UNIVERSITAT POLITÈCNICA DE VALÈNCIA

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

TESIS DOCTORAL

Autoconfiguración de redes ad hoc basadas en confianza

Autor: José Vicente Sorribes Díaz

Directores: Dra. Lourdes Peñalver Herrero

Dr. Jaime Lloret Mauri

Valencia

Noviembre 2021

Resumen

En la presente tesis se aborda el problema del descubrimiento de vecinos en redes estáticas inalámbricas ad hoc, redes sin infraestructura. En estas redes los nodos deben descubrir los vecinos como un primer paso tras el despliegue. Además se aborda el problema de creación de redes espontáneas basadas en la confianza, y el de selección de vecinos. Se han presentado distintos algoritmos que solucionan el problema siguiendo unas premisas que se irán relajando a lo largo de la tesis. Se han implementado esos algoritmos en Castalia 3.2 para su validación y comparación con otros protocolos de referencia. Los protocolos determinísticos presentan mejores resultados de simulación, aunque requieren que sigan una planificación en la transmisión. En cuanto al protocolo aleatorio CDPRR (Collision Detection Probabilistic Round Robin) no sigue una planificación pero requiere el conocimiento del número de nodos de la red. El protocolo CDH (Collision Detection Hello) permite el desconocimiento del número de nodos. Ambos logran el descubrimiento de los vecinos con probabilidad 1 mediante la detección de colisiones. Ambas propuestas obtienen mejores prestaciones que los protocolos usados como referencia. Tanto para CDH como para CDPRR se ha realizado un modelo analítico de determinadas métricas. Se ha propuesto un protocolo aleatorio basado en líder que permite obtener buenas prestaciones y se puede usar de forma asíncrona aunque solo permite su uso en entornos *one-hop*. Finalmente, se presenta un protocolo aleatorio consciente de la energía que permite buenos resultados para altos *duty cycles* y redes compuestas de pocos nodos. En cuanto a la creación de redes espontáneas basadas en la confianza, se ha propuesto un modelo que usa el protocolo CDPRR como base. Este protocolo de creación de redes espontáneas basadas en la confianza permite premisas más realistas y mejora un protocolo deter-

minístico de referencia. Finalmente, se propone un protocolo que combina el descubrimiento y la selección de vecinos con el objetivo de proporcionar nodos favoritos. Estos nodos permiten el envío de información al exterior de la red o en futuras operaciones como el encaminamiento.

Resum

En la present tesi s'aborda el problema del descobriment de veïns en xarxes estàtiques sense fil ad hoc, xarxes sense infraestructura. En estes xarxes els nodes han de descobrir els veïns com un primer pas després del desplegament. A més s'aborda el problema de creació de xarxes espontànies basades en la confiança, i el de selecció de veïns. S'han presentat distints algoritmes que solucionen el problema seguint unes premisses que se n'aniran relaxant al llarg de la tesi. S'han implementat eixos algoritmes en Castalia 3.2 per a la seua validació i comparació amb altres protocols de referència. Els protocols determinístics presenten millors resultats de simulació, encara que requerixen que seguisquen una planificació en la transmissió. Quant al protocol aleatori CDPRR (Collision Detection Probabilistic Round Robin) no seguix una planificació però requerix el coneixement del nombre de nodes de la xarxa. El protocol CDH (Collision Detection Hello) permet el desconeixement del nombre de nodes. Ambdós aconseguixen el descobriment dels veïns amb probabilitat 1 per mitjà de la detecció de col·lisions. Ambdós propostes obtenen millors prestacions que els protocols usats com a referència. Tant per a CDH com per a CDPRR s'ha realitzat un model analític de determinades mètriques. S'ha proposat un protocol aleatori basat en líder que permet obtindre bones prestacions i es pot usar de forma asíncrona encara que només permet el seu ús en entorns *one-hop*. Finalment, es presenta un protocol aleatori conscient de l'energia que permet bons resultats per a alts *duty cycles* i xarxes compostes de pocs nodes. Quant a la creació de xarxes espontànies basades en la confiança, s'ha proposat un model que usa el protocol CDPRR com a base. Este protocol de creació de xarxes espontànies basades en la confiança permet premisses més realistes i millora un protocol determinístic de referència. Finalment, es proposa un

protocol que combina el descobriment i la selecció de veïns amb l'objectiu de proporcionar nodes favorits. Estos nodes permeten l'enviament d'informació a l'exterior de la xarxa o en futures operacions com l'encaminament.

Abstract

This thesis addresses the neighbor discovery problem in static wireless ad hoc networks, infrastructure-less networks. In these networks the nodes must discover the neighbors as a first step after the deployment. Furthermore, the thesis addresses the problem of creation of spontaneous networks based on trust, and the neighbor selection. Several algorithms have been presented that solve the problem following some assumptions that will be relaxed throughout the thesis. Those algorithms have been implemented in Castalia 3.2 for validation and comparison with other reference protocolos. The deterministic protocols provide better simulation results, although they require a transmission schedule. As for the randomized protocol CDPRR (Collision Detection Probabilistic Round Robin), it does not follow a schedule but it requires the knowledge of the number of nodes in the network. The CDH (Collision Detection Hello) protocol allows the ignorance of the number of nodes. They both achieve the discovery of the neighbors with probability 1 by detecting collisions. Both proposals achieve better performance than the protocols used as reference. For CDH and CDPRR an analytical model has been carried out regarding several metrics. A randomized protocol based on leader has been proposed that achieves a good performance and it can be used in an asynchronous way although it can only be used in one-hop environments. Finally, an energy-aware randomized protocol is proposed, which achieves good results for high *duty cycles* and networks composed of a small number of nodes. As for the creation of spontaneous networks based on trust, a model has been proposed which is based on the CDPRR protocol. This protocol for the creation of spontaneous networks based on trust allows more realistic assumptions and outperforms a determinsitic protocol used as reference. Finally, a protocol

is proposed which combines the discovery and selection of neighbors aiming at providing favourite nodes. These nodes allow sending information towards outside the network or in future operations such as routing.

Índice general

Índice de figuras

Índice de tablas

Índice de Algoritmos

Abreviaturas y acrónimos

ND	Neighbor Discovery
WSN	Wireless Sensor Network
MANET	Mobile Ad hoc Network
VANET	Vehicular Ad hoc Network
V2V	Vehicle to Vehicle
V2I	Vehicle to Infrastructure
FANET	Flying Ad hoc Network
UAV	Unmanned Aerial Vehicles
DTN	Delay Tolerant Networks
AC	Autoridad Certificadora
CDF	Cumulative Distribution Function
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
NACK	Negative Acknowledgement
IoT	Internet of Things
CDPRR	Collision Detection Probabilistic Round Robin
CDH	Collision Detection Hello
PRR	Probabilistic Round Robin
PSBA	Prime-set-based neighbor discovery algorithm
Panda	Power Aware Neighbor Discovery Asynchronously protocol
EH	Energy Harvesting
Hedis	Heterogeneous Discovery Quorum-based protocol
Todis	Triple-Odd based discovery co-primality based protocol
BLE	Bluetooth Low Energy
AI	Advertisement Interval

TDMA	Time Division Multiple Access
GPS	Global Positioning System
MAC	Medium Access Control
MSN	Mobile Sensor Network
KPND	Kalman Prediction-based Neighbor Discovery
AODV	Ad hoc On-Demand Distance Vector
CSMA	Carrier Sense Multiple Access
MTC	Machine-type communication
CRA	Completely Random Algorithm
SBA	Scan Based Algorithm
P2P	Peer-to-peer
AES	Advanced Encryption Standard
ECC	Error-Correcting Code
WWW	World Wide Web
RSA	Rivest Shamir Adleman
SPSNC	Secure Protocol for Spontaneous Network Creation
FMA	Friend Management Algorithm
CH	Cluster Head
MADM	Fuzzy multiple attribute decision-making
MODM	Pareto optimal technique
LEACH	Low-energy adaptive clustering hierarchy
TDM	Time Division Multiplexing
BAN	Body Area Network
LECDH	Low Energy Collision Detection Hello
EAH	Energy Aware Hello
DC	Duty Cycle
NDSP	Neighbor Discovery and Selection Protocol
NS-PRR	Neighbor Selection-PRR
NS-Hello	Neighbor Selection-Hello

Capítulo 1

Introducción

1.1 Redes inalámbricas ad hoc

Las redes inalámbricas ad hoc no tienen ningún tipo de infraestructura de comunicaciones tras su despliegue. Además, los dispositivos (nodos) que las forman están equipados con transceptores de radio de alcance limitado (típicamente por debajo de 500 metros) para llevar a cabo comunicaciones [1, 2]. Por lo tanto, los nodos sólo pueden enviar mensajes directamente hacia el resto de nodos en su rango de transmisión (conocidos también como vecinos *one-hop*). Otros nodos necesitan múltiples nodos intermedios que pasan los mensajes que no están destinados a su propio uso (de una manera *multi-hop*). Así, cada nodo también debe ser capaz de actuar como *router* [3, 4].

Los nodos no conocen qué otros nodos están en su rango de transmisión (vecinos). Así, cada nodo debe descubrir sus vecinos como un primer paso.

Por tanto, justo tras el despliegue, todos los nodos deben autoconfigurarse de forma autónoma para establecer y proporcionar una infraestructura de comunicaciones. Esta será útil para posteriores fases como el encaminamiento. Por lo tanto, cada nodo debe llegar a conocer cuántos y cuáles son los nodos en su rango de transmisión (vecinos). Para solucionar ese problema cada nodo debe descubrir los nodos en su rango de transmisión, esto es, se deben desa-

rrollar técnicas de descubrimiento de vecinos ND (*Neighbor Discovery*). Este descubrimiento representa un primer paso tras el despliegue [5, 6].

En cuanto a las ventajas de las redes inalámbricas ad hoc, se encuentran el rápido y fácil despliegue, su poca dependencia con la infraestructura y su bajo coste económico.

El medio de transmisión será compartido, por lo que pueden producirse colisiones cuando varios nodos intentan transmitir al mismo tiempo. Además, se puede producir pérdida de paquetes debido a errores en la transmisión. Otro punto importante es que hay restricciones energéticas dado que los dispositivos se alimentan generalmente de baterías que se descargarán en un tiempo delimitado. Finalmente, este tipo de redes presenta problemas de seguridad.

En entornos estáticos, los nodos no pueden moverse en el área de despliegue. Un ejemplo es una WSN (*Wireless Sensor Network*), cuyos nodos son desplegados en un campo para averiguar la cantidad necesaria de agua [7]. Por otro lado, en redes móviles MANETs (*Mobile Ad hoc Networks*) los nodos pueden entrar y salir de la red o entrar y salir del rango de transmisión de otros nodos. Un posible ejemplo podría ser una red vehicular ad hoc usada para monitorizar condiciones meteorológicas [8].

Dado que los dispositivos usan baterías, los algoritmos de descubrimiento de vecinos desarrollados deben considerar el consumo energético como métrica para proporcionar una alta eficiencia energética. En los siguientes Capítulos se tendrá en cuenta el consumo energético para desarrollar los protocolos. Más en concreto, en el Capítulo 6 se presenta un protocolo de descubrimiento consciente de la energía que trata de minimizar el consumo energético.

Hay muchas áreas de aplicación [9] para las redes inalámbricas ad hoc. Entre ellas, la industrial (e.g., comunicación entre sensores, robots, y redes digitales), la médica (e.g. monitorización de pacientes), y los negocios (e.g., encuentros, control de stocks). Además hay aplicaciones militares (e.g. entornos duros y hostiles), en agricultura, y en la enseñanza.

Como posibles ejemplos de aplicación, tenemos una red de sensores inalámbricos en un bosque para detectar fuego por un periodo de tiempo determinado. Otra posible aplicación son sensores en un puente con el objetivo de contar el número de vehículos y su velocidad. Finalmente, se podría tener sensores desplegados en un lago para estudiar la calidad del agua durante un periodo determinado de tiempo.

En la Figura 1.1, se muestra un ejemplo de red inalámbrica ad hoc.

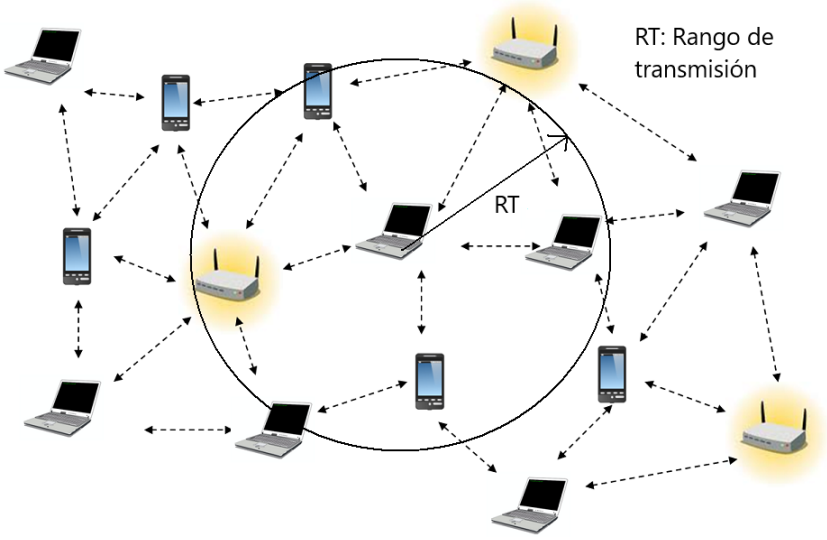

Figura 1.1: Red inalámbrica ad hoc

1.1.1 Tipos de redes inalámbricas ad hoc

En primer lugar, las redes inalámbricas ad hoc pueden ser estáticas. En ellas, los nodos no se pueden mover en el área de despliegue. Como ejemplo, tenemos una red de sensores en la cuál los nodos se lanzan de un avión a un bosque [7]. Los nodos se pueden situar en el campo para determinar varios parámetros como el fuego o la humedad. En segundo lugar, tendremos redes móviles. En ellas, los nodos se pueden mover en un área determinada. Como ejemplo podemos tener robots móviles o vehículos equipados con transceptores de radio [8] con el objetivo de intercambiar información. El último tipo también recibe el nombre de MANET. En ellas, los nodos pueden entrar y salir de la red incluso entrar y salir en el rango de transmisión unos nodos de otros.

Recientemente, las redes inalámbricas ad hoc se han visto utilizadas y mejoradas para distintos escenarios, como:

- VANET (*Vehicular Ad hoc Network*): con el objetivo de comunicación entre vehículos V2V (*Vehicle to Vehicle*) y entre vehículos y dispositivos fijos de tráfico V2I (*Vehicle to Infrastructure*). Generalmente, el objetivo es ofrecer seguridad y confort a los conductores y pasajeros. Otro objetivo es alertar de colisión con otros vehículos, información sobre el

estado de carreteras, meteorológica, acceso a Internet, información local, o mantenimiento del coche.

- FANET (*Flying Ad hoc Network*): compuesta por UAV (*Unmanned Aerial Vehicles*). Los UAV ambién son conocidos como drones, dispositivos voladores no tripulados para vuelos programados o controlados de forma remota. Entre posibles ejemplos se encuentran, la distribución de paquetes a domicilio, o la contribución a crear DTN (*Delay Tolerant Networks*) para ciudades inteligentes.

- WSN móvil no estacionaria: consta de un conjunto sensores autónomos distribuidos espacialmente, conectados vía una infraestructura de comunicaciones. Entre sus aplicaciones están el monitorizar, procesar y almacenar condiciones del entorno como temperatura, sonido, presión, humedad, etc. Hay 3 tipos principales: (i) sensores urbanos, personales, móviles, (ii) sensores autónomos remotos, (iii) sensores embebidos en infraestructuras y edificios/puentes.

- MANET Geosociales: supone una siguiente innovación en Redes Sociales al anticiparse a trabajar sobre dispositivos móviles utilizando información de localización.

- MANET para entornos de misión crítica, militares y para cuerpos de seguridad (policía, bomberos, etc.): Son redes compuestas por dispositivos que no necesitan infraestructura, utilizados en entornos militares. Ejemplos posible son dispositivos portátiles o implantados como prótesis, también en dispositivos de comunicación en vehículos y aeronaves.

1.2 Redes espontáneas basadas en la confianza

En las redes espontáneas ad hoc, un concepto introducido por primera vez en [10], los dispositivos (nodos) que la forman son autónomos y están equipados con transceptores de radio de alcance limitado. Por tanto, algunos nodos pueden comunicar directamente con los nodos en su rango de transmisión (vecinos). Sin embargo, otros nodos necesitan de múltiples nodos intermedios que pasan la información que no está destinada a su propio uso de una manera *multi-hop*. Con este objetivo, cada nodo debe actuar como un *router* [11, 12].

Este tipo de redes no presentan una infraestructura de comunicaciones a-priori y los vecinos son desconocidos, justo tras el despliegue, y son usadas durante un periodo de tiempo en una localización determinada. Por tanto, en la creación

de las redes espontáneas ad hoc, el descubrimiento de los vecinos [5, 6] se hace necesario para encontrar qué nodos están en rango de transmisión. Además, hay ausencia de un servidor central, por tanto no hay una AC (Autoridad Certificadora) centralizada disponible y cada nodo debe actuar como una AC.

Las redes espontáneas son un tipo especial de redes ad hoc que presenta las siguientes características [13]:

- Los nuevos servicios estarán disponibles sin intervención del usuario.
- Los nodos pueden coincidir en una localización física durante un tiempo determinado. Esos nodos colaboran en todo momento para proporcionar servicios como comunicación de grupo, seguridad, etc.
- Los nodos pueden entrar o salir de la red cuando quieran en cualquier instante de tiempo. Los dispositivos pueden venir de cualquiere sitio.
- Estas redes están compuestas de nodos móviles, por tanto no existe topología fija.
- Estas redes emulan las relaciones humanas para lograr la creación y su funcionamiento.
- Estas redes están compuestas por un conjunto de nodos que a veces no se conocen el uno del otro.
- Este tipo de redes deben tener un nivel de seguridad similar a las redes cableadas tradicionales.
- Cada nodo actúa como un *router*.
- Los nodos tienen un rango de transmisión limitado hacia otros nodos.
- Los nodos tienen recursos limitados, como CPU, memoria y energía (baterías).
- Los nodos móviles pueden moverse libremente en un área dada incluso dentro y fuera del rango de transmisión de otros nodos.
- El medio físico de transmisión es compartido.
- Las diferentes identidades son dadas por direcciones IP obtenidas dinámicamente.
- No hay administración central.

En cuanto a las diferencias entre una red ad hoc espontánea y una red ad hoc, la primera es usada en una localización determinada durante un periodo de tiempo, y no depende de un servidor central. El usuario no se requiere que sea un experto, esa red imita las relaciones humanas para trabajar juntos en grupos, con mínima intervención del usuario.

A menudo, las redes espontáneas ad hoc usan relación de confianza [14], imitando cómo los humanos interactúan, en la creación y gestión. Construyen así una cadena de confianza (también conocida como *trust net*).

El principal objetivo de crear una red espontánea ad hoc es establecer un servicio de gestión de clave distribuida a través del uso de una red de confianza. Por tanto, las claves públicas sólo será necesario obtenerla cuando sea necesario en operaciones futuras.

Más adelante tras la creación de la red espontánea, y cuando se requiera encaminamiento ad hoc, si un nodo confía en un segundo nodo puede enviar mensajes directamente a él. En el caso de que el segundo nodo no confíe en el primer nodo la comunicación no está permitida. Cuando un nodo quiere emitir mensajes hacia un nodo no-confiable lo tiene que hacer a través de un nodo de confianza.

1.3 Protocolos de descubrimiento de vecinos

Los protocolos de descubrimientos de vecinos tienen como objetivo principal el descubrimiento de todos los nodos en rango de transmisión.

En primer lugar, los algoritmos de descubrimiento de vecinos pueden ser aleatorios. En ellos, cada nodo elige transmitir en un tiempo elegido de forma aleatoria o transmitir de acuerdo con una probabilidad, en un *round*. Pueden descubrir todos los vecinos en un tiempo determinado con alta probabilidad (diferente de 1). En segundo lugar, en los algoritmos determinísticos, cada nodo transmite de acuerdo con una planificación en la transmisión predeterminada. Esto les permite descubrir todos los vecinos en un tiempo dado con probabilidad 1. El determinismo requiere a menudo que los nodos sigan premisas poco realistas como la sincronización y conocimiento a priori del número de vecinos. Sin embargo, hay protocolos determinísticos asíncronos en la literatura. Por otro lado, la aleatorización a menudo requiere que el algoritmo presente un tiempo ranurado, conocidos como *rounds* y que los nodos estén transmitiendo o escuchando de forma aleatoria en un *round*. A menudo se requiere de un gran número de *rounds* para lograr un descubrimiento fiable.

Los dispositivos en redes inalámbricas ad hoc a menudo usan baterías que se pueden gastar en un tiempo dado. Por tanto, los algoritmos desarrollados deben tener como objetivo proporcionar una alta eficiencia energética. Así, en esta tesis se tiene en cuenta el consumo energético.

Muchos protocolos de descubrimiento de vecinos de la literatura, esto es, los determinísticos, necesitan una planificación en la transmisión para el descubrimiento. Por otro lado, algunos protocolos aleatorios requieren premisas poco realistas. Entre ellas, no proporcionan detección de la terminación del proceso de descubrimiento o requieren conocer el número de vecinos, y no logran descubrir todos los vecinos con probabilidad 1. Además, no logran manejarse adecuadamente en presencia de colisiones. En cuanto a los modelos de creación de redes espontáneas basadas en la confianza existentes en la literatura, no consiguen manejarse adecuadamente en presencia de colisiones. En ambos casos, se propone protocolos mejorados que logren cumplir con los objetivos y proporcionar mejores prestaciones. Además, hay poco análisis teórico disponible en la literatura. No se suele usar el *throughput*, el *overhead*, el *packet delivery ratio*, o el porcentaje de descubrimientos por round. Tampoco se suele usar el porcentaje de descubrimientos por paquete enviado, la función de distribución acumulada CDF (*Cumulative Distribution Function*) de descubrimientos o el porcentaje de *idle slots*.

El problema a qué nos enfrentamos al proponer protocolos de descubrimiento de vecinos es:

- Los nodos deben poder operar en entornos estáticos.
- Los nodos tienen transceptores de radio de alcance limitado.
- Solo se dispone del modo *half-duplex*.
- Los nodos son desplegados de forma aleatoria en un área dada.
- Los nodos deben operar de forma asíncrona.
- Pueden existir colisiones.
- Los nodos deben tratar con colisiones, bien sea evitándolas o detectándolas y gestionando esas colisiones.
- Los nodos pueden detectar colisiones.
- Los nodos deben poder detectar energía.

- El número de nodos debe ser desconocido.
- Los nodos deben poder iniciar la transmisión en diferentes instantes de tiempo.
- Los nodos deben poder descubrir todos los vecinos con probabilidad 1.
- Los nodos deben saber cuando terminar el proceso de descubrimiento cuando todos los vecinos han sido descubiertos.
- Los protocolos deben poder ser usados tanto en entornos *one-hop* como *multi-hop*.
- El consumo energético también debe ser considerado. Esto es debido a que normalmente los dispositivos se alimentan de baterías que se pueden descargar en un tiempo determinado.
- El protocolo debe poder obtener mejores prestaciones que protocolos existentes.

1.4 Protocolos de selección de vecinos

En las redes inalámbricas ad hoc, son necesarios métodos de selección de vecinos para formar la topología de la red. Principalmente se usan en descubrir a qué vecino se debe enviar la información en *one-hop* para que llegue a su destino.

Tras la selección de vecinos [15], los nodos pueden enviar paquetes a sus destinos situados fuera del rango de transmisión. Para ello se utilizan nodos intermedios de forma que los nodos pasan la información al vecino que conecta con su destino.

Hay muchos parámetros usados para determinar la topología de la red tales como el número de nodos, el número de conexiones, el grado de los nodos y el diámetro de la red [16]. Para obtener la topología de la red tiene que haber una estrategia que los nodos deben seguir para elegir sus vecinos. *Grouping nodes* es una estrategia que proporciona muchos beneficios. Los algoritmos de selección de vecinos decidirán qué parámetros son usados y sus valores, para seleccionar el mejor vecino.

Como ejemplos de métodos de selección de vecinos, en [15] se utiliza selección de vecinos para descubrir un camino óptimo con el objetivo de proporcionar comunicaciones de extremo a extremo. En [17] se permite a cada nodo pasar

el mensaje solo a un pequeño conjunto de vecinos one-hop que cubren todos los vecinos *two-hop* del nodo. Típicamente, cada nodo intenta minimizar su energía eligiendo un adecuado conjunto de vecinos, como se muestra en [16]. Los autores en [18] presentan dos algoritmos de selección de vecinos que logran eficiencia energética y que pueden funcionar con protocolos basados en CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*). Teniendo en cuenta la movilidad, en [19] se presenta la selección de vecinos en escenarios urbanos ad hoc logrado mediante el intercambio de mensajes.

1.5 Detección de colisiones

En la presente tesis se exponen protocolos que hacen uso de la detección de colisiones para su funcionamiento. Una colisión se da cuando dos o más nodos transmiten simultáneamente. Cuando se detecta una colisión, se sabe que la transmisión no tuvo éxito. La detección de colisiones permite a cada nodo saber cuando ha sido descubierto por sus vecinos. Por lo tanto, los nodos pueden dejar de transmitir una vez que han sido descubiertos. Así, la detección de colisiones permite a cada nodo realizar un seguimiento del número de nodos que aún no se han descubierto. Estos nodos podrían adaptar su probabilidad de transmisión en cada ranura de tiempo.

En [5] se hace uso de la detección de colisiones. Los nodos pueden distinguir entre colisiones y ranura inactiva. Un mecanismo de detección de colisiones permite a los nodos distinguir entre el caso cuando dos o más están transmitiendo y cuando ningún nodo transmite.

En [20] se describe una arquitectura de capa física que permite a los receptores detectar colisiones, y presenta un mecanismo de *feedback* que permite que la información de colisión llegue a los transmisores. Esto permite a los nodos dejar de transmitir paquetes tan pronto como saben que hay una recepción con éxito de los mensajes de descubrimiento. Además, se propone un mecanismo fiable y práctico de detección de colisión que proporciona información de colisión a los transmisores. Un transmisor estará seguro de la recepción de sus mensajes de descubrimiento a través de los *feedback* de los receptores. Además, se propone un algoritmo en el cual los nodos dejan de transmitir paquetes tan pronto como saben de la recepción de sus paquetes por parte de los vecinos.

