

**EVALUACIÓN Y OPTIMIZACIÓN DE CRITERIOS PARA LA
SELECCIÓN DE RUTA EN REDES INALÁMBRICAS DE
SENSORES**

TÉSIS DE MÁSTER

Diego Fernando Bermúdez Garzón
dieberg1@posgrado.upv.es

Director

Juan Vicente Capella Hernández



UNIVERSIDAD
POLITECNICA
DE VALENCIA

INDICE

1. INTRODUCCIÓN	4
2. JUSTIFICACIÓN	8
3. OBJETIVOS	9
4. BASES TEORICAS	10
4.1. Protocolos de Encaminamiento	10
4.1.1. Ad-hoc On-demand Distance Vector – AODV	12
4.1.2. Dynamic Source Routing - DSR.....	15
4.1.3. Quick Dirty and Routing Protocol - QDRP	18
4.2. Criterios para elegir un protocolo dirigido a RIS	22
5. HERRAMIENTA DE TRABAJO	24
5.1. Herramienta de trabajo.....	24
5.2. Network Simulator 2 - NS2	24
6. OPTIMIZACION DE UN PROTOCOLO PARA REDES INALÁMBRICAS DE SENSORES	27
6.1. Evaluación de la propuesta	32
6.2. Simulación	38
6.3. Resultados.....	40
7. CONCLUSIONES Y TRABAJO FUTURO	49
8. BIBLIOGRAFIA	50

1. INTRODUCCIÓN

En la actualidad, el avance de la tecnología electrónica y las comunicaciones inalámbricas han evolucionado de tal manera que hoy en día podemos adquirir mecanismos cada vez más pequeños y con mayores prestaciones que hace unos pocos años, además de un coste relativamente bajo. Gracias a estas posibilidades podemos obtener nodos sensores de bajo consumo, multifuncionales y con comunicación inalámbrica de corto alcance que nos permite formar redes ad-hoc. Cabe mencionar que estos dispositivos al ser de un tamaño reducido tienen una limitación en la vida de sus baterías.

Cada uno de estos elementos consiste en una unidad de muestreo, una unidad para el procesamiento de datos y otra para la comunicación. Estos pequeños nodos los podemos usar en cualquier lugar de trabajo, formando una red entre ellos para obtener información del medio en que interactúan y posteriormente enviarla a un destino, aun sin disponer de una infraestructura de red; es lo que se denomina una red de inalámbrica de sensores (RIS).

Todas las RIS sin importar su finalidad tienen algo en común, y es que a la hora de realizar una formación en la red se distinguen dos tipos de nodos, los cuales son:

- ✓ **Nodo sensor:** Todo el tiempo están recolectando información y enviándola a un nodo padre de nivel superior hasta que llegue al destino.
- ✓ **Nodo Sumidero:** Siempre están recibiendo/absorbiendo la información que envían los nodos sensores para ser procesada en la estación principal. Los sumideros a diferencia de un nodo sensor no están limitados en sus recursos hardware ni en energía.

Esta clase de redes se caracterizan por su facilidad de despliegue y por ser autoconfigurables, pudiendo convertirse en todo momento en emisor, receptor, ofrecer servicios de encaminamiento entre nodos sin visión directa, así como registrar datos referentes a los sensores locales de cada nodo. Otra de sus características es su gestión eficiente de la energía, que les permite obtener una alta tasa de autonomía que las hacen plenamente operativas por largos periodos de tiempo.

Las RIS tienen una serie de características propias y otras adaptadas de las redes ad-hoc:

- ✓ **Topología:** En una RIS con movilidad, la topología siempre es cambiante y los nodos tienen que adaptarse para poder comunicar nuevos datos adquiridos. Sin embargo en una RIS sin movilidad la topología va de la mano con el escenario y con la forma en que el protocolo elige las rutas.
- ✓ **Infraestructura de Red:** Una red sensora no tiene necesidad alguna de infraestructura para poder operar, ya que sus nodos pueden actuar de emisores, receptores o enrutadores de la información. Sin embargo, hay que destacar en el concepto de red sensora la figura del nodo recolector (denominados nodo sumidero), que es el nodo que recolecta/absorbe la información y por el cual se recoge la información generada normalmente en tiempo discreto. Esta información generalmente es adquirida por un ordenador conectado a este nodo y es sobre el ordenador que recae la posibilidad de transmitir los datos por tecnologías inalámbricas o cableadas según sea el caso.
- ✓ **Tolerancia a Errores:** Un dispositivo sensor dentro de una red sensora tiene que ser capaz de seguir funcionando a pesar de tener errores en el propio sistema o buscar nuevas alternativas cuando se pierde la conexión con su padre.
- ✓ **Comunicaciones:** En aplicaciones de RIS siempre es característico el uso de algún protocolo que permita comunicaciones multi-salto, aunque también es muy común utilizar mensajería basada en broadcast. Entre los protocolos más usados en este ámbito están el DSR, DSDV, AODV y el QDRP, un nuevo protocolo que ha sido diseñado específicamente para RIS.
- ✓ **Consumo Energético:** Este es uno de los factores más sensibles debido a que tienen que conjugar autonomía con capacidad de proceso, ya que actualmente cuentan con una unidad de energía limitada.

Un nodo sensor tiene que contar con un procesador de consumo ultra bajo así como de un transceptor radio con la misma característica, a esto hay que agregar un software que también conjugue esta característica haciendo el consumo aún más restrictivo.

- ✓ **Limitaciones Hardware:** Para poder conseguir un consumo ajustado, se hace indispensable que el hardware sea lo más sencillo posible, así como su transceptor radio, esto nos deja una capacidad de proceso limitada.

A pesar de las limitaciones que encontramos en estos dispositivos, existen varias aplicaciones a nivel funcional, entre ellas se destacan:

- ✓ **Agricultura:** Las RIS favorecen una reducción en el consumo de agua y pesticidas, contribuyendo a la preservación del entorno. Adicionalmente, pueden alertar sobre la llegada de heladas, así como ayudar en el trabajo de las cosechadoras. Esta disciplina cubre múltiples prácticas relativas a la gestión de cultivos y cosechas, árboles, flores y plantas, ganado, etc. Por medio de sensores estratégicamente situados, se pueden monitorizar parámetros tales como el clima, la temperatura o la humedad de las hojas, con el fin de detectar rápidamente situaciones adversas y desencadenar los tratamientos apropiados.

- ✓ **Aplicaciones sociales y sanitarias:** El cuidado de personas mayores requiere en la mayor parte de los casos de un seguimiento exhaustivo de sus actividades, lo cual limita su privacidad y al mismo tiempo supone una excesiva carga de trabajo para los cuidadores. Mediante el uso de una RIS situados en puntos estratégicos del domicilio del anciano, así como en objetos de uso cotidiano, los cuidadores pueden monitorizar en tiempo real el comportamiento de las personas mayores, evitando la realización de tareas tediosas y centrándose en aspectos más importantes como es la mejora de su calidad de vida.

Adicionalmente, el cuidado médico tanto en hospitales como fuera de los mismos, por ejemplo la rehabilitación de pacientes, también se beneficia del uso de esta tecnología. Un ejemplo de ello es el proyecto CodeBlue, desarrollado en la Universidad de Harvard.

En este caso se han implementado distintos tipos de sensores para la monitorización de parámetros vitales: tasa de latidos del corazón, concentración de oxígeno en sangre, datos EKG de electrocardiograma, etc. Toda esta información se recoge por los sensores y se distribuye de forma inalámbrica a una PDA u ordenador portátil para su procesamiento. De este modo, cualquier señal de alerta puede detectarse a distancia en tiempo real.

- ✓ **Aplicaciones medioambientales:** El mantenimiento y cuidado de espacios y parques naturales resulta complejo en gran medida por las especiales características de los mismos. Se trata de áreas de grandes dimensiones, en algunos casos de difícil acceso, que están repletos de especies vegetales y animales que hay que preservar, por lo que la supervisión de los mismos debe realizarse empleando métodos lo menos intrusivos posibles. Nuevamente las redes inalámbricas de sensores pueden resultar de gran ayuda en este tipo de tareas. Los sensores, de pequeño tamaño, pueden disimularse con el entorno, procesando los datos de diversos parámetros ecológicos y transmitiendo la información de forma inalámbrica hasta un centro de control, situado

normalmente en la caseta de los guardias forestales. De este modo, se evita en la medida de lo posible la circulación de personas y vehículos por el parque. Entre los parámetros a monitorizar podemos enumerar: temperatura, humedad, crecimiento de árboles y arbustos, desplazamientos de especies, conteo de animales, caudales de ríos, etc.

- ✓ **Aplicaciones civiles:** IrisNet es una arquitectura software desarrollada por Intel para la gestión de redes mundiales de sensores de diversos tipos, incluyendo vídeo, permitiendo el acceso distribuido a dichos sensores de una forma potente y eficiente. Cuando el número de dispositivos crece de forma significativa, como es el caso de telarañas mundiales de sensores, resulta clave disponer de herramientas eficientes para el acceso a los mismos, en especial por el elevado consumo de ancho de banda. En el proyecto IrisNet se han demostrado distintas aplicaciones prácticas que hacen uso de dicha herramienta. Entre ellas se incluyen las siguientes:
 - Localización de plazas libres en parking
 - Vigilancia de niños y personas mayores mediante videocámaras
 - Seguridad del hogar
 - Avisador de riesgo de epidemias (gripe, fiebres, etc.)
 - Monitorización de redes de ordenadores
 - Observatorios terrestres y marítimos (costas)

- ✓ **Aplicaciones militares:** Además de las propias aplicaciones civiles, las redes de sensores inalámbricos encuentran un importante campo de aplicación en misiones militares. Por ejemplo, en la identificación y seguimiento de tropas o vehículos militares, así como en la detección de armas químicas y biológicas.

- ✓ **Aplicaciones de tráfico, incendios, vigilancia, ingeniería civil, etc.**

2. JUSTIFICACIÓN

El propósito de este proyecto es evaluar y optimizar los criterios de selección de ruta en redes inalámbricas de sensores aplicados a un protocolo en enrutamiento basado en conexiones tipo árbol, donde cada uno de los nodos debe establecer una conexión con un nodo padre para alcanzar un único destino, el sumidero.

Hallando estos criterios podremos estudiar los parámetros del algoritmo para mejorar las prestaciones entregadas por el protocolo; brindando menor retardo en la información, menor número de pérdida de paquetes, tiempo de estabilización de la red y sobre todo un encaminamiento más directo con el nodo destino.

Basado en las valoraciones realizadas, podremos aplicar los parámetros óptimos hallados a un protocolo que cumpla con las características necesarias para el enrutamiento de datos en una red inalámbrica de sensores, cuyo fin sea dirigir la información de todos los nodos a una estación base ó sumidero.

3. OBJETIVOS

- ✓ Analizar los diferentes algoritmos de encaminamiento para redes de sensores inalámbricas.
- ✓ Optimizar un protocolo para obtener la mejor ruta de encaminamiento entre los nodos fuente y destino.
- ✓ Reducir el tiempo que tarda la red en estabilizarse.
- ✓ Implementar un nuevo proceso/mecanismo que permita buscar mejores rutas para optimizar las actuales.
- ✓ Mejorar las latencias de los paquetes entre la fuente y destino; con ello prolongaremos el tiempo de vida de la red evitando enviar paquetes por rutas no óptimas.

4. BASES TEORICAS

4.1. Protocolos de Encaminamiento

Como bien sabemos en la actualidad existe un gran numero de protocolos usados para el manejo de la información, pero no todos los protocolos existentes son aplicables a las RIS ya que el medio en el que emplean es totalmente diferente. En las redes cableadas la pérdida de información se debe a la gran cantidad de tráfico que corre por el medio, mientras que en las RIS el medio usado para la comunicación es más hostil, el aire, donde los dispositivos y las transmisiones están expuestas a las implicaciones medioambientales, al ruido y todo tipo de inconveniente que se pueda dar en el medio, todo esto sin mencionar que los dispositivos deben “luchar” por acceder al medio de comunicación. Es por ello que el enfoque que se le da a un protocolo para redes ad-hoc es muy diferente al usado en redes cableadas, ya que en este ultimo caso los protocolos no están pensados para trabajar con tanto “stress”.

Los protocolos usados en las redes inalámbricas se clasifican en tres bloques como se puede apreciar en la figura 4.1.1. Estos protocolos se han desarrollado ante la necesidad de controlar el encaminamiento en las redes ad-hoc, teniendo en cuenta las limitaciones que tienen los dispositivos.

