado para alcanzar al enrutador más alejado del núcleo del grupo, cada enrutador tiene uno o múltiples caminos para seleccionar un núcleo. Cada receptor se conecta al núcleo elegido a través de los caminos más cortos entre el receptor y el núcleo. Todos los nodos en el camino más corto entre cualquier receptor y el núcleo forman colectivamente la malla. Un transmisor envía paquetes de datos al grupo a través del camino más corto entre el transmisor y el núcleo. Cuando el paquete de datos alcanza un miembro de la malla, éste hace inundación dentro de la malla y los nodos mantienen el ID del paquete en cache para descartar los paquetes duplicados.

PUMA usa paquetes de control para todas sus funciones de anuncios multicast (multicast announcements). Cada anuncio multicast especifica un número de secuencia, la dirección de un grupo (group ID), la dirección del núcleo (core ID), la distancia al núcleo, una bandera de miembro de malla que se prende cuando el nodo que envía pertenece a la malla y un padre que es el vecino preferido para alcanzar el núcleo. Los anuncios multicast sucesivos tienen un número de secuencia más alto que previos anuncios multicast enviados por el mismo núcleo. Con la información contenida en estos anuncios, los nodos eligen núcleos, determinan las rutas para las fuentes fuera del grupo multicast al grupo, notifican a otros acerca de uniones o salidas de la malla de un grupo y mantienen la malla del grupo [8].

La Figura 2.3 muestra la difusión de anuncios multicast.



Connectivity List at node 6
Core Id = 11 Group Id = 224.0.0.1, Seq No 79

Neighbor	Multicast Announcement		Time (ms)
	Distance To Core	Parent	
5	1	11	12152
1	1	11	12180
7	2	5	12260

Figura 2.3 Diseminación de anuncios multicast

2.2 Protocolos basados en árboles

2.2.1 AMRoute. Ad hoc Multicast Routing.

AMRoute asume la existencia de un protocolo de enrutamiento unicast en la red. Este protocolo tiene dos fases principales: creación de la malla y creación del árbol. Después de que el núcleo forma la malla, éste crea periódicamente un árbol virtual multicast sobre la malla. Este árbol multicast usa túneles unicast para conectar grupos de miembros. Debido a la malla que yace debajo, no hay necesidad de reajustes frecuentes del árbol, por lo tanto existe resiliencia contra la movilidad [13].

2.2.2 AMRIS. Ad hoc Multicast Routing protocol utilizing Increasing id-numberS

AMRIS es un protocolo bajo demanda, iniciado por la fuente, basado en árboles. En este protocolo, cada nodo en una sesión multicast genera un *multicast sesión member id* (msm-id), después de recibir el mensaje *NEW-SESSION* desde su nodo padre. Este tipo de mensaje es transmitido por un nodo especial llamado *Sid*, en donde el árbol tiene su raíz. El *msm-id* se incrementa desde la raíz hasta los nodo hoja, lo cual indica el flujo de datos. El protocolo envía paquetes periódicos para determinar si hay enlaces rotos. Si encuentra alguno, ejecuta el proceso de reconstrucción de rama (branch reconstruction) para mantener el árbol multicast [14].

2.3 Protocolos Híbridos

2.3.1 MMARP. Multicast MANet Routing Protocol.

El protocolo MMARP tiene como propósito hacer enrutamiento entre las redes móviles ad hoc y las redes fijas, como se muestra en la Figura 2.4

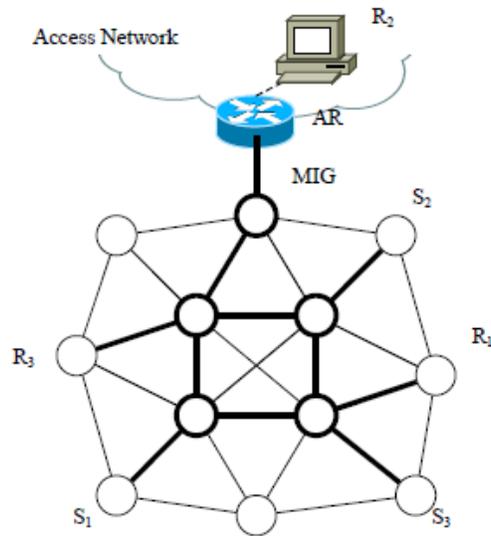


Figura 2.4 Red Híbrida

Aunque este protocolo está principalmente diseñado para las MANETs, es compatible con el modelo multicast estándar de IP. Las direcciones IP multicast se encuentran en el rango de direcciones 224.0.0.0 y 239.255.255.255 y son llamadas direcciones IP de clase D. El enrutamiento se hace en el nivel de la capa de red, permitiendo a los enrutadores crear caminos de distribución óptimos [5].

MMARP permite que nodos que se encuentran en redes alámbricas puedan comunicarse con nodos que se encuentran en una red MANET. Esto se debe a que MMARP soporta el protocolo IGMP (Internet Group Management Protocol). El protocolo IGMP, definido en el RFC 3376, es utilizado por computadoras y enrutadores para intercambiar información sobre pertenencia a los grupos multicast en una LAN. IGMP aprovecha la naturaleza de difusión de las LAN para proporcionar una técnica eficiente para el intercambio de información entre múltiples estaciones y enrutadores. En general, IGMP ofrece dos funciones principales: El envío de mensajes desde las estaciones a los enrutadores para suscribirse y para abandonar grupos multicast definidos por una dirección multicast dada. Y la comprobación periódica de los enrutadores sobre qué grupos multicast interesan a qué estaciones [3].

MMARP es un protocolo que usa estructura basada en mallas, la cual ofrece buena protección contra la movilidad de los nodos. Las partes proactiva y reactiva del protocolo son responsables de la construcción de la malla.

La parte reactiva del protocolo es la que se encarga de realizar el enrutamiento entre los nodos móviles de la red, es decir, en la MANET. Su funcionamiento consiste en la fase de solicitud y respuesta. Cuando un nodo tiene nueva información que enviar, éste emite un mensaje *MMARP_SOURCE* el cual inunda dentro de la red

entera actualizando el estado de los nodos intermediarios y las rutas multicast. Estos mensajes tienen un identificador el cual permite a nodos intermedios detectar duplicados y evitar innecesarias retransmisiones. Cuando un mensaje es recibido por un nodo por primera vez, éste guarda la dirección IP del salto anterior y retransmite el paquete. Cuando uno de esos mensajes llega al receptor o a un vecino del receptor, éste emite un mensaje *MMARP_JOIN* incluyendo la dirección IP del salto anterior hacia la fuente. Cuando un nodo detecta su dirección IP en un mensaje *MMARP_JOIN*, éste reconoce que está en el camino entre la fuente y el destino. Este nodo entonces activa su bandera *MF_FLAG* y retransmite el mensaje *MMARP_JOIN* incluyendo lo almacenado previamente del próximo salto hacia la fuente. De esta forma, el camino multicast más corto es creado entre la fuente y el destino.