1.6 Detección de energía

La detección de energía puede ser usada para permitir que determinados nodos que emiten su reconocimiento indiquen a otros nodos si han logrado transmitir con éxito.

La detección de energía también fue usada en [5] para detectar si se recibieron *feedbacks* o no.

En [20] se propone un mecanismo de detección de energía en la capa física que le permite estimar a los nodos su estado de recepción. Los nodos no solo son capaces de distinguir entre canales ocupados o inactivos, sino que también son capaces de discernir entre errores y colisiones. Un receptor que utiliza este mecanismo puede distinguir entre recepción con éxito, colisión, error de transmisión, y el caso de canal inactivo. Esto se consigue usando 2 umbrales y comprobando si se puede decodificar la señal para distinguir entre éxito y colisión. Si se logra decodificar bien la señal, esto significa que la transmisión tuvo éxito.

Además de usar el mecanismo de detección de energía, se propone un mecanismo simple de *feedback* que proporciona información de colisión a los transmisores. Con el *feedback*, solo se necesita llevar un bit de información a los transmisores, para indicar si ocurrió colisión durante la transmisión o no. Los receptores simultáneamente transmiten paquetes NACK (*Negative Acknowledgement*) durante una ranura reservada para *feedback* si detectan colisión en la ranura previa de transmisión. Si un transmisor detecta energía durante la ranura de *feedback* concluye que ocurrió una colisión durante su transmisión. Los nodos que transmitieron durante el primer *sub-slot*, escuchan el canal durante el segundo *sub-slot* y realizan detección de energía. Si detectan su estado de recepción como éxito, colisión, o error, concluyen que los paquetes NACK fueron transmitidos en el segundo *sub-slot* y una colisión ocurrió en el primer *sub-slot* de tiempo. En teoría solo se necesita un bit de *feedback* por tanto el *overhead* es despreciable. Por tanto, si un nodo transmite un mensaje de descubrimiento y no recibe NACKs de los receptores, concluye que su transmisión tuvo éxito. Una vez que un nodo sabe que su paquete ha sido recibido por sus vecinos, puede dejar de transmitir mensajes de descubrimiento. Sin embargo, debe continuar escuchando el canal para recibir mensajes de descubrimiento de sus vecinos.

1.7 Motivación

En la actualidad, gran cantidad de dispositivos necesitan ser desplegados en redes inalámbricas sin infrestructura de comunicaciones, tanto en redes ad hoc, como en IoT (*Internet of Things*) o *smart cities*. En los próximos años se espera que el uso de este tipo de redes se incremente. Por ello, se requiere que los dispositivos se autoconfiguren para ser usados en operaciones futuras como el encaminamiento. Las aplicaciones han aumentado, pudiendo aplicar los protocolos en entornos médicos, agrícultura, o enseñanza.

La mayoría de comunicaciones en la actualidad se basan en redes con infraestructura pero la demanda de redes sin infraestructura ha crecido en los últimos años. Esto es debido a que ciertas aplicaciones requieren el uso de redes inalámbricas ad hoc, no basándose en redes con una infraestructura tras el despliegue, con limitados recursos. Este requerimiento se debe a la naturaleza de la aplicación, esto es, despliegue en bosques, montañas, entornos militares, etc. Así, es necesario el desarrollo de protocolos de autoconfiguración en redes inalámbricas ad hoc.

1.8 Objetivos

El objetivo principal de la presente tesis es proponer y evaluar protocolos de descubrimiento de vecinos, de creación de redes espontáneas basadas en la confianza y de selección de vecinos. Estos protocolos serán desarrollados en el contexto de redes inalámbricas ad hoc. Las soluciones no deben seguir una planificación en la transmisión, deben ser capaces de operar adecuadamente en presencia de colisiones, siguiendo premisas más realistas. Aún así deben lograr un bajo consumo energético y obtener mejores prestaciones que las soluciones existentes en la literatura. El motivo principal es debido a que, aunque hay muchos trabajos que solucionan el descubrimiento, aún falta por abordar el diseño de protocolos que sigan premisas más realistas. Esto permitirá que se puedan usar en entornos prácticos. Además, las prestaciones se pueden mejorar.

El primer objetivo es desarrollar y evaluar protocolos de descubrimiento de vecinos determinísticos y su modelo analítico. Esto nos permitirá determinar sus prestaciones, concluyendo que obtienen óptimas prestaciones pero requiere premisas poco realistas. A continuación se centra el estudio en el objetivo de desarrollar protocolos aleatorios, que no sigan una planificación en la transmisión, modelarlos analíticamente y obtener las prestaciones. Con el objetivo de

hacer que los protocolos sigan premisas más realistas se propone un protocolo aleatório basado en líder, que principalmente cumple el requisito de no requerir sincronización. También se tiene como objetivo el tener en cuenta el consumo energético proponiendo un protocolo consciente de la energía que mejora esta métrica en determinadas circunstancias. A continuación, se tiene el objetivo de aplicar un protocolo de descubrimiento desarrollado para diseñar un protocolo para la creación de una red ad hoc basada en la confianza. Finalmente, con el objetivo de que la red tenga un *gateway* que permita la disponibilidad de datos con el exterior se propone un protocolo para el descubrimiento y selección de vecinos.

1.9 Estructura de la tesis

La presente tesis está organizada en los siguientes capítulos:

- **Capítulo 2. Estado del arte**: se presentan trabajos relacionados con el descubrimiento de vecinos, con la autoconfiguración en redes espontáneas basadas en la confianza, y con la selección de vecinos.
- **Capítulo 3. Protocolos de descubrimiento de vecinos determinísticos evitando colisiones**: se presentan 2 protocolos de descubrimiento determinísticos que dependen de una planificación predeterminada. Después se analizan sus prestaciones de forma analítica y se compara las prestaciones con dos protocolos de referencia.
- **Capítulo 4. Protocolos de descubrimiento de vecinos aleatorios basados en detección de colisiones**: se exponen 2 protocolos basados en detección de colisiones que permiten el descubrimiento siguiendo premisas más realistas. A continuación se compara con 2 protocolos de la literatura. La principal aportación es que ambos protocolos no dependen de una planificación predeterminada.
- **Capítulo 5. Protocolo de descubrimiento de vecinos asíncrono basado en líder con detección de colisiones**: se presenta un protocolo de descubrimiento aleatorio en el cual el descubrimiento lo coordina un líder. Después se compara con un protocolo determinístico basado en líder y otro protocolo aleatorio de referencia. En este caso, la principal aportación es que el protocolo ya no requiere de sincronización dado que el líder sincroniza el resto de nodos.

- **Capítulo 6. Protocolo de descubrimiento de vecinos aleatorio consciente de la energía basado en detección de colisiones**: se presenta un protocolo con el objetivo de mejorar el consumo energético. A continuación se compara con un protocolo consciente de la energía de la literatura. En este caso, se requiere sincronización pero se centra en mejorar el consumo energético.

- **Capítulo 7. Modelo analítico para protocolos de descubrimiento de vecinos aleatorios basados en la detección de colisiones**: se presenta un modelo analítico con el objetivo de evaluar las dos propuestas aleatorias. Para ello se modela según varias métricas y se procede a su representación gráfica, y se comparan con dos protocolos de referencia.

- **Capítulo 8. Protocolo para la creación de redes espontáneas inalámbricas ad hoc basadas en la confianza**: se presenta un protocolo para iniciar una red espontánea usando el protocolo CDPRR presentado en el capítulo 4 en dos fases. Permite el intercambio de claves y la comprobación de una firma para determinar qué vecinos son de confianza.

- **Capítulo 9. Descubrimiento y selección de vecinos basada en la gestión de prioridades**: se presenta un protocolo que combina el descubrimiento de vecinos con la selección de nodos favoritos para ser usados por ejemplo como *gateway*. Este protocolo se basa en difundir las prioridades y en informar de qué nodos son los favoritos.

- **Capítulo 10. Comparación cualitativa de propuestas**: se presenta una clasificación de las propuestas incluidas en la tesis, así como una comparación cualitativa de las propuestas de descubrimiento de vecinos.

- **Capítulo 11. Conclusiones**: se exponen las conclusiones de todo el trabajo llevado a cabo, y las contribuciones de la presente tesis. Finalmente se enumeran los posibles trabajos futuros como extensión de esta tesis, y las publicaciones resultantes del trabajo desarrollado.

Capítulo 2

Estado del arte

En este capítulo se presentan trabajos relacionados desarrollados para lograr el descubrimiento de vecinos y la creación de redes espontáneas basadas en la confianza. También se presentan protocolos de selección de vecinos. En la literatura, hay muchos trabajos que tratan sobre el descubrimiento de vecinos y la creación de redes espontáneas basadas en la confianza. Además, en la literatura hay protocolos de selección de vecinos. Algunos de ellos se detallan a continuación.

2.1 Protocolos aleatorios

En [5] se describe el *Coupon Collector's problem*, un problema clásico usado en redes inalámbricas, y se presentan algoritmos prácticos para el descubrimiento de vecinos. Se considera el protocolo *ALOHA-like* y se explica el funcionamiento cuando el protocolo no tiene detección de colisiones, reduciéndose al *Coupon Collector's problem*. También se considera el hecho de que exista detección de colisiones, y se propone un algoritmo basado en el envío de *feedbacks* por parte del receptor.

En [21] se presenta un protocolo de descubrimiento eficiente que tiene en cuenta el consumo energético. El protocolo tiene éxito al lograr ahorrar energía durante el despliegue y está diseñado para redes estáticas inalámbrias ad hoc. Los protocolos presentados pertenecen a los *Birthday protocols*, una familia de protocolos probabilísticos. Se presentan 3 modos disponibles. También se

presenta el *PRR* (*Probabilistic Round Robin*), que consiste en una analogía probabilística del algoritmo determinístico de planificación *round robin*. *PRR* logra maximizar la probabilidad de descubrimiento de vecinos cuando el ahorro de energía no es importante. Sin embargo, no presenta buena eficiencia energética, y puede fallar al descubrir algunos de los vecinos en redes densas. El protocolo *PRR* será usado como referencia con el objetivo de compararlo con las propuestas presentadas en la presente tesis.

En [22] los autores se centran en el impacto de colisiones e interferencias en el descubrimiento de vecinos en redes inalámbricas estáticas *multi-hop*. Se presentan también dos protocolos, conocidos como *Basic Hello protocol* y *Energy-aware Hello protocol*, este último logra reducir el consumo energético. Además, se han considerado tres modelos de radio en el cual las colisiones e interferencias se manejan de diferentes formas. Ambos protocolos se usarán como referencia para compararlos con las propuestas presentadas en esta tesis.

En [23] se presentan varios protocolos aleatorios para redes estáticas ad hoc, discute las prestaciones resultantes que dependen de las premisas tenidas en cuenta. Se presenta el protocolo *ALOHA-like algorithm* para redes *one-hop* de N nodos, que logra descubrir todos los vecinos en $O(N \ln N)$. Un protocolo orden óptimo en redes *one-hop*, que permite descubrir todos los vecinos en $O(N)$, un resultado razonable logrado incluso cuando los nodos no pueden detectar colisiones. También hay disponible una extensión a un escenario *multi-hop* general, el cual mejora al algoritmo *ALOHA-like*. A continuación se incluyen los resultados obtenidos relajando dos premisas. Los autores concluyen que la ausencia de un estimador del número de vecinos N o la falta de sincronización producen peores prestaciones. En concreto, resulta como mucho en un desaceleramiento de no más de un factor de 2 en comparación con cuando los nodos conocen N o cuando los nodos están sincronizados. En conclusión, algunos de los protocolos permiten a los nodos iniciar la ejecución del descubrimiento en diferentes instantes de tiempo. Además, se les permite a los nodos saber cuándo terminar el descubrimiento tras descubrir todos los vecinos. Finalmente, hay disponible una extensión a un entorno inalámbrico *multi-hop* más general. Esta extensión logra mejores prestaciones que el protocolo *ALOHA-like*.

FRIEND [24, 25] es un protocolo aleatorio síncrono *full-duplex* basado en *pre-handshaking* para redes estáticas. También se presenta la operación *half-duplex*, escenarios *multi-hop*, y redes *duty cycled*. De acuerdo con resultados analíticos y de simulación, los protocolos en [24, 25] logran mejorar el tiempo consumido en el descubrimiento de vecinos. En concreto, lo mejoran hasta un 68% en comparación con los protocolos *ALOHA-like* presentados previamente en [26].

Además, el protocolo *FRIEND* logra reducir la probabilidad de generar *idle slots* y colisiones.

Direct Algorithm y Group Testing with Binning, dos protocolos prácticos y escalables desarrollados desde el punto de vista *group testing* para redes estáticas se presentan en [27]. La complejidad del Direct Algorithm es $O(k(\log k)^2 \log\log k)$. El protocolo presenta buenas prestaciones a medida que el número total de nodos es grande, sin embargo su complejidad puede ser mejorada. Por esta razón, se propone el Group Testing with Binning. Este protocolo usa *binning* para partir el descubrimiento en varios problemas más pequeños. Cada uno de ellos implica muchos menos candidatos. El protocolo proporciona una complejidad resultante de $O(\lceil \frac{1}{\beta} \rceil max\{k^\beta (\log k^\beta)^2 \log\log k^\beta, \lceil k^{1-\beta} \rceil\})$. Sin embargo, se puede diseñar un sistema tal que k^β es una constante con complejidad $O(k \log k)$. Ambas propuestas logran alta precisión de descubrimiento y presentan un menor consumo temporal que esquemas de descubrimiento de acceso aleatorio similar al algoritmo Birthday-listen-and-transmit [21].

PSBA (*Prime-set-based neighbor discovery algorithm*)[28], es un protocolo aleatorio basado en un conjunto de primos que funciona bien en WSNs móviles con bajo *duty cycle*. Cada nodo elige aleatoriamente un primo p de un conjunto de primos (relacionado con el *duty cycle*) y que será usado como planificación. Los nodos se despiertan cada p ranuras en un ciclo y esto se repite por un periodo predefinido. PSBA mejora la cola larga de los algoritmos probabilísticos. De acuerdo con los resultados, los autores concluyen que PSBA mejora al Birthday protocol [21] en latencia promedio cuando el *duty cycle* es 1%. Además, la típica cola larga de los protocolos probabilísticos se mejora. PSBA también mejora al Birthday protocol, Disco [29] y SearchLight [30], en latencia promedio cuando el *duty cycle* es de 1-5%. Además, los autores concluyen que las prestaciones son mejores para PSBA en latencia promedio y consumo energético a medida que el *duty cycle* se reduce, en comparación con algoritmos existentes.

Panda [31] (*Power Aware Neighbor Discovery Asynchronously protocol*), es un protocolo probabilístico generalizado. Representa el primer protocolo de descubrimiento de vecinos disponible para nodos EH (*Energy Harvesting*). Panda-D está también disponible en [31], una versión que ajusta la tasa de descubrimiento de vecinos. Esta versión extiende el protocolo para funcionar bien en recolección de energía no homogénea. Por tanto sigue premisas más realistas. De acuerdo con los resultados, los autores concluyen que para más altos presupuestos de energía, la latencia de descubrimiento es mejorada. Panda mejora el SearchLight-E de baja energía (SearchLight [30] para presupuesto de

energía) y BD-E de baja energía (Birthday [21] para presupuesto de energía). Esta mejoría se produce en más de $x3$ en tasa de descubrimiento promedio. Además, Panda mejora al protocolo SearchLight-E en el caso peor de latencia de descubrimiento hasta un 40%. Panda y Panda-D tienen similar consumo energético y tasas de descubrimiento en un escenario *one-hop* con presupuestos de energía homogéneos. Sin embargo, en un escenario *multi-hop* la tasa de descubrimiento entre nodos está alrededor del 1% de la tasa de descubrimiento analítica para redes *one-hop*. También hay disponible una implementación de Panda en un prototipo de nodo único EH de muy baja energía basado en TI eZ430-RF2500-SEH. Un resultado importante es que Panda es altamente práctico y puede ser usado cuando los nodos se alimentan de baterías no recargables. El presupuesto de energía se ha fijado basándose en el tiempo de vida deseado, por tanto es adecuado para entornos reales.

Nihao [32] es un protocolo asíncrono centrado en la eficiencia energética para escenarios tanto simétricos como asimétricos. Este es el primer trabajo que usa la métrica COR, producto del *duty cycle*, latencia y tasa de ocupación de canal. S-Nihao (*Simplified Nihao*) es una versión que usa sólo una ranura de *wake-up* en un ciclo de planificación, y garantiza el descubrimiento bidireccional. De acuerdo con resultados analíticos los autores concluyen que S-Nihao es mejor que el LL-Optimal (Combinatoric) [33], dado un *duty cycle*, y proporciona un límite de latencia menor. S-Nihao ampliamente mejora soluciones existentes cuando sólo se consideran el *duty cycle* y la latencia ($x10$ mejor con *duty cycle* 5% y $x50$ mejor con *duty cycle* 1%). G-Nihao (*Generic Nihao*) es otra versión que garantiza el descubrimiento y proporciona una buena granularidad de *duty cycle* en el caso asimétrico. B-Nihao (*Balanced Nihao*) es otra versión, la más apropiada para aplicaciones prácticas en el caso simétrico con las mejores prestaciones. También hay disponible una implementación para Nihao en TinyOS 2.1.2 y y IEEE 802.15.4 radio compatible. De acuerdo con resultados del mundo real, B-Nihao es más rápido que Birthday [21], Disco [29], U-Connect [34] y SearchLight [30], para *duty cycles* de 1% y 5% y logra el límite de latencia más bajo. G-Nihao presenta mejor latencia que Disco, U-Connect, SearchLight y BlindDate [35], para *duty cycles* de 1% y 5%, ya que tiene la menor latencia caso peor.

En [36] se presenta un algoritmo de descubrimiento de vecinos basado en *gossip* y un algoritmo de encaminamiento para MANETs. El protocolo usa antenas direccionales inteligentes para optimizar el consumo energético. Se han propuesto dos protocolos. El primero de ellos, tiene como objetivo incrementar el número de vecinos descubiertos teniendo en cuenta los nodos que están situados en el segundo salto. El segundo protocolo tiene como objetivo reducir el

número de saltos en una ruta entre origen y destino logrado mediante antenas direccionales. Se implementa un mecanismo de *handshake* donde los nodos usan el sector de antena para descubrirse mutuamente. También se asume que los nodos están sincronizados. Se han obtenido resultados de simulación mediante Matlab. Estos muestran la reducción en el consumo temporal y el incremento en el *throughput* en comparación con otros protocolos de encaminamiento reactivos.

En [37] se presenta un algoritmo de descubrimiento de vecinos adaptativo basado en información histórica y que usa *backoffs*. El protocolo utiliza detección de colisiones y mecanismo de *feedback*. Se permite ajustar la ventana de contención, decrementándola para acelerar el proceso de descubrimiento. El protocolo se ha evaluado en cuanto a tiempo de descubrimiento y consumo energético, y se ha averiguado el tamaño óptimo de la ventana de contención.

La Tabla 2.1 y la Tabla 2.2 resumen las características de los protocolos aleatorios de descubrimiento de vecinos de la literatura, presentados en esta sección.

Según la Tabla 2.1, todos los protocolos presentados en ella son asíncronos y operan con bajos *duty cycles*. Además, todos permiten su uso en entornos tanto simétricos como asimétricos. En Panda y Nihao hay una implementación disponible, mientras que PSBA permite su uso en MANETs.

Tabla 2.1: Comparación cualitativa de protocolos aleatorios de descubrimiento de vecinos.

	[21]	[22]	[28]	[31]	[32]
Asíncrono	✓	✓	✓	✓	✓
Bajos *duty cycles*	✓	✓	✓	✓	✓
Parámetros	p_s, τ	ω, s	primo p		
Duty cycle		✓	relacionado con p	✓	✓
Periodo de planificación	1 round (τ)	1 round ω	1 periodo	✓	✓
Asimétrico	✓	✓	✓	✓	✓
Simétrico	✓	✓	✓	✓	✓
Encender radio	En un round (duración τ)	$\omega - t_w/t_w \in [0, s]$	p ranuras	Periodo activo	1 ranura en un ciclo
Primos balan/no balan					
Implementación				TI eZ430-RF2500-SEH	TinyOS 2.1.2 (ATMega128RFA1 MCU)
Uso en MANETs			✓		

Tabla 2.2: Comparación cualitativa de protocolos aleatorios de descubrimiento de vecinos.

	[21]	[22]	[23]	[24]	[27]	[28]	[31]	[32]	[36]
Red móvil									✓
Tiempo ranurado	✓	✓	✓	✓	✓	✓	✓	✓	✓
Aleatorio	✓	✓	✓	✓		✓	✓		✓
Asíncrono	✓	✓	✓			✓	✓	✓	
One-hop	✓	✓	✓	✓	✓	✓	✓	✓	✓
Multi-hop		✓	✓	✓		✓	✓	✓	✓
N desconocido		✓	✓			✓	✓		
Maneja colisiones	✓	✓	✓	✓	✓			✓	✓
Full-duplex				✓					
Pre-handshaking				✓					
Group testing					✓				

Según la Tabla 2.2, sólo el basado en *gossip* se puede usar en MANETs, el tiempo está ranurado para todos los protocolos, y casi todos excepto FRIEND, *group-testing* y el basado en *gossip* son asíncronos. Todos los protocolos excepto el Birthday protocol y *group-testing* se pueden usar en entornos *multi-hop*. Además, todos los protocolos manejan las colisiones excepto PSBA y Panda.

2.2 Protocolos determinísticos

Se presentan varios protocolos que tratan con el consumo energético y abordan el descubrimiento de vecinos.

Disco [29] es un protocolo de descubrimiento de vecinos asíncrono para aplicaciones de detección móviles. Como es práctico, puede fácilmente ser usado en entornos reales. De acuerdo con este protocolo, los nodos operan en bajos *duty cycles* ahorrando energía y logra descubrir los vecinos durante encuentros oportunistas. Además, es rápido y fiable. Para este propósito, los nodos proceden a elegir un par de números primos. La suma de los recíprocos de estos números es igual al *duty cycle* elegido. Además, cada nodo tiene un contador local, el protocolo lo incrementa y el nodo enciende su *radio* por un tiempo dado si su valor es divisible por cualquiera de los números primos. Luego transmite o escucha o hace ambas cosas. Un descubrimiento de vecinos ocurre cuando dos nodos encienden sus *radios* durante el mismo periodo de tiempo. De acuerdo con sus prestaciones, este protocolo logra latencias de descubrimiento determinísticas mucho mejores que las logradas por Quorum [38] y la familia de Birthday protocols [21]. Esta mejoría se produce en escenarios asimétricos y

las latencias se reducen en un 30-50%. En cuanto al caso simétrico, las prestaciones son similares a Quorum. Una ventaja notable es que Disco aborda un conjunto de problemas de descubrimiento de vecinos más general, y evita la necesidad de protocolos probabilísticos, tales como ALOHA o Birthday.

SearchLight [30] es un protocolo de descubrimiento de vecinos asíncrono basado en sondeos, que combina componentes determinísticas y probabilísticas. Para abordar las prestaciones, considera un compromiso entre latencia y consumo energético. Para el caso simétrico, logra unas buenas prestaciones. En el caso promedio es comparable a los del protocolo probabilístico y mejora a los protocolos determinísticos en los límites del caso peor. En cuanto a los casos asimétricos, sus prestaciones son similares a las logradas por los protocolos determinísticos cuando hay un alto grado de asimetría. Se han llevado a cabo simulaciones y los resultados muestran que SearchLight mejora a los protocolos existentes en eficiencia energética. También mejora en cuanto a latencia promedio hasta un 25% sobre protocolos existentes para muy bajos *duty cycles*. Su comportamiento es similar a los otros protocolos en los restantes casos. SearchLight mejora a otros protocolos existentes en el caso simétrico, mientras que sus prestaciones son similares en el caso asimétrico. Los autores proponen dos protocolos para determinar la planificación: SearchLight-S (*Sequential probing*) y SearchLight-R (*Randomized probing*). SearchLight-R mejora al SearchLight-S en todos los escenarios, y este hecho demuestra el beneficio de usar la propuesta aleatoria para determinar la planificación.

U-Connect [34] es un protocolo asíncrono cuyo objetivo es lograr eficiencia energética y baja latencia. Logra solucionar tanto los casos simétrico como asimétrico. Según el protocolo, el tiempo está ranurado y el periodo de tiempo entre consecutivas ranuras de escucha viene dada por el *duty cycle* deseado. A los nodos se les permite elegir diferentes *duty cycles*, por tanto eligen diferentes números primos dado que el *duty cycle* y los números primos están relacionados. Así, los nodos se despiertan en tiempos que son múltiplos de números primos. Para evaluar el protocolo, los autores usan la métrica del producto de potencia y latencia. En cuanto a las prestaciones, U-Connect mejora protocolos existentes logrando mucho menores latencia para un *duty cycle* dado en WSNs. Para el caso asimétrico, U-Connect logra al menos las mismas prestaciones que Disco [29]. En el escenario simétrico, el comportamiento asintótico de U-Connect es mucho mejor. Además, los resultados muestran que U-Connect logra bajo consumo energético en comparación con Quorum [38] y Disco. Además, la latencia caso peor para U-Connect es la misma en los casos asimétricos y simétricos. Un punto importante es que U-Connect se

puede ver como un protocolo unificado que se puede aplicar tanto en escenarios simétricos como asimétricos.

Centron [39] logra mejorar la probabilidad de descubrimiento con éxito y minimizar las colisiones y energía en regiones compuestas de muchos nodos. El protocolo se compone de dos partes. En la primera, conocida como formación de núcleo, se intercambian mensajes de invitación para construir un pequeño grupo de núcleo que actuará como un gran nodo móvil que genera su propio *duty cycle*. En la segunda fase, los miembros de núcleo lanzarán el descubrimiento de vecinos por turnos. El creador y cada miembro tendrán otra negociación para compartir sus tablas de vecinos. Se han obtenido resultados matemáticos usando Matlab. De acuerdo con estos resultados los autores concluyen que Centron mejora protocolos existentes en consumo energético. Además, los resultados para escenarios asimétricos y simétricos son similares. También se han obtenido resultados de simulación usando NS-3, *Stockholm data* y IEEE 802.11 protocolos de descubrimiento ad hoc en un entorno *one-hop*. A través de estas simulaciones, los autores concluyen que Centron mejora protocolos existentes hasta un 20% en latencia promedio. También mejora protocolos existentes en consumo energético y eficiencia en el descubrimiento en regiones muy concurridas, reduciendo las colisiones que no son necesarias.