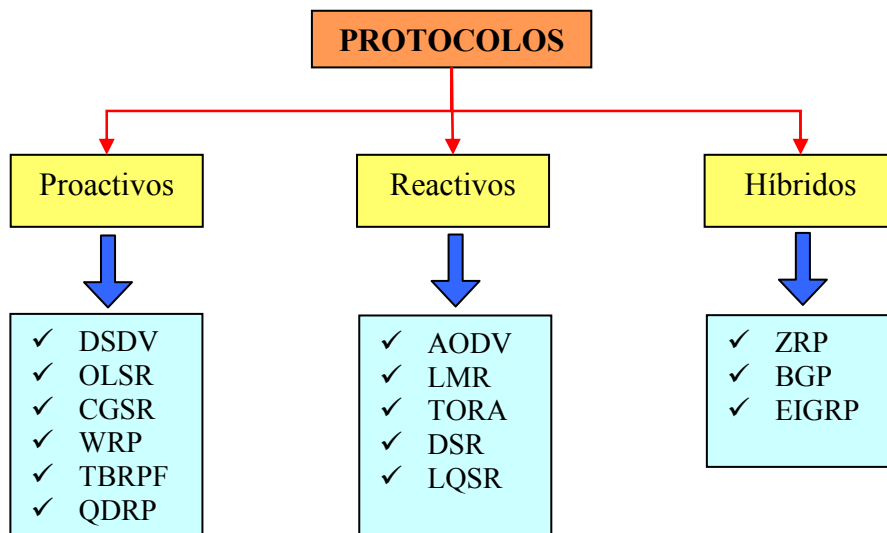


Figura 4.1.1. Clasificación de los protocolos según su funcionamiento.

Protocolos Proactivos

Son aquellos que tienen periódicamente actualizadas las tablas de encaminamiento de todos los nodos de la red aunque no estén enviando información. Cuando surge algún cambio entre las conexiones de la red la tabla de encaminamiento se actualiza y el protocolo elige la ruta más óptima para enviar información. Este proceso se debe al intercambio de mensajes de control, con lo que se incrementa el consumo de la batería, el número de paquetes enviados por la red.

Protocolos Reactivos

Estos protocolos únicamente tienen rutas de encaminamiento en sus tablas cuando un nodo origen tiene que realizar una comunicación con otro nodo en la red. Los protocolos reactivos al iniciar una comunicación y no tener una ruta para llegar al nodo destino, se envía un mensaje de descubrimiento de ruta y cuando recibe la respuesta a dicho mensaje, añade esta ruta en su tabla de encaminamiento. Es entonces cuando es posible la comunicación con el destino. El mayor inconveniente es la latencia que se añade al primer paquete de la transmisión por esa nueva ruta, pero a su vez mejora las prestaciones de la batería en los nodos. Dentro de estos protocolos existen dos clases de protocolos:

- **Encaminamiento origen:** La ruta de los nodos por donde tiene que pasar la información es almacenada en las cabeceras de los paquetes, de este modo los nodos intermedios no necesitan tablas de encaminamiento, ya que solo basta con leer las cabeceras para saber a quien reenviar la información. Este tipo de encaminamiento no es aconsejable usarlo en redes extensas, ya que a medida que el mensaje pasa por cada nodo se incrementa la cabecera del paquete.
- **Encaminamiento salto a salto:** En este caso la ruta la escoge cada nodo en cualquier momento, ya que cuando se envía la información la cabecera del paquete contiene la dirección del nodo destino y la dirección del siguiente salto. En este caso se adapta más rápido a los cambios de la topología pero se genera un gasto superior de recursos en los nodos intermedios ya que tienen que almacenar en tablas de encaminamiento las rutas correspondientes.

Protocolos Híbridos

Los protocolos híbridos son una mezcla de los protocolos proactivos con los reactivos. El fin de éstos es usar las mejores características que ofrecen ambos. Los

protocolos dividen las redes en zonas, y los nodos que están mas lejos del destino utilizan encaminamiento reactivo, mientras que los que están mas cerca utilizan encaminamiento proactivo; como es el caso de ZRP^[9] (The Zone Routing Protocol). Otros ejemplos de protocolos híbridos son IS-IS^[10] de OSI (Intermediate System to Intermediate System) y EIGRP^[11] (Enhanced Interior Gateway Routing Protocol) de CISCO.

Teniendo en cuenta los aspectos de cada uno de los enrutamientos mencionados anteriormente, daremos a continuación una descripción del funcionamiento de algunos de los protocolos mas usados en RIS, a su vez que un nuevo protocolo que se adapta a las características de un enrutamiento basado en conexiones tipo árbol, diseñado específicamente para redes inalámbricas de sensores.

4.1.1. Ad-hoc On-demand Distance Vector – AODV

AODV es un protocolo de encaminamiento para redes móviles ad-hoc (MANETs) y redes ad-hoc inalámbricas. AODV es la evolución de su anterior protocolo DSDV (Destination-Sequenced Distance-Vector). El DSDV inundaba la red de mensajes de control, de forma que la red se congestionaba y limitaba la duración de las baterías de los terminales.

El AODV es uno de los protocolos más utilizados de los algoritmos reactivos, siendo ideal para las redes Ad-Hoc. Este protocolo intercambia mensajes cuando necesita establecer una comunicación, es decir, envía mensajes a los vecinos para calcular cada ruta. Gracias a las mejoras incorporadas en AODV se evita la problemática que tiene DSDV, pero por el contrario se incrementa la latencia en el primer paquete a enviar cada vez que se calcula la ruta.

Las características del protocolo son:

- ✓ Señalización de control baja
- ✓ Señalización de procesamiento mínima
- ✓ Prevención de bucles
- ✓ Funciona sólo con enlaces bidireccionales

Cada nodo tiene asociada una tabla de encaminamiento que utiliza para poder establecer enlaces con otros nodos. Estas tablas de encaminamiento contienen los siguientes campos:

- ✓ Dirección IP Origen
- ✓ Tiempo de Vida (TTL)
- ✓ Dirección IP Destino
- ✓ Número secuencia Destino
- ✓ Contador de saltos (hop count)

Aparecen los campos de las direcciones IP de la fuente y de la IP del destino para saber en todo momento de donde vienen los paquetes y hacia donde se deben dirigir.

También aparece un campo con el número de secuencia del nodo destino que sirve para distinguir entre información nueva e información antigua y de esta forma evitar formación de bucles y transmisiones de rutas antiguas.

Otro parámetro que se almacena en las tablas de encaminamiento es el tiempo de vida. Este sirve para evitar que viajen paquetes perdidos por la red y utilizar enlaces de los que no se conoce su estado desde hace mucho tiempo. Cuando a un destino le llegan dos paquetes desde la misma fuente por caminos distintos, el campo hop count muestra el número de saltos que han tenido que hacer para cada una de las rutas. De esta forma se sabe cual de ellas es la ruta más corta y la que tiene que seleccionarse para hacer el envío de información. Cada vez que se quiere comunicar una fuente con un destino, se inicia un proceso de descubrimiento de ruta, que finaliza cuando recibe un paquete con la ruta calculada.

Existe otro concepto conocido como mantenimiento de ruta, que sirve para actuar en caso de que se rompa un enlace a lo largo de una ruta. Esto se consigue dando tiempo de vida a las rutas descubiertas antes de considerarlas como invalidas.

Descubrimiento de Rutas

Cuando un nodo quiere transmitir un paquete a un destino, lo primero que debe hacer es buscar en su tabla de encaminamiento a ver si existe una ruta hacia este destino previamente calculada. En el caso de encontrarla no iniciaría ningún proceso de descubrimiento de ruta, supondría que la que tiene almacenada en su tabla de encaminamiento es correcta y está actualizada. En el caso contrario, comenzará el proceso de descubrimiento para encontrar un camino válido.

El proceso comienza con el envío de un paquete RREQ (Route Request) en modo broadcast. Este paquete llega a los nodos vecinos que se encuentran a un salto de distancia y estos a su vez lo reenvían a sus vecinos y así sucesivamente hasta llegar al destino. Cualquier nodo que durante el proceso de búsqueda conozca la ruta hacia

el destino, puede contestar con un paquete de RREP al nodo origen indicando la ruta que necesita.

Mientras se va realizando el proceso de búsqueda, todos los nodos van actualizando las tablas de encaminamiento.

En el formato del paquete RREQ del protocolo de encaminamiento AODV, nos encontramos los siguientes campos:

- ✓ Dirección IP Origen
- ✓ Número de Secuencia del Origen
- ✓ Dirección IP del Destino
- ✓ Número de Secuencia del Destino
- ✓ RREQ Identificador
- ✓ Contador de saltos (hop count)

Uno de los campos es el identificador que se va modificando cada vez que se genera un envío de RREQ. Esto sirve para que los nodos que lo vayan recibiendo (nodos intermedios) sepan si el paquete es idéntico al anterior (tiene el mismo identificador) y deben descartarlo, o por el contrario, si deben retransmitirlo (porque el identificador de paquetes es distinto).

Mantenimiento de Rutas

Cuando una ruta es encontrada se le da un tiempo de vida y se considera útil hasta que un temporizador asociado a esta ruta no expire. Esto se utiliza para no tener que iniciar un descubrimiento de ruta para cada mensaje de información que se quiere enviar. Durante una comunicación entre el nodo fuente y el destino puede ocurrir que alguno de los nodos modifique su posición. Esto puede dar lugar a que se rompa el enlace y que la ruta quede inutilizada. El nodo vecino al enlace roto debe ser el encargado de informar al resto de nodos sobre dicho suceso. Para ello se utiliza el envío del mensaje RERR (Route Error).

El mensaje viene a ser igual que el mensaje RREP pero con un número de salto igual a infinito. Es decir, el nodo que detecta roto el enlace envía un RERR con valor de hop count hacia la fuente de valor infinito, lo que hace que cualquier otra ruta sea mejor y deban reencaminarse los paquetes por otro sitio. De esta manera, el nodo fuente decide si ha terminado la comunicación con el nodo destino o si por el contrario debe iniciar un nuevo proceso de descubrimiento de ruta.

AODV presenta una serie de opciones de optimización, como la posibilidad de reparar a nivel local un enlace roto que forma parte de una ruta activa. Cuando se rompe un enlace, en lugar de enviar un paquete de RERR a la fuente, el nodo que ha detectado la rotura puede intentar repararlo localmente enviando un RREQ con el número de secuencia del destino incrementado en uno hacia ese destino. Los paquetes de datos se quedan almacenados en este nodo esperando recibir un RREP con una nueva ruta disponible hacia el destino. Si este nuevo procedimiento de Descubrimiento de Ruta no tiene éxito y el RREP no llega, entonces sí que será necesario informar a la fuente acerca de la rotura del enlace enviándole un paquete RERR.

4.1.2. Dynamic Source Routing - DSR

DSR es un protocolo reactivo creado específicamente para redes Ad-Hoc, que envía información únicamente cuando es necesario, con lo que hay un ahorro de energía en la red, disminuyendo el número de transmisiones y ahorrando batería. Usando DSR la red no necesita infraestructura ni administración ya que posee auto-organización y auto-configuración. Como contrapartida tenemos la latencia inicial al descubrir una nueva ruta, y al ser un protocolo basado en encaminamiento fuente, la cabecera del paquete va creciendo a medida que va pasando por los nodos intermedios, cosa que conlleva a un notable incremento en el tamaño de las cabeceras de los paquetes.

Este protocolo consta de dos mecanismos: el descubrimiento de ruta y el mantenimiento de ruta.

Descubrimiento de ruta

Este mecanismo se ejecuta cuando el nodo emisor quiere enviar un paquete y no tiene en su tabla de encaminamiento una ruta hacia el nodo destino, si no es así, simplemente hace uso de su tabla.

En caso de no tener una ruta de encaminamiento en su tabla el nodo emisor tiene que utilizar el mecanismo de descubrimiento de ruta para encontrar dinámicamente una nueva hacia el nodo destino; el emisor envía un paquete RREQ (Route Request) en modo difusión (broadcast). Este paquete contiene la dirección del nodo fuente, la dirección del nodo destino y un identificador asociado al paquete RREQ. Cualquier nodo intermedio que recibe un paquete RREQ “no duplicado” agrega su dirección a la cabecera del paquete.

Cuando el nodo destino recibe el paquete, este envía un mensaje RREP (Route Reply) de regreso al origen, (Ver figura 4.1.2.1.). Sin embargo los nodos de la red pueden tener información de rutas almacenadas en su caché, fruto de anteriores descubrimientos de ruta y de paquetes enviados. Por eso, si un nodo intermedio recibe un RREQ, y este nodo tiene en su tabla de encaminamiento el nodo destino solicitado, el nodo envía un RREP a la fuente con una copia del registro de la ruta acumulada en el RREQ, más la ruta que contenga su tabla de encaminamiento. Si en su defecto es el nodo destino, únicamente se transmite el paquete RREP con una copia del registro de las rutas acumuladas más la suya.

Una vez el paquete alcanza el destino éste deberá enviar el paquete RREP, el cual tiene que ser enviado por la misma ruta que le indica el mensaje RREQ pero en sentido inverso, pudiendo utilizar rutas bidireccionales.