En cambio la parte proactiva del protocolo se encarga del enrutamiento en la sección alambrada de la red. La parte proactiva está basada simplemente en los avisos periódicos de los MIGs (Multicast Internet Gateways, por sus siglas en inglés). El MIG es la puerta de enlace a la red fija. Los MIGs emiten periódicamente un mensaje *MMARP_DFL_ROUTE* el cual es inundado a la red ad hoc entera. La recepción de este mensaje informa a nodos intermedios sobre el camino hacia fuentes multicast. Cuando el mensaje *MMARP_DFL_ROUTE* alcanza al receptor o un vecino del receptor, este nodo inicia un proceso de unión parecido al descrito en la parte reactiva, solo que en este caso la unión se realiza hacia el MIG y no hacia la fuente. Cuando el MIG recibe el mensaje *MMARP_JOIN*, éste entonces envía un reporte IGMP hacia el enrutador que se encuentra a un salto del nodo en la red fija que quiere hacer comunicación dentro de la red ad hoc. Este enrutador es el FHMR (First Hop Multicast Router, por sus siglas en inglés). De esta forma la comunicación entre la red ad hoc y la red alambrada es realizada [5].

2.4 Propuesta de solución

Los protocolos de enrutamiento multicast para MANETs anteriormente descritos como soluciones, tienen el inconveniente de que hacen continuamente inundación, la cual es costosa, generando un pobre rendimiento en la entrega de paquetes de datos. Por lo que estos protocolos no son escalables. Por las anteriores razones proponemos un protocolo que utilice regiones de interés y además sea escalable. Este protocolo lo llamaremos SSMRP - Scalable Source-initiated Multicast Routing Protocol.

El protocolo SSMRP tiene una similitud con los protocolos expuestos en este capítulo, principalmente con los basados en mallas. Pues utiliza las mallas para el encuentro de rutas del origen al destino. Es mejor utilizar mallas que árboles ya que cuando en un árbol se pierde una ruta, porque se fue un nodo o hubo una desconexión en algún segmento de la red, al árbol se tiene que recomponer para crear de nuevo las rutas ya que principalmente están basados en núcleos. En cambio en una malla la recomposición de una segmentación en la red es mucho más fácil porque existen múltiples caminos de la fuente al destino y no tiene que

recomponerse toda la malla para el envío de paquetes. Por eso es más costoso mantener un árbol que una malla.

La diferencia fundamental de SSRMP y los demás protocolos es que utiliza regiones de interés (nodos relevantes para la disseminación de información de un flujo, ya sean receptores, fuentes o nodos de reenvío que se encuentran localizados entre las fuentes y los receptores), y es escalable pues al no tener que inundar toda la red, sino solo las regiones de interés, emplea menos mensajes de control conforme aumenta el tamaño de la red.

Resumen

En este capítulo se describieron los protocolos de enrutamiento multicast para redes MANET y sus principales deficiencias. En el siguiente capítulo se describirá el diseño del algoritmo SSMRP que utiliza las regiones de interés que ayudan a superar estas deficiencias.

Capítulo 3

DISEÑO DEL ALGORITMO SSMRP

En este capítulo se describe el diseño del algoritmo basado en regiones de interés así como sus estructuras y tablas que maneja.

3.1 Descripción del algoritmo

En el protocolo SSMRP (Scalable Source-Initiated Multicast Routing Protocol) miembros del grupo y tablas multicast de ruteo se establecen y actualizan por la fuente bajo demanda. Similar a protocolos de enrutamiento unicast bajo demanda, una fase de petición y respuesta forma parte del protocolo. Mientras una fuente multicast tenga paquetes que enviar, ésta hace difusión (broadcast) a la región de interés. Esta actualización periódica actualiza los miembros del grupo y sus rutas como se explica a continuación. Cuando un nodo recibe un paquete de petición que no esté duplicado, es decir, que el mismo paquete no sea enviado por varios nodos, éste guarda el identificador del nodo que está en dirección a la fuente y retransmite el paquete a la región de interés. Mientras existan entradas válidas en la tabla de miembros, los nodos envían sus *tablas de unión* a sus vecinos. Cuando un nodo recibe tablas de unión de otros nodos, éste verifica si su propio identificador se encuentra en esas tablas. Si es así, el nodo se da cuenta que se encuentra en el camino a la fuente y entonces forma parte del grupo de reenvío, de tal manera que las tablas de unión son propagadas a la región de interés hasta que alcancen la fuente multicast vía el camino más corto. Este proceso construye o actualiza las rutas de las fuentes a los receptores construyendo una malla de nodos.

El grupo de reenvío es un conjunto de nodos que se encarga de enviar paquetes multicast. Éste soporta caminos más cortos entre cualquiera de los miembros. Todos los nodos dentro de la región de interés reenvían paquetes multicast de datos. Se debe notar que un receptor multicast también puede ser parte del grupo de reenvío si está entre la fuente y otro receptor. Por lo que este nodo pertenece al grupo de receptores y al grupo de reenvío. Este tipo de nodos tiene dos funciones, recibir y reenviar. La malla proporciona una variada conectividad entre los nodos multicast y la inundación provee redundancia ayudando a la propagación a través de los nodos.

3.2 Región de interés

Como anteriormente se mencionó una región de interés o enclave para un flujo multicast o unicast es un componente conectado de red que contiene aquellos nodos relevantes para la disseminación de información de un flujo, ya sean receptores, fuentes o nodos de reenvío que se encuentran localizados entre las fuentes y los receptores. Ya que todos los nodos en la región de interés de un flujo tienen interés en el flujo, los nodos participan proactivamente en la señalización necesaria para mantener la información de enrutamiento. Por la misma razón, nodos localizados fuera de la enclave definidos por un destino (unicast o multicast) no participan en el proceso de enrutamiento de paquetes para ese destino; por lo tanto, la transmisión y recepción de paquetes con respecto a ese destino se vuelve una sobrecarga innecesaria [15].

Para el caso de un flujo unicast de un destino determinado, los nodos con interés en tal flujo son: el destino unicast, las fuentes activas con el tráfico hacia el destino y los nodos que forman el camino más corto entre las fuentes y el destino. Debido a que el destino unicast es único, los nodos fuera del enclave del destino unicast simplemente detienen la propagación de los paquetes de control. Para un destino multicast otros nodos pueden enviar paquetes a un grupo multicast sin ser parte del grupo. Un enclave, está definido por un destino multicast activo, que incluye las fuentes, el destino (incluye receptores y la malla multicast) y los nodos de reenvío entre ellos. Los paquetes de control son enviados dentro del enclave con mucha más frecuencia que fuera del enclave. Esta frecuencia decrece exponencialmente con respecto a la distancia en saltos de un nodo al límite del enclave.

Las regiones de interés son establecidas con el soporte de los paquetes de datos que fluyen de las fuentes a los destinos. El algoritmo de la Figura 3.1, se utiliza para decidir si un nodo tiene que transmitir un paquete de control al destino D (Join Query Destiny). Para un destino unicast, el algoritmo regresa verdadero si el nodo es transmisor (*sender, sd*) o un nodo en la ruta (*pathnode, pn*), si se ha transmitido un paquete desde cualquier fuente *sd* durante los segundos anteriores, y falso de lo contrario [15].

```

Algoritmo ENCLAVE(JQ)
if AddressType(JQ.D) = multicast then
    if rc  $\vee$  sd  $\vee$  mm  $\vee$  pn then
        else
            if  $r \bmod R = 0$  then
                r++;
            else
                r++;
                return false;
        else
            if pn  $\vee$  sd then
                else
                    return false;
    return true;

```

Figura 3.1 Algoritmo Región de Interés

Para un destino multicast, el algoritmo regresa verdadero si el nodo es un receptor (*rc*), una fuente (*sd*), miembro de una malla (*mm*), o si el nodo está en la ruta (*pn*). Como en el caso anterior, un nodo está en la ruta si ha retransmitido un paquete desde cualquier fuente durante los segundos anteriores. De otra manera, el algoritmo comprueba para el valor $r \bmod R$ y regresa verdadero si es igual a cero y falso de lo contrario. El valor $r \bmod R$ es usado para reducir la frecuencia con la cual un nodo fuera del enclave transmite paquetes. El valor r es inicializado a 0.