Hedis (*Heterogeneous Discovery Quorum-based protocol*) y Todis (*Triple-Odd based discovery co-primality based protocol*) [40] permiten el grano fino de *duty cycle* heterogéneo. Esto significa que cada nodo puede operar en un *duty cycle* diferente. Ambos protocolos logran un descubrimiento de vecinos asíncrono en un entorno heterogéneo y también límites superiores de latencia. Hedis es un protocolo de descubrimiento de vecinos periódico basada en ranura en el cual la latencia promedio es $O(nm)$ siendo n,m enteros positivos. En cuanto a Todis, satisface la propiedad de pares co-primos para *duty cycles* prácticos y presenta la más baja latencia de descubrimiento. También usa un compromiso entre latencia y energía. Este protocolo es más simple que el previo en el diseño y proporciona una mayor cantidad de números a elegir en comparación con Disco [29] y U-Connect [34]. Se han llevado a cabo simulaciones para averiguar las prestaciones y los autores concluyen que tanto Hedis como Todis ampliamente mejoran a todos los protocolos existentes. Ambos ahorran energía con una granularidad más fina de *duty cycle* y optimizan esa granularidad. Además, ambas propuestas proporcionan *duty cycles* prácticos, que permiten decrementar la energía consumida por tanto las baterías durarán más tiempo. U-Connect mejora Hedis y Todis en latencia, mientras que ambas propuestas se comportan de forma similar en latencia. Un punto importante es que Hedis es uno de los pocos protocolos cuyas prestaciones están por encima del promedio

en latencia tanto en los casos heterogéneos como homogéneos. Hedis tiene una granularidad más fina que Todis. Además, Todis presenta la latencia más alta en los escenarios caso peor mientras que su latencia es baja en la mayoría de los casos. Ambos protocolos se han implementado en un dispositivo *smartphone* Xiaomi Mi-Note en Android y soporta BLE (*Bluetooth Low Energy*). La implementación en el mundo real ofrece resultados de latencia variando el *duty cycle* y concuerdan con los resultados de simulación. En cuanto a Todis, presenta las mejores prestaciones en latencia en entornos heterogéneos mientras que tiene las segundas mejores prestaciones en el caso homogéneo. En conclusión, Hedis es el protocolo más apropiado para ser usado en WSNs, mientras que soporta *duty cycles* prácticos en una granularidad fina. Aún así proporciona un buen límite de latencia, y prolonga la vida de la batería.

Quorum-based [38] es un protocolo de descubrimiento de vecinos asíncrono multi-canal basado en *handshake* para redes cognitivas de *radio* auto-organizadas ad hoc móviles (MANETs). Cada nodo puede transmitir o recibir señales escaneando y encontrando su propia conjunto de canales de frecuencia disponible. Cuando dos vecinos potenciales sintonizan sus transceptores el mismo canal de frecuencia durante un AI (*Advertisement Interval*) se logra un descubrimiento con éxito. Esto se produce en redes síncronas o al menos una parte de él para redes asíncronas. Sin embargo, el protocolo puede ahorrar energía cuando los nodos descubren que algunos de los canales no están disponibles. Se establece un compromiso de forma que se puede lograr un descubrimiento más rápido y con menos consumo energético. Hay disponibles dos protocolos, esto es, Grid Techniques y Sync Grid Techniques. Sync Grid Techniques proporciona menor consumo energético y es apropiado para redes síncronas aunque puede ser usado en escenarios asíncronos. Se han llevado a cabo simulaciones y los autores concluyen que en escenarios síncronos y asíncronos ambas técnicas logran casi los mismos resultados. Un resultado importante es que el protocolo es casi independiente del conjunto de canales disponible o la densidad de red. Los posibles conflictos pueden ser despreciados usando un tiempo *backoff* aleatorio antes de cada transmisión. Sin embargo, este protocolo requiere de un *hardware* dedicado debido a su operación multi-canal, y los dispositivos deben tener transceptores capaces de receibir en dos canales independientes al mismo tiempo. En el modo transmitiendo no se necesita transmisión simultánea en dos canales independientes, y no puede tratar con los casos asimétricos.

ND_HC [41] es un algoritmo de descubrimiento de vecinos *cross-layer* para redes inalámbricas grandes, que hace uso de TDMA (*Time Division Multiple Access*), agrupación en clústeres de red hexagonal normal, y GPS (*Global Positioning System*). Los mensajes de Hello se generan en la capa MAC (*Me-*

dium Access Control) y son transmitidos siguiendo una manera TDMA con un *backoff* aleatorio. ND_HC logra reducir las colisiones y mejora el *throughput*, mientras que está libre de colisiones. De acuerdo con los resultados de simulación a través de NS-2, ND_HC mejora al ND 802.11 en eficiencia de descubrimiento.

En [42], se lleva a cabo un estudio que se centra en el descubrimiento de vecinos contínuo para WSNs móviles con bajo *duty cycle*. En él, se presenta el descubrimiento de vecinos contínuo y se discute el uso de U-Connect [34], Disco [29], Hedis y Todis [40], SearchLight [30], PBD [43], y otras propuestas. También se resume los protocolos de la literatura.

Panacea [44] es un protocolo eficiente para WSNs, que logra baja latencia y consumo energético, teniendo en cuenta las colisiones. De acuerdo con los resultados, los autores concluyen que hay un límite de $O(N \ln N)$ en la latencia para diferentes *duty cycles* en Panacea-NCD (sin detección de colisión). Cuando la detección de colisiones es posible, en Panacea-WCD hay también un límite en la latencia de $O(N \ln N)$. Además, las evaluaciones coinciden con los resultados analíticos.

La Tabla 2.3 resume las características de los protocolos determinísticos de descubrimiento de vecinos de la literatura, presentados en esta sección.

Según la Tabla 2.3, todos los protocolos son asíncronos y operan con bajos *duty cycles*. Todos excepto el Quorum permiten su uso en entornos asimétricos, mientras que todos se pueden usar en entornos simétricos. Hay una implementacién en Disco, SearchLight, U-Connect, Hedis y Todis. Todos los protocolos pueden ser usados en MANETs.

Tabla 2.3: Comparación cualitativa de protocolos determinísticos de descubrimiento de vecinos

	[29]	[30]	[34]	[38]	[40]	[39]
Asíncrono	✓	✓	✓	✓	✓	✓
Bajos *duty cycles*	✓	✓	✓	✓	✓	✓
Parámetros	$(p_{i1}, p_{i2}), (p_{j1}, p_{j2})$	$t \in Z$ (un primo)	p (un primo)	$m \in Z^+$	$n \in Z^+$ misma paridad / n, m impar	
Duty cycle	$\frac{1}{p_1} + \frac{1}{p_2}$	$\frac{2}{t^i}$	$\frac{3 \times p+1}{2 \times p^2}$	$\frac{2 \times m-1}{m^2}$	$\frac{2}{n}$ / $\frac{3}{n}$	
Periodo de planificación	$T = p_1 \times p_2$	$\frac{t^2}{2}$	$T = p^2$	$T = m^2$ AI (Intervalo de Anuncio)	$n \times (n-1)$ / $(n-2) \times n \times (n+2)$	
Asimétrico	✓	✓	✓		✓	✓
Simétrico	✓	✓	✓	✓	✓	✓
Encender *radio*	Múltiplos de primos	Ranura de ancla (0) y sonda	1 de cada p ranuras	Canal de frecuencia en un AI	ancla y sonda / múltiplos de $n-2$ or n or $n+2$	
Primos balan/no balan	✓					
Implementación	Telos motes (Tiny OS)	Smartphones Android G1	FireFly Badge		Mi-Note Android (BLE)	
Uso en MANETs	✓	✓	✓	✓	✓	✓

2.3 Basados en *Wake-up*

En [45] se presenta un protocolo de descubrimiento de vecinos para MANETs con reconocimiento de información social. Los nodos incluyen tanto un *radio wake-up* como transceptor de radio que permite operación *half-duplex*. La señal de *radio wake-up* y los mensajes de Hello son difundidos y luego los receptores cambiarán de modo inactivo a activo. El *framework* pasivo de descubrimiento introducido permite el uso en aplicaciones sociales móviles. De acuerdo con las simulaciones a través de NS-2, la propuesta mejora Disco [29], U-Connect [34] y SearchLight [30] en latencia y consumo energético. Además, hay disponible una implementación en un *smartphone*.

PWEND [46] es un protocolo de descubrimiento de vecinos para MSNs (*Mobile Sensor Networks*). Puede proporcionar mejor latencia, logra reducido consumo energético a través de un mecanismo basado en *wake-up*. Además, la latencia de descubrimiento caso peor puede ser reducida. De acuerdo con las simulaciones a través de Matlab, PWEND mejora las soluciones existentes, tales como G-Nihao [32], $QConnect_A$ [47], Disco [29] y SearchLight (*stripe*) [30] en latencia y consumo energético.

2.4 MANETs altamente dinámicas

KPND (*Kalman Prediction-based Neighbor Discovery*) [48] es un protocolo que tiene como objetivo mejorar la latencia y eficiencia y puede ser usado en MANETs altamente dinámicas. KPND está basado en un modelo de predicción de movilidad usando la teoría de filtro Kalman y mensajes Hello, y GPS permite detectar cuando los vecinos se unen y dejan la red. De acuerdo con resultados de simulación a través de NS3.28 y Mobisim, KPND mejora HP-AODV, ARH [49] y ROMSG [50].

En [51] se presenta un protocolo de descubrimiento de vecinos adecuado para MANETs altamente dinámicas donde los nodos tienen recursos limitados. El protocolo combina encaminamiento, planificación y descubrimiento de vecinos. La propuesta usa versiones de AODV (*Ad hoc On-Demand Distance Vector*) y CSMA (*Carrier Sense Multiple Access*) para lograr descubrimiento de ruta ciega y reenvío de paquetes al mismo tiempo. De acuerdo con resultados de simulación, la propuesta presenta un comportamiento apropiado. Además, el protocolo es robusto en movilidad, fallo, o en el caso de nodos uniéndose a la red.

En [43] se presenta un protocolo de descubrimiento de vecinos para redes oportunistas móviles, que es consciente de la movilidad. El protocolo reduce el esfuerzo en el escaneo. Un análisis teórico evalúa la eficiencia energética y reenvío de datos a través de simulaciones teniendo en cuenta varios modelos de movilidad.

2.5 Antena y radar

En [52] se presenta un protocolo de descubrimiento de vecinos para ser usado en redes MTC (*Machine-type communication*) inalámbricas ad hoc que usa las capacidades del radar. De acuerdo con resultados numéricos, la latencia del protocolo es mejor con información previa del radar. Se concluye que el proceso puede ser acelerado cuando se usan mecanismos *stop-discovery* y *non-response*. Además, la propuesta mejora CRA (*Completely Random Algorithm*) [53] en latencia. Sin embargo, el radar y la comunicación deben ser integrados, mientras que se debe asumir sincronización y operación *half-duplex*.

En [54], se presenta un algoritmo de descubrimiento de vecinos modelado como un autómata de aprendizaje. En él, los nodos pueden aprender sobre su entorno y de observaciones previas y logra un descubrimiento más rápido en redes densas. La propuesta inteligente basada en el aprendizaje se basa en un autómata de aprendizaje de estado finito (FLA) y logra el descubrimiento con una alta probabilidad. Los nodos incluyen una antena direccional orientable y usan una manera ALOHA-like para transmitir. De acuerdo con los resultados de simulación, la propuesta presenta mejor latencia y una clara mejora con respeto al *2-way random handshaking protocol* [55] y al *scan based algorithm* [56].

RCI-SBA [57] es un algoritmo de descubrimiento de vecinos *two-way handshaking* basado en escaneo para redes ad hoc, que se centra en la eficiencia energética, integra radar y comunicaciones. En RCI-SBA, los nodos incluyen antenas direccionales, y hace uso de señales de radar y comunicación y GPS. Los resultados de análisis matemático prueban que la propuesta logra mejor consumo energético. De acuerdo con los resultados de simulación, el consumo energético de CRA [53] es peor que el de RCI-SBA. El protocolo RCI-SBA mejora al SBA (*Scan Based Algorithm*) [56] en consumo energético.

La Tabla 2.4 resume las características de protocolos recientes de descubrimiento de vecinos de la literatura, presentados en esta sección.

Tabla 2.4: Comparación cualitativa de protocolos recientes de descubrimiento de vecinos.

	[48]	[52]	[51]	[41]	[54]	[57]	[45]	[46]
Red móvil	✓	✓	✓		✓		✓	✓
Tiempo ranurado	✓	✓		✓	✓	✓	✓	✓
Aleatorio	✓	✓	✓		✓	✓		
Asíncrono	✓		✓	✓	✓	✓	✓	✓
One-hop	✓	✓	✓	✓	✓	✓	✓	✓
Multi-hop	✓	✓	✓					
N desconocido	✓	✓	✓					
Maneja colisiones	✓	✓	✓	✓	✓	✓	✓	✓
Full-duplex								
Pre-handshaking								
Group testing								

Según la Tabla 2.4, todos los protocolos se pueden usar en redes móviles excepto el ND_HC y RCI-SBA. El tiempo está ranurado excepto en el protocolo en [51]. Todos excepto el protocolo en [52] son asíncronos. Solo se permite su uso en entornos *multi-hop* a los protocolos [48], [52] y [51]. Finalmente, todos los protocolos son capaces de manejar las colisiones.

2.6 Protocolos seguros

En [58] los autores se centran en la seguridad de muchas sesiones del protocolo que lanza monedas además de primitivas criptográficas estándar contra un adversario DolevYao. Los autores se centran en el secreto, para determinar si un adversario puede determinar un secreto. También se centran en la indistinguibilidad, para determinar si la probabilidad de observar es la misma para diferentes observadores bajo el mismo adversario. Ambas métricas son *coNP-complete* para protocolos no aleatorios. Sin embargo, los autores demuestran que, para protocolos aleatorios, el secreto e indistinguibilidad son ambos decidibles en coNEXPTIME. También existe un límite inferior para el problema de secreto logrado al reducir el problema de la no satisfacibilidad de lógica de primer orden monádica sin igualdad.

2.7 Control de acceso

En [59] se incluye un *survey* para control de acceso en IoTs, aborda diferentes aplicaciones y necesitan una gran cantidad de información privada del usuario por tanto pueden surgir problemas de seguridad. En este contexto, el control de acceso se usa para asegurar el acceso de los usuarios a recursos de información autorizado bajo condiciones legítimas. El *survey* analiza los principales problemas y retos del control de acceso en entornos heterogéneos altamente dinámicos de la vida real. Para este propósito, [59] proporciona una guía teórica y una técnica para el control de acceso en IoT, y analiza futuras direcciones del control de acceso en IoTs.

2.8 Redes espontáneas basadas en la confianza

En [60] se explican las diferencias entre redes ad hoc y redes espontáneas. Se identifican 5 retos clave introducidos por el entorno de redes espontáneas. Uno de los puntos principales que marca la diferencia entre una red espontánea y redes fijas o móviles es que hacen fácil la integración de servicios y dispositivos. Además, se fijan nuevos servicios y parámetros de configuración de dispositivos. Deben ser llevados a cabo sin intervención del usuario o interferencia en la operación de la red. La mala operación o fallo en uno de los dispositivos o servicios no compromete la viabilidad de la comunidad.

En [61] se propone SCOPE, un prototipo para redes sociales espontáneas P2P (*Peer-to-peer*). Por debajo de la capa de red, SCOPE sigue el modo ad hoc 802.11 y no necesita infraestructura. SCOPE sigue el modelo jerárquico P2P. Algunos nodos con una capacidad de cómputo mayor se convierten en super-nodos. Los super-nodos forman un *overlay* y proporcionan el sistema de administración de datos distribuido para las redes sociales P2P. Los nodos cliente conectan con los super-nodos y confían en ellos para compartir sus contenidos o acceder a información compartida.

Un método para unir redes espontáneas se propone en [62]. Los autores presentan una propuesta para unir implícitamente redes espontáneas siguiendo un modo de movilidad de grupo. El protocolo de encaminamiento entre células evita cuellos de botella en los enlaces. Algunos protocolos de encaminamiento jerárquico se basan en la elección de una célula *cluster-head* (o nodo punto de referencia).

En [63] se pueden encontrar algunos ejemplos sobre IoTs en el HP Labs CeNSE project. Los autores se centran en el despliegue de una red de sensores

de extensión mundial para crear un "Sistema nervioso central para la Tierra". También se centran en el proyecto "A Smarter Planet", una estrategia desarrollada por IBM. Este proyecto considera los sensores como base fundamental en sistemas inteligentes de gestión del agua y ciudades inteligentes.

En [64] los autores se centran en un reto específico: el actual modelo de conectividad entre la WSN e Internet. Los autores intentan responder si los nodos sensores deberían delegar todas las comunicaciones de Internet a un conjunto de sistemas centrales de gestión. O, por el contrario, si deberían convertirse en ciudadanos de primera clase de Internet implementando la pila entera TCP/IP y también otros estándares como servicios web.

Un análisis de un protocolo seguro para redes espontáneas inalámbricas ad hoc se presenta en [65], centrándose en dispositivos con recursos limitados. De acuerdo con el protocolo, se pueden intercambiar servicios y recursos. La seguridad para este tipo de redes usa un esquema de gestión de clave híbrido simétrico/asimétrico para intercambiar los datos. Las tarjetas de identidad son cifradas antes del intercambio, y un esquema de clave simétrico se usa para cifrar los datos. El esquema de criptografía de clave simétrico usa el algoritmo AES (*Advanced Encryption Standard*). El intercambio de certificado será cifrado usando un esquema de criptografía de clave asimétrico algoritmo ECC (*Error-Correcting Code*) debido a sus mejores resultados. La confianza se obtiene por contacto visual o por procedimiento de autenticación usando una clave de sesión.

En [66], se presenta un protocolo seguro para la creación de redes espontáneas inalámbricas ad hoc para acceder a las IoTs. El protocolo se basa en la interacción directa P2P y comunidades, y es usado por diferentes tipos de dispositivos con recursos limitados. El protocolo tiene como objetivo mejorar la comunicación en Intranet y en Internet, y la integración entre diferentes comunidades con bajos recursos. Este protocolo permite a los usuarios acceder de forma segura a la WWW (*World Wide Web*) por conexión compartida a Internet entre comunidades a través de un solo o varios nodos que usan TCP/IP. Se han llevado a cabo simulaciones usando Castalia. En los experimentos, un servidor web se conecta a una nube IP y simula un comportamiento Internet y diferentes redes espontáneas se conectan también a esta nube IP. De acuerdo con los resultados de simulación, se obtiene un 61% de mejora con respecto a una arquitectura convencional, el tráfico es más estable y muestra menores fluctuaciones. Finalmente, hay disponible un prototipo en [66].

Un protocolo de creación de red espontánea móvil ad hoc de computación en la nube [67], permite compartir recursos de computación y aplicaciones. Se

propone un servicio AC distribuido. La gestión de seguridad se basa en una infraestructura de clave pública para autenticación de usuario. Cada usuario mantiene un repositorio local de certificados de clave pública y sus valores de confianza. La propuesta usa un resumen SHA-1, un esquema de cifrado de clave asimétrico usa RSA (*Rivest Shamir Adleman*) y ECC, principalemente usado en el proceso de autenticación de usuario. El cifrado de clave simétrico usa el algoritmo AES, y se usa como una clave de sesión. Se usa Bluetooth en el proceso de autenticación. Se han obtenido resultados de simulación usando Castalia 2, logrando buena eficiencia y prestaciones incluso con un número de nodos elevado. Se ha implementado un prototipo para simular la creación de un sistema de computación en la nube móvil usando una red espontánea ad hoc.

Un protocolo seguro auto-configurado [68] para la creación y administración de redes espontáneas inalámbricas ad hoc distribuidas y descentralizadas, se centra en dispositivos de baja potencia. Cuando un nuevo nodo se une a la red, usa una tarjeta de identidad, *hash* SHA-1 y certificado. Utiliza la confianza entre usuarios para intercambiar la información. Los recursos son compartidos y se ofrecen nuevos servicios. Se ha elegido el algoritmo AES para esquema de cifrado simétrico. Por otro lado, se ha usado ECC y RSA para esquema de cifrado asimétrico. Más adelante, cuando la red ha sido creada, los servicios son compartidos por medio de conexiones TCP usando la tecnología IEEE 802.11b/g. Bluetooth o ZigBee. Esto permite la autenticación de nodos cuando se unen a la red. Se ha desarrollado un prototipo usando Java (J2ME). También hay disponible una implementación real en un dispositivo móvil Nokia E65 en una red espontánea. Se han llevado a cabo varios *tests* para validar la operación del protocolo y comparar el protocolo con otros protocolos para red espontánea ad hoc. Los tiempos de respuesta obtenidos son adecuados para su uso en entornos reales, incluso cuando los dispositivos tienen recursos limitados. Los autores concluyen que las necesidades de almacenamiento y memoria volátil son bastante bajas y el protocolo puede ser usado en dispositivos con restricciones de recursos.

Un protocolo seguro completamente auto-configurado se presenta en [69] para la creación de redes espontáneas inalámbricas ad hoc. El protocolo usa un mecanismo de pre-distribución de clave basado en la confianza del usuario para intercambiar información inicial y las claves secretas. También comparte servicios y recursos, y se centra en dispositivos con recursos limitados. Un usuario puede crear sus propios recursos o puede pedirlos de sus vecinos. Para lograr autenticación de nodo, se requiere de un mecanismo de intercambio de clave. En la creación de la red, el primer paso tiene lugar cuando un nuevo nodo

se une a la red e intercambia tarjetas de identidad. Luego, tiene lugar una fase de acceso a servicio. Finalmente, se forma una cadena de confianza. Además, la propuesta usa el algoritmo AES como esquema de cifrado simétrico, y un esquema asimétrico. De acuerdo con los resultados de simulación, los tiempos de ejecución y el consumo energético puede ser mejorado por este protocolo. Además, una ventaja destacable es que los autores presentan una técnica de detección de intrusión.

Se presenta un protocolo seguro completo auto-configurado de clave simétrica en [70] para la creación y administración de redes espontáneas inalámbricas ad hoc móviles independientes y descentralizadas. El protocolo tiene como objetivo el mejorar la comunicación e integración entre diferentes centros de estudio de comunidades con bajos recursos. Este protocolo se usa para compartir recursos y muchos servicios de Internet a toda la red, donde solo un nodo se conecta a Internet. Se usa un esquema de detección de intrusos para miembros que se unen a la red. Esta propuesta usa criptografía asimétrica para identificación de dispositivo, y criptografía simétrica para compartir claves de sesión. En un paso de unión, el sistema maneja tarjetas de identidad y certificados. Se lleva a cabo infraestructura de clave pública y la clave pública se usa como una clave de sesión. Los dispositivos deben colaborar en la Intranet o en la Internet. La conexión puede ser compartida y el primer nodo de la red proporcionará acceso a la WWW si tiene conexión a Internet. Sin embargo, para el acceso a Internet podrían haber más de un nodo y cada nodo puede compartir diferentes servicios. Los autores muestran el diseño y simulación de un modelo que permite acceso óptimo a red espontánea usando un mecanismo de almacenamiento en caché. Una propuesta analítica, una validación a través de simulaciones y comparación con arquitecturas regulares y los protocolos más similares de la literatura, está disponible en [70].

En [71] se presenta un protocolo de seguridad auto-configurado completo. El protocolo se basa en la confianza de usuario para la creación y administración de redes espontáneas inalámbricas ad hoc móviles distribuidas y descentralizadas. Este protocolo permite compartir servicios y recursos. También proporciona mecanismos de intercambio de clave para autorización de nodo y autenticación de usuario para lograr una comunicación fiable. En un paso de unión del nodo a la red, el protocolo usa tarjetas de identidad, certificados y *hash* SHA-1. El primer nodo de la red crea la red y una clave de sesión provisional. El cifrado simétrico usa el algoritmo AES para compartir claves de sesión. Los tiempos de ejecución y consumo energético en los procedimientos de criptografía son adecuados para dispositivos de baja potencia. El esquema de cifrado de clave asimétrica usado es ECC y RSA para identificación de dis-

positivo. Los nodos deben colaborar en la Intranet o en Internet. El acceso a WWW está disponible si un usuario tiene conexión a Internet. Sin embargo, podría haber más de un nodo para el acceso a Internet, donde cada nodo podría compartir diferentes servicios.

Un protocolo completo seguro auto-configurado ligero se presenta en [72] para redes espontáneas inalámbricas. El protocolo usa un esquema híbrido simétrico/asimétrico y confianza entre usuarios para intercambiar la clave de sesión y las claves para cifrar la información. Es capaz de crear la red y compartir servicios y recursos seguros para ser usado en dispositivos con recursos limitados. Los tiempos de respuesta obtenidos son adecuados para ser usado en entornos reales, y el almacenamiento y memoria volátil requerido es bastante bajo.

Un protocolo completo seguro es presentado en [73] para redes espontáneas inalámbricas ad hoc, basado en la confianza entre nodos colaboradores. El protocolo usa cifrado simétrico AES, mientras que el esquema de clave asimétrica usa RSA, para autenticación de usuario distribuida. Ha sido diseñado para ser usado en dispositivos móviles con recursos limitados, y requiere espacio de memoria limitado y energía. Proporciona un mecanismo de detección de intrusión, usado para detectar los nodos que pueden ser atacantes. Un usuario autenticado puede mostrar los nodos, actualizar la información, procesar una solicitud de autenticación, responder a una solicitud de información, enviar datos a un nodo, o dejar la red.

Un protocolo seguro auto-configurado [74] para la creación de redes espontáneas ad hoc, usa un esquema híbrido simétrico/asimétrico, y confianza entre usuarios para intercambiar datos. Las claves secretas son compartidas para cifrar los datos que transmiten. Puede ser usado en dispositivos con recursos limitados. El protocolo permite compartir recursos y servicios en la red, y la confianza se logra por solo los nodos de nivel cero y primer nivel. Los certificados para todos los nodos que se unen a la red se obtienen a partir de un nodo de confianza, y son usados para comunicar con otros nodos. Un metodo de firma tiene como objetivo el proteger contra un ataque de repudio.