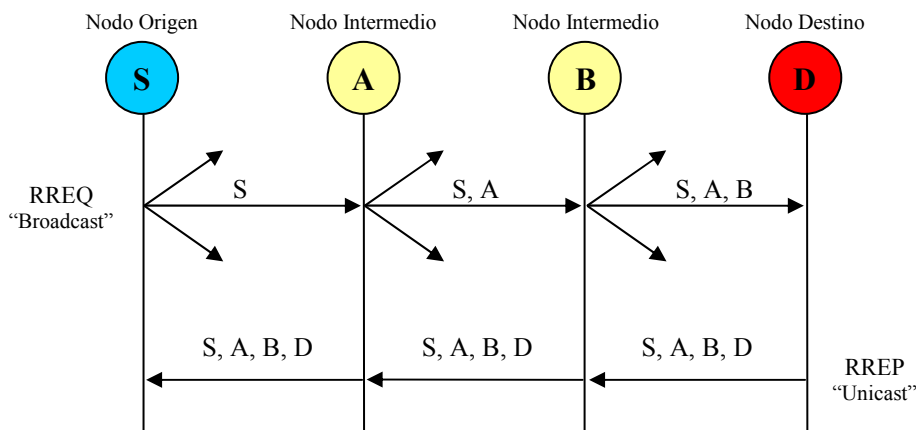


Figura 4.1.2.1. Mecanismo de descubrimiento de ruta del protocolo DSR.

Una vez aprendida una ruta gracias al paquete RREQ, se guarda en una caché temporal, la cual tiene un tiempo asignado para borrarse en el caso de no ser utilizada. Este temporizador tiene asignados 300 segundos según la especificación de DSR para que la ruta caduque y poder eliminarla. Si una ruta que esta en caché se usa antes de que se venza el temporizador, dicho temporizador se inicializa nuevamente a 300 segundos, con lo que se estaría logrando ahorrar consumos en el dispositivo y en el medio.

Mantenimiento de ruta

El mecanismo de mantenimiento de ruta es el encargado de avisar si una ruta se rompe. Cuando se está transmitiendo, cada nodo es el encargado de controlar el estado del tramo en el que se encuentra, es decir, verifica que el siguiente nodo está recibiendo la información.

En el momento que se descubre que una ruta ha caído, el nodo que lo descubre envía un paquete RRER al nodo fuente, y éste al recibirlo procesa el RRER y borra la ruta de su caché de encaminamiento. Si hay otra ruta en la caché se envían los datos por dicha ruta, en caso contrario se realiza el mecanismo de descubrimiento de ruta.

En el draft se especifican tres formas para el descubrimiento de rotura de una ruta. Reconocimientos de la capa de enlace, reconocimientos pasivos, y Reconocimientos de la capa de red.

- ✓ **Reconocimientos de la capa de enlace:** Estos reconocimientos se realizan desde la capa de enlace y se utilizan siempre que es posible. A medida que envían datos, el siguiente receptor, ya sea el final o sea un nodo encaminador, tiene que ir recibiendo reconocimientos de sus paquetes enviados. El emisor no debe borrar ningún paquete hasta que no llegue su reconocimiento del siguiente nodo.
- ✓ **Reconocimientos pasivos:** Este mantenimiento se puede utilizar cuando los reconocimientos de capa de enlace no están disponibles.

El mecanismo que utiliza es el siguiente: Un terminal envía un paquete a un nodo adyacente al que no sea el destino final, cuando lo recibe, tiene que reenviar el paquete según su tabla de encaminamiento. Como el medio es compartido por todos los nodos, el paquete que ha reenviado el nodo intermedio, será recibido por el terminal que inicio la transmisión y por el que indica en la tabla de encaminamiento del nodo intermedio. De esta forma el nodo transmisor se asegura que al siguiente nodo le ha llegado el paquete transmitido por él.

Para detectar una pérdida de ruta se tiene en cuenta el tiempo de reconocimiento pasivo. Cuando el temporizador expira se reenvía el paquete sin necesidad de notificarlo a la capa de red. Si no se recibe ninguna retransmisión después de un número de intentos, se crea el paquete RRER para notificarlo al emisor.

- ✓ **Reconocimientos de la capa de red:** Este modo de detección de rupturas de rutas se realiza cuando no hay ningún otro mecanismo de detección posible. El

nodo que transmite tiene que solicitar un reconocimiento en la capa de red del siguiente nodo cada determinado tiempo. Para realizar tal función el nodo tiene que introducir en la cabecera de opciones del paquete DSR la opción de petición de reconocimiento. En el momento que recibe un paquete que contenga una petición de reconocimiento éste tiene que mirar si contiene su dirección IP para procesarla y enviarle una respuesta con un reconocimiento, en caso contrario no se procesa el paquete. Una vez que pase un tiempo determinado (basado en el algoritmo de estimación del RTT que se usa en TCP) y no llegue ningún paquete de reconocimiento, se realizarán 2 retransmisiones de petición de reconocimiento más, para asegurarse que es una ruptura. En este caso el nodo que se percata de la ruptura envía al anterior un paquete RRER indicándole la ruptura del enlace.

4.1.3. Quick Dirty and Routing Protocol - QDRP

QDRP es un protocolo de enrutamiento diseñado para redes inalámbricas de sensores (redes ad-hoc), donde los nodos solo se dedican a enviar información de forma periódica a un nodo destino en común, de modo que dos nodos de la red no compartirán información entre sí, siendo el destino el único lugar donde la información es recolectada para su posterior análisis.

Cada uno de los nodos que componen la red debe alcanzar una conexión con un padre para encaminar por medio de éste los paquetes hasta el destino. En un protocolo donde las conexiones entre dispositivos son tipo “árbol”, los nodos solo deben enviar los paquetes a su padre en un nivel superior y así sucesivamente hasta alcanzar la raíz del árbol, donde se encuentra el nodo sumidero.

El funcionamiento del protocolo se basa en iniciar/activar cada uno de los dispositivos, luego cada uno de ellos pasa por una máquina de estados donde se va completando su configuración de acuerdo a los parámetros con que se inicie el dispositivo. Éste puede ser un nodo sensor (nodo sin funciones de coordinador) o un nodo sumidero.

Este protocolo a diferencia de los otros se basa en que los nodos no tienen que preocuparse por saber quien es el nodo destino, solo basta con saber quien es el padre del nodo, de esta forma solo se limita a reenviar información a los niveles superiores del árbol hasta alcanzar el origen del mismo.

Al iniciarse los dispositivos éstos van pasando por una máquina de estados. Los posibles estados en los que puede estar un nodo son los siguientes:

- OFF
- BOOTING
- OFFLINE
- CONNECTING
- ONLINE
- DISCONNECTING

En los estados se hace uso del envío y recepción de mensajes pertenecientes al protocolo. Estos mensajes son:

- ✓ **SHOW_OFF:** Se emite este mensaje cuando el nodo está en estado ONLINE y el máximo de hijos aceptados por nodo no ha sido aun alcanzado. Emitido solo en el estado ONLINE.
- ✓ **CONNECT:** Se utiliza para solicitar al candidato a nodo padre que acepte nuestra conexión. Emitido solo en el estado OFFLINE.
- ✓ **CONNECTION_ACCEPTED:** Cuando el padre recibe un CONNECT y puede aceptar más conexiones responde con este mensaje. Si recibe un CONNECTION_RENEW responde de igual manera con un CONNECTION_ACCEPTED para renovar la conexión con un nodo que ya era hijo suyo. Emitido solo en el estado ONLINE.
- ✓ **CONNECTION_REJECTED:** Este mensaje es emitido cuando el padre no puede aceptar más conexiones. Emitido solo en el estado ONLINE.
- ✓ **CONNECTION_RENEW:** Se usa para renovar una conexión con un nodo hijo. Emitido solo en el estado ONLINE.
- ✓ **DISCONNECT:** Se envía DISCONNECT para finalizar una conexión con el nodo padre. Emitido solo en el estado ONLINE.
- ✓ **DISCONNECTION_OK:** La desconexión con el nodo padre se ha realizado satisfactoriamente. Este responde con un DISCONNECTION_OK al nodo que era su hijo. Emitido solo en el estado ONLINE.
- ✓ **ROUTE_CRASH:** Si el mensaje es de nuestro padre, se ejecuta el proceso de desconexión, y se informa a todos los hijos de dicha rotura, y por consiguiente la caída de toda la rama del árbol de conexión. Emitido solo en el estado ONLINE.

Ahora, conociendo los mensajes usados por el protocolo veremos una breve descripción de cada estado. (Ver relación de estados en la figura 4.1.3.1.)

- ✓ **Estado OFF:** En este punto el dispositivo se encuentra apagado, de modo que no se tiene en cuenta ningún tipo de envío ni recepción de paquetes.
- ✓ **Estado BOOTING:** El dispositivo se ha encendido/iniciado. Si éste es un nodo sensor, pasa al estado OFFLINE donde activará sus temporizadores para buscar un nodo padre al cual conectarse.

Si el nodo es un sumidero se configura con sus respectivos parámetros y pasa automáticamente al estado ONLINE, ya que al actuar como sumidero no depende de una conexión con ningún otro nodo, siendo este quien recibe toda la información de los nodos hijos.

En este estado tampoco se tiene en cuenta ningún tipo de envío ni recepción de paquetes.

- ✓ **Estado OFFLINE:** Cuando el nodo pasa a éste estado, activa un temporizador para conectarse a la red. Cada vez que un nodo recibe un mensaje SHOW_OFF de un nodo vecino, toma sus datos y los guarda en una lista de posibles conexiones, y una vez vencido el temporizador para unirse a la red revisa si hay nodos en la lista que puedan servir como candidatos a padre, de no ser así recarga de nuevo su temporizador para buscar en el siguiente ciclo de tiempo otro candidato.

En el caso de existir uno o más candidatos el nodo pasa al estado CONNECTING.

Cualquier tipo de paquete diferente a SHOW_OFF es descartado, ya que no es posible procesar o encaminar información en este estado.

- ✓ **Estado CONNECTING:** Al iniciar el presente estado se comprueba si la lista de posibles conexiones contiene al menos un nodo al cual conectarse, sino, retorna al estado OFFLINE.

Si la lista tiene uno o más candidatos se elige entre ellos el que este más cerca del sumidero. Si entre la lista hay uno o más nodos con igual número de saltos al sumidero se escoge el que tenga menor número de hijos y se le envía un mensaje CONNECT para solicitar nuestra conexión con dicho nodo.

Tras enviar el mensaje de conexión activamos un temporizador para esperar por la respuesta del padre. Si el mensaje de confirmación no llega en 1/6 del tiempo especificado en el temporizador se le reenvía el mensaje, y así hasta cumplir 6 ciclos de tiempo. Si en este tiempo no hay una conexión realizada se elimina de la lista al candidato a padre y se prueba de nuevo lanzando el estado CONNECTING.

Si por el contrario, recibimos un mensaje de CONNECTION_ACCEPTED, actualizamos nuestra información en base a la del padre y pasamos al estado ONLINE.

- ✓ **Estado ONLINE:** Estado en el que se encuentra un nodo después de lograr una conexión exitosa con el padre. En este estado los nodos pueden transmitir y encaminar paquetes de datos hacia el nodo sumidero. Todos los paquetes que recibe un nodo sensor es reenviado en sentido ascendente por el árbol hasta que se alcanza el destino (sumidero).

Al iniciarse este estado se activa un temporizador para emitir mensajes de SHOW_OFF de forma periódica. El mensaje es lanzado siempre y cuando el nodo tenga espacio para recibir a más hijos. Por defecto el número de hijos está establecido en 5 para un nodo sensor. Un nodo sumidero puede aceptar tantos hijos como quiera o como se le especifique ya, que dispone de recursos ilimitados.

El mantenimiento de ruta en este estado se realiza a través de un temporizador de conexión asociado a este nodo, que expira cada 30 segundos. Para evitar que el temporizador expire, cada 5 segundos se envía un mensaje de CONNECT al padre, y espera como respuesta un mensaje CONNECTION_ACCEPTED. Recibido este mensaje se renueva el temporizador a 30 segundos. En caso de no recibir el mensaje de respuesta por parte del padre el nodo vuelve a intentar la conexión 5 veces más.

Si el temporizador no es renovado por el padre, el nodo cancela todos los temporizadores asociados a él y envía un mensaje de ROUTE_CRASH para comunicar a sus hijos la caída de la ruta usada, luego el nodo pasa al estado DISCONNECTING.

- ✓ **Estado DISCONNECTING:** Estado en el cual el nodo cancela todos los temporizadores con el padre y sus hijos antes de pasar al estado OFFLINE.

Un nodo que se encuentre en este estado no puede enviar ni encaminar paquetes, ya que no tienen una conexión asociada.

La relación entre todos los estados mencionados se puede apreciar en la figura 4.1.3.1.