3.3 Reenvío de datos

Después de establecer el grupo, la región de interés y la construcción de rutas, una fuente multicast puede transmitir paquetes a los receptores seleccionando rutas y grupos de reenvío. Los paquetes de control son enviados periódicamente solo si existen aún paquetes de datos.

Cuando un nodo recibe un paquete de datos multicast, el nodo lo reenvía si éste no está duplicado y la bandera de reenvío no ha expirado.

3.4 Flexibilidad

No se necesita enviar paquetes adicionales para dejar o unirse al grupo. Si una fuente multicast quiere dejar el grupo, ésta simplemente cesa de enviar peticiones por que no tiene más datos que enviar.

Si un receptor no quiere pertenecer más a un grupo multicast en particular, éste remueve la entrada

correspondiente en su tabla de miembros y no transmite su tabla de unión a ese grupo. Nodos que pertenecen a un grupo de reenvío son degradados a nodos de no-reenvío si no se actualiza su tiempo de espera (timeout).

3.5 Selección de los valores del temporizador

Los valores de intervalo del temporizador para la actualización de ruta y el grupo de reenvío pueden tener impacto en el rendimiento de SSMRP. La selección del temporizador del estado suave (softstate) debe ser acorde al ambiente de la red (tipo de tráfico, carga de tráfico, patrones de movimiento, velocidad de movimiento, capacidad del canal, etcétera). Cuando se usan intervalos de actualización pequeños, éstos provocan que los miembros del grupo generen más paquetes causando congestión en la red. En cambio, si se utilizan intervalos de actualización grandes, aunque generan menos tráfico de paquetes de control, los nodos no están actualizados constantemente o tardan en ser actualizados. Ya que estas son redes altamente dinámicas, utilizar intervalos grandes de actualización puede provocar un pobre rendimiento del protocolo.

De la misma forma el tiempo de espera de un nodo que pertenece al grupo de reenvío debe ser cuidadosamente seleccionado. En redes con carga de tráfico pesado, se deben usar valores pequeños para que los nodos que no estén siendo utilizados, su tiempo de espera expire de forma rápida. En situaciones con alta movilidad en cambio, se deben usar valores grandes para que existan más rutas alternativas.

Es importante notar que el tiempo de espera debe ser más grande (de 3 a 5 más veces) que el valor del intervalo del temporizador para la actualización de las rutas.

3.6 Enrutamiento Unicast

Una de las grandes fortalezas de este protocolo es su capacidad para el enrutamiento unicast. Además de que puede coexistir con otro protocolo que realice este enrutamiento, éste puede operar como protocolo unicast. Así, una MANET equipada con SSMRP no requiere otro protocolo unicast por separado para poder funcionar.

3.7 Estructura de datos

3.7.1 El protocolo SSMRP contiene las siguientes estructuras de datos

Tabla de Miembros: Cada receptor multicast guarda información de la fuente en su tabla de miembros. Para cada grupo en que el nodo participa, guarda el identificador de la fuente y el tiempo del último paquete de

control. Si un paquete de unión no se recibe desde la fuente dentro del periodo de actualización, ese registro es removido de la tabla de miembros.

Tabla de enrutamiento: La tabla de enrutamiento es creada bajo demanda y es soportada por cada nodo. Una entrada es insertada o actualizada cuando se recibe un paquete de solicitud de unión que no esté duplicado. El nodo guarda el destino (por ejemplo algún nodo que envió el paquete de solicitud unión), el próximo salto al destino (el último nodo que reenvió el paquete de solicitud de unión) y los nodos que se encuentren en el 1-vecindario, es decir, a un salto de distancia, para formar la región de interés.

Tabla de reenvío de grupo: Cuando un nodo es parte del grupo de reenvío de un grupo multicast, éste mantiene la información en la *tabla de reenvío de grupo*. Se registra el ID del grupo multicast y la última actualización del tiempo de espera.

Mensajes en cache: Los mensajes en cache son mantenidos por cada nodo para detectar duplicados. Cuando un nodo recibe un nuevo paquete de solicitud de unión o de datos, éste guarda el ID de la fuente y el número de secuencia del paquete. Esquemas como FIFO (First In First Out) o LRU (Least Recently Used) permiten remover entradas viejas y prevenir que la cache sea extensa.

3.8 Criterio de selección de ruta

En SSMRP, un receptor multicast selecciona las rutas basado en el mínimo retardo (por ejemplo, seleccionar una ruta por el primer Join Query recibido). Se aplica otro método de selección cuando se utiliza predicción de movilidad. La idea está inspirada por el protocolo ABR [9] (Associativity-Based Routing, por sus siglas en inglés) el cual escoge rutas estables. En nuestro algoritmo, en lugar de usar el camino con el mínimo retardo, podemos escoger la ruta que sea más estable (ejemplo, la que lleve más tiempo conectada). Para seleccionar una ruta, un receptor multicast debe esperar un tiempo apropiado después de recibir el primer Join Query, para que todas las posibles rutas y *calidad* de ruta sean conocidas. El receptor entonces, escoge la ruta más estable y transmite su Join Reply.

3.9 Operación

3.9.1 Reenvío de grupo

3.9.1.1 Generando un *JoinQuery*

Cuando una fuente multicast tiene paquetes que enviar pero no hay ruta conocida, ésta genera un paquete

JoinQuery. El campo *Type* debe ser puesto a uno. El TTL (Time to Live Value), debe ser ajustado al tamaño y diámetro de la red. El número de secuencia (Sequence Number) debe ser grande, suficiente para prevenir la ambigüedad y el *Hop Count* (cuenta de saltos) debe ser inicializado a cero. La fuente escribe su dirección IP en los campos *Source IP Address* y *Last Hop IP Address*.

Para la información de ubicación y movimiento, se establece el campo *MIN_LET* (Tiempo de expiración del enlace), al valor *MAX_LET_VALUE* ya que la fuente no tiene un nodo anterior a ella. Cuando la fuente recibe *JoinReplies* de los receptores multicast, ésta selecciona el mínimo *RET* (Tiempo de expiración de la ruta) entre todos los *JoinReplies* recibidos. Entonces la fuente puede construir nuevas rutas generando un *JoinQuery* antes de los resultados del siguiente mínimo RET.

3.9.1.2 Procesando un *JoinQuery*

Cuando un nodo recibe un paquete *JoinQuery*:

1. Revisa si está duplicado comparando la dirección IP de la fuente y el número de secuencia con las entradas que se encuentran en cache. Si es así, se descarta el paquete.
2. Si no está duplicado, inserta una entrada en cache con la información que trae el paquete (número de secuencia, dirección IP) e inserta o actualiza la entrada para la tabla de enrutamiento.
3. Si el nodo es miembro del grupo multicast, éste origina un paquete *Join Reply* y le adjunta un valor *RET*.
4. Incrementa el *Hop Count* en 1 y decrementa el TTL en 1.
5. Si el valor del TTL es menor o igual a 0, descarta el paquete.
6. Si el valor del TTL es mayor que 0, escribe la dirección IP del nodo en el campo *Last Hop IP Address* y reenvía el paquete.

3.9.1.3 Generando un *Join Reply*

Un receptor multicast transmite un paquete *Join Reply* después de seleccionar la ruta multicast. Cada dirección IP de un transmisor y dirección IP del próximo salto (next hop) se incluyen en el paquete *Join Reply*. También se incluye el tiempo de expiración de la ruta.