Un protocolo seguro auto-configurado es presentado en [75] para crear y administrar redes espontáneas distribuidas y descentralizadas. El objetivo es la distribución de datos, recursos y servicios compartidos entre los usuarios. El protocolo permite a dispositivos de diferentes tipos unirse y dejar la red en cualquier momento. Una red de confianza se puede construir para obtener una AC distribuida entre los usuarios que confían en un nuevo usuario. Se proporciona criptografía asimétrica (RSA) y criptografía simétrica (AES) para intercambiar las claves de sesión. Cada dispositivo tiene un par de claves públi-

ca y privada para identificación de dispositivo y no hay usuarios anónimos. Se usa un resumen SHA-1 para crear una firma. El nivel de confianza, esto es, o de confianza o no de confianza, se establece mirando físicamente (visión directa). Este nivel de confianza puede cambiar dependiendo del comportamiento del nodo, incluso dejar de confiar. Se han obtenido resultados de simulación usando NS-2 para validar el protocolo, con respecto a *packet delivery ratio*, *throughput*, y promedio de consumo energético.

EESCSP [76] es un protocolo seguro auto-configurado que se centra en la eficiencia energética para creación y administración de redes espontáneas. El protocolo está basado en el establecimiento de confianza cara a cara entre nodos que se unen y autentican, proporcionando seguridad total. Proporciona seguridad mientras que la unión y acceso de servicios y recursos en la red sin conexión Internet usa mecanimos de establecimiento de nivel de confianza que serán seguros. Además el protocolo tiene como objetivo usuarios no expertos. Los usuarios pueden unirse o dejar la red. El protocolo tiene como objetivo ahorrar energía de nodos en el momento de que un nuevo nodo se una. Se usa RSA algoritmo de cifrado asimétrico para autenticación mientras que se usa AES algoritmo de clave simétrico para comunicación y la clave de sesión para cifrar mensajes. Se crean certificados por resumen SHA-1. El protocolo puede construir una red de confianza para obtener la AC distribuida y usa ordenadores portátiles como dispositivos móviles. Su objetivo es la creación y administración realizando integración automáticamente con poca intervención del usuario. Hay una implementación en Java 1.6 o superior en Windows 7, que crea una red espontánea inalámbrica LAN usando técnicas de Wi-Fi en vez de usar Bluetooth entre ordenadores portátiles. SPSNC (*Secure Protocol for Spontaneous Network Creation*) se compara con el protocolo seguro con eficiencia energética (SPSNC-EE), con respecto a latencia, *packet hop count* y *packet delivery rate*. Se obtienen mayores valores para SPSNC-EE en las 3 métricas.

Un protocolo seguro autónomo distribuido y ligero es presentado en [77]. Su objetivo es la creación, comunicación y administración de redes espontáneas inalámbricas ad hoc. Para ello usa un esquema híbrido simétrico/asimétrico y la confianza entre usuarios para intercambiar los datos iniciales y las claves secretas. Se basa en una red social imitando el comportamiento de las relaciones humanas. La confianza se basa en el primer contacto visual entre usuarios, está diseñado para dispositivos con recursos limitados, y proporciona seguridad fácil de usar. Una sola clave secreta compartida se usa para crear una red criptográfica de comunicación para autenticar al titular como parte del grupo seguro. El protocolo permite compartir recursos y servicios de forma

segura. Se proporciona criptografía asimétrica en la cual cada dispositivo tiene un par de claves pública y privada para identificación de dispositivo. Se usa criptografía simétrica para intercambiar claves de sesión entre nodos, y cifrar los datos usando la clave de sesión compartida. Las tarjetas de identidad son compartidas usando un algoritmo de criptografía. La revocación de usuarios malintencionados también se realiza proporcionando seguridad mejorada. Hay disponible una implementación usando NS-2. Se han obtenido resultados con respecto al *normalized routing overhead*, *throughput*, *packet delivery ratio*, *delay* promedio, tanto para el protocolo con revocación como sin revocación. Se concluye que el protocolo se comporta mejor con el método de revocación que sin revocación.

Un protocolo seguro completo independiente auto-configurado descentralizado y distribuido para la creación de redes espontáneas inalámbricas ad hoc se presenta en [78]. El protocolo usa un esquema híbrido de clave pública y privada. También usa la confianza entre usuarios para intercambiar los datos iniciales e intercambiar las claves secretas que serán usadas para cifrar los datos. La creación, comunicación y administración de la red, y hay disponible detección de intrusión. Permite operación del usuario fácil de usar, adaptado para diferentes dispositivos, crear la red y compartir servicios y recursos seguros. La propuesta tiene como objetivo mejorar la comunicación e integración entre diferentes centros de estudio de comunidades con bajos recursos. Se usa criptografía asimétrica (RSA) para identificación de dispositivo y autenticación, y criptografía simétrica (AES) para compartir claves de sesión entre nodos. Los autores se centran en la detección de intrusión usando una técnica de detección de firma para realizar un seguimiento de los intrusos. La clave de sesión es revocada periódicamente para evitar la inundación de la red. Hay disponible una implementación para probar el protocolo y compararlo con otros protocolos para red espontánea ad hoc. De acuerdo con los resultados, el retraso promedio en la propuesta es mejor que en arquitecturas regulares.

Un protocolo seguro completo auto-configurado se presenta en [79] para redes espontáneas inalámbricas ad hoc descentralizadas y distribuidas. El protocolo usa un esquema híbrido de clave pública y privada, y la confianza entre usuarios para intercambiar los datos iniciales e intercambiar las claves secretas. Su objetivo son los dispositivos con espacio de memoria limitado y limitada energía. Permite compartir servicios y recursos seguros, y distribución de datos seguros entre usuarios autorizados de una manera fácil de usar. Se usa clave simétrica como clave de sesión para cifrar el mensaje por AES. Por otro lado, se usa clave asimétrica para autenticación de usuario y distribución de clave de sesión por RSA. Hay disponible un sistema de detección de intrusión para

detectar diferentes tipos de ataques, para proteger la red, analizar y descubrir intrusiones. Hay una implementación para probar el protocolo y compararlo con otros protocolos de red espontánea ad hoc. Se requiere mínima implicación por parte del usuario para configurar el dispositivo principalmente para establecer confianza. El protocolo también realiza revocación de clave de sesión para evitar la inundación de la red.

Un protocolo seguro completo auto-configurado ligero se presenta en [80] para redes espontáneas inalámbricas ad hoc distribuidas y descentralizadas. El protocolo usa un esquema híbrido simétrico/asimétrico, con poca intervención del usuario y la integración de diferentes dispositivos. El objetivo es crear y gestionar tales redes, y compartir datos en dispositivos con limitados recursos. Un usuario sin conocimiento técnico avanzado puede construir una red espontánea y participar, y el protocolo proporciona facilidad de uso por parte del usuario. Las direcciones IP identifican a cada nodo, se comparten servicios seguros usando conexiones TCP, y la red se construye usando tecnología IEEE 802.11b/g para compartir recursos. Bluetooth permite la autenticación de nodos cuando se unen a la red. También hay disponibles mecanismos de revocación de clave de sesión y detección de intrusos. Se proporciona una implementación usando J2ME y una máquina virtual rápida KVM. La implementación de protocolos de comunicación se ha hecho tanto en Wi-Fi como Bluetooth. *Crypto*, esto es, una solución *Bouncy Castle Lightweight API* se ha elegido dado que proporciona una API criptográfica ligera *open source*. En cuanto a los resultados, los tiempos de respuesta obtenidos son adecuados para ser usados en entornos reales. Por otro lado, las necesidades de almacenamiento y memoria volátil son bastante bajas y el protocolo se puede usar en dispositivos con recursos limitados.

El protocolo en [81] crea una red segura espontánea ad hoc usada por diferentes dispositivos, y permite a los nodos usar los servicios disponibles. Tras la creación, los nodos están agrupados en clústeres y un *cluster head* es asignado a cada clúster. Cuando un nodo en un clúster necesita acceder a un servicio un método es usado para encontrar y adquirir la mejor calidad de servicio disponible de otros nodos que han usado el servicio. Estos nodos proporcionan información tales como *delay* y tasa de transmisión. Basándose en esta información el valor de confianza será calculado para esos nodos. Para ello, se tiene en cuenta estos valores de confianza los nodos que necesitan servicio deciden desde qué nodo debe ser accedido el servicio. Cuando un nodo proporcionando un servicio se mueve a otro clúster la gestión de la historia del servicio proporciona información sobre el nodo migrado. La unión de nodos a la red depende de la tarjeta de identidad, y el nivel de confianza se establece mirando

físicamente. Una clave de sesión creada aleatoriamente es distribuida a todos los nodos de la red. Los servicios usados en esta propuesta son la transmisión de ficheros. Cuando el nivel de potencia de la batería del actual *cluster head* cae por debajo de un umbral predeterminado o sirve por un periodo de tiempo predeterminado, difunde (en el clúster) un nuevo mensaje de elección. Todos los nodos votan por un nuevo *cluster head* y el *cluster head* decide el ganador basándose en mayoría simple. Se han obtenido resultados experimentales para su comparación. De acuerdo con el porcentaje que los nodos puede acceder a la calidad de servicio, la propuesta es mejor que el sistema existente. En cuanto al *overhead* frente a número de nodos, la propuesta tiene menos *overhead* en comparación con el sistema existente.

En la Tabla 2.5 se muestra una comparación cualitativa de las soluciones para redes espontáneas presentadas en esta sección y la propuesta, que será presentada en el Capítulo 8.

Tabla 2.5: Comparación cualitativa de protocolos de trabajos relacionados con las redes espontáneas y la propuesta.

	[66]	[67]	[68]	[69]	[70]	[71]	[72]	[73]	[74]	Propuesta
IoTs	✓									
Red en la nube		✓								
Llamada a la red va al servidor web conectado a la nube IP	✓									
Red móvil		✓		✓	✓	✓	✓	✓	✓	
Red espontánea inalámbrica ad hoc	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Crea red	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Gestiona red	✓	✓	✓	✓	✓	✓	✓	✓		
Crea recursos		✓		✓	✓	✓	✓	✓	✓	
Comparte datos seguros				✓	✓		✓	✓	✓	
Comparte servicios y recursos		✓		✓	✓	✓	✓	✓	✓	
Ofrece servicios seguros		✓		✓	✓	✓	✓	✓	✓	
Prototipo desarrollado	✓	✓	Java(J2ME) con KVM Mobile Nokia E65							
Despliegue real										
Dispositivos con recursos limitados	✓		✓	✓	✓		✓	✓	✓	✓
Sistemas heterogéneos (diferentes dispositivos)	✓		✓		✓	✓	✓	✓	✓	✓
Comunidades con bajos recursos	✓									
Dispositivo con identidad única	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Comunidad con identidad de grupo única	✓									
Simulación	Castalia/ OPNET	Castalia	✓							Castalia
Fase de descubrimiento de vecinos	✓	✓								
Umbral de vecino	✓	✓								
Lista de tarjetas de vecino	✓	✓	✓	✓	✓	✓				✓
Tarjeta de identidad	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

	[66]	[67]	[68]	[69]	[70]	[71]	[72]	[73]	[74]	Propuesta
Par de claves pública-privada	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Certificado firmado por clave privada	✓	✓	✓	✓	✓	✓	✓	✓	✓	
Intercambio de tarjetas de identidad	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Función de resumen hash	✓	SHA-1	SHA-1	✓	✓	SHA-1	SHA-1	SHA-1	SHA-1	SHA-1
Repositorio local de certificados de clave pública y valores de confianza		✓	✓			✓				✓
Mínima interacción de usuario (aplicación amigable de usuario)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Usuarios no expertos	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Fase autenticación				✓	✓	✓	✓	✓		✓
Fase de preautenticación	✓									
Confianza establecida por un usuario	✓	✓	✓	✓	✓	✓	✓	✓	✓	
Confianza establecida automáticamente	✓						✓	✓	✓	✓
Pre-autenticación de usuario decide el nivel de confianza	✓									
Cadena de confianza	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Rango de confianza	✓									
Solo dos niveles de confianza		✓	✓	✓	✓	✓	✓	✓	✓	✓
Cambio de valores de confianza	✓		✓		✓	✓	✓			
Confianza modificable basándose en el comportamiento	✓		✓		✓	✓	✓			
Puede también dejar de confiar						✓	✓	✓		
Claves pública obtenidas a traves de red de confianza	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Administración de servicio distribuida a través de red de confianza	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Infraestructura de clave pública	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Cada nodo actúa como cliente/servidor	✓	✓	✓			✓		✓		
Los nodos se unen y dejan la red a voluntad en cualquier momento	✓	✓	✓	✓	✓	✓	✓	✓	✓	
El primer nodo crea y establece la red			✓		✓		✓	✓	✓	
Acceso a Internet a todos los nodos	✓	✓			✓	✓				
Comparte muchos servicios de Internet		✓			✓					
Colabora en la Intranet/Internet	✓	✓			✓	✓				
Redes distribuidas	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Administración central										
Redes independientes					✓	✓	✓	✓	✓	✓
Red autoconfigurada			✓	✓	✓	✓	✓		✓	✓
Valor de confianza basado en relaciones humanas	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

	[66]	[67]	[68]	[69]	[70]	[71]	[72]	[73]	[74]	Propuesta
Dispositivos comportamiento similar a relaciones humanas	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Algoritmo de pre-distribución de clave				✓						
Cifrado de clave simétrico		AES	AES	AES	✓	AES	✓	AES	AES	
Cifrado de clave asymétrico		RSA/ ECC	RSA/ ECC	✓	✓	RSA/ ECC	✓	RSA	RSA/ ECC	
Criptografía adecuada para dispositivos de baja energía						✓	✓	✓	✓	
Clave de sesión		✓	✓		✓	✓	✓	✓	✓	
AC distribuida (cada nodo actúa como AC)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Servicio de nombres distribuido						✓				
La AC para un nodo cualquiera de los nodos de confianza	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Relación de confianza puede ser asimétrica						✓	✓	✓		
Confianza basada en proximidad física (primer contacto visual)	✓		✓	✓	✓	✓	✓	✓	✓	
Asignación de dirección IP úmica	✓	✓	✓	✓	✓	✓	✓	✓	✓	
Detecta existencia de direcciones IP duplicadas	✓		✓	✓		✓		✓	✓	
Direcciones obtenidas dinámicamente		✓					✓	✓	✓	
Autenticación usando direcciones IP			✓					✓	✓	
Autenticación intercambiando claves				✓		✓	✓	✓	✓	✓
Autorización intercambiando claves				✓		✓	✓	✓	✓	
Key management	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Intercambio inicial de datos/claves				✓	✓	✓	✓	✓	✓	✓
Intercambio de claves secretas para cifrar datos				✓	✓	✓	✓	✓	✓	
Cada nodo solicita servicios de sus nodos de confianza				✓		✓	✓		✓	
Grupos trabajan de una forma colaborativa para el mantenimiento de la red	✓					✓	✓	✓	✓	
Solo un nodo se requiere que esté conectado a Internet	✓	✓			✓	✓				
Más de un nodo puede conectarse para proporcionar acceso a Internet	✓				✓	✓				
Conexión compartida si un usuario tiene conexión a Internet	✓	✓			✓	✓				
Acceso a la WWW si un usuario tiene conexión a Internet	✓				✓	✓				

	[66]	[67]	[68]	[69]	[70]	[71]	[72]	[73]	[74]	Propuesta
Los mejores nodos llevan a cabo comunicaciones a través de Internet	✓				✓	✓				
Servicios compartidos usando conexiones TCP			✓							
Protocolos TCP/IP	✓									
Red construida usando IEEE 802.11b/g			✓					✓	✓	
Autenticación a través de Bluetooth o ZigBee			✓	✓			✓	✓	✓	✓
Basada en redes sociales	✓									✓
Técnica de detección de intrusión				✓	✓			✓		
Técnicas de caché para evitar sobrecarga de los nodos					✓					

2.9 Protocolos de selección de vecinos

El *neighbor selection game* para redes inalámbricas ad hoc en los cuales los nodos eligen sus vecinos de forma que minimizan su consumo energético se presenta en [16]. Además, los autores presentan dos propuestas distribuidas para selección de vecinos: Global Best-Response Algorithm y Local Algorithm. En el primero, los nodos tienen información topológica completa sobre la conectividad de todos los otros nodos de la red. Este algoritmo da como resultado una topología *NE spanning tree*, y tiene un límite de eficiencia energética global. En la segunda propuesta, cada nodo está consciente sólo de los nodos con una limitada distancia de salto, lo que representa un caso más realista. Este algoritmo da como resultado una topología conectada NE. Además, en la selección de vecinos tanto la potencia de transmisión como el conjunto de vecinos son desconocidos.

En [82] hay disponible un estudio sobre diferentes métodos de selección de vecinos, y los autores desarrollaron un mecanismo de selección de vecinos rápido y adecuado para WSNs basadas en grupo. Esta propuesta está basada en el parámetro capacidad definido por los autores, energía y la nueva distancia de vecino. También hay disponible una arquitectura para WSNs, y el protocolo presenta un método para ser usado cuando un nuevo nodo se une al grupo y tiene que seleccionar sus vecinos. De acuerdo con el protocolo, un emisor envía información a su grupo, y envía datos al sensor de frontera para llegar a grupos vecinos. Cuando el sensor en el grupo vecino recibe la información, lo encamina hacia todos los sensores en su grupo. La información se envía rápidamente a los otros grupos. A continuación, el nuevo sensor elige el mejor

sensor del mismo grupo para tener un enlace con él, y el nuevo sensor sabrá a qué grupo se ha unido.

En [19] se presenta un algoritmo de selección de vecinos para redes ad hoc centrándose en movilidad urbana altamente dinámica, llamado FMA (*Friend Management Algorithm*). Cada nodo tiene como objetivo seleccionar vecinos estables en entornos urbanos ad hoc, sólo intercambiando sus mensajes entre vecinos. Para cada nodo en el mismo camino, FMA considera los nodos vecinos que están presentes durante una última duración determinada como "amigos" y filtra los otros. Además, los nodos que están lejos de los vecinos son excluidos de la lista de "amigos". Para cada nodo en una intersección, FMA comienza a monitorizar mensajes para encontrar nuevos "amigos" que posiblemente se muevan juntos en el mismo camino. En movilidad urbana, algunos vecinos partirán en las intersecciones mientras que los nuevos nodos se convertirán en nuevos vecinos. Los vecinos podrían ser "amigos" durante mucho tiempo, lo que supone una mejor calidad de servicio.

En [17] se presenta una propuesta que tiene como objetivo reducir las retransmisiones redundantes del método de inundación. El protocolo permite a cada nodo pasar el mensaje sólo a un conjunto más pequeño de vecinos *one-hop*. Las contribuciones del problema *minimum forwarding set* que tiene como objetivo seleccionar vecinos en redes inalámbicas ad hoc son las siguientes. Un algoritmo con un tiempo exacto $O(N \log^2 N)$, un algoritmo con un tiempo 2-aproximación $O(N \log N)$ cuando todos los vecinos *two-hop* están en el mismo cuadrante con respecto al nodo emisor. Un algoritmo exacto $O(N^2)$, con 6-aproximación y un tiempo $O(N \log N)$, cuando N es el número total de vecinos *one-hop* y *two-hop*. Un algoritmo de 3-aproximación con tiempo $O(N \log^2 N)$. Además, también se proporcionan una aproximación de factor constante para el *Disk Cover problem* y un algoritmo basado en partición que soluciona el *1-Hop Disk Cover problem*. Además, se conluye que cuando los nodos son capaces de ajustar el rango de transmisión, es posible reducir más la congestión.

En [15] se propone una técnica de selección de vecinos segura para su uso en MANETs, que usa *machine learning* inteligente y recurrente basada en recompensas para descubrir nodos de ruta óptima. Resultados experimentales muestran que la propuesta logra mejorar el *throughput*, *packet delivery ratio*, y *ratio* de detección, logrando mejores tiempos. La presencia de atacantes en la ruta produce una degragación en las prestaciones, y por tanto el origen necesita seleccionar vecinos robustos y fiables. En la propuesta de selección de vecinos, los estados de los nodos son pre-clasificados, y se estima y analiza a continuación la recompensa de los vecinos para seleccionar nodos de ruta óptima.

Los autores en [83] presentan unos mecanismos de descubrimiento de vecinos y selección de vecinos fiable. Estos mecanismos se basan en la función de confianza usando una toma de decisiones basado en red de neuronas artificiales (AF). Estos mecanismos están diseñados para WSNs con recursos limitados para mejorar la seguridad de la red. El AFTNS es un proceso dinámico de evaluación de confianza auto-adaptativo que identifica nodos de confianza basándose en sus atributos de una forma dinámica. AFTNS mejora las prestaciones de la red a través de la mejora de una tasa de detección maliciosa y retención del tiempo de vida de la red. La selección de nodos filtra nodos para ser seleccionados basándose en la confianza y energía. Si el valor de confianza de la ruta es mayor que la de los vecinos, se evalúa la secuencia de las rutas. Esto es debido a que algún nodo que se comporta mal está presente en la ruta.

En [84] se analizan algoritmos de selección de vecinos eficientes en energía para el encaminamiento en redes de sensores inalámbricos. Se ha considerado una topología de red plana donde todos los nodos tienen la misma responsabilidad y capacidad. Este trabajo considera protocolos planos por tanto no incluyen descripciones de protocolos jerárquicos. Se presenta un protocolo basado en negociación en tres etapas junto con una subrutina que lo hace eficiente en energía. Los autores presentan dos propuestas: Selección de vecinos teniendo la energía más alta (HE), y Selección de ruta que consume la mínima energía (MECRT). Los resultados muestran que MECRT supera HE con respecto al tiempo de vida de la red y consumo energético. Tras un análisis experimental los autores concluyen que MECRT es mejor para tamaño de red medio y grande en comparación con el algoritmo HE. Sin embargo, ambas técnicas no garantizan la selección de la ruta más corta o el mecanismo de encaminamiento rápido.

En [85] se presenta una propuesta difusa de eficiencia energética para la toma de decisiones para la selección de CHs (*Cluster Heads*) en WSNs. El protocolo MADM (*Fuzzy multiple attribute decision-making*) se usa para seleccionar CHs. Tiene en cuenta la energía residual, número de vecinos, movilidad, y la distancia a la estación base de los nodos, considerado para optimizar el número de clústeres o CHs. Las propuestas MODM (*Pareto optimal technique*) y MADM *fuzzy TOPSIS* se usan para seleccionar CHs. Todos los CHs seleccionados envían mensajes de anuncio en la red declarando su presencia como CHs. Cada nodo mide la distancia a todos los CHs. El nodo se une al CH con mínima distancia y envía un mensaje al CH más cercano. Los resultados de simulación demuestran que *fuzzy TOPSIS* es más eficaz en prolongar el tiempo de vida de la red y ahorro energético en entornos homogéneos WSN.

En [86] se propone un algoritmo de selección de CH eficiente energéticamente para adaptación de clústeres. El modelo propuesto es una extensión del algoritmo de selección de CH estocástico LEACH (*Low-energy adaptive clustering hierarchy*). Para ello se modifica la probabilidad de cada nodo en convertirse en CH basándose en el nivel de energía restante de nodos sensores. El protocolo tiene como objetivo reducir el total de consumo energético de sensores y prolongando el tiempo de vida de la red. Los resultados de simulación muestran que el modelo propuesto podría implementar mejor equilibrio de carga y prolongar el tiempo de vida de la red. La propuesta supera al LEACH con respecto a número de nodos vivos, el número de mensajes de datos recibidos en la estación base y la cantidad total de energía restante en la red. También prolonga el tiempo de vida de la red.

Los autores en [87] proponen un *framework* distribuido basado en la confianza. También presentan un mecanismo para la elección de CHs confiables basado en el consumo de energía. Se usa en el contexto de redes de sensores inalámbricos basados en clúster. El mecanismo basado en la confianza tiene como objetivo principalmente la prevención de que nodos adversos o nodos comprometidos se conviertan en CHs. Tan pronto como se han establecido los clústeres, los CHs crean una planificación de tiempo TDM (*Time Division Multiplexing*) e informan a cada miembro del clúster. A través de escucha pasiva los nodos son capaces de desarrollar relaciones de confianza con sus vecinos. Cuando el nivel de energía de la batería del actual CH cae por debajo de un umbral predeterminado o sirve por un periodo de tiempo predeterminado, difunde (en el clúster) un nuevo mensaje de elección. Todos los nodos votan a un nuevo CH usando un voto secreto. El actual CH decide el ganador basándose en mayoría simple. Finalmente, el CH envía a través de *multicast* el ganador a todos los miembros del clúster.

2.10 Conclusiones

En este capítulo se han presentado diferentes soluciones para el descubrimiento de vecinos, creación y gestión de redes ad hoc basadas en la confianza, y selección de vecinos.

Como se observa, el descubrimiento de vecinos se ha realizado utilizando distintas estrategias, en principio usando técnicas determinísticas y probabilísticas. También se ha usado un mecanismo de *wake-up* para reducir el consumo energético. Algunos autores se centran en MANETs altamente dinámicas, mientras que otros trabajos se basan en el uso de antena y radar.

En cuanto a las redes espontáneas basadas en la confianza, se presentan varias estrategias para diferentes escenarios.

La selección de vecinos se logra de forma que se minimiza el consumo energético, y se basan en grupos y clústeres. Algunos se centran en la movilidad, y otros en reducir las retransmisiones redundantes.

Capítulo 3

Protocolos de descubrimiento de vecinos determinísticos evitando colisiones

En este capítulo se presentan dos propuestas para solucionar el descubrimiento de los vecinos en entornos estáticos one-hop en la presencia de colisiones. Se han llevado a cabo simulaciones con Castalia 3.2. El objetivo es comparar las prestaciones de las propuestas con las de dos protocolos de la literatura, es decir, PRR y Hello. Se han evaluado de acuerdo con seis métricas. De acuerdo con los resultados de simulación, la propuesta Leader-based ($O(N)$) supera a los otros protocolos con respecto a tiempo de descubrimiento de vecinos, y throughput. También los supera en descubrimientos por paquetes enviados, y el ratio de paquetes recibidos por paquetes enviados. La propuesta TDMA-based es la más lenta ($O(N^2)$) y presenta los peores resultados con respecto a consumo energético, y descubrimientos por paquetes enviados. Sin embargo, ambas propuestas siguen una planificación de la transmisión predeterminada que les permite descubrir todos los vecinos con probabilidad 1. También usan un mecanismo de feedback. Además, se ha llevado a cabo un estudio analítico para ambas propuestas de acuerdo con varias métricas. Además, la solución Leader-based solo puede funcionar adecuadamente en entornos one-hop. La propuesta TDMA-based es apropiada para su uso en entornos multi-hop.

3.1 Introducción

Nuestra hipótesis es que se pueden proponer protocolos determinísticos más rápidos y eficientes para redes estáticas ad hoc, y averiguamos sus prestaciones para concluir esto.