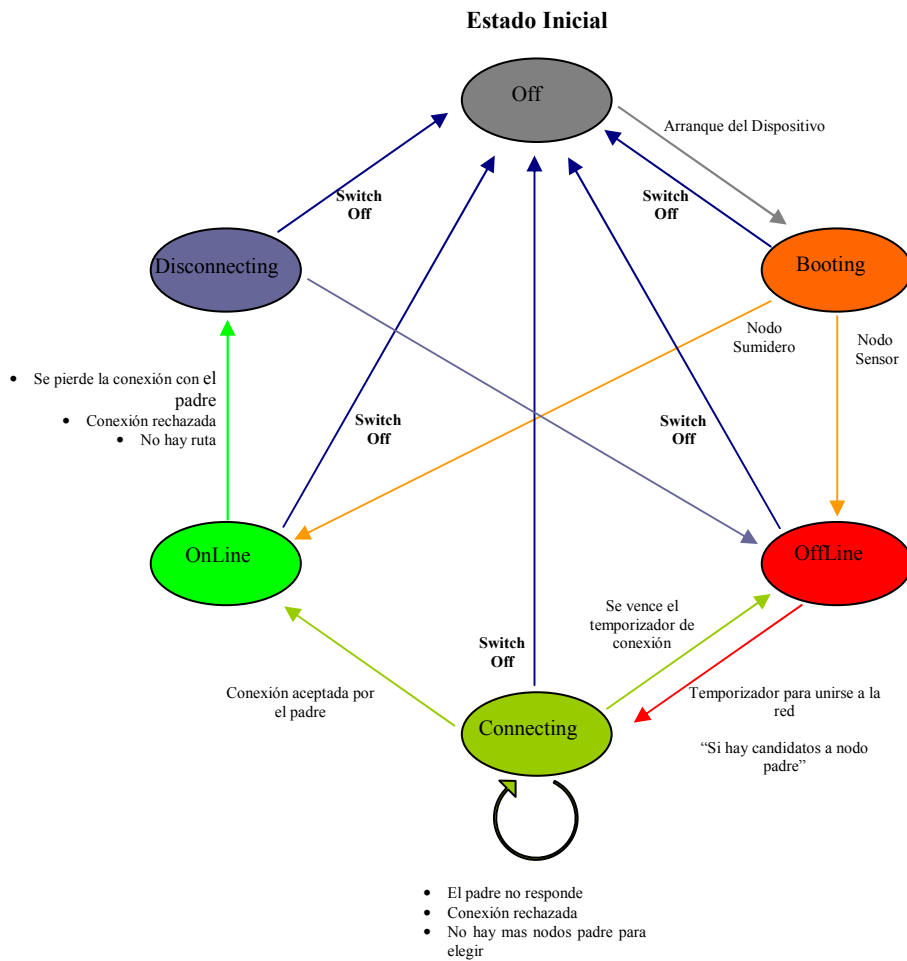


Figura 4.1.3.1. Diagrama general de estados del protocolo QDRP.

4.2. Criterios para elegir un protocolo dirigido a RIS

Como bien sabemos, las redes inalámbricas de sensores están compuestas por dispositivo de un tamaño muy pequeño, en algunos casos diminutos, lo cual limita el hardware que lo compone. Pese a estas limitaciones se debe tener en cuenta una de mayor importancia que es la vida útil que tiene la batería con la que funciona el dispositivo.

Es por este motivo que el protocolo que se usa en una RIS debe ser muy específico y cumplir con unos requerimientos mínimos que nos brinde un funcionamiento óptimo y sobre todo que administre de la mejor forma el hardware para que la batería del dispositivo perdure el mayor tiempo posible. Todo ello se puede lograr reduciendo al máximo las transmisiones que genera cada nodo para comunicarse con sus vecinos.

Entre los criterios mas importantes para seleccionar un protocolo de esta índole se debe tener en cuenta que:

- ✓ Se obtengan las mejores rutas de encaminamiento
- ✓ Halla un mínimo de transmisión de mensajes
- ✓ Presente baja latencia de los paquetes entre la fuente y el destino
- ✓ La red se estabilice en el menor tiempo posible
- ✓ No se sobrecargue la red con mensajes de control
- ✓ Sea tolerante a fallos

Si un protocolo cuenta con estas características será idóneo para implantar en sistemas que cuente con bajos recursos.

En la actualidad tenemos diversos protocolos que se usan para trabajar con RIS pero no todos cumplen con las especificaciones necesarias. Por otra parte, Quick and Dirty Routing Protocol (QDRP) es un protocolo que fue diseñado específicamente para trabajar con redes de sensores inalámbricas que se adapta a nuestras necesidades a la hora de simular los escenarios de trabajo, ya que presenta baja latencia en la información (extremo a extremo), tiene un tiempo de respuesta bajo en cuanto la estabilización de red y sobre todo, la forma para establecer sus conexiones lo hace mas optimo en cuanto a otros protocolos. Ver estudios preliminares^[8].

5. HERRAMIENTA DE TRABAJO

5.1. Herramienta de trabajo

Actualmente existen diversas posibilidades a la hora de elegir un simulador de redes. Aunque existen varias opiniones ante el funcionamiento y la calidad de cada uno, la gran parte de la comunidad académica y científica que han evaluado diferentes simuladores, coinciden en ratificar que NS2 es una herramienta que ofrece unos resultados óptimos y que además se trata de un programa de código abierto (desarrollado bajo los términos de GPL) frente a la mayoría de opciones existentes de pago. Como pasa en todo el software de licencia pública, cada uno tiene la posibilidad de aportar sus ideas para ir mejorando y desarrollando esta herramienta, que aunque ya se encuentre entre una de las principales, es una buena apuesta a tener en cuenta de cara al futuro debido a su utilización y continua mejora por parte del sector científico; por tanto, todas estas razones son las que han hecho que nos inclinemos por NS2.

5.2. Network Simulator 2 - NS2

“Network Simulator 2” ó NS2 es un simulador de redes basado en eventos discretos que está dirigido a la investigación de redes y que proporciona soporte para la simulación de enrutamiento, protocolos multicast y protocolos IP, tales como TCP, UDP, RTR y SRM sobre redes cableadas y redes inalámbricas (locales ó satelitales).

NS2 tiene muchas ventajas que hacen de él una herramienta muy útil, tales como la capacidad de detallar gráficamente el tráfico de la red y la de monitorizar los paquetes que hay en ella a través del Network Animator, (ver figura 5.2.1.), además del soporte para múltiples protocolos, entre los que encontramos los de uso más común como lo son:

- Ad hoc On-Demand Distance Vector Routing – AODV
- Dynamic Source Routing – DSR
- Destination-Sequenced Distance Vector – DSDV

Adicionalmente, NS2 soporta varios algoritmos de encaminamiento y manejo de colas (queuing). El enrutamiento LAN y broadcast son parte de los algoritmos de

encaminamiento. Estos algoritmos incluyen manejo de colas como “fair queuing”, “deficit round-robin” y FIFO.

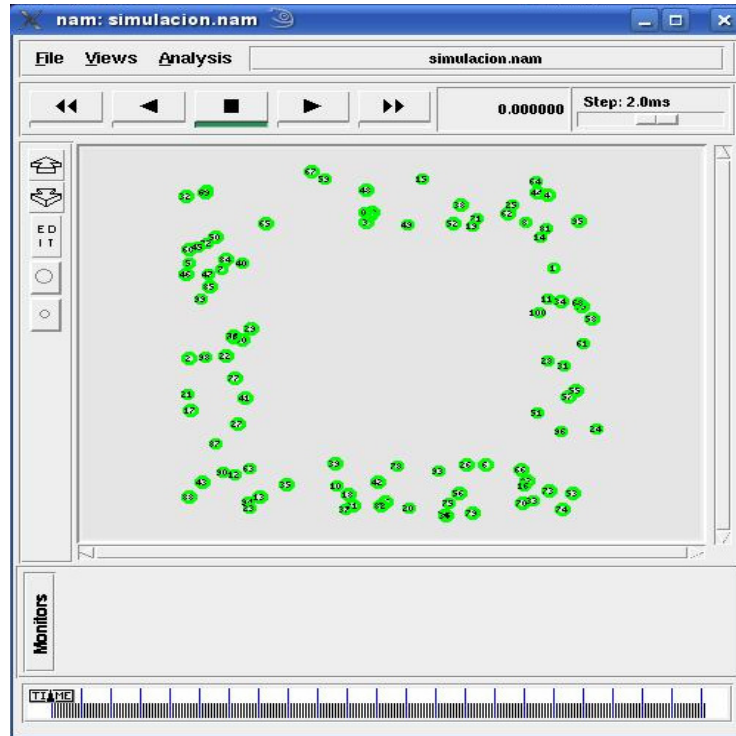


Figura 5.2.1. Network Animator – NAM.

El funcionamiento interno del NS2 se puede observar de forma simplificada mediante la figura 5.2.2.

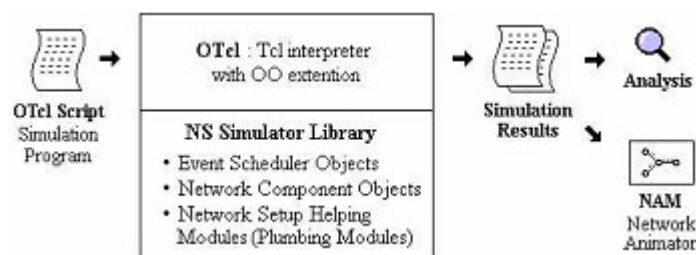


Figura 5.2.2. Vista simplificada del funcionamiento de NS2.

Como se puede observar, se inicia con un script en OTcl que viene a hacer lo que el usuario codifica para simular. Es la única entrada que da el usuario al programa. El resto es el procesamiento interno de NS. La simulación queda en un archivo que

puede ser bastante incomodo de leer o analizar para el usuario, sin embargo, usando una aplicación especial como NAM se puede mostrar mediante una interfaz grafica los resultados obtenidos en la simulación.

El script es un archivo escrito en Tcl orientado a objetos, es decir, OTcl, que tiene diversos componentes internos que se muestran en el cuadro del medio de la figura 5.2.2.

En estos componentes se configura la topología de la red, calendariza los eventos, carga las funciones necesarias para la simulación, planifica cuando iniciar o terminar el tráfico de un determinado paquete, entre otras cosas.

6. OPTIMIZACION DE UN PROTOCOLO PARA REDES INALÁMBRICAS DE SENSORES

Como se ha citado anteriormente, la selección del camino óptimo hacia el nodo sumidero resulta fundamental para obtener las mejores prestaciones en el sistema. En este trabajo se propone un nuevo protocolo (EQDRP), que mejora sustancialmente el criterio de selección de nodo padre en el árbol de encaminamiento. Este nuevo criterio se basa en la formula:

$$k * \left(\frac{1}{Hops_to_sink + 1} \right) + (k_1 * LQI) + k_2 * \left(\frac{PNode_energy}{Num_child_nodes + 1} \right)$$

Donde se tienen en cuenta los siguientes factores:

- ✓ Saltos al sumidero, (Hops_to_sink)
- ✓ Numero de hijos del nodo, (Num_child_nodes)
- ✓ Nivel de calidad del enlace, (LQI – Link Quality Indicator)
- ✓ Porcentaje de energía (batería) del nodo, (PNode_energy)

Esta expresión presenta determinados índices (k, k₁, k₂) que van a condicionar el peso de cada uno de estos factores en la decisión final.

El rango de valor de las variables se obtuvo mediante simulaciones, encontrando que para cualquier otro valor de k, k₁ ó k₂ menor al primer índice todos los valores serán iguales, comportándose de igual manera para el índice mayor.

$$k = 102, 178, 255$$

$$k_1 = 0.2, 0.5, 1$$

$$k_2 = 0.15, 1, 2$$

Teniendo entonces un máximo y un mínimo de dichos valores, tomamos un tercero para tener un valor intermedio del rango.

El valor final para cada uno de los índices se halló mediante una serie de simulaciones donde se prueba con todos los valores de k, k₁ y k₂ obtenidos, en las cuales se medía el tiempo de estabilización de la red y la latencia promedio

sobre dos escenarios diferentes. El primer escenario corresponde a la figura 6.1. y el segundo escenario a la figura 6.2.

En ambos escenarios se trabaja con unos parámetros generales para el archivo de simulación “tcl” correspondiente, estos parámetros son:

MAC	802.15.4
Beacons	No habilitado
Alcance de los transeptores de radio	10 mts
Cantidad de sumideros	1
Id nodo sumidero	32
Id nodo Coordinador WPAN	32
Modo de inicio de los nodos	Consecutiva
Tiempo del inicio primer nodo	0.0 seg.
Tiempo de inicio entre cada nodo	0.05 seg.
Trafico utilizado en la simulación	Constant Bit Rate (CBR)
Tamaño de los paquetes	32 bytes
Inicio de transmisión del trafico	100.0 seg.
Fin de transmisión del trafico	400.0 seg.

Tabla 6.1. Parámetros generales usados en el archivo de simulación.