3.9.1.4 Cuando se recibe un *Join Reply*

1. El nodo busca la dirección IP del próximo salto en las entradas del paquete *Join Reply* que se recibió. Si no existen entradas que coincidan con la dirección IP del nodo, no hace nada. Esto es porque el nodo no se encuentra en el camino a la fuente.
2. Si una o más entradas coinciden con la dirección IP del nodo, entonces prende su bandera *FG_FLAG* y construye su propio paquete *Join Reply*. La dirección IP del próximo salto puede ser obtenida de la tabla de enrutamiento.
3. Reenvía el paquete *Join Reply* a los nodos vecinos.

Resumen

En este capítulo se describió el diseño del algoritmo SSRMP para un protocolo de enrutamiento multicast que utiliza las regiones de interés.

En el siguiente capítulo se describirá la implementación del protocolo en el simulador de red NS2.

Capítulo 4

IMPLEMENTACIÓN DEL ALGORITMO SSMRP

En este capítulo se describe el funcionamiento del simulador de eventos discretos NS2 para protocolos de enrutamiento, así como la implementación de SSMRP en NS2.

4.1 El simulador de red NS2

Se hace uso del simulador de eventos discretos NS2 [16] versión 2.34, el cual provee simulaciones realistas de la capa física y es un estándar de facto para la evaluación de protocolos de comunicaciones. Para las simulaciones de ODMRP se hizo uso del código del "Proyecto Monarca" [17]. Cada simulación fue ejecutada diez veces con diez semillas distintas. Con el objetivo de tener simulaciones válidas, los algoritmos multicast usan el mismo periodo de tiempo de tres segundos para actualizar sus estructuras de enrutamiento.

Tabla 4.1 Entorno de simulación.

Número de nodos	100	Posición de los nodos	Aleatoria
Área de simulación	1400m x 1400m	Tiempo de simulación	150s
Protocolo MAC	802.11b	Tasa de transmisión	20480 bps
Modelo de movilidad	Randomwaypoint	Paquetes por fuente	1000
Pausa del nodo	10s	Vel. min.-max nodo	1-10 m/s

4.2 El protocolo SSMRP

El protocolo SSMRP está contenido principalmente en la clase *ssmrp.cc* la cual llama a la clase agente *SSMRPAgent.cc* que se encarga del enrutamiento del protocolo. Esta clase se encarga del envío de paquetes en NS2 y contiene tres funciones principales del método *command()* (Figura 4.1). *Start*, que se encarga del inicio de envío de paquetes desde el nodo fuente; *Tracetarget* para el registro de trazas; y *Port-Dmux* para clasificar y enviar paquetes basados en el puerto de la capa de transporte, esto es atribuido al puerto clasificador que en NS2 se llama *demultiplexer* [18].

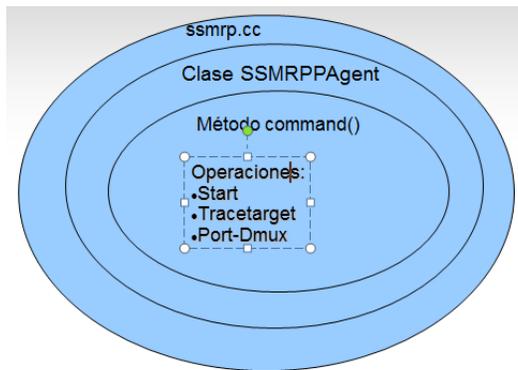


Figura 4.1 Clase ssmrp.cc

Es fundamental el entendimiento de la clase *ssmrp.cc* para evitar errores en las simulaciones en NS2.

Al correr las simulaciones del protocolo SSMRP antes de que fueran satisfactorias se generaron varios errores en NS2. Esto sucede principalmente cuando se agrega un protocolo nuevo al simulador. Se ponen como ejemplo los siguientes errores y su explicación:

```
(_o14 cmd line 1)
  invoked from within
"_o14 cmd addr"
```

Empezando de arriba hacia abajo este bloque de tres líneas se repite consecutivamente en la salida de errores.

Lo siguiente es otro ejemplo de error:

```
(_o14 cmd line 1)
  invoked from within
"_o14 cmd addr"
  invoked from within
"catch "$self cmd $args" ret"
  invoked from within
"if [catch "$self cmd $args" ret] {"
```

invoked from within

Este error es a causa de llamar otro procedimiento. NS2 provee un mecanismo de capa de errores de nivel por nivel hasta llegar al error real en el código. Cada error es separado por la frase “invoked from within”, esto quiere decir que la línea de arriba es invocada de la línea de abajo siguiente:

```
"if [catch "$self cmd $args" ret] {"
```

Esta es la parte exacta donde se causa el error en NS2. El formato es el siguiente:

categoría [argumento] número de línea

Categoría puede ser el archivo, procedimiento o nombre de la clase donde se origina el error.

Argumento, puede ser opcional y especifica la ubicación de la categoría.

Número de línea, muestra exactamente la ubicación del argumento [18].

4.3 Corridas del simulador.

A continuación se muestran ejemplos de simulaciones hechas en NS2 versión 2.34.

```
warning: no class variable ARPTable::maxdelay_

*** NOTE: no power scaling scenario specified.
*** NOTE: no connection pattern specified.
using backward compatible Agent/CBR; use Application/Traffic/CBR instead
using backward compatible Agent/CBR; use Application/Traffic/CBR instead
Starting Simulation...
completed through 6.0 secs...
SSMRPAgent: join-group succeeded for group 32769
completed through 12.0 secs...
SORTING LISTS ...DONE!
channel.cc:sendUp - Calc highestAntennaZ_ and distCST_
highestAntennaZ_ = 1.5, distCST_ = 550.0
completed through 18.0 secs...
completed through 24.0 secs...
NS EXITING...
vancho@vancho-laptop:~/NSv/ns-allinone-2.34j/ns-2.34$
```

Figura 4.2 Simulación 1 de NS2

```

vancho@vancho-laptop:~/NSv/ns-allinone-2.34j/ns-2.34$ ./ns scripts/run.tcl
*** no error model specified.
Seeding Random number generator with 496

num_nodes is set 30
UniAgent: net_id is 0
warning: no class variable ARPTable::mindelay_

        see tcl-object.tcl in tclcl for info about this warning.

warning: no class variable ARPTable::maxdelay_

INITIALIZE THE LIST xListHead
UniAgent: net_id is 1
warning: no class variable ARPTable::mindelay_

        see tcl-object.tcl in tclcl for info about this warning.

warning: no class variable ARPTable::maxdelay_

UniAgent: net_id is 2
warning: no class variable ARPTable::mindelay_

        see tcl-object.tcl in tclcl for info about this warning.

warning: no class variable ARPTable::maxdelay_

```

Figura 4.3 Simulación 2 de NS2

En la primera figura se puede observar cómo se van uniendo nodos al grupo con Id 32769 y en la segunda cómo se inicializa la lista de nodos que participan en la simulación y la asignación de su identificador *net_id*.

4.4 Implementación de algoritmo Enclave

Como se mencionó en el capítulo 3 el algoritmo Enclave (Figura 3.1) y que se reproduce a continuación es usado para decidir si un nodo tiene paquetes que retransmitir para un destino D. En caso de un destino unicast el algoritmo Enclave regresa verdadero si el nodo es un transmisor (sd) o está en el camino hacia el destino (pn – path node) de lo contrario regresa falso [15].