En el presente capítulo se centra el estudio en el descubrimiento de vecinos en redes estáticas inalámbricas ad hoc. Se presentan 2 protocolos proactivos determinísticos de descubrimiento de vecinos, esto es, TDMA-based y Leader-based, en la presencia de colisiones. En ambos se asume que siguen una planificación predeterminada.

El leader-based es apropiado para redes *one-hop*, esto es, todos los nodos están en el rango de transmisión de todos los demás. En cuanto al protocolo TDMA-based, también puede ser usado y funciona bien en entornos *multi-hop*, aunque en esos escenarios las prestaciones se degradarían.

Las dos propuestas determinísticas presentadas permiten descubrir todos los vecinos con probabilidad 1. Funciona incluso en redes densas, lo que significa que los nodos tienen una gran cantidad de vecinos, y lo logra en una cantidad de tiempo reducida. Las propuestas tienen como objetivo evitar colisiones, y lograr prestaciones óptimas en entornos estáticos ad hoc. También logran solucionar el problema de los protocolos aleatorios, que no descubren todos los vecinos con probabilidad 1. Soluciona también el problema de los protocolos determinísticos presentados en el capítulo 2 que no se centran en tratar con colisiones.

Como referencia, se ha decidido seleccionar dos algoritmos aleatorios de la literatura, ya que se han usado para comparar en otros trabajos. Estos algoritmos son: PRR [21], que pertenece a la familia de los *Birthday protocols*, y el *Hello protocol* [22].

Entre los problemas encontrados en [21] y [22], se resaltan los siguientes: no se proporciona condición de terminación a menos que se fije un número de *rounds*, y los vecinos no son descubiertos con probabilidad 1. En el protocolo en [21] el número de nodos debe ser conocido. Por tanto, el principal objetivo es proponer protocolos que sepan cuando terminar el proceso de descubrimiento y mejorar la probabilidad de descubrir todos los vecinos.

Las principales contribuciones de este capítulo son: (i) Leader-based, una propuesta determinística que logra descubrir todos los vecinos con probabilidad 1. Sigue una planificación en la transmisión predeterminada, incluye un nodo especial conocido como líder que inicia el descubrimiento. Termina el descubri-

miento de acuerdo con la planificación. Sólo puede usarse en entornos *one-hop*, aunque debe conocer el número total de nodos de la red, (ii) TDMA-based, una propuesta determinística que también logra descubrir todos los vecinos con probabilidad 1. Sigue una planificación en la transmisión predeterminada, termina el descubrimiento de acuerdo con la planificación. Se puede usar tanto en entornos *one-hop* como *multi-hop*, aunque debe conocer cuántos nodos hay en la red, (iii) Una comparación cualitativa de protocolos determinísticos de la literatura, (iv) Una comparación cualitativa de Hello, PRR y las propuestas, (v) Un estudio analítico de las propuestas en cuanto a consumo temporal, consumo energético, y *throughput*. También número de descubrimientos por paquetes enviados, y paquetes recibidos por paquetes enviados, (vi) Una implementación del Leader-based, TDMA-based y protocolos de referencia. Se ha llevado a cabo con Castalia 3.2 [88] para comparar las prestaciones de esos protocolos en cuanto al número de vecinos descubiertos y las otras cinco métricas usadas en el estudio analítico.

Más adelante, en la Tabla 3.1 se presenta una comparación cualitativa de protocolos de descubrimiento de vecinos, los dos protocolos de referencia y las dos propuestas. Se resalta las principales características de cada protocolo. Entre las más importantes: Hello y PRR son aleatorios y el tiempo está ranurado, también son asíncronos aunque requieren sincronización en los límites de ranura. Pueden ser usados tanto en *one-hop* como *multi-hop*, aunque ninguno de los dos es capaz de descubrir todos los vecinos con probabilidad 1. Por otro lado, las propuestas Leader-based y TDMA-based no son aleatorias, son determinísticas, el tiempo no está ranurado, y son síncronos. Siguen una planificación en la transmisión. Están diseñados para ser usados en entornos *one-hop*. Sin embargo, el TDMA-based puede también ser usado en entornos *multi-hop*, degradando sus prestaciones. Ambos son capaces de descubrir todos los vecinos con probabilidad 1.

3.2 Protocolos determinísticos basados en la evitación de colisiones

A continuación, procedemos a presentar un protocolo TDMA-based (con un tiempo de descubrimiento de vecinos cuadrático $O(N^2)$). Además se presenta un protocolo Leader-based (con un tiempo de descubrimiento lineal $O(N)$).

3.2.1 Premisas

Las premisas para los nodos que se deben considerar para ambas propuestas, son las siguientes:

- Los nodos no se permite que se muevan en el área de despliegue, ni entrando y saliendo de la red ni entrando y saliendo del rango de transmisión de otros nodos. Por tanto, no son adecuados para su uso en MANETs.
- Los nodos son desplegados aleatoriamente una vez en un área delimitada.
- Los nodos requieren sincronización, lo que significa que no pueden funcionar de una forma asíncrona.
- Los nodos deben transmitir siguiendo una planificación predeterminada.
- Los nodos tienen identificadores únicos, que les permite destinguirse de otros nodos en la red, por ejemplo el número de serie del fabricante.
- Los nodos incluyen un transceptor de radio cuyo rango de transmisión es limitado. Todos los nodos tienen el mismo rango de transmisión. El transceptor permite a los nodos transmitir o recibir pero no simultáneamente, es decir, solo está disponible el modo *half-duplex*.
- Los nodos hacen uso de una memoria interna, en este caso una tabla de vecinos.
- El número de nodos N debe ser conocido por todos los nodos que forman la red.

En relación con su uso, el protocolo leader-based está diseñado para entornos *one-hop*. En cuanto al protocolo TDMA-based también se comporta bien en escenarios *multi-hop*, aunque las prestaciones se empeorarían.

La Tabla 3.1 muestra información con más profundidad sobre las propuestas y los protocolos de referencia.

3.2.2 Modelo protocolo Leader-based

La propuesta considera la existencia de colisiones, por tanto el principal objetivo es evitarlas y buscar prestaciones óptimas.

De acuerdo con la Figura 3.1, el modelo consiste en tres fases. Primero, un tipo especial de nodo, llamado líder, es elegido de forma aleatoria y luego inicia

Tabla 3.1: Comparación cualitativa de protocolos de descubrimiento de vecinos de referencia y las propuestas.

	[22]	[21]	Leader	TDMA
Entornos estáticos	✓	✓	✓	✓
Entornos móviles				
Protocolo aleatorio	✓	✓		
Tiempo ranurado	✓	✓		
N desconocido	✓			
Requiere sincronización en límites de ranura	✓	✓	✓	✓
Requiere planificación en la transmisión			✓	✓
Half-duplex	✓	✓	✓	✓
Entornos *one-hop*	✓	✓	✓	✓
Entornos *multi-hop*	✓	✓		✓
Modo *sleep* disponible				
Colisiones consideradas	✓	✓	✓	✓
Las colisiones no pierden la transmisión			✓	✓
Detección de pérdida de paquetes				
Líder necesario			✓	
Inicia transmisión en diferentes instantes de tiempo	✓			
Descubre todos los vecinos			✓	✓
Con mecanismo de *feedback*			✓	✓
Requiere gran número de ranuras		✓		
Requiere número de nodos N grande		✓		

el descubrimiento difundiendo su identificador hacia los vecinos potenciales. Tan pronto como el paquete *BROADCAST* llega a sus destinos, comienza una segunda fase en la cual los vecinos deben enviar reconocimientos al líder. Estos reconocimientos se envían uno tras otro de acuerdo con una planificación en la transmisión predeterminada. En ella cada vecino envía un paquete *ACK* con su identificador hacia el líder en una duración fija de N *sub-slots*. En cada *sub-slot* un vecino diferente envía reconocimientos. En el momento en que cada paquete *ACK* llega al nodo líder, éste procede a actualizar su tabla de vecinos con el identificador del vecino en el paquete. Cuando todos los reconocimientos han sido recibidos, una tercera fase inicia y el líder envía un paquete *BROADCAST* conteniendo la tabla de vecinos construida en fases previas y el líder finaliza. Además, tan pronto como este último *BROADCAST* llega a los vecinos, éstos proceden a guardar esta tabla en sus tablas de vecinos locales y finalizan.

Este protocolo se ha diseñado para evitar posibles colisiones, y es un protocolo proactivo. Es apropiado sólo para redes *one-hop*, un caso simple pero útil cuando el rango de transmisión de los nodos es alto, por ejemplo, 500 metros.

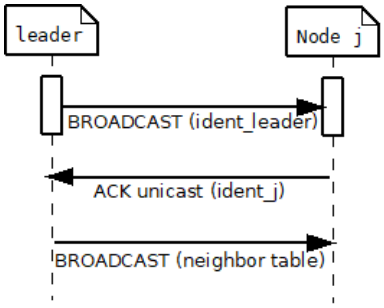

Figura 3.1: Protocolo Leader-based

Con el objetivo de evitar colisiones debidas a varios nodos transmitiendo al mismo tiempo, en la fase 2 las respuestas al líder se llevan a cabo en orden. En este caso, los vecinos envían reconocimientos uno tras otro siguiendo una planificación en la transmisión predeterminada. Por tanto, el Leader-based es una propuesta libre de colisiones. También se ha descubierto que el consumo temporal es lineal $O(N)$, como se verá en la sección 3.5.2. Sin embargo, el protocolo Leader-based está diseñado para escenarios *one-hop*, la cual es una desventaja importante ya que no se permite su uso en entornos *multi-hop*. Además, sólo es adecuado para ser usado en entornos estáticos, lo que significa que no puede ser usado en MANETs.

A continuación, se muestran los resultados analíticos obtenidos para el protocolo Leader-based.

El tiempo de descubrimiento de vecinos total viene dado en la ecuación 3.1, siendo N definido como el número total de nodos de la red, y τ definido como el tiempo que un nodo está transmitiendo.

$$T = (N + 2) \times \tau \tag{3.1}$$

Por tanto, el tiempo de descubrimiento sigue una tendencia lineal $O(N)$.

El promedio del consumo energético por nodo viene dado en la ecuación 3.2.

$$E = \frac{1}{N} \times [(N + 1) \times E_{tx} + (N^2 + N - 1) \times E_l] \tag{3.2}$$

siendo E_{tx} la energía consumida por un solo nodo cuando transmite por segundo y E_l la energía consumida por un solo nodo cuando escucha por segundo.

El *throughput* por nodo viene dado en la ecuación 3.3.

$$Thr = \frac{N^2 + N - 1}{N \times (N + 2) \times \tau} \tag{3.3}$$

En cuanto al número de descubrimientos por paquetes enviados, viene dado por la ecuación 3.4.

$$ratio_1 = \frac{N - 1}{N + 1} \tag{3.4}$$

Finalmente, se muestra la ecuación 3.5 para los paquetes recibidos por enviados.

$$ratio_2 = \frac{N^2 + N - 1}{N \times (N + 1)} = \frac{N^2 + N - 1}{N^2 + N} \tag{3.5}$$

A continuación, el Algoritmo 1 muestra con detalle el funcionamiento de la propuesta Leader-based.

Un problema puede surgir cuando el *BROADCAST* que contiene la tabla de vecinos se pierde, por tanto la tabla de vecinos no será recibida por ningún nodo vecino. En este caso, el descubrimiento de vecinos falla. Una mejora consistiría en enviar simultáneos *UNICASTs*, conteniendo la tabla de vecinos, hacia cada vecino.

3.2.3 Modelo protocolo TDMA-based

La propuesta funciona en dos fases llevadas a cabo por todos los nodos. En la primera, cada nodo envía un paquete *BROADCAST*, que contiene su identificador y llega a todos los vecinos potenciales. En la segunda, justo tras recibir el paquete, cada vecino reconoce con un paquete de respuesta *ACK* que contiene su identificador y se envía hacia el emisor del *BROADCAST*. Cada respuesta sigue un orden planificado, en una duración total de N *sub-slots*. Un vecino diferente envía su paquete de reconocimiento en cada *sub-slot*. Cuando un *ACK* es recibido por un nodo i del vecino j, el nodo i procede a actualizar su tabla de vecinos almacenando el identificador del nodo j, es decir, $ident_j$.

Algoritmo 1 Propuesta Leader-based

Entrada τ tiempo que un nodo está transmitiendo, N número de nodos de la red

1: Elegir aleatoriamente un líder k
2: k transmite $BROADCAST(ident_{leader})$ a los vecinos potenciales
3: k espera $Timer_0 = (N+1) \times \tau$ segundos para que el $BROADCAST$ llegue y para las respuestas de los vecinos
4: **para cada** $Vecino\ j$ **hacer**
5: Cuando el $BROADCAST$ llega al vecino j:
6: j espera $Timer_j = j \times \tau$ segundos a su instante adecuado para enviar reconocimientos
7: Cuando $Timer_j$ ha expirado:
8: j transmite $ACK(ident_j)$ hacia el nodo k
9: **si** $ACK(ident_j)$ es recibido por por el líder k **entonces**
10: líder k guarda $ident_j$ en su tabla de vecinos (NT)
11: **fin si**
12: **fin para**
13: Cuando $Timer_0$ ha expirado:
14: líder k transmite *BROADCAST(NT)* hacia los vecinos y finaliza

Tan pronto como los *ACKs* enviados por todos los vecinos han llegado al emisor del *BROADCAST*, este nodo finaliza el proceso. El siguiente nodo, de acuerdo con un orden planificado, lleva a cabo las fases 1 y 2.

Resaltar que cada nodo envía un paquete *BROADCAST* uno tras otro, de acuerdo con un orden planificado que es implementado en el dispositivo. El funcionamiento se puede observar en la Figura 3.2, en la cual se muestran dos tiempos. Primero, T_i es el tiempo que un determinado nodo tiene que esperar a su momento adecuado para enviar el paquete *BROADCAST*. Esto se produce cuando los nodos previos ya hayan transmitido sus paquetes *BROADCAST* y recibido todos los paquetes *ACK*. En segundo lugar, cada vecino tiene que esperar un tiempo T_j a su momento adecuado para transmitir el paquete *ACK* hacia el emisor del *BROADCAST*. Esto se produce cuando los vecinos previos hayan ya transmitido sus reconocimientos. Luego envía los paquetes *ACK* de nuevo uno tras otro, de acuerdo con un orden planificado implementado en el código del dispositivo. La propuesta TDMA-based es por tanto libre de colisiones ya que todas las transmisiones se llevan a cabo en orden siguiendo una planificación, de forma que las colisiones son evitadas.

A continuación, esos dos tiempos T_i y T_j, se presentarán.

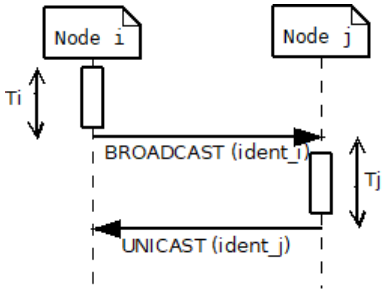

Figura 3.2: Propuesta TDMA-based.

En primer lugar, el tiempo T_i que un nodo i tiene que esperar para enviar el paquete *BROADCAST* se muestra en la ecuación 3.6. T_0 se define como el tiempo en que el descubrimiento de vecinos inicia, N se define como el número de nodos, y τ se define como el tiempo que un nodo está transmitiendo.

$$T_i = T_0 + i \times (N + 1) \times \tau \tag{3.6}$$

En segundo lugar, la ecuación 3.7 muestra el tiempo T_j que un vecino j tiene que esperar para enviar reconocimiento.

$$T_j = j \times \tau \tag{3.7}$$

A continuación, se procede a mostrar los resultados analíticos para la propuesta.

El tiempo total de descubrimiento de vecinos se puede encontrar en la ecuación 3.8.

$$T = N \times (N + 1) \times \tau \tag{3.8}$$

Por lo tanto, el tiempo de descubrimiento de vecinos sigue una tendencia cuadrática $O(N^2)$.

El promedio de energía consumida por nodo viene dada por la ecuación 3.9. E_{tx} es la energía consumida por un solo nodo cuando transmite por segundo y E_l la energía consumida por un solo nodo cuando escucha por segundo.

$$E = N \times E_{tx} + N^2 \times E_l \tag{3.9}$$

El *throughput* por nodo viene dado en la ecuación 3.10.

$$Thr = \frac{N}{(N+1) \times \tau} \tag{3.10}$$

El número de descubrimientos por paquetes enviados viene dada en la ecuación 3.11.

$$ratio_1 = \frac{N-1}{N^2} \tag{3.11}$$

A continuación, mostramos la ecuación 3.12 para los paquetes recibidos por enviados.

$$ratio_2 = \frac{N^2}{N \times N} = 1 \tag{3.12}$$

El funcionamiento de la propuesta TDMA-based se muestra en la Figura 3.3. Las transmisiones de todos los nodos, es decir, *BROADCASTs* y *ACKs*, se llevan a cabo en orden de forma que se evitan las colisiones. Por tanto, el TDMA-based es un protocolo libre de colisiones. Sin embargo, el tiempo de descubrimiento de vecinos sigue una tendencia cuadrática $O(N^2)$ como se muestra en la ecuación 3.8. El tiempo de descubrimiento de la propuesta Leader-based es mejor (tendencia lineal $O(N)$).

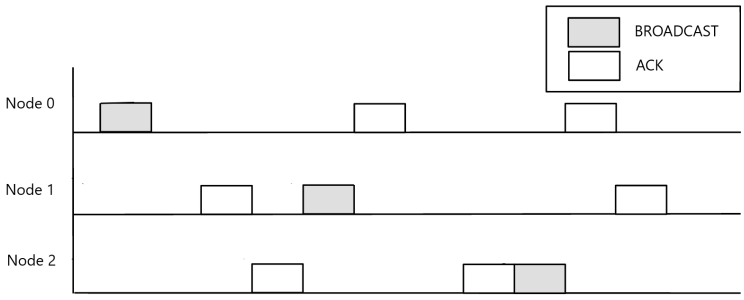

Figura 3.3: Protocolo TDMA-based (línea de tiempos).

A continuación, se presenta el Algoritmo 2, que muestra con detalle cómo funciona la propuesta TDMA-based.

Algoritmo 2 Propuesta TDMA-based.

Entrada T_0 tiempo en que inicia el descubrimiento de vecinos, N número de nodos, τ tiempo en que un nodo está transmitiendo.

1: Espera T_0 segundos para iniciar el descubrimiento de vecinos
2: **para cada** $Nodo\ i$ **hacer**
3: $T_i = T_0 + i \times (N + 1) \times \tau$
4: i espera T_i segundos hasta su momento de transmisión
5: i envía $BROADCAST(ident_i)$ a los vecinos potenciales
6: **para cada** $Vecino\ j$ **hacer**
7: Cuando $BROADCAST(ident_i)$ llega al vecino j:
8: $T_j = j \times \tau$
9: j espera $Timer_j = T_j$ segundos a su momento adecuado para enviar reconocimientos
10: Cuando $Timer_j$ ha expirado:
11: j transmite $ACK(ident_j)$ hacia el nodo i
12: Cuando $ACK(ident_j)$ llega al nodo i:
13: i guarda $ident_j$ en su tabla de vecinos.
14: **fin para**
15: **fin para**

3.3 Elección del simulador

Existen muchos simuladores disponibles que podrían usarse para evaluar las prestaciones de protocolos para redes ad hoc. Entre ellos podríamos nombrar: OPNET, OMNET++, NS-2, NS-3, Netlogo, DARS, QualNet, INET, INETMANET, GloMoSim, MobiWan, WSNet, ShoX, GrubiX, Mixim, Truetime, BonnMotion, SIMUTools, WNS3, SSF-based, MobiSim, Opnet Modeller, JIST/SWANS, JSIM, NETSIM5.0, NCTUns.

En nuestro caso, para realizar las simulaciones hemos elegido el simulador Castalia versión 3.2 [88], que básicamente permite simular WSN y BAN (*Body Area Networks*). Comprobamos que Castalia 3.2 reúne los requisitos para la evaluación de protocolos de descubrimiento de vecinos en redes inalámbricas ad hoc estáticas. Además, este simulador tiene utilidades para simular MANETs. Recientemente, se ha difundido Castalia 3.3.

3.4 Protocolos de referencia

Con el objetivo de comparar, elegimos dos protocolos de la literatura.

El Hello [22] presenta el tiempo ranurado de tamaño ω, conocidos como *rounds*. En cada *round* todos los nodos eligen de forma aleatoria un tiempo t_i ($0 \leq t_i \leq \omega - \tau$). Cada nodo difunde un paquete iniciando en t_i durante una duración τ y escucha durante el resto de la ranura. Cuando ocurre una transmisión con éxito, decimos que un vecino ha sido descubierto. Un número de *rounds* tras el cuál el protocolo finaliza, tiene que ser elegido. Resaltar que el protocolo es *one-way*, esto es, no incluye ningún mecanismo de *feedback*.

El PRR [21] también presenta el tiempo ranurado (*rounds*) de tamaño τ. En un *round* los nodos eligen transmitir con probabilidad $\frac{1}{N}$ o escuchar con probabilidad $1 - \frac{1}{N}$. De nuevo, cuando tiene lugar una transmisión con éxito, un descubrimiento de vecino ocurre. El número de *rounds* es un parámetro que debe ser fijado cuidadosamente. PRR también es un protocolo *one-way*, esto es, no incluye ningún mecanismo de *feedback*.

3.5 Simulación y resultados

3.5.1 Escenario de simulación

En la sección 3.5.2 comparamos las prestaciones de las propuestas con la de dos protocolos de referencia: Hello y PRR. Para obtener resultados de simulación se ha usado Castalia 3.2 [88].

Se han establecido los mismos parámetros para las propuestas y para los protocolos de referencia. Para ello se han fijado diferentes tamaños de red (escalabilidad), diferentes modelos de colisión (colisiones). Se fija un número de *rounds* específicos para los protocolos de referencia dado que tras un número determinado de *rounds* esos protocolos finalizan. También fijamos una duración de *round* para Hello a $\omega = N \times \tau$, con un τ a 0.07 segundos. Sin embargo, para Hello se establece una duración de $0.5N$ *rounds*. Para PRR se establece una duración de $10N$ *rounds*. Ambos protocolos de referencia son *one-way* por tanto no se ha implementado ningún mecanismo de *feedback*.

En cuanto al área de despliegue, se ha fijado con una extensión de 10mx10m en un modo *one-hop* y los nodos se han desplegado de acuerdo con mallas $M \times M$.

Tabla 3.2: Parámetros de simulación.

Parámetro	Valor
Static	True
Modelo de radio	CC2420
Modelo de colisión	2
Potencia de transmisión	-5 dBm
Tasa de paquetes	5 packet/s
Tamaño de paquete	2500 bytes
Duración del *round* para Hello	$\omega = N \times \tau$
τ	0.07s
Tamaño *one-hop*	10mx10m
Despliegue	Malla MxM
Número de *rounds* para PRR	$10 \times N$
Número de *rounds* para Hello	$0.5 \times N$

Como se indicó, se han manejado las colisiones haciendo uso del parámetro disponible en Castalia 3.2, esto es, el *collisionModel*. Este parámetro se puede fijar a los siguientes valores: 0 (sin colisiones), 1 (modelo simplista para colisiones), o 2 (modelo de interferencia aditiva). Para la mayoría de las gráficas, se decidió usar el modelo de colisiones más realista, es decir, el modelo de interferencia aditivo.

Los protocolos de descubrimiento de vecinos tienen principalmente como objetivo descubrir todos los vecinos proporcionando un bajo consumo temporal. Por ello, las simulaciones fueron llevadas a cabo para obtener el *tiempo de descubrimiento de vecinos*, y el *número de vecinos descubiertos*. Además, dado que los nodos tienen baterías que limitan el tiempo de vida del dispositivo, se obtuvo el *consumo energético*. Además, consideramos interesante obtener el *throughput*, *descubrimientos por paquetes enviados* y *paquetes recibidos por paquetes enviados*.

Usamos el modelo de radio *ZigBee*, esto es, *CC2420*. Para una potencia de transmisión de $-5dBm$, E_{tx}, la energía consumida por un solo nodo cuando transmite por segundo es 0.0522J. E_l, la energía consumida por un solo nodo cuando escucha por segundo, es 0.068J.

Los parámetros usados para obtener los resultados de simulación se pueden encontrar en la Tabla 3.2.

3.5.2 Resultados

A continuación, se procede a presentar y discutir los resultados obtenidos a través de simluación en un entorno *one-hop*, y comparar los resultados de las dos propuestas con los de Hello y PRR.

Tiempo de descubrimiento de vecinos

Esta métrica se refiere al tiempo que tarda un protocolo en finalizar. Para la propuesta Leader-based, como se muestra en la sección 3.2.2, el tiempo de descubrimiento de vecinos sigue una tendencia lineal $O(N)$. Como resultado óptimo, el tiempo de descubrimiento de vecinos está cercano a $7s$ para redes compuestas de 100 nodos.

En relación con la propuesta TDMA-based el tiempo de descubrimiento sigue una tendencia cuadrática $O(N^2)$. Crece cuando el número de nodos se incrementa y presenta peores resultados que la propuesta Leader-based.

La Figura 3.4 muestra los resultados habiendo fijado el modelo de interferencia aditiva para colisiones, esto es, el modelo de colisiones más realista. La métrica es el tiempo de descubrimiento de vecinos. Se puede concluir que el Leader-based supera al PRR y el Hello, y se puede probar que se pueden obtener resultados similares para los otros dos modelos de colisión. A continuación, PRR es mejor que Hello fijando la duración de PRR a 10N *rounds*. Hello con 0.5N *rounds* (tamaño de ranura N) presenta mejores resultados que la propuesta TDMA-based. El tiempo de descubrimiento de vecinos para todos los protocolos presenta una tendencia creciente. Esto es debido a que a medida que el número de nodos crece, más tiempo se requiere para descubrir los vecinos, es decir, el tiempo de descubrimiento depende de N.

Además, se demuestra que los resultados obtenidos a través de simulaciones coinciden con los resultados analíticos presentados en la sección 3.2.2 y 3.2.3, es decir, ecuaciones 3.1 y 3.8.

Número de vecinos descubiertos

En esta sección, se presentan y discuten tres gráficas, comparando los cuatro protocolos con respecto al número de vecinos descubiertos. La Figura 3.5 muestra los resultados usando el modelo de colisión 0, es decir, sin colisiones. En cuanto a los resultados para modelos de colisión 1 y 2, se muestran en la Figura 3.6 y Figura 3.7.