La única diferencia en los parámetros usados en las dos simulaciones, radica en la cantidad de nodos.

La simulación dura 400 segundos. El tráfico de datos CBR comienza a partir de los 100 segundos. Con el fin de evitar que todos los nodos de la red comiencen a transmitir al mismo tiempo, se ha decidido que las transmisiones de datos CBR comiencen en distintos instantes en cada nodo, según la siguiente fórmula:

$$\text{tiempo_inicio_trafico_CBR}(\text{nodo } \$i) = 100 + ((\$i \% 20) * 0.250)$$

donde \$i es el número identificador del nodo y % indica operación módulo. De esta manera evitamos una sincronización irreal en el inicio de transmisión de datos.

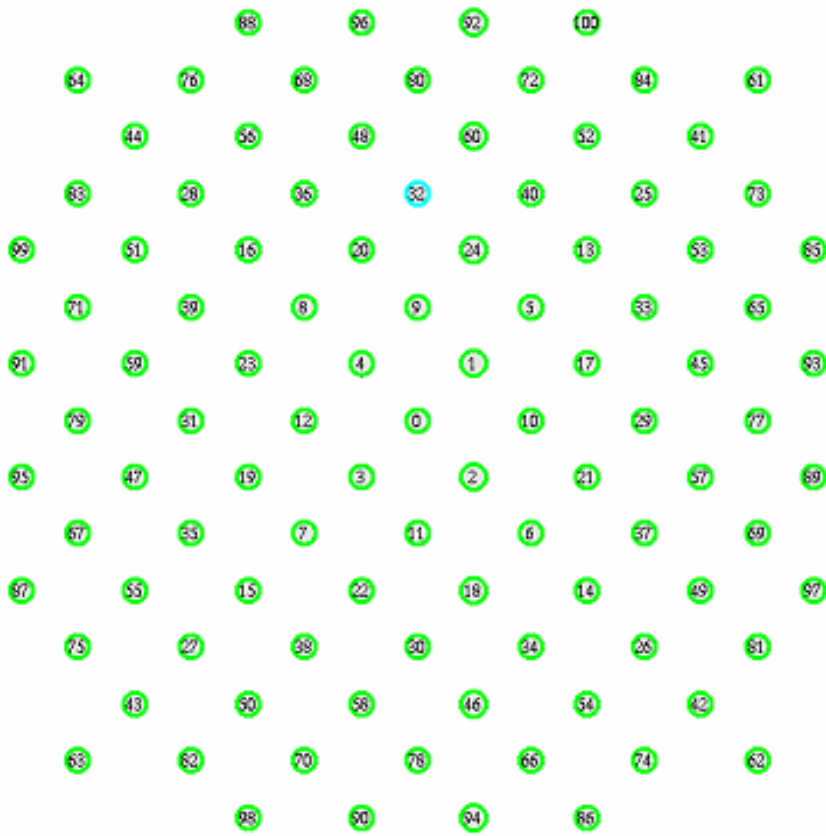


Figura 6.1. Topología de pruebas N° 1.

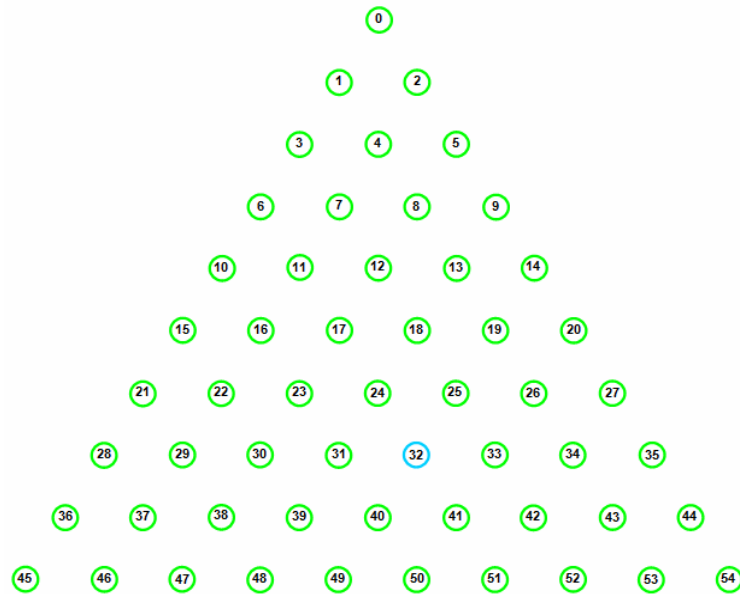


Figura 6.2. Topología de pruebas N° 2.

Ahora, lanzando una simulación sobre el escenario de pruebas N° 1 (figura 6.1) por cada uno de los valores de k , k_1 y k_2 obtenemos una serie de resultados donde se muestra la latencia y el tiempo de estabilización de la red. Ver tabla 6.2.

k	k₁	k₂	Tiempo de Estabilización de la red	Latencia promedio -Tráfico CBR-
102	0,2	0,15	23,1904 seg.	0,0762 seg.
102	0,2	1	23,005 seg.	0,07364 seg.
102	0,2	2	23,005 seg.	0,07364 seg.
102	0,5	0,15	23,005 seg.	0,07364 seg.
102	0,5	1	23,005 seg.	0,07364 seg.
102	0,5	2	23,016 seg.	0,07372 seg.
102	1	0,15	23,005 seg.	0,07364 seg.
102	1	1	23,005 seg.	0,07364 seg.
102	1	2	23,199 seg.	0,07373 seg.
178	0,2	0,15	22,86 seg.	0,076 seg.
178	0,2	1	22,92 seg.	0,07243 seg.
178	0,2	2	23,001 seg.	0,07362 seg.
178	0,5	0,15	22,86 seg.	0,07349 seg.
178	0,5	1	22,99 seg.	0,07356 seg.
178	0,5	2	23,005 seg.	0,07364 seg.
178	1	0,15	23,1904 seg.	0,07373 seg.
178	1	1	23,005 seg.	0,07364 seg.
178	1	2	23,005 seg.	0,07364 seg.
255	0,2	0,15	22,8514 seg.	0,06822 seg.
255	0,2	1	22,93 seg.	0,07112 seg.
255	0,2	2	23,005 seg.	0,07364 seg.
255	0,5	0,15	23,164 seg.	0,07371 seg.
255	0,5	1	23,005 seg.	0,07364 seg.
255	0,5	2	23,005 seg.	0,07364 seg.
255	1	0,15	23,1904 seg.	0,07373 seg.
255	1	1	23,005 seg.	0,07364 seg.
255	1	2	23,005 seg.	0,07364 seg.

Tabla 6.2. Tiempos de estabilización de la red y latencias promedio obtenidas para el escenario N° 1

De los valores obtenidos, podemos notar que para $k=255$, $k_1=0.2$ y $k_2=0.15$ es cuando cada uno de los tiempos se muestra óptimo.

De igual forma, se lanzan nuevamente todas las simulaciones sobre el escenario de pruebas N° 2 para obtener más datos. Ver tabla 6.3.

k	k ₁	k ₂	Tiempo de Estabilización de la red	Latencia promedio -Tráfico CBR-
102	0,2	0,15	20,8785	0,05333
102	0,2	1	20,862	0,05316
102	0,2	2	20,862	0,05316
102	0,5	0,15	20,862	0,05316
102	0,5	1	20,862	0,05316
102	0,5	2	20,875	0,05328
102	1	0,15	20,862	0,05316
102	1	1	20,862	0,05316
102	1	2	20,8785	0,05325
178	0,2	0,15	20,807	0,05315
178	0,2	1	20,82	0,05038
178	0,2	2	20,858	0,05324
178	0,5	0,15	20,807	0,05307
178	0,5	1	20,826	0,05309
178	0,5	2	20,862	0,05316
178	1	0,15	20,8785	0,05325
178	1	1	20,862	0,05316
178	1	2	20,862	0,05316
255	0,2	0,15	20,795	0,05086
255	0,2	1	20,807	0,05267
255	0,2	2	20,862	0,05316
255	0,5	0,15	20,877	0,05323
255	0,5	1	20,862	0,05316
255	0,5	2	20,862	0,05316
255	1	0,15	20,8785	0,05325
255	1	1	20,862	0,05316
255	1	2	20,862	0,05316

Tabla 6.3. Tiempos de estabilización de la red y latencias promedio obtenidas para el escenario N° 2

Como podemos observar los valores de $k=255$, $k_1=0.2$ y $k_2=0.15$ son los que ofrecen mejores prestaciones frente a los demás.

Teniendo los valores indicados para cada índice, éstos pasaran a multiplicar cada factor de la expresión.

En el primer factor tenemos involucrados los saltos al sumidero; a mayor número de saltos menor valor tendrá el término delimitado entre paréntesis. En el denominador de este término se suma 1 para evitar divisiones por cero en el caso de que quien responda sea el propio sumidero. Al índice k se le es asociado un valor mayor que a los otros índices, para poder maximizar el primer término, de manera que no sea despreciable ante los otros factores de la expresión.

En el segundo factor nos encontramos con el LQI, el cual varía en función de la calidad del enlace. Este valor es recibido en la capa MAC de la 802.15.4. y varía entre el rango de 0 a 255, donde 0 es la peor calidad y 255 la mejor. El rango típico de este valor está entre 40 y 95; en nuestro caso definimos un umbral medio para estandarizar todos los valores por encima de 70 a 70. Ahora para minimizar este término y hacerlo del rango del primer factor lo multiplicamos por k_1 .

En el último factor están relacionados el porcentaje de la batería del posible nodo padre sobre la cantidad de nodos hijos que éste tenga; de esta forma a mayor número de hijos menor rendimiento tendrá la batería. A todo ello lo multiplicamos por el índice k_2 para minimizar el término y hacer que este en el rango de los otros factores.

El protocolo Enhanced Quick and Dirty Routing Protocol (EQDRP) surge de la evolución del antiguo protocolo QDRP. En esta nueva versión se realizan algunos cambios para mejorar aspectos tales como: tiempo de estabilización de la red resultado de las rutas de encaminamiento, latencias, uso de mensajes innecesarios y agregar un nuevo mecanismo que busque nuevas rutas.

6.1. Evaluación de la propuesta

Basado en estudios preliminares ^[8] notamos que los parámetros usados por QDRP no son los más certeros para trabajar de forma óptima en el proceso de encaminamiento, es por ello que procedemos a evaluar los parámetros influyentes de QDRP y a mencionar los cambios realizados.

Para realizar un análisis del protocolo a la vez que se mencionan los cambios a realizar sobre éste, trabajaremos sobre el escenario usado en la implementación del protocolo QDRP. Ver figura 6.1.1.

Los parámetros generales establecidos en el archivo de simulación “tcl”, correspondiente al escenario mencionado se describen en la tabla 6.1.1.

A continuación pasaremos a analizar los siguientes aspectos sobre QDRP

- ✓ Inicio de los nodos
- ✓ Tiempo de estabilización de la red
- ✓ El uso de mensajes SHOW_OFF
- ✓ Establecimiento de rutas

Mencionando a su vez el uso del nuevo mensaje SENDME_SHOW_OFF y la búsqueda de nuevas rutas.

Inicio de los nodos

El tiempo que toma un nodo en iniciarse se basa en la expresión “Id.nodo*0.050seg.”, de modo que el arranque de cada uno será de la siguiente forma

- Nodo 0: 0.00 seg.
- Nodo 1: 0.05 seg.
- Nod0 2: 0.10 seg.
- Nodo 3: 0.15 seg.
- Nodo 31: 1.55 seg.
- Nodo 32: 1.60 seg.
- Nodo 100: 5.00 seg.

Como ven, el inicio de los nodos se realiza con un tiempo razonable entre ellos, sin embargo hay un pequeño inconveniente y es que el nodo sumidero (en nuestro caso el nodo 32) se debe iniciar antes que todos los otros de esta forma reducimos el tiempo de conexión de los nodos vecinos hacia él, y por consiguiente el tiempo de estabilización de la red.

Tiempo de estabilización de la red

El tiempo de estabilización es lo que tarda en estar conectado cada uno de los nodos, desde el momento que inicia el primer nodo hasta el último de ellos. El tiempo que emplea el primer nodo en conectar con un padre depende del tiempo en que se inicie el sumidero.

Tal como se sugirió en el apartado anterior, solo cambiando el tiempo de inicio del sumidero se pueden conseguir mejores tiempos de conexión y evitar enviar mensajes innecesarios, ya que si el nodo raíz del árbol no está en línea no se iniciará ninguna conexión en los otros nodos.

Por medio de la figura 6.1.1. vemos el tiempo máximo que tarda QDRP para estabilizar la red.