Algoritmo ENCLAVE(JQ)

```
if AddressType(JQ.D) = multicast then  
    if rc V sd V mm V pn then  
        else  
            if r mod R = 0 then  
                r++;  
            else  
                r++;  
                return false;  
  
        else  
            if pn V sd then  
                else  
                    return false;  
  
return true;
```

Para el caso de destinos multicast, como se muestra en el pseudo código anterior, el algoritmo *Enclave* regresa verdadero si el nodo es un receptor (*rc*), un transmisor (*sd*), un miembro de una malla (*mm*), o si el nodo está en el camino (*pn*). Un nodo *pn* es un nodo que ha retransmitido un paquete desde cualquier fuente a *D* durante el último periodo de segundos. Sino, *Enclave* revisa el valor de $r \bmod R$ y regresa verdadero si es igual a 0 y falso de lo contrario. El valor $r \bmod R$ es usado para reducir la frecuencia con la cual un nodo fuera de la enclave transmite JQ. El valor de *r* es inicialmente puesto a 0 [15].

En el caso de que nodos de red se desconecten, ya que el protocolo está basado en mallas puede encontrar el destino en los múltiples caminos que existan si necesidad de recomponer toda la malla.

Para el caso de que la región de interés sea muy grande al grado que sea del mismo tamaño que el total de la red, el rendimiento es el mismo que ODMRP.

Resumen

En este capítulo se describió la implementación del protocolo SSMRP en el simulador NS2. Los problemas

encontrados y cómo fueron resueltos.

En el siguiente capítulo se describirá las pruebas y resultados obtenidos de SSRMP en el simulador NS2.

Capítulo 5

PRUEBAS Y RESULTADOS

5.1 Pruebas

En este capítulo se presentan los resultados obtenidos al comparar el tráfico multicast del algoritmo propuesto SSMRP contra ODMRP. Se utilizó ODMRP para los experimentos ya que es ampliamente usado en los análisis de desempeño para los algoritmos multicast. Se calculó la tasa de paquetes entregados PDR (Packet Delivery Ratio), el retardo de extremo a extremo EED (End to End Delay), la sobrecarga de control CO (Control Overhead) y la sobrecarga total (Total Overhead) como métricas de desempeño. El porcentaje de paquetes entregados es una de las métricas más importantes en la evaluación del desempeño de un algoritmo de enrutamiento ya que mide la efectividad del algoritmo para establecer y mantener rutas que conecten a la fuente con sus receptores. Para el caso del tráfico multicast el porcentaje de paquetes entregados se calcula de la siguiente manera:

$$PDR = \frac{\sum_{i=1}^{n_s} n_r(i)}{\sum_{i=1}^{n_s} n_m(i)} \quad (5.1)$$

Donde n_s es el número total de paquetes enviados, $n_r(i)$ es el número de nodos que reciben el paquete i , $n_m(i)$ es el número de nodos que son miembros del grupo multicast D al momento que el paquete i con destino D es enviado.

El retardo extremo a extremo es el promedio de tiempo que les toma a los paquetes de datos desplazarse desde el nodo fuente hasta llegar a sus destinos. Esta métrica está relacionada con la longitud de los caminos que conectan a las fuentes con los receptores. Se calcula de la siguiente manera:

$$EED = \frac{1}{n_r} \sum_{i=1}^{n_r} t_r(i) - t_s(i) \quad (5.2)$$

Donde n_r es el número total de paquetes recibidos, $t_s(i)$ es el tiempo en el que el paquete i fue enviado, y $t_r(i)$ es el tiempo en el que el paquete i fue recibido por alguno de los nodos miembros del grupo multicast D .

La sobrecarga de control es el número total de paquetes de control que transmite un algoritmo de enrutamiento para establecer y mantener las rutas. La sobrecarga total es el total de bytes transmitidos a lo

largo de todo el experimento. Estas métricas se utilizan para evaluar la cantidad de recursos que los algoritmos consumen para proveer las rutas.

Para las pruebas se hace uso del protocolo MAC IEEE 802.11 para acceder al medio, y todos los paquetes de control se envían en modo broadcast. Los nodos usan el modelo de movilidad “randomwaypoint”, en el cual escogen una posición y velocidad aleatoria para desplazarse. Una vez que alcanzan el destino seleccionado, permanecen un tiempo de pausa en la misma posición y cuando termina ese lapso de tiempo el nodo se vuelve a desplazar.

En la simulación los nodos seleccionan una posición aleatoria dentro de un área de 1400 m x 1400 m, donde permanecen 10 segundos sin movimiento, al finalizar este tiempo de pausa, se desplazan a otra posición y velocidad aleatoria en un intervalo de 1 a 10 metros por segundo. El tiempo de simulación para los experimentos es de 150 segundos. Ver Tabla 5.1.

Los nodos fuente de tráfico son designados al azar, al igual que los miembros del grupo multicast. Se hace uso de la aplicación CBR (Constan Bit Rate) para generar datos a transmitir. CBR permite establecer la frecuencia de transmisión, el destino y el tamaño de paquetes, y no tiene ninguna aplicación en particular en los nodos más que generar y recibir paquetes de datos. En los experimentos cada fuente de datos transmite un máximo de 1000 paquetes a una frecuencia de 10 paquetes por segundo. Cada paquete generado tiene un tamaño de 256 bytes.

El formato del identificador del destino del paquete CBR es simplemente la dirección del grupo multicast deseado.

Con lo anterior dicho y conforme a los objetivos planteados en el capítulo 1 de diseñar un protocolo iniciado por el transmisor, diseñar experimentos para su evaluación e implementarlo en un simulador de redes se muestra a continuación los escenarios para lograr los objetivos. El protocolo como se vio en el capítulo 4 se implementó en el simulador de redes NS2 versión 2.34.

Cada simulación fue ejecutada seis veces con seis semillas distintas. Con el objetivo de tener simulaciones válidas, los algoritmos multicast usan el mismo periodo de tiempo de tres segundos para actualizar sus estructuras de enrutamiento (periodo *joinquery* para ODMRP)

Tabla 5.1: Entorno de simulación

Número de nodos	100	Posición de los nodos	Aleatoria
Área de simulación	1400m x 1400m	Tiempo de simulación	150s
Protocolo MAC	802.11b	Tasa de transmisión	20480 bps
Modelo de movilidad	Randomwaypoint	Paquetes por fuente	1000
Pausa del nodo	10s	Vel. min.-max nodo	1-10m/s
Tamaño de los paquetes	256 bytes	Número de fuentes	20

5.2 Desempeño con tráfico multicast

En esta sección se presentan los resultados obtenidos al comparar SSMRP con ODMRP.

5.2.1 Incrementando el número de fuentes

En el primer experimento, el escenario cuenta con un solo grupo multicast que consta de 20 nodos, y se va incrementando el número de fuentes activas, es decir, el número de fuentes que envían datos hacia el grupo multicast. Para este experimento las fuentes no son miembros del grupo. Los resultados se presentan en las Figuras 5.1-a- 5.4.