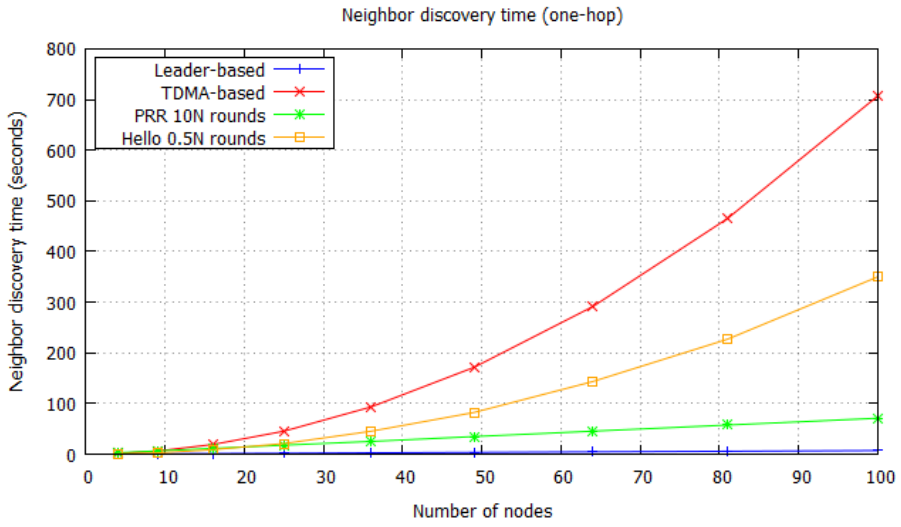

Figura 3.4: Tiempo de descubrimiento de vecinos, comparación (collisionModel 2).

La Figura 3.5 nos permite concluir que todos los protocolos logran descubrir todos los vecinos, es decir, presentan un comportamiento ideal para el modelo sin colisiones.

En cuanto a la Figura 3.6, muestra los resultados para el modelo de colisión 1, esto es, el modelo simplista para colisiones. Ambas propuestas logran descubrir los N-1 vecinos, superando a ambos protocolos de referencia. Hello 0.5N *rounds* no logra descubrir todos los vecinos para número de nodos por debajo de 50 y PRR 10N *rounds* no logra descubrir todos los vecinos.

La Figura 3.7 muestra que, fijando el modelo de interferencia aditiva para colisiones, se obtienen similares resultados que los de la Figura 3.6. Las propuestas logran descubrir los N-1 vecinos, superando a ambos protocolos de referencia. Hello 0.5N *rounds* no logra descubrir todos los vecinos para un número de nodos por debajo de 40, y PRR 10N *rounds* no logra descubrir todos los vecinos.

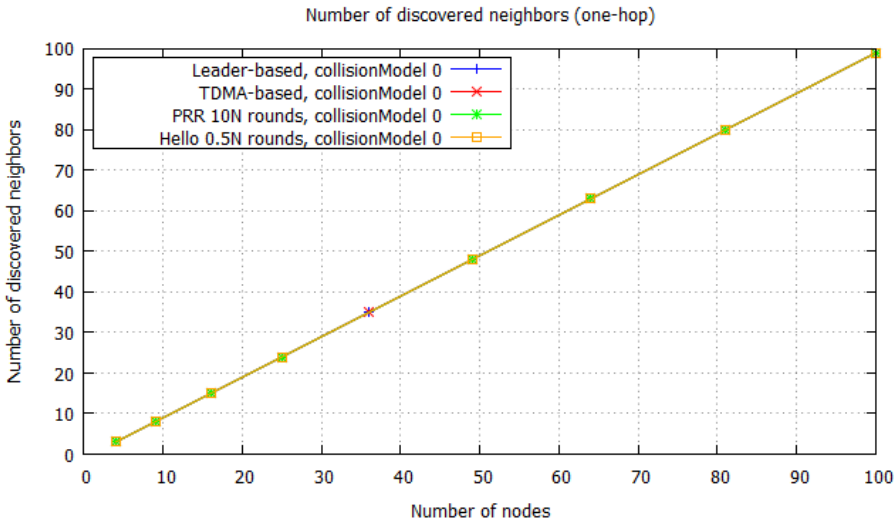

Figura 3.5: Número de vecinos descubiertos, comparación (collisionModel 0).

Consumo energético

En cuanto al consumo energético, según la Figura 3.8, para todos los protocolos bajo prueba el consumo energético aumenta a medida que el número de nodos crece, de forma similar a la Figura 3.4. TDMA-based claramente presenta los peores resultados. Esto es debido a que se requiere más tiempo por tanto el consumo energético es peor. Se obtienen resultados similares para las otras soluciones. Para redes formadas por 100 nodos, el PRR 10N *rounds* es el mejor, consumiendo 4.824J por nodo, luego el Leader-based consume 6.92J por nodo. Finalmente el Hello 0.5N *rounds* consume 23.734J por nodo. Además, los resultados de simulación coinciden con los resultados analíticos. Estos resultados se muestran para el Leader-based en la ecuación 3.2 y para el TDMA-based en la ecuación 3.9.

Throughput

En cuanto al *Throughput*, mostrado en la Figura 3.9, ambas propuestas claramente logran los mejores resultados. Comienzan aproximadamente en 28000 byte/s para 4 nodos y convergen a 35360 byte/s para 100 nodos. Además, los paquetes recibidos por segundo es el máximo. El Hello 0.5N *rounds* supera al PRR 10N *rounds*, y ambos siguen una tendencia decreciente. Esto se

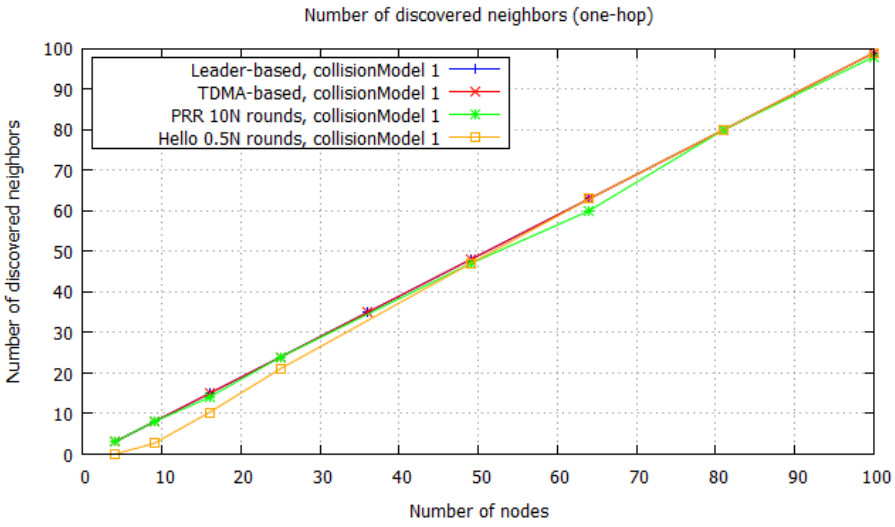

Figura 3.6: Número de vecinos descubiertos, comparación (collisionModel 1).

debe a que a medida que el número de nodos crece más colisiones aparecen, menor número de paquetes son recibidos. Por tanto el consumo temporal se incrementa y el *throughput* decrece, Hello 0.5N *rounds* inicia con 4440 byte/s para 4 nodos a 36 byte/s para 100 nodos. PRR 10N *rounds* inicia con 2200 byte/s para 4 nodos a 38.8 byte/s para 100 nodos. De nuevo, los resultados de simulación coinciden con los valores teóricos. Estos resultados se muestran para el Leader-based en la ecuación 3.3 y para el TDMA-based en la ecuación 3.10.

Número de descubrimientos por paquetes enviados

Como se muestra en la Figura 3.10, el Leader-based presenta los mejores resultados con respecto a vecinos descubiertos por total de paquetes enviados. Se trata de un resultado óptimo, dado que el consumo de tiempo es menor, enviando menos paquetes, por tanto el *ratio* es mayor para el mismo número de descubrimientos. Comienza con 0.6 para 4 nodos y converge a 0.98 para 100 nodos. A continuación, PRR 10N *rounds* es mejor que las otras soluciones para número de nodos por encima de 16, seguido por Hello 0.5N *rounds*, y finalmente el TDMA-based es el peor. Resaltar que este orden es el mismo que para el tiempo de descubrimiento de vecinos en la Figura 3.4. Hello 0.5N

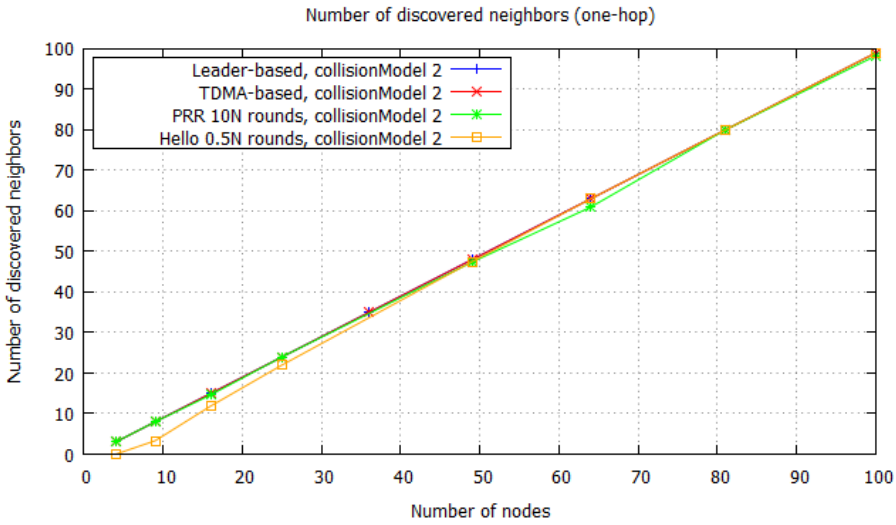

Figura 3.7: Número de vecinos descubiertos, comparación (collisionModel 2).

rounds y TDMA-based siguen una tendencia decreciente a medida que el número de nodos crece. Los resultados de simulación coinciden con los resultados analíticos en las ecuaciones 3.4 y 3.11.

Paquetes recibidos por paquetes enviados

Según la Figura 3.11, la propuesta TDMA-based supera a las otras soluciones, proporcionando un *ratio* de paquetes recibidos por paquetes enviados de 1. Este es el valor óptimo, mientras que la propuesta Leader-based llega a este valor para número de nodos por encima de 10. A continuación, PRR 10N *rounds* supera al Hello 0.5N *rounds*, que es el peor. Ambos protocolos de referencia siguen una tendencia decreciente. De nuevo, los resultados de simulación coinciden con los resultados analíticos en las ecuaciones 3.5 y 3.12.

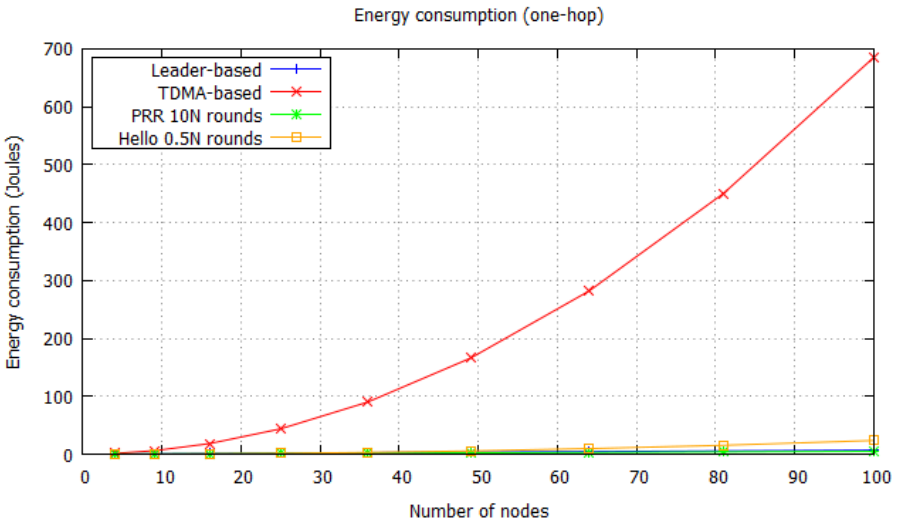

Figura 3.8: Consumo energético promedio por nodo, comparación (collisionModel 2).

3.6 Protocolos de referencia con feedback

En esta sección se presentan y analizan los resultados obtenidos comparando las dos propuestas con los protocolos de referencia incluyendo un *feedback*.

Para Hello y PRR fijamos un número de *rounds* a $3N$ y $6N$, e incluimos un mecanismo de *feedback*, fijando un tiempo de $ACKs$ de $N \cdot \tau$.

3.6.1 Configuración de la simulación

En la siguiente sección procedemos a realizar las simulaciones de ambas propuestas determinísticas en comparación con los dos protocolos de referencia, esto es, Hello y PRR.

El escenario de simulación es el mismo para todos los protocolos a simular. Para obtener los resultados de simulación, hemos variado diferentes modelos de colisión, y diferente número de nodos.

Además, para PRR y Hello fijamos el número de *rounds*, dado que tras un número finito de *rounds* esos algoritmos finalizan, a $3 \cdot N$ y $6 \cdot N$ *rounds*. Para el protocolo Hello fijamos un tamaño de *round* $\omega = N \cdot \tau$, donde N es el número de nodos de la red y τ es el tiempo que un nodo está transmitiendo. El valor de

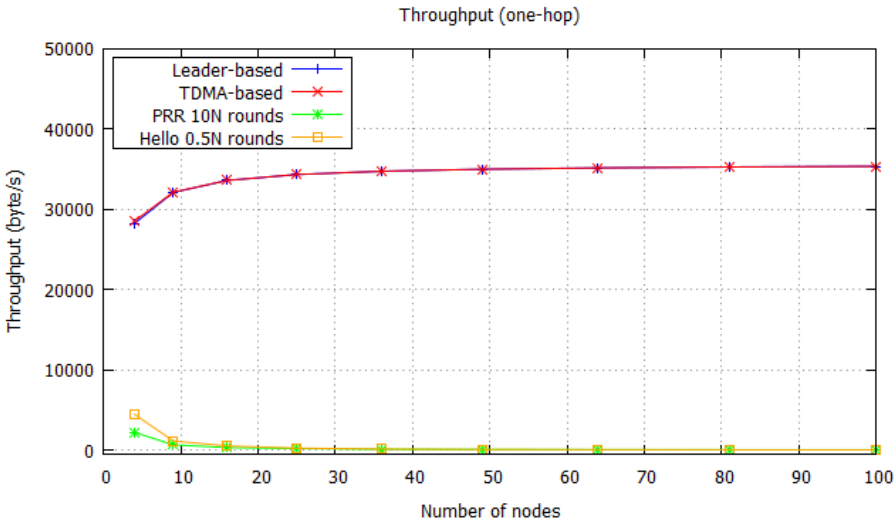

Figura 3.9: *Throughput* por nodo, comparación (collisionModel 2).

τ lo fijamos a 0.07 segundos para todos los protocolos que simulamos. Además, para Hello y PRR incluimos un mecanismo de *feedback* para el envío de los ACK, durante un tiempo $N \cdot \tau$.

En cuanto al area de despliegue, la fijamos a 10mx10m en un escenario *one-hop*, esto es, todos los nodos están en el rango de transmisión de todos los demás. Los N nodos son desplegados en mallas MxM.

En cada Figura, hemos usado un modelo de colisión distinto, haciendo uso del parámetro *collisionModel* que nos proporciona Castalia 3.2. Este parámetro puede tomar los valores 0 (sin colisiones), 1 (modelo simplístico para colisiones), o 2 (modelo de interferencia aditiva).

El principal objetivo de los protocolos de descubrimiento de vecinos es descubrir todos los vecinos, o casi todos, en un tiempo reducido. Por lo tanto, las simulaciones se centran en obtener 2 métricas, esto es, el Tiempo de Descubrimiento de Vecinos y el Número de Vecinos Descubiertos.

Proponiendo 2 protocolos determinísticos, nuestro objetivo es descubrir todos los vecinos con probabilidad 1. Esto es debido a que los protocolos aleatorios logran el descubrimiento con alta probabilidad (pero distinta de 1). Otro

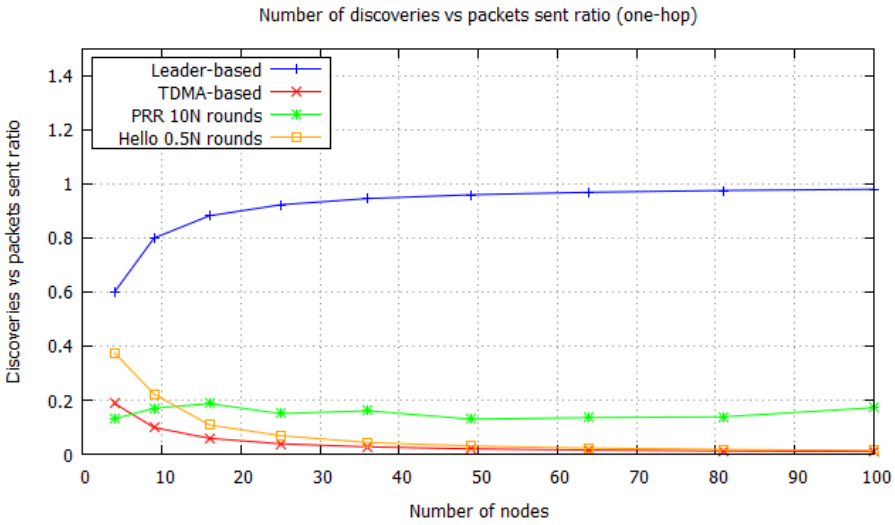

Figura 3.10: Descubrimientos por total de paquetes enviados *ratio*, comparación (collisionModel 2).

objetivo es mejorar el tiempo de descubrimiento, evitando colisiones de una forma proactiva.

En la Tabla 3.3 se muestran los parámetros de simulación.

3.6.2 Resultados

Tiempo de descubrimiento de vecinos

En primer lugar, el tiempo de descubrimiento de vecinos se ha obtenido para el Leader-based y concluimos que sigue una tendencia creciente lineal $O(N)$ a medida que el número de nodos crece. Para redes compuestas por 100 nodos el tiempo de descubrimiento es aproximadamente 7 segundos, un resultado óptimo.

Tanto los resultados de simulación como teóricos obtenidos fijando el *collisionModel* a 2 (el más realista) para el Leader-based son comparados presentando ambos resultados idénticos. Se puede probar que los resultados son similares para los modelos de colisión 0 y 1.

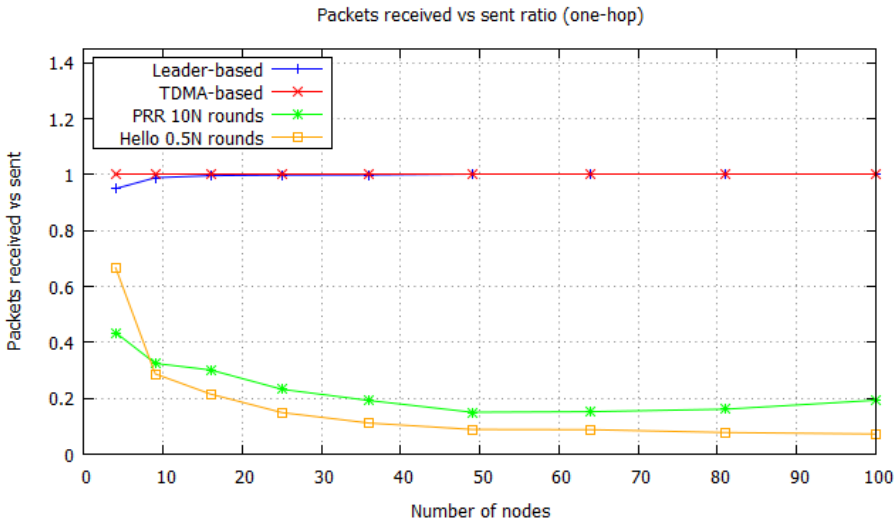

Figura 3.11: Paquetes recibidos por paquetes enviados ratio, comparación (collisionModel 2).

En cuanto al TDMA-based, sigue una tendencia creciente $O(N^2)$ con el número de nodos. De nuevo, tanto los resultados teóricos como de simulación obtenidos fijando el *collisionModel* a 2 para el TDMA-based son comparados, presentando ambos resultados idénticos. Se puede probar que los resultados son similares para los modelos de colisión 0 y 1.

En conclusión, los resultados de simulación obtenidos coinciden con los resultados teóricos obtenidos mediante el modelo analítico presentado en secciones previas. La ecuación 3.1 para el Leader-based y la ecuación 3.8 para el TDMA-based.

Como se muestra, la solución TDMA-based presenta peores resultados en cuanto a tiempo de descubrimiento que la solución Leader-based. Sin embargo, el Leader-based puede funcionar de forma adecuada sólo en escenarios *one-hop*.

A continuación, se presenta en la Figura 3.12 una comparación del tiempo de descubrimiento de vecinos de las propuestas y protocolos de referencia (PRR, Hello, Leader-based, TDMA-based).

En cuanto al nivel de prestaciones logrado, la Figura 3.12 muestra los resultados para *collisionModel* 2 (el más realista). Se concluye que las propuestas superan

Tabla 3.3: Parámetros de simulación.

Parámetro	Valor
Static	True
Modelo de radio	CC2420
Potencia de transmisión	-5 dBm
Tasa de paquetes	5 packet/s
Tamaño de paquete	2500 bytes
Duración del *round* para Hello	$\omega = N \times \tau$
Duración ranura de *feedback* para Hello y PRR	$N \cdot \tau$
τ	0.07s
Tamaño *one-hop*	10mx10m
Despliegue	Malla MxM
Número de *rounds* para PRR	$3 \times N$ y $6 \times N$
Número de *rounds* para Hello	$3 \times N$ y $6 \times N$

al PRR y al Hello, siendo los resultados similares para los otros dos modelos de colisión. También concluimos que PRR es más rápido que el Hello fijando el tamaño de ranura a N tanto para $3Nrounds$ como $6Nrounds$. PRR con 6N *rounds* y Hello con 3N *rounds* (tamaño de ranura N) presentan resultados similares. Analíticamente, concluimos que el Hello presenta un tiempo de descubrimiento de $O(6N^2)$ para 3N *rounds*, y $O(12N^2)$ para 6N *rounds*. En cuanto al PRR, los tiempos de descubrimiento son $O(3N^2)$ para $3Nrounds$, y $O(6N^2)$ para $6Nrounds$.

Número de vecinos descubiertos

En cuanto al número de vecinos descubiertos para todos los protocolos, se sigue de nuevo un procedimiento de simulación similar al descrito anteriormente. A continuación, se muestran 3 figuras, que comparan el número de vecinos descubiertos para todos los protocolos. En concreto, los resultados para el modelo sin colisiones se presentan en la Figura 3.13, y las gráficas para los otros dos modelos de colisión se presentan en la Figura 3.14 y Figura 3.15.

La Figura 3.13 muestra que, en casi todos los protocolos, todos los nodos descubren sus N-1 vecinos en el escenario *one-hop*, excepto el PRR con 3N *rounds*, que es ligeramente más bajo.

Considerando el modelo de colisión 1 mostrado en la Figura 3.14, el Leader-based y TDMA-based superan a las otras soluciones. También, se concluye que PRR con 6N *rounds* es mejor que PRR con 3N *rounds*. De forma similar,

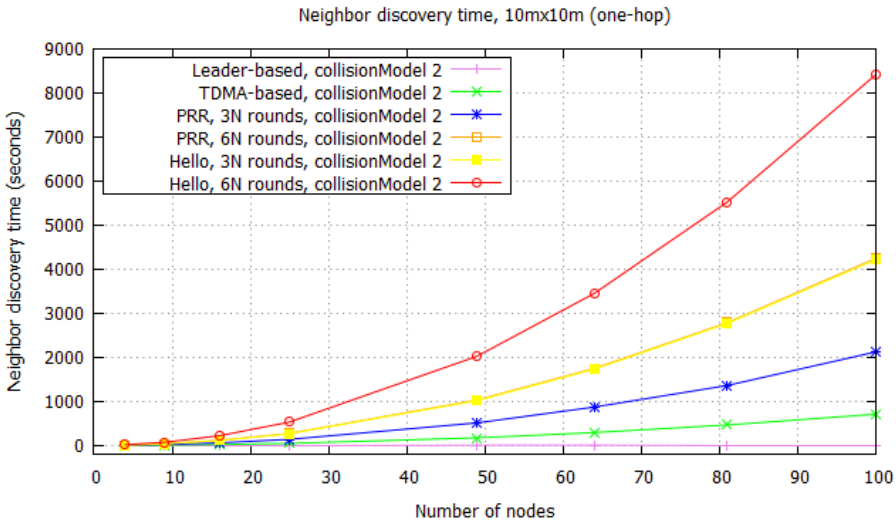

Figura 3.12: Tiempo de descubrimiento, comparación.

Hello con 6N *rounds* es ligeramente mejor que el Hello con 3N *rounds*. Resaltar que PRR 6N *rounds* es mejor que ambas variantes de Hello, mientras que PRR 3N *rounds* es peor que ambas.

De acuerdo con la Figura 3.15, donde el modelo de colisión se fijo a 2, de nuevo, el Leader-based y TDMA-based superan a las otras soluciones. PRR 6N *rounds* es también mejor que PRR 3N *rounds*. A continuación, Hello 6N *rounds* supera al PRR 3N *rounds*, y Hello 3N *rounds* es el peor. Recordar que PRR con 6N *rounds*, y Hello con 3N *rounds* (y tamaño de ranura N), tienen un resultado similar en cuanto a tiempo de descubrimiento, como podemos observar en la Figura 3.12.