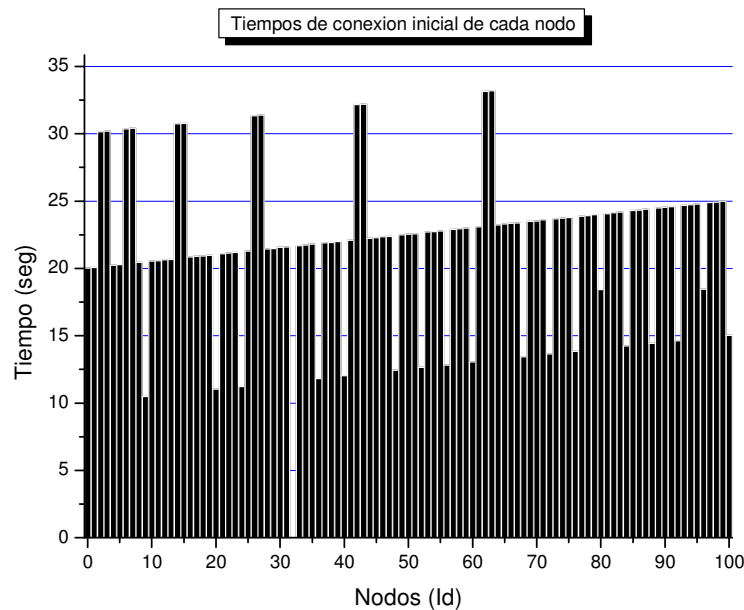


Figura 6.1.1. Tiempos iniciales de conexión en cada nodo – QDRP.

El uso mensajes SHOW_OFF

Cada vez que un nodo se inicia y está buscando una ruta éste espera por un mensaje SHOW_OFF de un nodo el cual ya se encuentre con una conexión establecida o del sumidero directamente. Con lo que se crea un uso erróneo de estos mensajes ya que debe ser el nodo que esta buscando la ruta quien inicie la comunicación y no un nodo con una conexión existente quien la ofrezca.

Siendo esta la forma de trabajo de QDRP nos preguntaríamos, ¿Cuánto tiempo estaría un nodo ONLINE enviando estos mensajes? La respuesta a ello es que mientras un nodo no tenga el máximo número de hijos da por entendido que puede aceptar más conexiones, y mientras pueda hacerlo estará enviando este mensaje.

Pero el planteamiento que se llevo a cabo para establecer este tipo de funcionamiento en la configuración no tiene en cuenta un factor muy importante, y es que si un nodo nunca llega a tener el máximo numero de hijos estará enviado todo el tiempo este mensaje, como es el caso de los nodos que se encuentran en las hojas del árbol.

Basado en este aspecto tan importante, se decide cambiar la forma de realizar las conexiones, teniendo en cuenta que estas deber ser iniciadas por un nodo en estado OFF_LINE hacia uno ONLINE y no en forma inversa.

Establecimiento de rutas

En todos los protocolos de encaminamiento usados para redes inalámbricas y de sensores uno de los factores más importantes es el establecimiento de ruta. Si un protocolo elige buenas rutas de encaminamiento ofrecerá menor latencia entre la fuente y el destino, por consiguiente se realizaran menos transmisiones de mensajes que a su vez se representa en consumo de energía.

QDRP tiene una forma muy particular de elegir las rutas de encaminamiento entre sus posibles candidatos, que se basa en elegir de su lista al nodo que se encuentre mas cerca del sumidero y en el caso que dos o mas nodos tengan el mismo numero de saltos entonces se elige de ellos al que tenga menor numero de hijos y de ser igual este último se elige a cualquiera de ellos.

Este tipo de elección no es la más certera en todos los casos ya que para elegir una ruta aparte del número de saltos al destino y el número de hijos se deben tener en cuenta otros factores como la batería con la que cuenta el posible padre y la calidad del enlace que existe entre ellos. Si bien, pasamos por alto estos factores obtendríamos un encaminamiento que no es el mas óptimo. Aplicando QDRP sobre el escenario de pruebas N° 1 obtendríamos los siguientes resultados

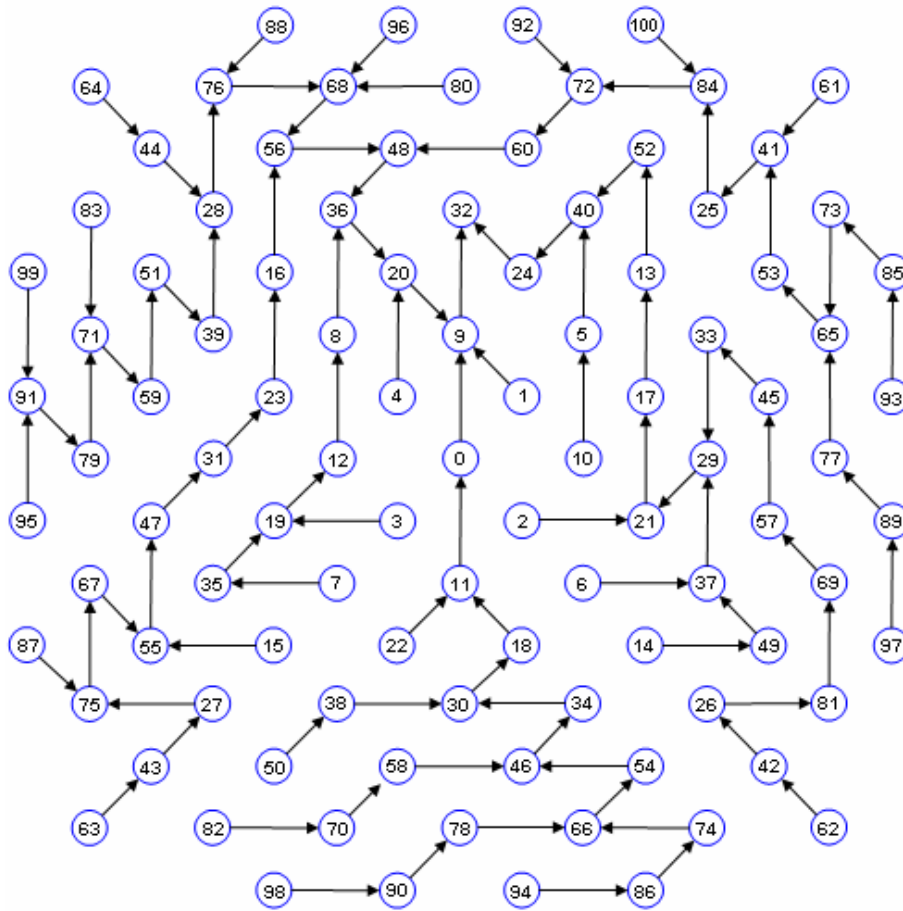


Figura 6.1.2. Rutas obtenidas por QDRP.

Observando la figura anterior podemos notar que las rutas ofrecidas en el escenario no son lo más adecuadas, ya que en un árbol que alcanza una profundidad de 8 niveles encontramos rutas de hasta 15 saltos.

Por lo tanto los valores esperados, en muchos de los casos van a ser mayores al óptimo, como puede ser el caso del nodo 63 en el que tenemos los siguientes datos.

Datos Óptimos		Datos Obtenidos	
Nodo Fuente	63	Nodo Fuente	63
Nodo Destino	32	Nodo Destino	32
Ruta (Saltos)	8	Ruta (Saltos)	15
Latencia	x	Latencia	Aprox. 2x

Tabla 6.1.1. Comparación simplificada de datos óptimos para un nodo frente a datos obtenidos.

Valorando este aspecto tan importante hemos notado que cambiando el proceso que realiza la elección de rutas podemos mejorar la latencia y la cantidad de transmisiones que se genera para enviar la información. A todo ello se le suma la configuración, que, el nodo sumidero al disponer de recursos “ilimitados” tendrá el doble de nodos hijos que un nodo sensor.

Sin embargo para poder establecer una ruta de la mejor forma se ha cambiado el uso de los mensajes SHOW_OFF (ver uso de mensajes SHOW_OFF) y se ha introducido el uso de mensajes SENDME_SHOW_OFF.

Los mensajes SENDME_SHOW_OFF

Como sabemos, los mensajes SHOW_OFF no se usaban de manera correcta y las comunicaciones no se iniciaban por quien buscaba una ruta sino porque quien ya la poseía.

Por ello se ha introducido el uso de mensajes SENDME_SHOW_OFF para ser usados por un nodo que este buscando una ruta. Un nodo en estado OFFLINE emite estos mensajes para buscar un nodo padre; cuando un nodo con una conexión establecida recibe este mensaje responde con un SHOW_OFF y el nodo OFFLINE que lo recibe lo agrega a la lista de posibles candidatos.

Búsqueda de rutas

QDRP en el establecimiento de rutas buscaba entre sus posibles candidatos un padre. Una vez elegido el padre, este seguía siendo su ruta durante todo el tiempo de la simulación a no ser que cayera la conexión con dicho padre.

Aunque la expresión sugerida para obtener mejores rutas nos garantice mejor selección de padres no siempre se consiguen las mejores en el primer intento, debido a que en muchos casos los nodos no se inician en el mismo tiempo o muchos de ellos no escuchan sus peticiones a tiempo y pasan por alto las solicitudes.

Es por ello que se ha implementado un nuevo proceso/mecanismo para buscar mejores rutas cada determinado tiempo, haciendo uso de los mensajes SENDME_SHOW_OFF.

El proceso de búsqueda de nuevas rutas funciona de la misma forma que el establecimiento de rutas, solo que esta vez el envío de mensajes

SENDME_SHOW_OFF lo lleva a cabo un nodo ONLINE y lo repite con menos frecuencia que cuando se desea establecer la ruta.

Ejecutando este proceso podemos mejorar las rutas de encaminamiento en caso de que existe un mejor camino.

6.2. Simulación

Aplicados los cambios mencionados al código fuente del protocolo procedemos a realizar la simulación con los nuevos parámetros establecidos.

Para los escenarios de simulación de este análisis se proponen las siguientes topologías:

- ✓ **Topología 1:** Escenario usado en el desarrollo de QDRP (ver figura 6.1.)
- ✓ **Topología 2:** Piramidal (ver figura 6.2.)
- ✓ **Topología 3:** Senoidal (ver siguiente 6.2.1.)

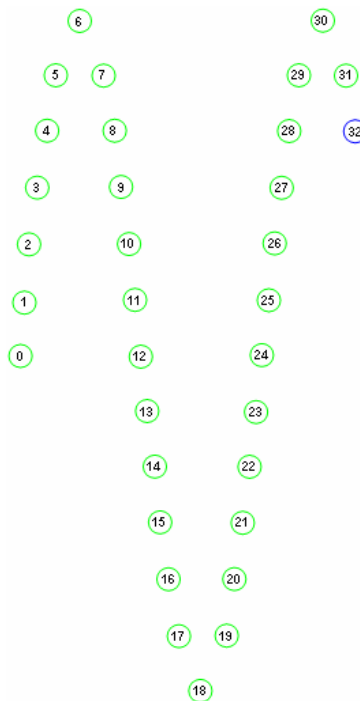


Figura 6.2.1. Topología de pruebas N° 3.

Todos los escenarios están configurados según los parámetros generales de simulación especificados en la tabla 6.1.

Bajo estos escenarios evaluaremos los parámetros abajo indicados, sobre QDRP y EQDRP. Ellos son:

✓ **Tiempo de estabilización de la red**

Tiempo que tardan en conectar todos los nodos de la red.

✓ **Establecimiento de ruta**

Ver las rutas obtenidas por ambos protocolos.

✓ **Búsqueda de rutas**

Observar si hay cambios de rutas sobre las ya establecidas.

✓ **Latencia entre extremos**

Tiempo transcurrido desde el instante que se envía un paquete hasta que es recibido por el sumidero.

- Media: valor promedio de todos los paquetes recibidos por el sumidero.
- Mediana: valor central de la medidas de latencia
- Desviación estándar: dispersión de las medidas con respecto a la media.

✓ **Cantidad de paquetes enviados**

Numero de paquetes CBR que se transmiten.

✓ **Cantidad de paquetes recibidos**

Numero de paquetes CBR que han sido recibidos por el sumidero

6.3. Resultados

✓ Escenario 1

MAC 802.15.4. Modo sin balizas

Alcance de los transceptores de radio, 10 mts

Topología: Grid uniforme con 101 nodos.

Estabilización de la red

En la siguiente figura podemos notar como disminuye el tiempo máximo que toma la red en estabilizarse en EQDRP. A su vez apreciamos que los tiempos de conexión inicial para los nodos más cercanos son más bajos en comparación con los tiempos que tomaba QDRP. Ver figura 6.3.1.