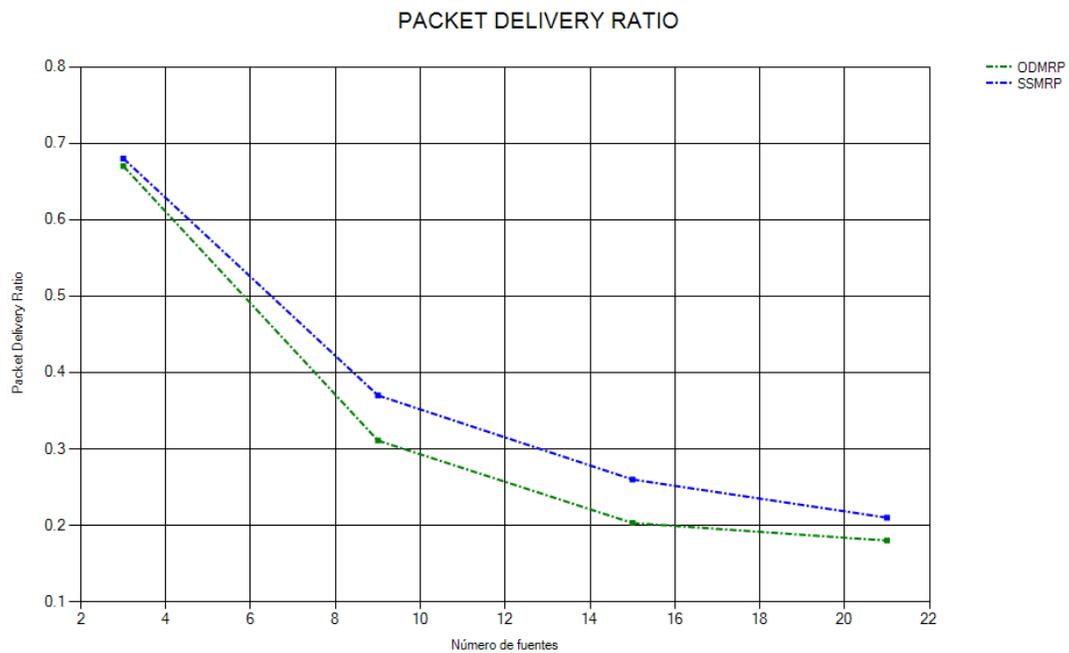


Figura 5.1. Desempeño al incrementar el número de fuentes vs Packet Delivery Ratio.

Estos resultados son particularmente positivos si analizamos la Figura 5.1 donde se ve que el número de paquetes de datos entregados por SSMRP es al menos igual a ODMRP y conforme el número de fuentes se incrementa, el número de paquetes entregados por SSMRP es mayor. Es claro ver que SSMRP es ligeramente mejor que ODMRP, esto es por que en el comienzo cuando son pocas fuentes no existe región de interés que pueda crear SSMRP por lo reducido de la malla, fuentes y nodos. Por lo que la entrega de paquetes es casi la misma en ambos protocolos.

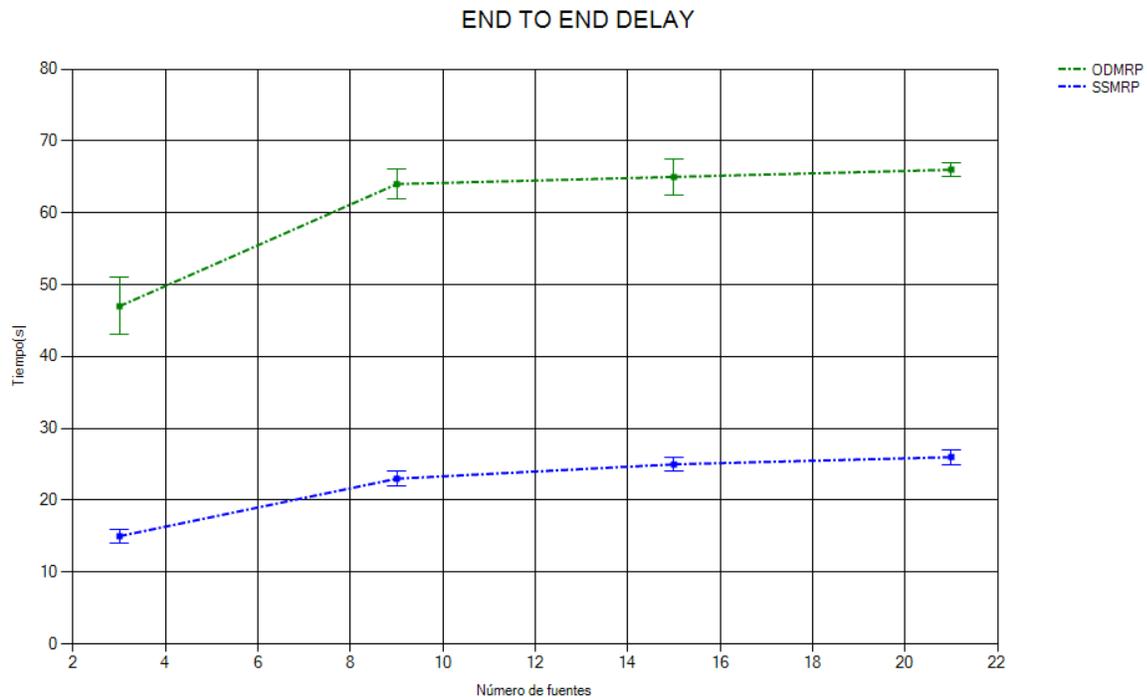


Figura 5.2. Desempeño al incrementar el número de fuentes vs End to End Delay.

También se observa que en la Figura 5.2 que el retardo de extremo a extremo en SSMRP es mucho menor al de ODMRP. Este retardo está relacionado con la cantidad de paquetes de control que son intercambiados. Ya que el medio de transmisión tiene una capacidad limitada y además los paquetes de control tienen mayor prioridad que los paquetes de datos, en algoritmos como ODMRP (con un alto intercambio de tráfico de control) los paquetes de datos tienen que ser almacenados dentro de las colas de transmisión, hasta que los paquetes de control hayan sido transmitidos.

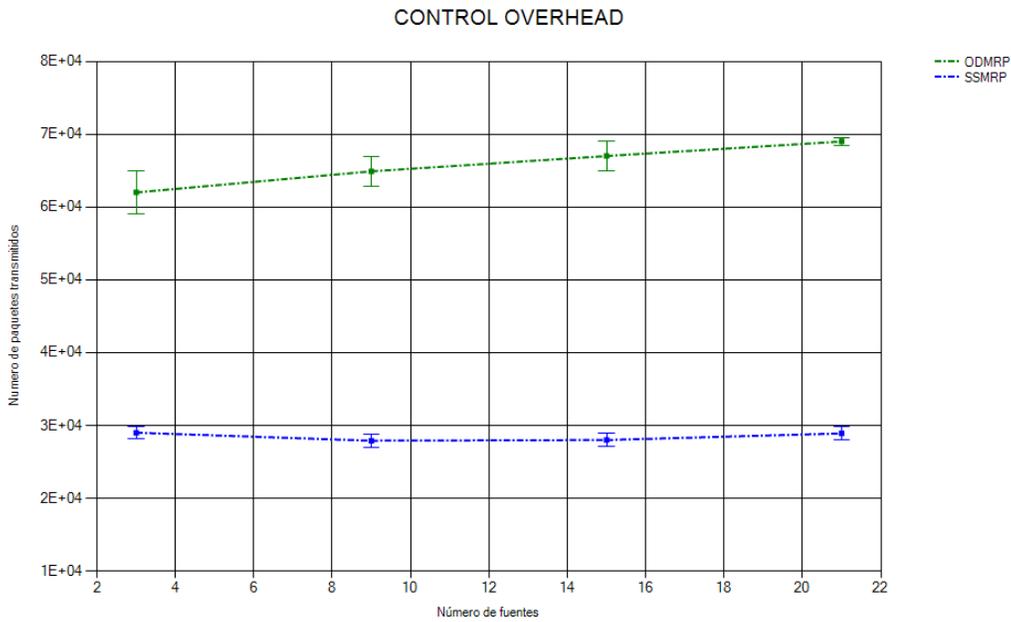


Figura 5.3. Desempeño al incrementar el número de fuentes vs Control Overhead.