3.7 Conclusiones

En este capítulo se ha realizado un estudio del problema de descubrimiento de vecinos en entornos estáticos *one-hop* inalámbricos ad hoc teniendo en cuenta la existencia de colisiones. Hello y PRR han sido elegidos para ser usados como referencia, y se han propuesto dos protocolos determinísticos y simulados con Castalia 3.2 para compararlos. Las propuestas hacen uso de un mecanismo

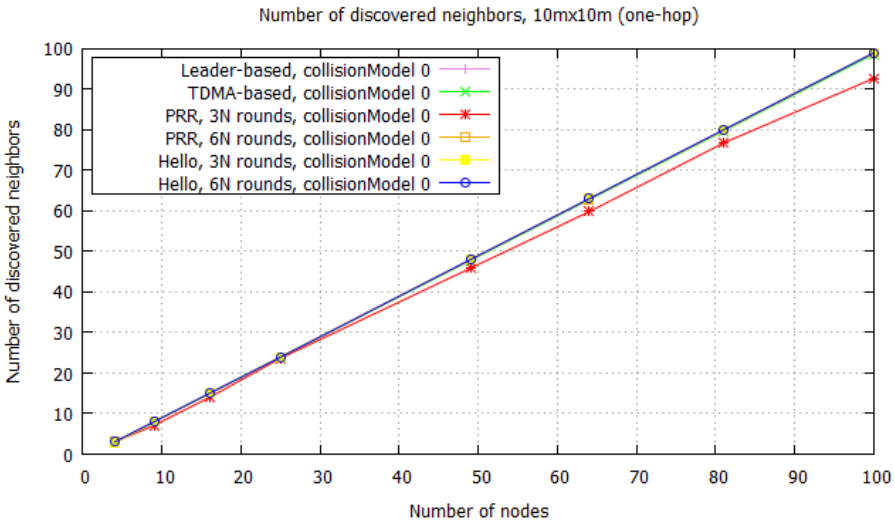

Figura 3.13: Número de vecinos descubiertos, comparación (collisionModel 0).

de *feedback* para mejorar su funcionamiento. Los protocolos de referencia son *one-way*. Para su comparación se han usado seis métricas.

También se llevó a cabo un estudio analítico para ambas propuestas. Las métricas usadas son el tiempo de descubrimiento de vecinos, el consumo energético, el *throughput*, el *ratio* de descubrimientos por paquetes enviados, y el *ratio* de paquetes recibidos por enviados.

Se ha demostrado que la propuesta Leader-based presenta un comportamiento óptimo con respecto a resultados temporales ($O(N)$). Supera al PRR con 10N *rounds*, que a su vez supera al Hello con 0.5N *rounds*, y el TDMA-based es el protocolo más lento ($O(N^2)$). Por tanto, el Leader-based logra una mejora en un factor de N con respecto al consumo temporal sobre el TDMA-based.

También fijamos diferentes modelos de colisión, tenemos como objetivo obtener el número de vecinos descubiertos. Los resultados permiten concluir que ambas propuestas también logran resultados óptimos y logran superar a las otras soluciones.

En cuanto al consumo energético, PRR 10N *rounds* es el mejor, seguido por el Leader-based, que a su vez supera al Hello 0.5N *rounds*, y finalmente el TDMA-based es el peor. En relación con el *throughput*, ambas propuestas claramente

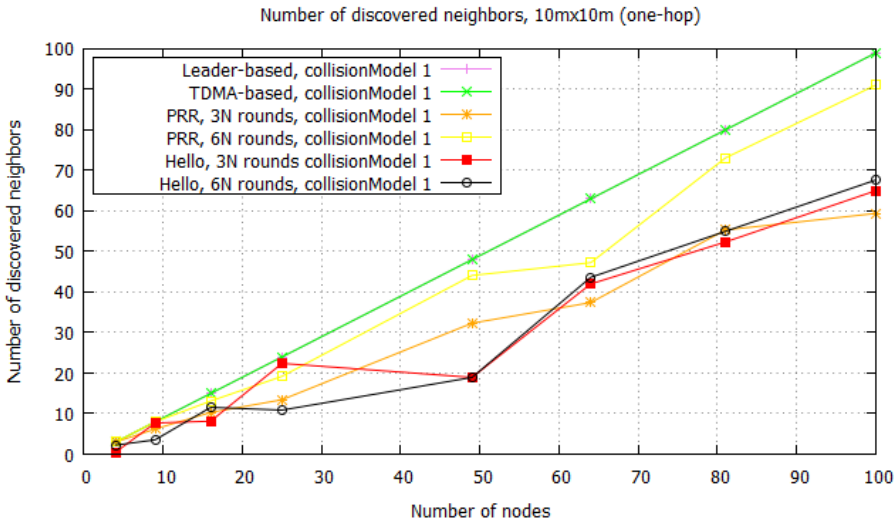

Figura 3.14: Número de vecinos descubiertos, comparación (collisionModel 1).

superan a los protocolos de referencia. La propuesta Leader-based presenta los mejores resultados en cuanto a los descubrimientos por paquetes enviados. PRR 10N *rounds* es mejor que el Hello 0.5N *rounds* y el TDMA-based es el peor. Sin embargo, TDMA-based presenta los mejores resultados con respecto a los paquetes recibidos por enviados, seguidos por el Leader-based. Luego el PRR 10N *rounds* y finalmente Hello 0.5N *rounds* es el peor.

La solución Leader-based puede sólo funcionar adecuadamente en un entorno *one-hop*, aunque logra comportamiento óptimo en escenarios estáticos. En cuanto al TDMA-based, también es adecuado para entornos *multi-hop*, pero en este caso sus prestaciones se degradarían. Ambas propuestas logran descubrir todos los vecinos con probabilidad 1, aunque se basan en una planificación en la transmisión para su funcionamiento, y se incluye un mecanismo de *feedback*.

Hay muchos protocolos para la elección de líder en la literatura y que podrían ser usados previo al descubrimiento de vecinos de la solución Leader-based.

Como futuro trabajo, se podría mejorar el consumo energético proponiendo protocolos determinísticos en entornos *multi-hop* con recursos energéticos limitados. Además, se podría investigar cómo se comportan los protocolos en entornos interiores [89, 90, 91].

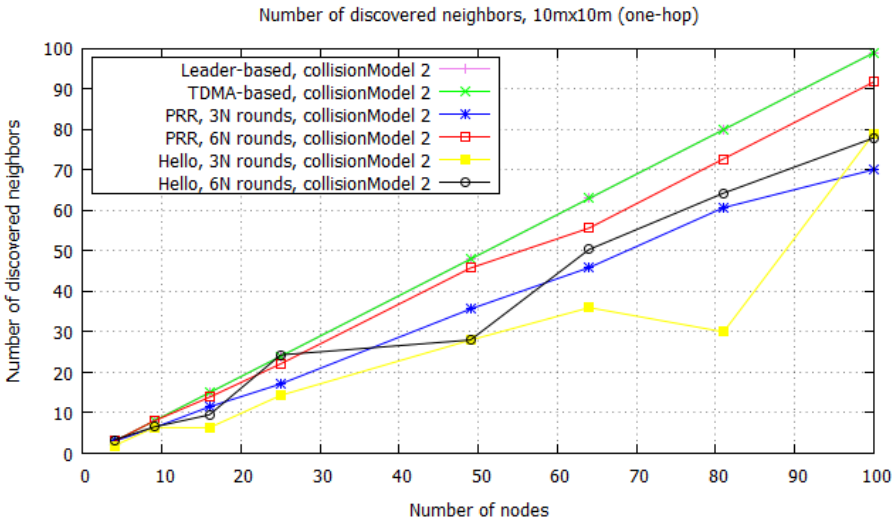

Figura 3.15: Número de vecinos descubiertos, comparación (collisionModel 2).

Capítulo 4

Protocolos de descubrimiento de vecinos aleatorios basados en detección de colisiones

En este capítulo, presentamos 2 protocolos de descubrimiento de vecinos aleatorios basados en detección de colisiones en escenarios estáticos one-hop y multi-hop. A estos protocolos los llamamos CDH y CDPRR. Se ha usado Castalia 3.2 para comparar nuestras propuestas con dos protocolos elegidos de la literatura y usados como referencia. Estos protocolos de referencia son PRR y Hello. Para su simulación, elegimos 5 métricas: el Tiempo de descubrimiento de vecinos, el Número de vecinos descubiertos, el Consumo energético. Además evaluamos el Throughput y el ratio Número de vecinos descubiertos por paquetes enviados. Los resultados de simulación nos permiten concluir que nuestras 2 propuestas presentan mejores prestaciones que ambos Hello y PRR en relación con las 5 métricas. Se han evaluado en la presencia de colisiones. Esta mejora se produce tanto para escenarios one-hop como multi-hop. En concreto, CDPRR presenta mejores resultados de tiempos, consumo energético, y ratio de número de vecinos descubiertos por paquetes enviados. Como novedad, en relación con los protocolos de referencia, ambas propuestas permiten descubrir todos los vecinos con probabilidad 1, y logran funcionar bajo premisas más realistas. Ambos protocolos conocen cuándo terminar el descubrimiento, lo que se produce cuando todos los nodos han sido descubiertos. CDH no necesita conocer el número de nodos de la red. Además, este capítulo incluye una comparación cualitativa de soluciones existentes y nuestras propuestas.

4.1 Introducción

Este capítulo se centra en el descubrimiento de vecinos en redes *multi-hop* estáticas inalámbricas ad hoc. Se proponen 2 protocolos aleatorios que funcionan de forma reactiva y no tienen una planificación predeterminada. Éstos solucionan el descubrimiento en la presencia de colisiones, que tienen lugar cuando 2 o más nodos transmiten simultáneamente. Además aprovechan las ventajas de la detección de colisiones.

Estos protocolos aleatorios son CDH y CDPRR.

Nuestras propuestas se comparan con 2 protocolos de la literatura, usados como referencia, Hello [22] y PRR [21]. Para compararlos los 4 protocolos se han implementado en el simulador Castalia 3.2 [88]. Se han usado 5 métricas: el Tiempo de descubrimiento de vecinos, el Número de vecinos descubiertos, el consumo energético, el *throughput* y el *ratio* número de vecinos descubiertos por paquetes enviados.

El problema principal de los protocolos de referencia es que no conocen cuando terminar el descubrimiento y los vecinos son descubiertos con alta probabilidad (pero no con probabilidad 1). Además, PRR necesita conocer el número de nodos de la red. Por tanto, nuestro objetivo es mejorar esos protocolos, mediante la detección de colisiones y detección de la terminación. Se debe permitir a los protocolos desconocer el número de nodos, y aún así obtener mejores prestaciones.

Este trabajo es diferente de otros trabajos recientes en la literatura como [48] que usa un filtro *Kalman* como modelo de predicción, combina mensajes Hello y predicción de movilidad de nodo. Usa el mecanismo *ALOHA-like* [26] con probabilidad de transmisión 0.5. En [52] se usa información previa de radar para acelerar la velocidad, integra radar y comunicación. El protocolo es *3-way* pero el *feedback* puede colisionar, y usa transmisión direccional y recepción direccional. En [51] el protocolo combina encaminamiento y descubrimiento de vecinos. No requiere conocimiento a priori de los parámetros de la red (tal como tamaño, topología o movilidad). En [41] se presenta un protocolo *cross-layer* que realiza el descubrimiento de vecinos en la capa MAC. Se envían mensajes Hello periódicamente tras un *backoff* aleatorio de una forma *TDMA*. Funciona con la ayuda de *clustering* hexagonal y GPS para actualizar la última información de vecinos. El protocolo [54] usa antenas direccionales, y modela el descubrimiento de vecinos como un autómata de aprendizaje de estado finito. Ese protocolo opera de una manera *ALOHA-like*, *2-way handshake* donde los nodos transmiten o reciben con igual probabilidad. En [57] se in-

tegra radar y comunicación, usando antenas direccionales. Hace uso de un *handshake two-way* para cada dirección enviando mensajes Hello. En [45] se presenta descubrimiento de vecinos con reconocimiento social bajo un *framework* de descubrimiento pasivo. Se difunde una señal radio *wake-up* antes de la difusión de mensajes Hello para cambiar del modo inactivo al modo activo. Los mensajes Hello son integrados con información social, y se realiza descubrimiento de vecinos en la capa MAC. En [46] las balizas son separadas de los *slots* activos. Una difusión de balizas periódica puede ser ajustada dinámicamente para acelerar el descubrimiento. Hay disponible un *wakeup* proactivo. La mayoría de estos protocolos pueden ser usados en entornos móviles.

En cuanto a las novedades de las propuestas presentadas en este capítulo en comparación con trabajos recientes destacan: no hay radar ni antena direccional. Son propuestas *2-way* usando transceptores de radio omnidireccionales. No se usa planificación, y el descubrimiento de vecinos se realiza en la capa de red. No se requiere conocimiento a priori de los parámetros de la red en CDH pero es necesario en CDPRR. Las propuestas han sido diseñadas para ser usadas en entornos estáticos.

Las principales contribuciones de este capítulo son: (i) CDH, una propuesta aleatoria basada en la detección de colisiones y en el protocolo Hello. Presenta un tamaño de ranura fijo y logra descubrir todos los vecinos con probabilidad 1. Logra terminar cuando todos los vecinos han sido descubiertos. Sigue premisas más realistas, tales como no requerir conocer el número de nodos de la red, y los nodos pueden transmitir en diferentes instantes de tiempo. Además, es adecuado para su uso tanto en entornos *one-hop* como *multi-hop*, (ii) CDPRR, una propuesta aleatoria basada en detección de colisiones y en el protocolo PRR. Presenta una probabilidad de transmisión $\frac{1}{N}$ fija durante todo el proceso de descubrimiento de vecinos. Logra el descubrimiento de todos los vecinos con probabilidad 1, terminando cuando todos los vecinos han sido descubiertos. Sin embargo, requiere conocer el número de nodos de la red, y es adecuado para su uso tanto en entornos *one-hop* como *multi-hop*, (iii) Una comparación cualitativa de los protocolos del estado del arte y de las dos propuestas, (iv) Implementación de ambas propuestas y los protocolos de referencia en el simulador Castalia 3.2 [88]. Se obtienen resultados en cuanto a tiempos, número de vecinos descubiertos, consumo energético, *throughput*, y el *ratio* número de vecinos descubiertos por paquetes enviados. Además, se concluye que las propuestas son más rápidas y más eficientes en energía que las soluciones existentes.

En la Tabla 2.4, se muestra una comparación cualitativa de protocolos recientes. Según la Tabla 2.4, la mayoría de protocolos se pueden usar en redes

Tabla 4.1: Comparación cualitativa de protocolos de referencia y las propuestas.

	[22]	[21]	CDH	CDPRR
Red estática	✓	✓	✓	✓
Red móvil				
Aleatorio	✓	✓	✓	✓
Tiempo ranurado	✓	✓	✓	✓
N conocido		✓		✓
Requiere sincronización				
Requiere planificación en la transmisión				
One-hop	✓	✓	✓	✓
Multi-hop	✓	✓	✓	✓
Sleep disponible				
Colisiones pierden la transmisión	✓	✓		
Detección de pérdida de paquetes				
Detección de colisión	✓		✓	✓
Detección de terminación			✓	✓
Inicia transmisión en diferentes instantes de tiempo	✓		✓	
Descubre todos los vecinos			✓	✓
Basado en *handshake*			✓	✓
Requiere gran número de ranuras		✓		
Requiere N grande para operación adecuada		✓		
Protegido contra pérdida de paquetes			✓	✓

móviles (MANETs), el tiempo está ranurado, se usan esquemas aleatorios, son asíncronos. Se pueden usar en entornos *one-hop*, y todos tratan con colisiones. Además, ninguno de ellos usa *full-duplex*, ni *pre-handshaking* ni *group testing*. En cuanto a las propuestas, según se muestra en la Tabla 4.1, sólo se pueden usar en entornos estáticos, el tiempo está ranurado, son aleatorios y asíncronos. Son adecuados para su uso tanto en *one-hop* como *multi-hop*, y pueden tratar con colisiones. En CDH, el número de nodos de la red puede ser desconocido.

Las dos propuestas aleatorias difieren de soluciones previas dado que el objetivo es descubrir todos los vecinos con probabilidad 1, incluso en redes densas. Por tanto, se soluciona el problema de protocolos aleatorios existentes que no descubren todos los vecinos con probabilidad 1. Las propuestas logran reducir el consumo temporal y consumo energético, incrementa el *throughput* y el *ratio* número de vecinos descubiertos por paquetes enviados. Además, son adecuadas para usar en entornos estáticos *multi-hop*.

El tiempo está ranurado en las 2 propuestas para mejorar la fiabilidad e incrementar la velocidad de descubrimiento.

4.2 Visión general del sistema

En la Figura 4.1 se muestra el funcionamiento del protocolo CDH. Como vemos, el protocolo es basado en *handshake* y su funcionamiento se basa en el intercambio de paquetes *BROADCAST* entre los nodos durante una ranura de tiempo (*round*). Según el ejemplo de operación mostrado en la Figura 4.1, el nodo i envía un paquete *BROADCAST* que contiene su identificador. Su destino son los nodos en su rango de transmisión, y lo emite en un tiempo elegido de forma aleatoria t_i. El nodo j realizará la misma operación. Para ello, envía un paquete *BROADCAST* que contiene su identificador en un tiempo elegido de forma aleatoria t_j. Al final del *round* (ω), ambos nodos proceden a enviar una serie de paquetes de *feedback*, como se explicará más adelante. Esto indica qué mensajes transmitieron con éxito.

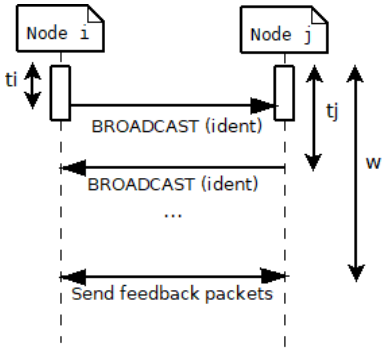

Figura 4.1: Sistema prototocolo CDH

En el CDPRR, cuyo funcionamiento se muestra en la Figura 4.2, el descubrimiento también se logra intercambiando paquetes *BROADCAST*, y el protocolo es basado en *handshake*. Según el protocolo, el nodo i envía un paquete *BROADCAST* que contiene su identificador si un estado elegido de forma aleatoria determina que está transmitiendo. De lo contrario, el nodo i permanece escuchando. Al final del *round* de tamaño τ, los nodos que escuchaban envían un paquete de *feedback* si el paquete *BROADCAST* de un nodo fue bien recibido. La Figura 4.2 también incluye un ejemplo del funcionamiento del protocolo CDPRR. En primer lugar, hay un *round* de tamaño τ en el que ambos nodos están escuchando y se supone que otro nodo transmitió con éxito. Al final del *round* ambos nodos envían 1 paquete de *feedback* hacia los nodos en rango de transmisión. Más adelante, el nodo i está en estado transmitiendo y envía un paquete *BROADCAST* con su identificador y suponemos que transmite con éxito. El nodo j está escuchando, y al final del *round*, el nodo j envía un

paquete de *feedback* hacia el nodo i indicando que el paquete *BROADCAST* llegó bien.

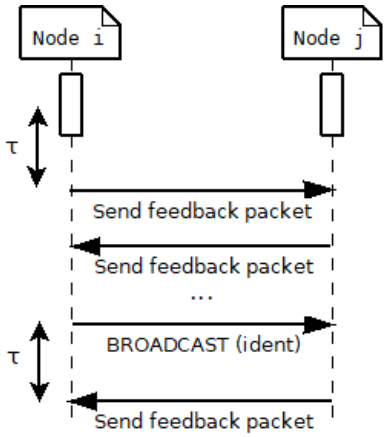

Figura 4.2: Sistema protocolo CDPRR

En las secciones 4.3.2 y 4.3.3 se explica con más detalle el funcionamiento de ambas propuestas.

4.3 Protocolos aleatorios basados en la detección de colisiones

En esta sección se presentan dos propuestas aleatorias basadas en la detección de colisiones, esto es, CDH y CDPRR.

4.3.1 Premisas

A continuación, se presenta las premisas que debe tener en cuenta las 2 propuestas:

- El tiempo está ranurado en *rounds* y todos los nodos conocen el tamaño de la ranura.
- No existe planificación predeterminada en la transmisión.
- Cada nodo está equipado con un transceptor de radio de alcance limitado.
- Todos los nodos tienen el mismo rango de transmisión.

- El funcionamiento está basado en el modo *half-duplex*, es decir, los nodos pueden transmitir o recibir pero no simultáneamente.
- Los nodos son desplegados aleatoriamente en un área dada.
- Los nodos son estáticos, esto es, no pueden moverse en el área de despliegue.
- Cada nodo tiene un identificador único que le distingue de los otros nodos, e.g., dirección MAC o número de serie de fabricante.
- Cada nodo debe conocer un identificador mínimo (ident_min) y un identificador máximo (ident_max). Ambos tienen el mismo valor para todos los nodos. Representan el mínimo y máximo identificador de todos los nodos posibles en la red. Los identificadores de los nodos deben estar en ese rango (entre ident_min e ident_max). Además, no es necesario que estén en la red desplegada todos los nodos ni los nodos con identificador ident_min ni ident_max.
- Los nodos requieren sincronización en los límites de las ranuras.
- Pueden existir colisiones.
- Los nodos pueden detectar colisiones y terminación.
- Los nodos pueden detectar energía.
- Ambos protocolos son basados en *handshake*.
- Cada nodo tiene una memoria interna para guardar información topológica local, en este caso la tabla de vecinos.
- El numero de nodos es conocido por todos los nodos en el CDPRR pero es desconocido por todos los nodos en el CDH.
- En el CDH, los nodos pueden iniciar la transmisión en diferentes instantes de tiempo.

4.3.2 CDPRR

En esta sección, se propone una solución aleatoria basada en la detección de colisiones, que llamamos CDPRR. Funciona tanto para redes *one-hop* como *multi-hop*. De acuerdo con la Figura 4.3, el protocolo CDPRR consiste en varios *rounds* y finaliza cuando todos los vecinos han sido descubiertos.

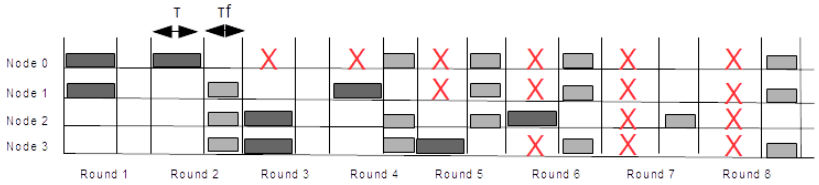

Figura 4.3: Protocolo CDPRR.

Este protocolo requiere sincronización en los límites de ranura, y usa transmisiones ranuradas con tamaño de ranura τ. Asumimos que los nodos pueden detectar colisiones cuando están escuchando. También se asume que en CDPRR el número de nodos de la red es conocido por todos los nodos. En el protocolo, hay dos *sub-slots*, el primero para enviar paquetes *BROADCAST* y el segundo se usa para enviar paquetes de *feedback*.

En CDPRR, el tiempo también está ranurado en *rounds* de tamaño τ. Cada nodo puede estar en uno de 3 posibles estados en cada *round*, esto es, T (transmitiendo), L (escuchando) o S (transmitió con éxito en *rounds* previos), como se muestra en la Figura 4.4. Esta Figura representa la máquina de estados de CDPRR y muestra el funcionamiento del protocolo.

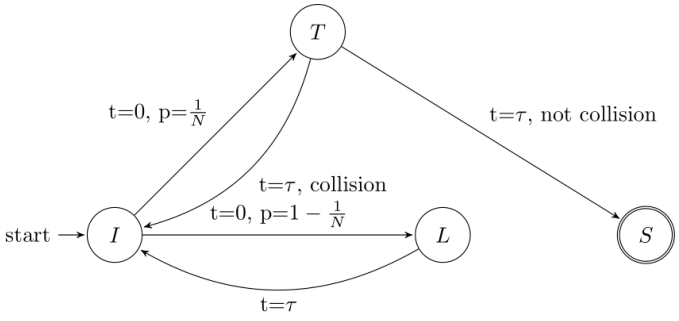

Figura 4.4: Máquina de estados de CDPRR

Como se muestra en la Figura 4.5, en el primer *sub-slot* al principio de un *round*, cada nodo elige de forma aleatoria un estado transmitiendo T con pro-

babilidad $\frac{1}{N}$ o escuchando L con probabilidad $1 - \frac{1}{N}$. Esta elección tiene lugar cuando el estado no es S. N es el número de nodos de la red. Si el estado es T, el nodo envía un paquete *BROADCAST* conteniendo su identificador hacia los nodos en su rango de transmisión durante un tiempo τ. En caso contrario se mantiene escuchando. Al final del *round*, ya se ha llevado a cabo detección de colisiones por los receptores dado que el modo *half-duplex* no permite al emisor que haga detección de colisiones. Luego, se abre un *sub-slot* de tamaño τ_f para enviar el *feedback* que consiste en un solo paquete de *feedback*. Cuando un solo nodo logra transmitir con éxito en un *round*, los receptores del *BROADCAST* no detectan ni colisión ni canal inactivo. Luego proceden a actualizar la tabla de vecinos con el identificador en el *BROADCAST* y envían un paquete de *feedback* hacia los otros nodos. En caso contrario, se detecta una colisión o el canal está inactivo y no envían el paquete de *feedback*. Al mismo tiempo, los nodos que transmitieron, escuchan el canal y cuando detectan energía en el canal, el estado cambiará a S. Esto significa que transmitió con éxito, inicia un nuevo *round*, y permanece en este estado hasta el final del algoritmo. Este nodo no continuará compitiendo en los siguientes *rounds* (una marca X en rojo en la Figura 4.3) y permanecerá escuchando hasta el final del algoritmo. Sin embargo, continuará enviando paquetes de *feedback* cuando sea necesario. En caso contrario, esto es, no se detecta energía, el nodo que transmitía inicia un nuevo *round* eligiendo un nuevo estado. Los nodos que recibieron el *BROADCAST* simplemente inician un nuevo *round*. En el caso de que el nodo esté en estado L, el nodo inicia un nuevo *round* y elige un nuevo estado. En caso contrario, esto es, si el nodo está en estado S, el nodo inicia un nuevo *round* pero no elige un nuevo estado. El tamaño del paquete de *feedback* es mucho más pequeño que el del *BROADCAST*, e indica que una transmisión con éxito tuvo lugar. El protocolo finaliza cuando todos los vecinos han sido descubiertos. Resaltar que en el proceso de *feedback*, los receptores sólo enviarán un paquete de *feedback*. Además, no provocan colisiones ya que las transmisiones de los paquetes de *feedback* están perfectamente sincronizadas y los transmisores sólo necesitan detectar energía.

Si hay una colisión en el primer *sub-slot*, esto significa que los nodos que la provocaron no han logrado transmitir con éxito. Por tanto en el segundo *sub-slot* los nodos en estado L o S no envían el paquete de *feedback*.

Resaltar que una colisión no puede tener lugar en el segundo *sub-slot* dado que sólo se detecta energía.