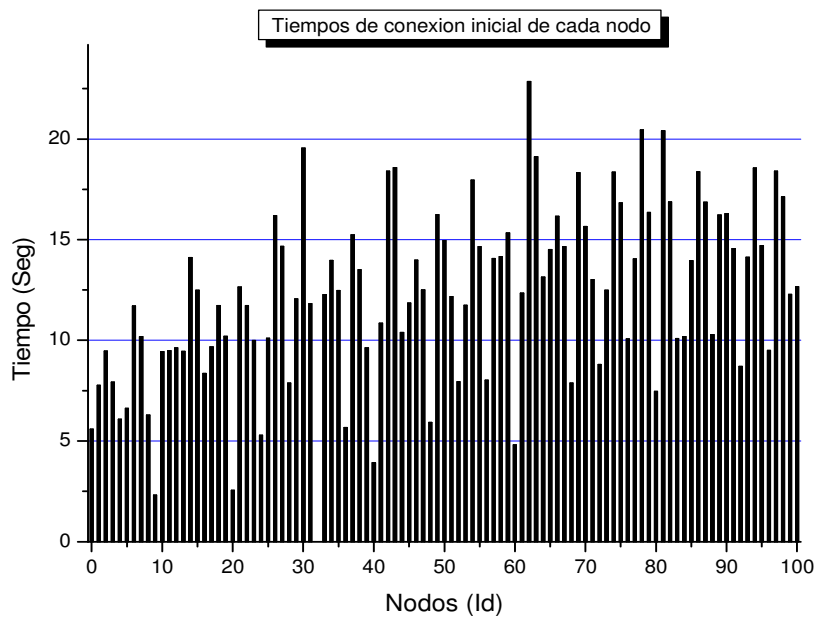


Figura 6.3.1. Tiempos iniciales de conexión – Escenario 1, EQDRP.

Establecimiento de rutas

Una vez que se estabiliza la red podemos ver que las rutas obtenidas en el primer intento de búsqueda son mas directas que las ofrecidas por QDRP aunque no todos los nodos obtengan la mejor ruta. Ver la siguiente figura.

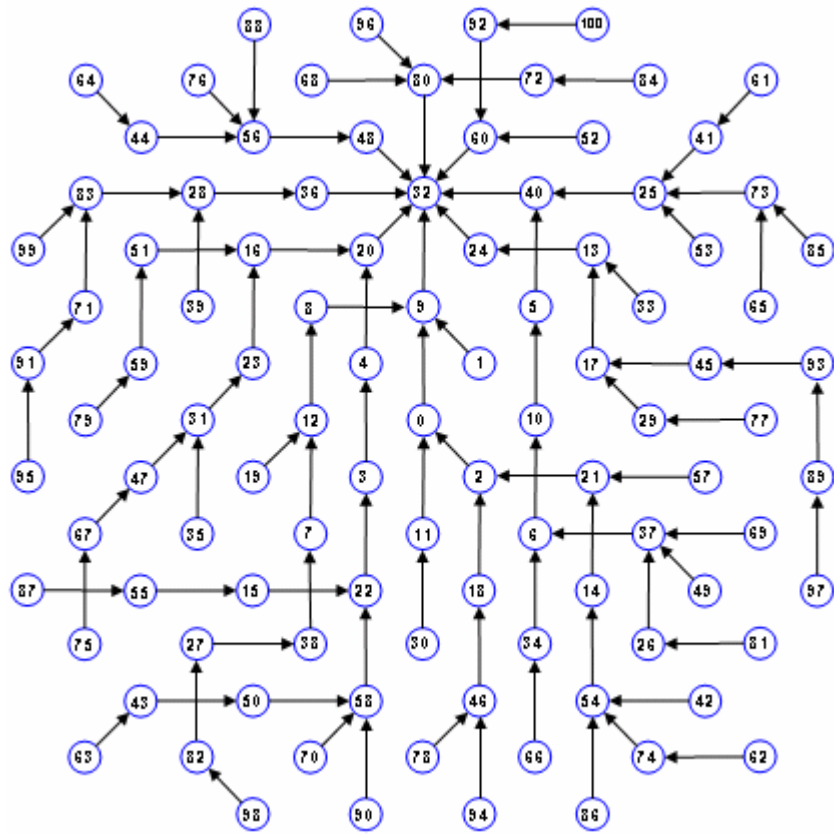


Figura 6.3.2. Rutas obtenidas por EQDRP.

Búsqueda de rutas

Finalizado el proceso de búsqueda de rutas podemos ver que no todos los nodos obtienen el mejor camino como es el caso del nodo 78. Pero transcurrido el tiempo establecido en los temporizadores para buscar una nueva ruta este establece como nuevo padre al nodo 30, el cual se encuentra a un salto menos que el nodo actual. Ver la siguiente figura.

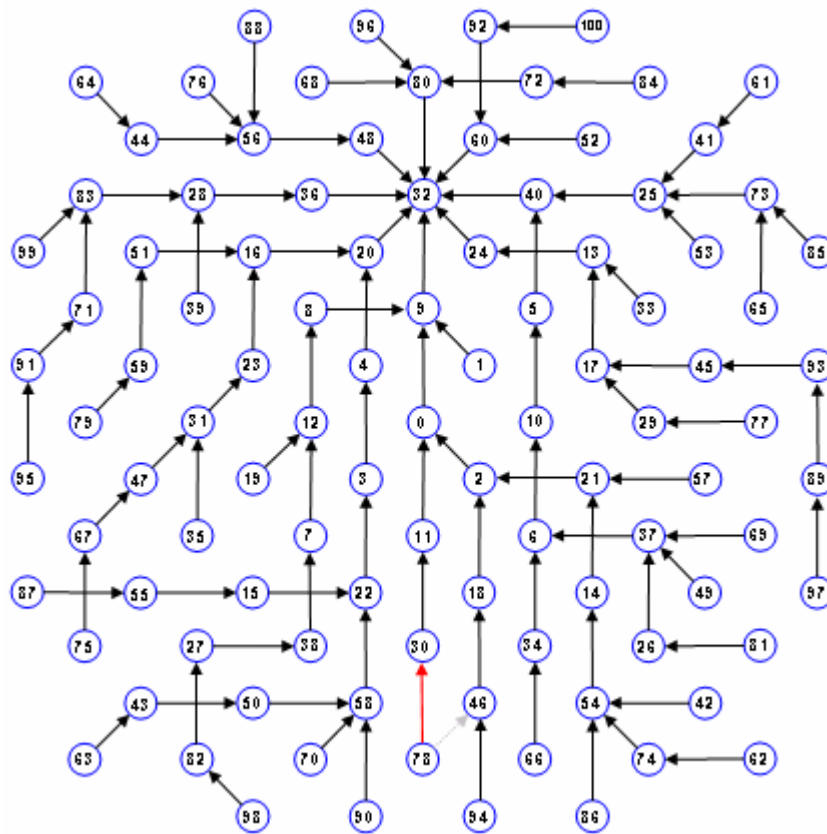


Figura 6.3.3. Cambio de ruta en los nodos, efectuado por EQDRP.

Como resultado a los análisis de las trazas podemos notar que no solo las rutas ni el tiempo de estabilización de la red son las únicas mejoras apreciadas.

En la tabla 6.3.1. podemos observar otros valores que también son representativos en un encaminamiento..

Protocolo	QDRP	EQDRP
Tiempo de estabilización de la red	33,184 seg.	22,8514 seg.
Latencia (Media)	0,13139 seg.	0,06822 seg.
Latencia (Desviación Estándar)	0.06377 seg.	0,13338 seg.
Latencia (Mediana)	0,13031 seg.	0,06253 seg.
Paquetes de datos enviados	5975	12062
paquetes de datos recibidos	4939	10802
Porcentaje de paquetes recibidos	82,66%	89,55%

Tabla 6.3.1. Comparación de datos entre QDRP y EQDRP, en el escenario N° 1.

Según los datos, vemos que el protocolo EQDRP es más fiable que su antecesor. La latencia media de los paquetes se ha reducido a la mitad del valor gracias a la obtención de mejores rutas de encaminamiento; a ello también se suma el incremento del porcentaje de paquetes recibidos en el sumidero.

✓ Escenario 2

MAC 802.15.4. Modo sin balizas

Alcance de los transceptores de radio, 10 mts

Topología: Distribución piramidal con 55 nodos.

Este escenario cuenta con 55 nodos distribuidos en forma piramidal, ver figura 6.2, el tráfico CBR usado es igual para todos los escenarios.

Estabilización de la red

El tiempo de estabilización de la red tarda cada vez mas en QDRP cuando los nodos no se encuentran agrupados, en este escenario los nodos más lejanos al sumidero tardan mucho más tiempo que en EQDRP. Ver figura 6.3.4.

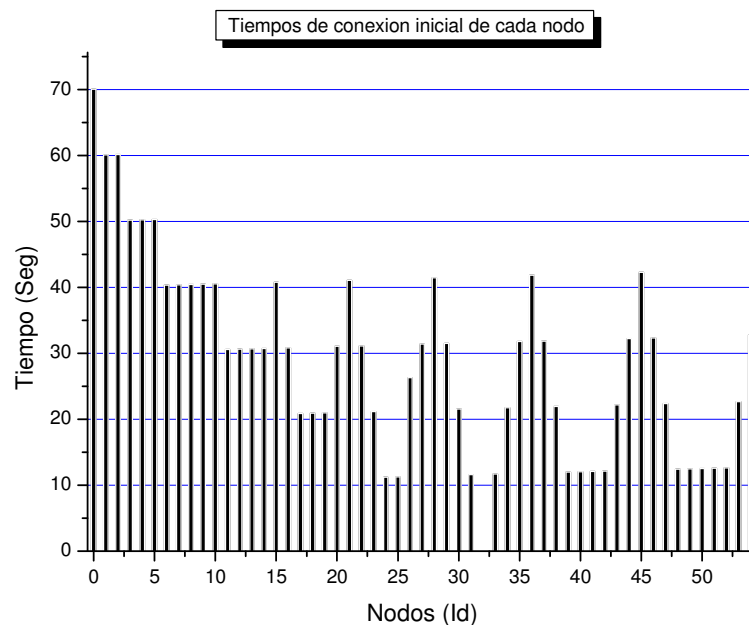


Figura 6.3.4. Tiempos iniciales de conexión – Escenario 2, QDRP.

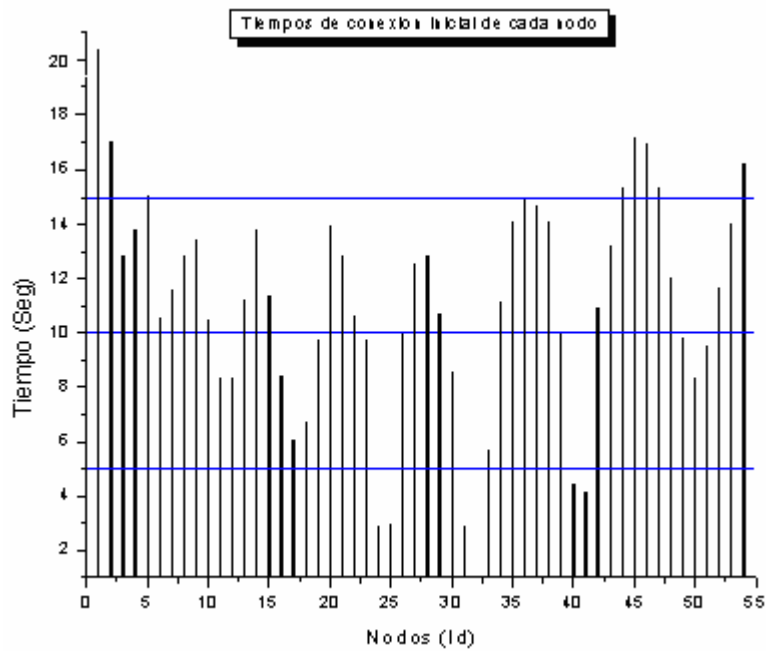


Figura 6.3.5. Tiempos iniciales de conexión – Escenario 2, EQDRP.

Establecimiento de rutas

El establecimiento de ruta se ve afectado drásticamente cuando se cambian los escenarios, y como consecuencia, al no tener un mecanismo para mejorar las rutas éstas seguirán activas por siempre.

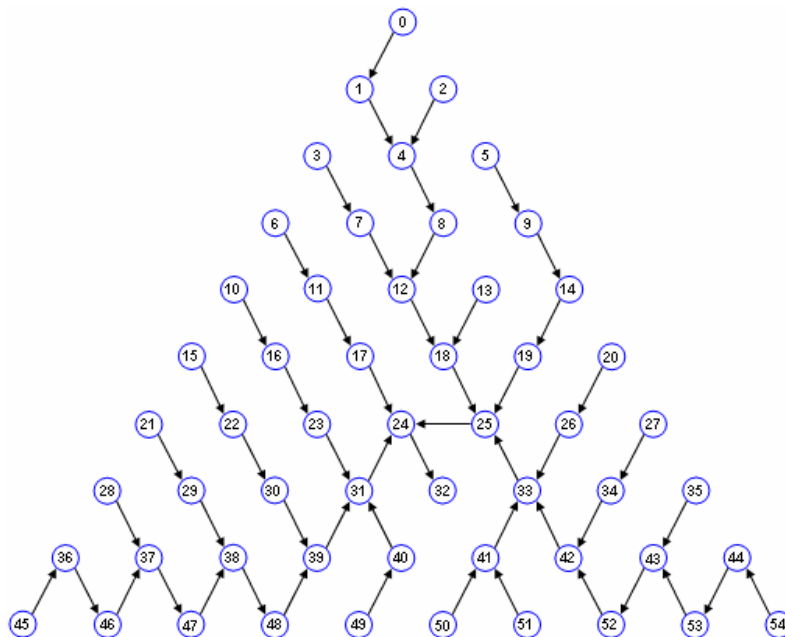


Figura 6.3.6. Rutas obtenidas por QDRP en el escenario N° 2.