Como puede observarse en la Figura 5.3, el número de paquetes de control generados por SSMRP es considerablemente menor que el generado por ODMRP. Esto es debido al uso de las regiones de interés en SSMRP, con las cuales se contiene el intercambio de tráfico de control solo a las porciones de la red donde es necesario, mientras que con ODMRP se inunda periódicamente la red completa con paquetes *JoinRequest* generados por cada fuente.

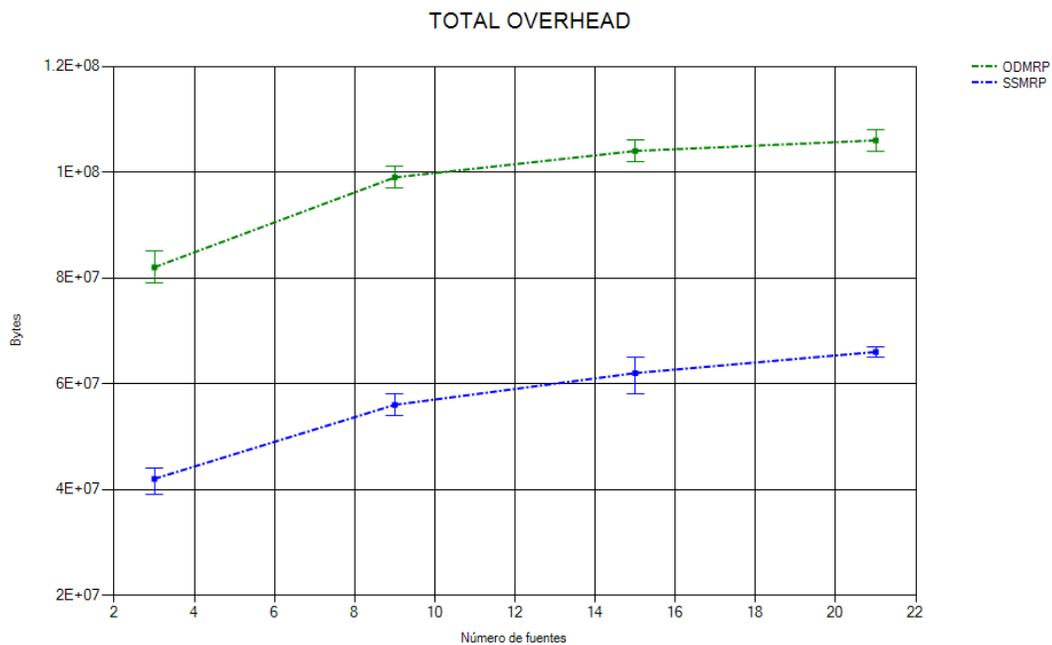


Figura 5.4. Desempeño al incrementar el número de fuentes vs Total Overhead.

En las Figuras 5.3 y 5.4 donde se muestran respectivamente, el número de paquetes de control transmitidos y el total de bytes transmitidos en toda la red, se observa que SSMRP hace un uso más eficiente de los recursos de la red que ODMRP.

Los resultados anteriores demuestran la efectividad de las regiones de interés para lograr una mayor reutilización espacial del ancho de banda lo que redundará en retardos más bajos y una mayor entrega de paquetes de datos.

5.3 Discusión

Como se menciona en esta tesis se utilizó el simulador NS2 versión 2.34, pero en un principio fue de difícil uso por el desconocimiento del mismo. Uno de los grandes problemas de este simulador es la falta de un IDE (Integrated Development Environment), pues cuando se empieza a programar un protocolo de red o se hace la implementación de alguno ya existente, si existe algún error en código es complicado dar con él por falta del IDE. Existe la forma de acoplar el NS2 a Eclipse pero no cumple completamente con el fin. Para la lectura de las trazas el NS2 no tiene la capacidad de hacerlo por lo que se tiene que utilizar otro lenguaje de programación al que ya se emplea en el simulador (C++, o tcl) como AWK. De la misma forma no se puede graficar directamente los resultados de las trazas del protocolo empleado. Se tiene que hacer uso de otras herramientas como MATLAB o .NET.

Sin embargo NS2 es usado ampliamente para la investigación de protocolos de red y lo más importante es gratis.

Se debe mencionar en esta tesis que los resultados obtenidos son más o menos realistas. Pues las simulaciones se realizaron en escenarios con 100 nodos. Hablando de comunidades pequeñas, se puede decir que en el país existen comunidades rurales donde no hay una infraestructura de red y su población ronda en 100 habitantes. Aquí el protocolo trabajaría conforme a los resultados mostrados.

Ahora que si fueran poblaciones de más de 1000 habitantes es muy probable que el protocolo no cumpla con su objetivo.

Resumen

En este capítulo se describieron los resultados obtenidos en las simulaciones del protocolo SSMRP, las gráficas resultantes en comparación con el protocolo ODMR y el comportamiento de ambos.

En el siguiente capítulo se presentan las conclusiones.

Capítulo 6

CONCLUSIONES

En este documento se ha presentado SSMRP, algoritmo de enrutamiento multicast para MANET's. En esta tesis se presenta una clase de algoritmos de enrutamiento basados en regiones de interés. En este tipo de enrutamiento solo si existen nodos que tengan interés en transmitir datos para algún destino, los nodos receptores de ese destino miembros del grupo multicast, anunciarán su presencia y formarán una malla dentro de la que se dará mantenimiento a las rutas para ese destino. Esta malla únicamente contiene a las fuentes, a los receptores y a los nodos intermedios que los conectan, y se desactivan cuando dejan de existir flujos de datos para el destino para el cual se formó la malla. A esta malla se le llama región de interés, es decir, esta región se activa bajo demanda, solo cuando un nodo requiere enviar datos para ese destino. Además los nodos dentro de la región de interés dan mantenimiento a las rutas hacia el destino.

6.1 Logros alcanzados

Conforme a los objetivos propuestos en el capítulo 1:

- *Analizar protocolos de enrutamiento para MANETs.* Como se mencionó en el capítulo 2 Estado del Arte, se estudiaron y analizaron varios protocolos para enrutamiento de MANETs. Como NSMP, DCMP, PUMA, AMRoute, AMRIS, MMARP etc. Todos estos son protocolos que están basados en mallas o basados en árboles, incluso híbridos. El estudio dio como resultado que es mejor un protocolo basado en mallas que basado en árboles pues las mallas son más redundantes y esto es mejor para redes inalámbricas por la constante movilidad de los nodos.
- *Diseñar un protocolo de enrutamiento del tipo iniciado por el transmisor y que sea escalable.* Se diseñó un protocolo Sender Initiated (Iniciado por el transmisor) y es escalable pues los mensajes de control no se inundan a toda la red sino solo a las regiones de interés. Esto lo podemos ver en las gráficas de resultados, conforme se van incrementando las fuentes la gráfica de SSMRP se mantiene pero la gráfica de ODMRP va en caída, esto confirma que el protocolo SSMRP si es escalable.
- *Implementar el protocolo propuesto en un simulador de redes.* Se implementó el protocolo propuesto en el simulador de redes NS2 como se especifica en el capítulo 4. Esto fue logrado después de varios

intentos para implementar ODMRP pues su código fuente se encontraba en una versión anterior de NS2. Después se tuvo que implementar el nuevo protocolo SSMRP pero en un principio hubo muchos problemas de compilación. Al final los dos protocolos sí pudieron ser simulados en NS2.