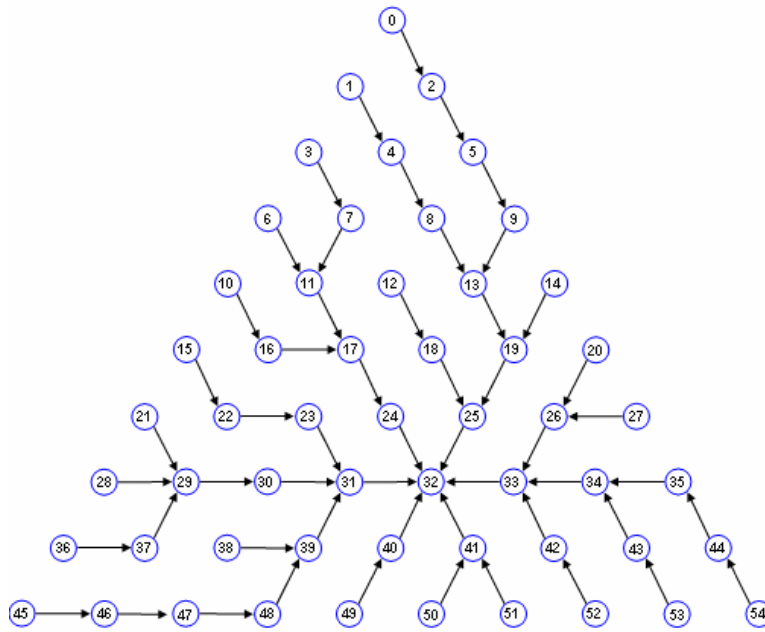


Figura 6.3.7. Rutas obtenidas por EQDRP en el escenario N° 2.

En QDRP conforme la distribución de los nodos se va extendiendo en forma no grupal, se van obteniendo mejores rutas en vista de que no existen más candidatos posibles, sin embargo cuando hay una pequeña agrupación de éstos se tiende a elegir rutas no óptimas, tal como se aprecia en la figura 6.3.7.

Búsqueda de rutas

Debido a que EQDRP ha encontrado en su fase inicial las mejores rutas, estas no cambiarán hasta que uno de los nodos baje sus prestaciones.

Otros datos a tener en cuenta son los siguientes:

Protocolo	QDRP	EQDRP
Tiempo de estabilización de la red	70,035 seg.	20,795 seg.
Latencia (Media)	0.07844 seg.	0,05086 seg.
Latencia (Desviación Estándar)	0.03266 seg.	0,10175 seg.
Latencia (Mediana)	0,07582 seg.	0,04656 seg.
Paquetes de datos enviados	3231	3202
paquetes de datos recibidos	3062	3155
Porcentaje de paquetes recibidos	94,76%	98.53%

Tabla 6.3.2. Comparación de datos entre QDRP y EQDRP, en el escenario N° 2.

✓ Escenario 3

MAC 802.15.4. Modo sin balizas

Alcance de los transceptores de radio, 10 mts

Topología: Distribución senoidal con 32 nodos

Este escenario cuenta con 33 nodos distribuidos en forma senoidal, ver figura 6.2.1. el trafico CBR usado es igual para todos los escenarios.

Estabilización de la red

Para esta simulación se ha planteado el escenario mostrado en la figura anterior, para ver como afecta la forma en que se distribuyen los nodos al tiempo de estabilización y las latencias de los paquetes; ya que las conexiones son mas directas pero secuenciales de modo que un nodo tendrá pocas opciones de encontrar varios candidatos a padre para establecer su conexiones.

Aunque las figura 6.3.8. y 6.3.9. sean muy similares, tenemos una gran diferencia de tiempos en la conexiones, ya que QDRP sobrepasa los 230 seg. mientras que EDQRP no llega a los 60 seg.

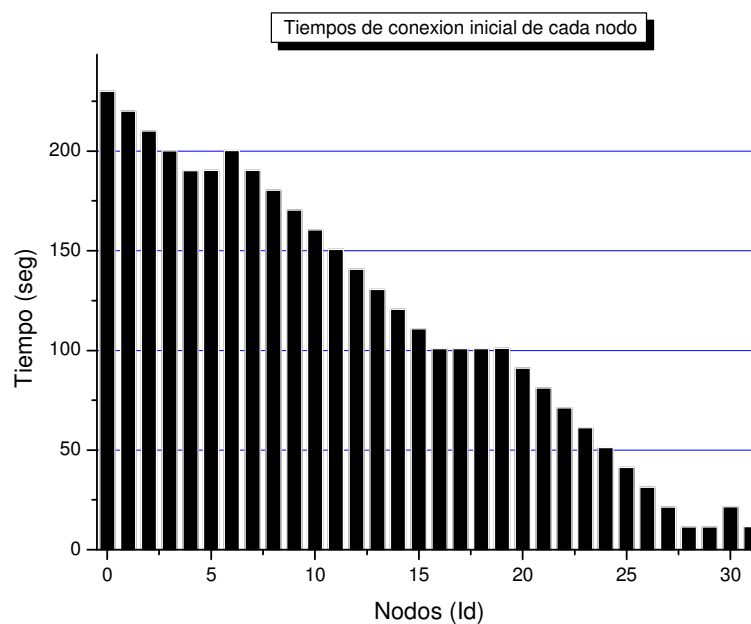


Figura 6.3.8. Tiempos iniciales de conexión – Escenario 3, QDRP.

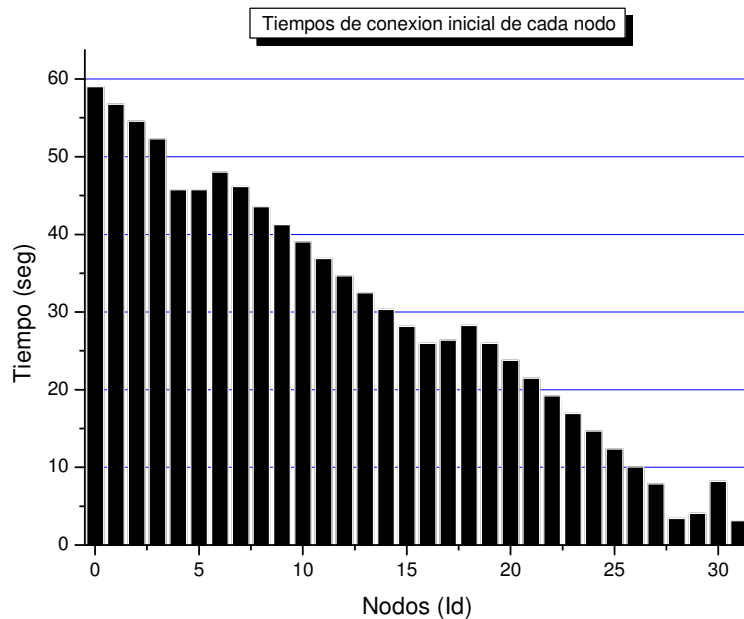


Figura 6.3.9. Tiempos iniciales de conexión – Escenario 3, EQDRP.

Establecimiento de rutas

El establecimiento de rutas se realiza de la misma manera en ambos casos, dado que en este escenario son muy pocas las opciones que se presentan a la hora de elegir a un nodo candidato a padre, y con la diferencia del tiempo de respuesta para establecer sus conexiones. Ver figura 6.3.10.

Búsqueda de rutas

Debido a la forma en que están dispuestos los nodos, el escenario no se presta para realizar cambios de ruta.

Como resultado de las simulaciones obtenemos los datos mostrados en la tabla 6.3.3.

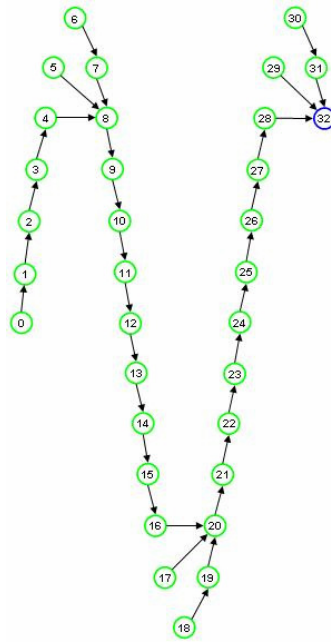


Figura 6.3.10. Rutas obtenidas por QDRP y EQDRP en el escenario N° 3.

Protocolo	QDRP	EQDRP
Tiempo de estabilización de la red	230,039 seg.	59,03 seg.
Latencia (Media)	0,18154 seg.	0,16011 seg.
Latencia (Desviación Estándar)	0,18999 seg.	0,10339 seg.
Latencia (Mediana)	0,17395 seg.	0,15392 seg.
Paquetes de datos enviados	3841	3834
paquetes de datos recibidos	3321	3806
Porcentaje de paquetes recibidos	86,46%	99,26%

Tabla 6.3.3. Comparación de datos entre QDRP y EQDRP, en el escenario N° 3.

7. CONCLUSIONES Y TRABAJO FUTURO

Analizando los resultados que se han presentado, podemos apreciar que los cambios aplicados sobre el protocolo muestran buenos resultados, mejorando de forma significativa así los aspectos planteados al inicio de este documento.

Los aportes realizados no solo pueden ser aplicados al protocolo QDRP, sino también a cualquier otro protocolo basado en encaminamiento en “árbol”, como puede ser el caso del protocolo QDRP-SC.

Como trabajo futuro se plantea la implementación de estos algoritmos en entornos reales y la validación experimental de los resultados obtenidos mediante simulación.

Por otro lado, sería interesante repetir la calibración de parámetros para diversos protocolos orientados a árbol, ajustando el valor de los mismos y comparando numéricamente el aumento de prestaciones obtenidas.

Asimismo, el impacto de este método de selección de rutas sobre las prestaciones se multiplicaría de forma muy importante en caso de contar con protocolos donde se contemplen periodos de inactividad por parte de los nodos, entrando durante los mismos en modo bajo consumo.

Finalmente, se propone evaluar la inclusión de nuevos factores especialmente orientados a redes con nodos móviles, tales como la dirección del movimiento, para la selección de padre, modificando la expresión obtenida con nuevos parámetros a ajustar.

8. BIBLIOGRAFIA

- [1] AODV Routing Protocol Implementation Design
URL: http://moment.cs.ucsb.edu/pub/wwan_chakeres_i.pdf
Fecha ultimo acceso: 29/10/2008

- [2] Dynamic Source Routing Protocol
URL: <http://www.cs.cmu.edu/~dmaltz/dsr.html>
Fecha ultimo acceso: 01/11/2008

- [3] DSR: The Dynamic Source Routing Protocol For Multi-Hop Wireless Ad Hoc Networks
URL: <http://www.monarch.cs.rice.edu/monarch-papers/dsr-chapter00.pdf>
Fecha ultimo acceso: 01/11/2008

- [4] Vint Project
URL: <http://www.isi.edu/vint/>
Fecha ultimo acceso: 06/10/2008

- [5] The Network Simulator NS-2
URL: <http://www.isi.edu/nsnam/ns/>
Fecha ultimo acceso: 07/10/2008

- [6] IEEE 802.15.4 MAC implementation in NS-2
URL: http://www.ee.washington.edu/research/funlab/802_15_4/
Fecha ultimo acceso: 06/11/2008

- [7] Changes made to the IEEE 802.15.4 NS-2 Implementation
URL: http://www.ee.washington.edu/research/funlab/802_15_4/ns2_changes.pdf
Fecha ultimo acceso: 06/11/2008

- [8] Sáez, Abel
Quick and Dirty Routing Protocol - Diseño en implementación de un algoritmo de enrutamiento para redes de sensores inalámbricas.
Tesina de master, diciembre de 2007

- [9] The Zone Routing Protocol (ZRP) for Ad Hoc Networks
URL: <http://tools.ietf.org/html/draft-ietf-manet-zone-zrp-04>
Fecha ultimo acceso: 27/11/2008
- [10] Intermediate System to Intermediate System Protocol (IS-IS)
URL: http://www.cisco.com/warp/public/cc/pd/iosw/prodlit/insys_wp.pdf
Fecha ultimo acceso: 27/11/2008
- [11] Enhanced Interior Gateway Routing Protocol (EIGRP)
URL: http://www.cisco.com/en/US/docs/internetworking/technology/handbook/Enhanced_IGRP.pdf
Fecha ultimo acceso: 28/11/2008