- *Diseñar una serie de experimentos para evaluar el desempeño del protocolo propuesto.* Se diseñaron varios escenarios en NS2 para poder evaluar el rendimiento del protocolo. La tabla del entorno de simulación se encuentra en el capítulo 5. Cada simulación fue ejecutada 10 veces con semillas distintas cada una para tener resultados más cercanos a la realidad.
- *Analizar e interpretar los resultados.* En el capítulo 5 se interpretaron los resultados obtenidos por medio de las gráficas mostradas. Se puede ver por medio de las gráficas que el protocolo SSMRP tiene mejoras en cuanto a Control Overhead, Total Overhead, End to End Delay y Paket Delivery Ratio.

Se realizaron experimentos que prueban el mejor desempeño de SSMRP comparado contra ODMRP, el protocolo estándar para enrutamiento multicast en MANET's. Los resultados muestran que el promedio de entrega de paquetes es similar o mayor al de ODMRP.

Estos resultados también muestran que el tráfico de control generado por SSMRP, el cual es necesario para establecer y mantener las regiones de interés, es menor que el tráfico que ODMRP genera para establecer y dar mantenimiento a la malla. También podemos ver que SSMRP hace un uso más eficiente del ancho de banda, pues el total de bytes transmitidos por SSMRP y la entrega de sus paquetes es menor al de ODMRP.

6.2 Trabajo a futuro

Como trabajo a futuro para esta tesis sería posible obtener las siguientes estadísticas: Porcentaje de rutas encontradas por el protocolo propuesto y porcentaje de número de saltos de la fuente al destino.

Hacer escenarios donde se incremente el número de grupos o fijar el número de fuentes e incrementar el número de nodos.

Finalmente para mostrar verdaderamente el desempeño del algoritmo propuesto implementado en NS-2, será necesario realizar su implementación en dispositivos reales.

6.3 Aportaciones

Dentro de las contribuciones de esta tesis es la utilización de las regiones de interés, ya que es una mejora en las redes móviles y protocolos de enrutamiento.

Se puede comprobar que las regiones de interés ayudan al escalamiento en redes móviles ya que el envío de paquetes es limitado a áreas y no inundan la red entera.

La implementación del protocolo se realizó en NS2 y proviene del estado del arte del protocolo ODMRP. Por lo que ambos ODMRP y SSMRP pueden ser consultados para futuras mejoras.

Bibliografía

- [1] Chalmeta Ugas, J. (2009). *Estudio y análisis de prestaciones de redes móviles Ad Hoc mediante simulaciones NS-2 para validar modelos analíticos*. Cataluña.
- [2] Menchaca Mendez, R., & Garcia-Luna-Aceves, J. *Scalable Multicast Routing in MANETs Using Sender-Initiated Multicast Meshes*. University of California, Santa Cruz.
- [3] Stallings, W. (2004). *Comunicaciones y redes de computadoras*. Prentice Hall.
- [4] Lee, S., & Kim, C. (2000). *Neighbor supporting ad hoc multicast routing protocol*. In *Mobihoc '00: Proc. of 1st ACM intl symp. on Mob. ad hoc net. & comp.* ACM.
- [5] Ruiz, P. M., & Gomez Skarmeta, A. F. (2004). *Reducing data overhead of mesh-based ad hoc multicast routing protocols by steiner tree meshes*. In *Proc. of IEEE SECON*.
- [6] Ballardie, A. J., Crowcroft, J., & Francis, P. (1993). *Core Based Trees (CBT)*.
- [7] S. Manoj, B., Kumar Das, S., & Ram Murphy, C. S. (2002). *A Dynamic Core Based Multicast Routing Protocol for Ad Hoc Wireless Networks*. In *MobiHoc '02: Proc. of the 3rd ACM intl. symp. on Mob. ad hoc net. & comp.*
- [8] Vaishampayan, R., & Garcia Luna Aceves, J. *Efficient and Robust Multicast Routing in Mobile Ad Hoc Networks*.
- [9] C.-K. Toh. *Associativity-Based Routing for Ad-Hoc Mobile Networks*. *Wireless Personal Communications Journal, Special Issue on Mobile Networking and Computing Systems*, Kluwer Academic Publishers, vol. 4, no. 2, Mar. 1997, pp. 103-139.
- [10] S.J. Lee, M. Gerla, and Chian, "On-demand multicast routing protocol", in *Proceedings of WCNC*, September 1999.
- [11] J. J. Garcia-Luna-Aceves and E. L. Madruga. *The core-assisted mesh protocol*. *IEEE Journal on Selected Areas in Communications*, Aug 1999.
- [12] E. M. Royer and C. E. Perkins. *Multicast operation of the ad-hoc on-demand distance vector routing protocol*. In *Mo-biCom '99: Proc. of the 5th annual ACM/IEEE intl. conf. on Mob. comp. and net.* ACM, 1999.

- [13] J. Xie and R.R. Talpade, A. McAuley, and M. Liu, "Amroute: Adhoc multicast routing protocol", Mobile Networks and Applications (MONET), December 2002.
- [14] C.W. Wu and Y.C. Tay, "Amris: A multicast protocol for ad hoc wireless networks," Proceedings of IEEE MILCOM, October 1999.
- [15] J.J Garcia-Luna-Aceves, and Rolando Menchaca-Mendez. "PRIME: An Interest-Driven Approach to Integrated Unicast and Multicast Routing in MANETs".
- [16] Network Simulator NS2 <http://www.isi.edu/nsnam/ns/>.
- [17] Proyecto Monarca <http://www.monarch.cs.rice.edu/ftp/multicast/README>.
- [18] Introduction to Network Simulator NS2. Teerawat Issariyakul, Ekram Hossain.
- [19] Manual NS2 <http://www.isi.edu/nsnam/ns/doc/index.html>

ANEXO : Glosario de términos

Conjunto conexo: es un subconjunto $C \subseteq X$ de un espacio topológico (X, \mathcal{T}) (donde \mathcal{T} es la colección de conjuntos abiertos del espacio topológico) que no puede ser descrito como unión disjunta de dos conjuntos abiertos de la topología.

Control Overhead: La sobrecarga de control es el número total de paquetes de control que transmite un algoritmo de enrutamiento para establecer y mantener las rutas.

EED: End to End Delay. El retardo extremo a extremo es el promedio de tiempo que les toma a los paquetes de datos desplazarse desde el nodo fuente hasta llegar a sus destinos.

Grafo: es un conjunto de objetos llamados vértices o nodos unidos por enlaces llamados aristas o arcos, que permiten representar relaciones binarias entre elementos de un conjunto.

NS2: Network Simulator. Software para simular protocolos de red.

ODMRP: On-Demand Multicast Routing Protocol. Protocolo de enrutamiento multicast bajo demanda.

PDR: Packet Delivery Ratio. Mide la efectividad del algoritmo para establecer y mantener rutas que conecten a la fuente con sus receptores.

SSMRP: Scalable Source-Initiated Multicast Routing Protocol. Protocolo de enrutamiento multicast escalable iniciado por la Fuente.

Total Overhead: La sobrecarga total es el total de bytes transmitidos a lo largo de todo el experimento. Estas métricas se utilizan para evaluar la cantidad de recursos que los algoritmos consumen para proveer las rutas.