La Figura 4.3 también incluye un ejemplo del funcionamiento del protocolo. Los nodos están desplegados en una malla de 2x2 en un escenario *one-hop*. En el *round* 1, los nodos 0 y 1 están en estado T, por tanto se produce una

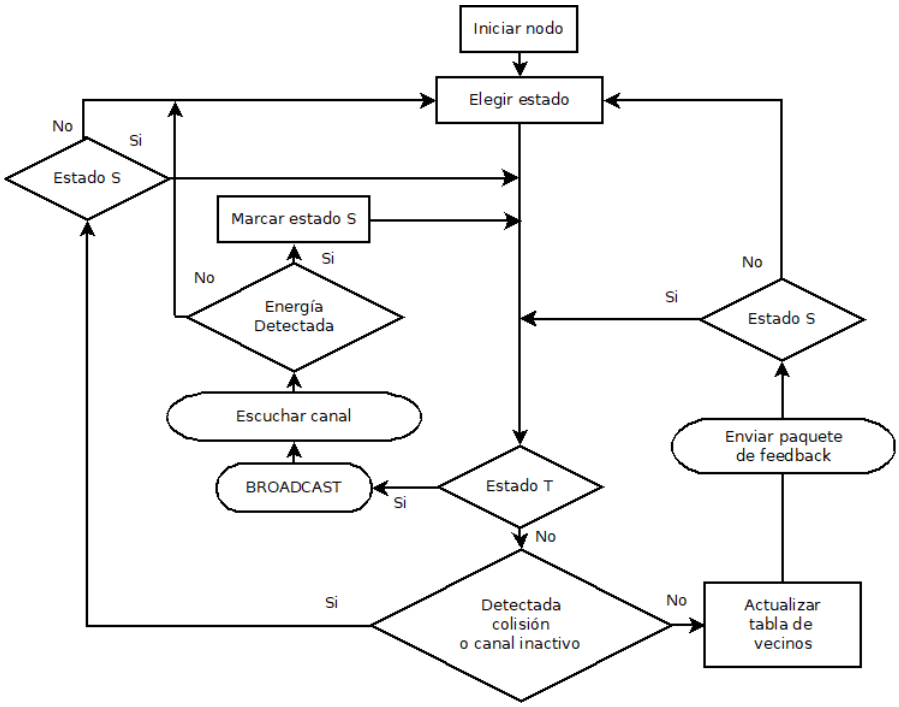

Figura 4.5: Diagrama de flujo CDPRR.

colisión y todos los nodos continúan compitiendo en el siguiente *round*. En el *round* 2, sólo el nodo 0 transmite con éxito, por tanto dejará de competir desde entonces y una marca X roja indica esta situación. En el *round* 3, se produce de nuevo una colisión. En el *round* 4, el nodo 1 logra transmitir con éxito, mientras que los nodos 2 y 3 continúan compitiendo en los siguientes *rounds*. En el *round* 5, el nodo 3 transmite con éxito. Finalmente, en el *round* 6, el nodo 2 logra transmitir con éxito. En el *round* 7 y *round* 8 se comprueba que todos los nodos transmitieron con éxito, lo que significa que el algoritmo finaliza.

En CDPRR, hay un mecanismo de detección de terminación, en el cual si todos los nodos transmitieron con éxito, esto es, están todos en estado S, el algoritmo finaliza. En caso contrario, un nuevo *round* comienza. En redes *one-hop*, cuando el número de nodos es conocido, el protocolo sabe cuando terminar si todos los nodos han descubierto todos sus vecinos $(N-1)$, consultando la tabla de vecinos. Cuando un nodo ha descubierto $N-1$ vecinos (el nodo sabe que $N-1$ nodos han sido descubiertos consultando el número de nodos

en su tabla de vecinos) continuará compitiendo. Tan pronto como el nodo que ha descubierto sus $N-1$ vecinos elige transmitir el *BROADCAST* los restantes nodos han descubierto $N-1$ vecinos y envían el paquete de *feedback*. Entonces el nodo detecta energía, cambia su estado a S y espera un único *round*. A continuación, en el siguiente *round*, el nodo debe sólo enviar el paquete de *feedback* y finaliza. Los restantes nodos que han descubierto sus $N-1$ vecinos detectan energía y esperan un único *round*. En el siguiente *round* para esos $N-1$ vecinos deben sólo enviar el paquete de *feedback* cada uno y finalizan. Por lo tanto, dos *rounds* adicionales son necesarios para el *handshake* de terminación.

De acuerdo con el ejemplo en la Figura 4.3, al principio del *round* 6 los nodos 0,1 y 3 han descubierto $N-2$ vecinos, mientras que el nodo 2 ha descubierto $N-1$ vecinos y continúa compitiendo. En el *round* 6, el nodo 2 transmite el *BROADCAST*, los nodos 0,1 and 3 envían los paquetes de *feedback*, por tanto, el nodo 2 detecta energía y espera al *round* 7. En el *round* 7, los nodos 0,1 y 3 han descubierto $N-1$ vecinos y el nodo 2 debe sólo enviar el paquete de *feedback* y para. Los nodos 0, 1 y 3 detectan energía, por tanto los nodos 0, 1 y 3 esperan al *round* 8. En el *round* 8, los nodos 0, 1 y 3 deben sólo enviar paquetes de *feedback* y paran. Entonces el protocolo ha finalizado para todos los nodos.

Sin embargo, en el caso *multi-hop*, el protocolo finaliza cuando en un número de *rounds* consecutivos (que se debe fijar adecuadamente), los nodos no reciben ningún *BROADCAST* (todos ellos están escuchando). Se utiliza esta condición ya que la probabilidad de que ningún nodo envíe un *BROADCAST* y estén en estado L en varios *rounds* consecutivos es muy baja. Por lo tanto concluimos que todos los nodos están en estado S. Por supuesto, en el caso *multi-hop*, la probabilidad de descubrimiento con éxito no es 1.

Resaltar que τ coincide con la duración del paquete *BROADCAST*.

En CDPRR se ha fijado la probabilidad de transmisión a $\frac{1}{N}$ ya que esa probabilidad maximiza la probabilidad de descubrimiento.

Si se pierde un paquete *BROADCAST*, el nodo continuará en el siguiente *round*. Además, si se pierde un paquete de *feedback* y los paquetes de *feedback* de otros nodos no se pierden, el protocolo funciona bien. Sin embargo, si todos los paquetes de *feedback* se pierden, los nodos continuará en el siguiente *round*.

CDPRR aborda el problema del nodo oculto (permitiendo la escalabilidad en escenarios *multi-hop*) como sigue. Sean 3 nodos A, B, y C. A y B están en rango de transmisión, B y C también están en rango de transmisión, mientras

que A y C están fuera del rango de transmisión el uno del otro. Si ni A ni B ni C envían *BROADCAST* todos continúan compitiendo. Si A y C envían *BROADCAST* hay una colisión en B (los 3 nodos continúan en el siguiente *round*). Si los 3 envían *BROADCAST* entonces los 3 continúan compitiendo. Si B envía *BROADCAST*, A y C envían *feedback* y B es descubierto (permanece escuchando) mientras que A y C continúan compitiendo. Si ni A ni B envían *BROADCAST* entonces ningún nodo es descubierto y los nodos continúan compitiendo en el siguiente *round*. Si A o C envían *BROADCAST* entonces A o C son descubiertos y B continúa compitiendo. Si A y B envían *BROADCAST* entonces se produce una colisión, mientras que C recibe el *BROADCAST* de B y C envía *feedback*. Por tanto B es descubierto por el nodo C y deja de competir. Así, A no descubrió el nodo B y B no es vecino de A pero lo es de C por tanto B no será descubierto por A cuando esta situación tiene lugar. Tanto A como C continúan compitiendo. Para concluir, debido al problema del nodo oculto algunos nodos pueden no ser descubiertos.

CDPRR puede ser extendido al caso *multi-hop* en el cual uno o más nodos pueden pertenecer a varias subredes. Se tendrá en cuenta que esos nodos pueden enviar *BROADCASTs* y paquetes de *feedback* hacia todas las subredes a las cuales pertenecen simultáneamente. El principal problema que ocurre es el problema del nodo oculto que abordamos antes. En los otros casos, el protocolo funciona adecuadamente de la misma forma que en el caso *one-hop*. En el caso *multi-hop* la condición de terminación cambia como se indicó antes. El protocolo finaliza cuando todos los nodos están escuchando en varios *rounds* consecutivos.

Un caso típico que puede ocurrir es una red compuesta de nodos A,B, y C. A y C están fuera de rango de transmisión el uno del otro, A está en rango de transmisión de B, y C está en rango de transmisión de B. Si B y C envían el *BROADCAST* simultáneamente, asumiendo un funcionamiento *half-duplex*, ambos están transmitiendo por tanto no se escuchan el uno al otro y tiene lugar una colisión. Sin embargo, si consideramos el efecto de captura RF que puede ocurrir en escenarios prácticos, se da lo siguiente. B podría ser capaz de recibir el *BROADCAST* de C correctamente en vez de detectar colisión. Además, el umbral para detectar transmisión de energía es normalmente más bajo que para una recepción correcta de paquetes. Por tanto, el *feedback* de B puede ser detectado tanto por A como por C. En este caso, C será descubierto por B, mientras que A no será descubierto por B. Sin embargo si A está compitiendo en este *round* A deja de competir, y C deja de competir por tanto A no será descubierto por ningún nodo de la red.

Se puede dar un escenario realista en el cual el número real de nodos en la red (N') es menor que el número de nodos conocido (N). En primer lugar, la condición de terminación explicada antes no es válida ya que cuando $N' < N$ los nodos no sabrán cuando terminar. Por lo tanto, la condición de terminación debería ser cambiada de forma que el protocolo finalize cuando en un número de *rounds* consecutivos todos los nodos están escuchando. El protocolo funcionará adecuádamente incluso cuando $N' < N$, dado que la probabilidad de transmisión es aún $\frac{1}{N}$. La probabilidad de escucha es aún $1 - \frac{1}{N}$, y tanto los *BROADCASTs* como los *feedbacks* serán enviados y recibidos sólo por los N' nodos.

Resaltar que en CDPRR se emiten paquetes de *feedback* y no 1 bit cuando un nodo transmite con éxito. La mayor duración de un paquete de *feedback* mejora a la hora de detectar energía.

4.3.3 CDH

En esta sección, se propone una solución aleatoria basada en la detección de colisiones, para redes tanto *one-hop* como *multi-hop*, que llamamos CDH.

En el protocolo CDH el tiempo está ranurado en *rounds* como se puede observar en la Figura 4.6, y hay dos *sub-slots* en un *round*. La duración en segundos del primer *sub-slot* es ω mientras que el tamaño del segundo *sub-slot* (*feedback*) es ω_f en segundos. Los tiempos ω y ω_f son fijos (los mismos para todos los *rounds*) y no dependen del número de nodos. Cada nodo puede estar en 3 posibles estados: *Transmit*, *Listen* o *Success*, como se observa en la Figura 4.7. Si el nodo está en el estado *Success* significa que el nodo logró transmitir con éxito en rounds previos.

En primer lugar, como se muestra en la Figura 4.6 y el diagrama de flujo en la Figura 4.8, el protocolo CDH consiste en varios *rounds* y finaliza cuando todos los vecinos han sido descubiertos.

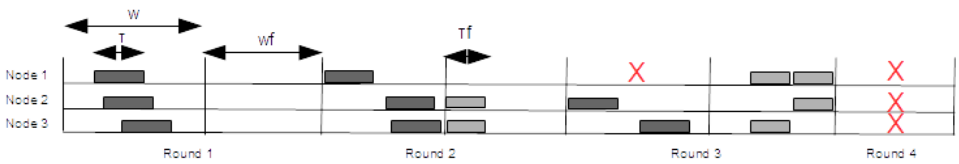

Figura 4.6: Protocolo CDH.

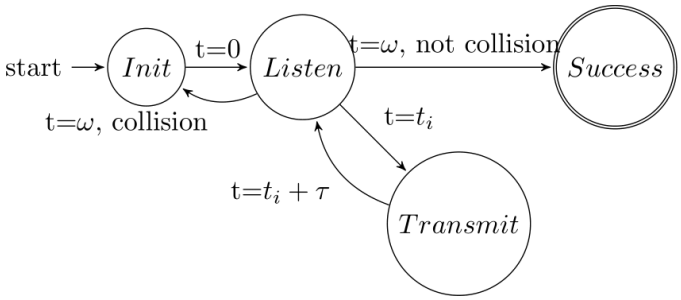

Figura 4.7: Máquina de estados de CDH

De acuerdo con la Figura 4.6 y la Figura 4.8, en el primer *sub-slot* de un *round*, cada nodo transmite un solo paquete de *BROADCAST* conteniendo el identificador. Esto se produce cuando el nodo no está en estado *Success*. El paquete se envía iniciando en un tiempo elegido de forma aleatoria t_i ($t_i \in [0, \omega - \tau]$ siendo ω la duración del primer *sub-slot*). Está transmisión se produce durante τ, que es el tiempo en que un nodo está transmitiendo. El nodo escucha los mensajes entrantes durante el resto de la ranura, esto es, $\omega - \tau$. Durante los periodos de escucha en el primer *sub-slot* de cada nodo, ese nodo realiza un proceso de detección de colisiones. Decimos que se ha detectado colisión cuando los paquetes *BROADCAST* emitidos por 2 o más nodos se solapan en el tiempo. En otro caso, decimos que un nodo transmitió con éxito. En el segundo caso, es decir, un nodo logró transmitir con éxito, el resto de nodos en rango de transmisión guardan el identificador del nodo que emitió el *BROADCAST* en su tabla de vecinos. Todos los nodos usan un segundo *sub-slot* para decir a los otros nodos qué nodos han transmitido con éxito. Para ello, envían una serie de paquetes de *feedback* uno tras otro desde la posición *ident_min* hasta la posición *ident_max*. Éstos son los identificadores mínimo y máximo de los posibles nodos de la red. El número máximo de paquetes de *feedback* fijado a través de *ident_min* e *ident_max* debe ser suficiente para considerar los identificadores de todos los nodos. Los identificadores podrían ser números no consecutivos. El orden de transmisión de los paquetes de *feedback* es de *ident_min* a *ident_max*. Resaltar que los IDs pueden ser números no consecutivos pero los *feedbacks* son transmitidos en orden (desde $ident_min$ hasta *ident_max*). Además se asume que los nodos no conocen la lista de IDs. Cuando el *jth feedback* es planificado para ser enviado, los nodos con identificador distinto de j enviarán un paquete de *feedback* si el nodo j transmitió con éxito. El nodo con identificador igual a j escuchará el canal. En caso contrario, el nodo identificador no enviará el paquete de *feedback* dado que el nodo j provocó una colisión. Si no fue detectada una colisión para el

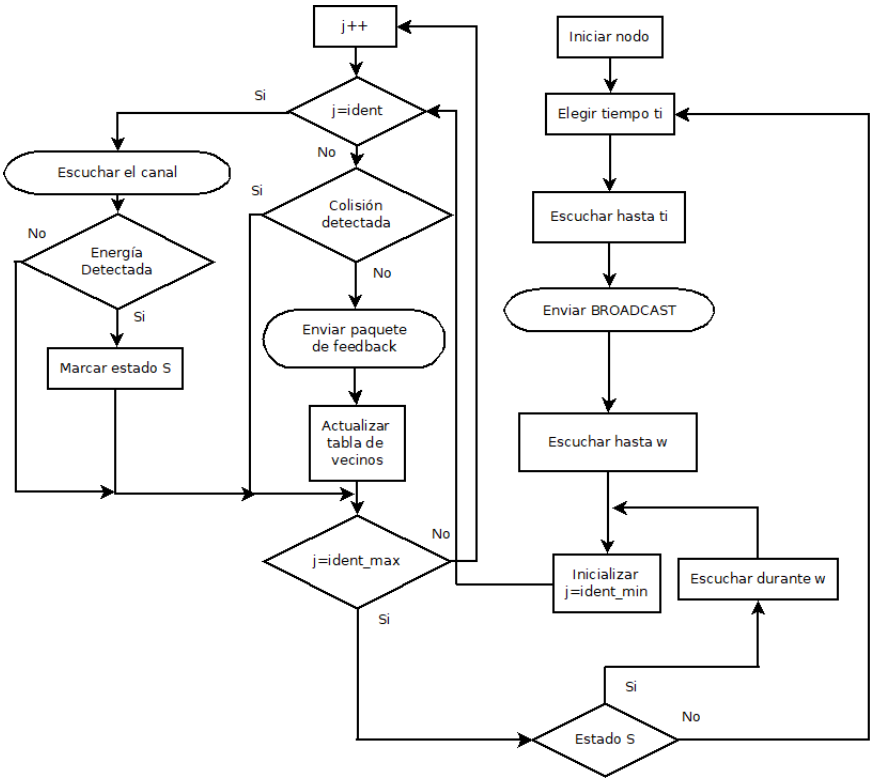

Figura 4.8: Diagrama de flujo CDH.

nodo j, lo cual significa que el nodo j transmitió con éxito, el resto de nodos actualizan sus tablas de vecinos con el identificador de j. Este identificador está en el *BROADCAST* que no colisionó. Además, envían el paquete de *feedback*. En otro caso, no envían el paquete de *feedback*. Los nodos con identificador igual a j ahora escuchan el canal y si se detecta energía, saben que transmitieron con éxito. Esto se detecta cuando hay un paquete de *feedback* en el canal. Una marca X roja en la Figura 4.6 indica esto. El nodo cambiará al estado *Success* hasta el final del algoritmo, no competirá en los siguientes *rounds*. Permanecerá escuchando los paquetes *BROADCAST* de otros nodos, aunque continuará enviando los paquetes de *feedback* cuando sea necesario. En caso contrario, esto es, ocurrió una colisión, continuará compitiendo en los siguientes *rounds*. Cuando j llega al $ident\_max$, si el estado del nodo es *Success* no enviará el paquete *BROADCAST* en los siguientes *rounds*, en caso contrario un nuevo *round* inicia y el nodo elegirá un nuevo t_i.

Los paquetes de *feedback* no colisionan dado que los nodos sólo deben detectar energía. Los paquetes de *feedback* son mucho más pequeños que los *BROADCASTs*.

Si un nodo que está escuchando detecta otra transmisión de un nodo j, el comportamiento dependerá de la detección de colisiones. Si no ocurrió colisión, el nodo enviará el jth paquete de *feedback* en el segundo *sub-slot*, en caso contrario no enviará un paquete de *feedback* en el segundo *sub-slot*.

Este protocolo incluye un mecanismo de detección de terminación. El protocolo finaliza cuando todos los nodos han logrado transmitir con éxito en *rounds* previos, esto es, todos los nodos están en estado *Success*. Esto se detecta cuando en un *round* no hay señal en el canal en el primer *sub-slot*, esto es, los nodos no enviaron sus paquetes de *BROADCAST*. Ésto indica que están todos en el estado *Success*. En ese caso, el algoritmo finaliza.

Si hay colisiones en el primer *sub-slot*, esto significa que los nodos que la provocaron no han logrado transmitir con éxito. Por lo tanto en el segundo *sub-slot* los nodos no envían un paquete de *feedback* para estos nodos que colisionaron. Resaltar que no puede tener lugar colisión en el segundo *sub-slot* ya que solo se quiere detectar energía y los nodos están perfectamente sincronizados.

Se ha asumido que los nodos pueden detectar colisiones cuando están escuchando.

Este protocolo requiere sincronización en los límites de ranura.

Si se pierde un paquete de *BROADCAST*, el nodo continúa en el siguiente *round*. Sin embargo, cuando éste es el único nodo que está compitiendo y se pierde, el protocolo asume que todos los nodos han transmitido con éxito. En este caso, el protocolo finaliza y este nodo no es descubierto. Además, si se pierde un paquete de *feedback* y los otros paquetes de *feedback* para el mismo nodo no se pierden, el protocolo funciona bien. Si todos los *feedbacks* para un nodo dado se pierden, entonces este nodo continuará en el siguiente *round*.

La Figura 4.6 muestra un ejemplo de funcionamiento del protocolo en un escenario *one-hop*, en el que todos los nodos están en rango de transmisión de todos los demás. En el *round* 1, los paquetes de los 3 nodos se solapan en el tiempo, provocando una colisión, por tanto todos continúan compitiendo en el siguiente *round*. En el *round* 2, el nodo 1 transmite con éxito, por tanto no continúa compitiendo en los siguientes *rounds* como se muestra con una marca X roja en los *rounds* 3 y 4. Los paquetes de los nodos 2 y 3 se solapan en el tiempo provocando una colisión, por tanto continúan compitiendo en el *round*

3. En el *round* 3 tanto el nodo 2 como el 3 transmiten con éxito. Al final del *round* 3, todos los nodos han transmitido con éxito, y el algoritmo finaliza en el *round* 4.

El paquete de *feedback* es mucho más pequeño que el *BROADCAST*. Esto significa una ventaja sobre los reconocimientos intercambiados cuando se tiene en cuenta el origen y destino que contienen los paquetes de *ACK*. Además los paquetes de *feedback* no provocan colisiones dado que sólo debe ser detectada energía.

CDH aborda el problema del nodo oculto como sigue. Sean 3 nodos A, B, y C. A y B están en rango de transmisión, B y C están en rango de transmisión. A y C están fuera de rango de transmisión el uno del otro. Si A y B envían un *BROADCAST*, una colisión tiene lugar, mientras que C recibe el *BROADCAST* de B, y C envía el *feedback*. Por lo tanto B es descubierto por el nodo C y B deja de competir. A no descubrió a al nodo B y B no es vecino de A pero B es vecino de C por tanto B no será descubierto por A cuando esta situación tiene lugar. Ambos A y C continúan compitiendo. Para concluir, debido al problema del nodo oculto algunos vecinos pueden no ser descubiertos.

CDH puede ser extendido al escenario *multi-hop* en el cual uno o más nodos pueden pertenecer a varias subredes. De la misma forma que para el escenario *one-hop*, estos nodos pueden enviar *BROADCASTs* y paquetes de *feedback* hacia todas las subredes a las cuales pertenecen simultáneamente. El principal problema que ocurre es el problema del nodo oculto que abordamos antes. En los otros casos, el protocolo opera adecuádamente de la misma forma que en escenarios *one-hop*. En el escenario *multi-hop*, la condición de terminación es la misma que en escenarios *one-hop*. El protocolo finaliza para un nodo cuando el canal está inactivo durante un *round*, es decir, todos los nodos en rango de transmisión de este nodo han logrado transmitir con éxito en *rounds* previos.

De igual forma a lo que sucede en CDPRR, un caso típico puede ocurrir en una red compuesta de los nodos A, B, y C. A y C están fuera del rango de transmisión el uno del otro. A está en el rango de transmisión de B. C está en el rango de transmisión de B. Si ambos B y C envían *BROADCASTs* al mismo tiempo, teniendo en cuenta que asumimos un modo *half-duplex*, y están ambos transmitiendo. Por tanto no se escuchan el uno al otro y una colisión tiene lugar. Sin embargo, considerando el efecto de captura de RF que puede darse en en entornos prácticos, B podría ser capaz de recibir el *BROADCAST* de C correctamente en vez de detectar una colisión. Además, el umbral para la detección de transmisión de energía es normalmente más bajo que para la recepción correcta de paquetes. Por lo tanto, el *feedback* de B puede ser

detectado tanto por A como por C. En este caso, C será descubierto por B. A no será descubierto por B. Sin embargo si A está aún compitiendo en este *round*, A deja de competir por tanto A no será descubierto por ningún nodo de la red.

Para concluir, los nodos que escuchan y recibieron un *BROADCAST* deben enviar paquetes de *feedback* en el segundo *sub-slot* correspondiendo a los emisores que emitieron con éxito. Con este objetivo, los *feedbacks* necesitan ser enviados con sincronización precisa y en el orden correcto. El emisor solo comprobará la transmisión de energía. El protocolo sólo necesita saber el identificador mínimo ($ident\_min$) y el identificador máximo ($ident\_max$) de los posibles nodos en la red. No necesita conocer el número de vecinos implicados. Además, los identificadores pueden ser números no consecutivos. Los nodos que no se despliegan en la red no serán considerados cuando se envían *BROADCASTs* y paquetes de *feedback*. Sin embargo, los *feedbacks* deben ser enviados de forma determinística. El protocolo no necesita saber el orden de cada ID de vecino. Así, el orden será determinado por el identificador.

4.4 Protocolos de referencia

En el presente capítulo, se ha decidido elegir 2 protocolos aleatorios de la literatura para ser usados como referencia. Ambos protocolos de referencia son similares a las dos propuestas. Mediante las dos propuestas comprobamos la ventaja de incorporar detección de colisiones. El objetivo de las dos propuestas será mejorar esos protocolos de referencia. El primero de ellos, es conocido como Hello [22], y requiere que el tiempo esté ranurado en *rounds* de tamaño ω. En cada *round*, los nodos transmiten un paquete *BROADCAST* en un tiempo elegido aleatoriamente t_i, tal que $0 \leq t_i \leq \omega - \tau$. Los nodos están transmitiendo durante un tiempo τ, esto es, el tiempo que un nodo está transmitiendo. Los nodos escuchan durante el resto de la ranura por un tiempo total $\omega - \tau$. El tiempo t_i puede ser diferente entre nodos de un mismo *round* y entre *rounds* de un mismo nodo. Se dice que una colisión tiene lugar cuando dos o más nodos transmiten al mismo tiempo, es decir, sus paquetes se solapan en el tiempo. En caso contrario, decimos que se ha producido una transmisión con éxito, y un vecino se ha descubierto. Al tratarse de un protocolo que no es basado en *handshake*, no sabremos cuando finaliza. Por ello, el número de *rounds* debe ser elegido adecuadamente, ya que tras un numero finito de *rounds* el protocolo finaliza.

En cuanto al segundo protocolo de referencia, conocido como PRR [21], el tiempo también está ranurado en *rounds* de tamaño τ. En cada *round*, los nodos eligen aleatoriamente un estado, esto es, transmitir (estado T) con probabilidad $\frac{1}{N}$ o escuchar (estado L) con probabilidad $1-\frac{1}{N}$. Los estados pueden ser diferentes entre nodos en un mismo *round* y entre *rounds* de un mismo nodo. De nuevo, una colisión se produce cuando dos o más nodos transmiten al mismo tiempo. En caso contrario, un nodo transmitió con éxito y por tanto un vecino es descubierto. De nuevo, el protocolo no es basado en *handshake* y el número de *rounds* debe ser fijado adecuadamente dado que tras ese número de *rounds* el protocolo finaliza.

A continuación, se presenta una comparación cualitativa de los protocolos de referencia y las dos propuestas, en la Tabla 4.1. Según se muestra en la Tabla, Hello y PRR son aleatorios, requieren sincronización en los límites de las ranuras, y pueden ser usados en escenarios *multi-hop*. Ninguno de los dos es capaz de descubrir todos los vecinos con probabilidad 1, y son *one-way*. Por otro lado, CDPRR y CDH son adecuados para su uso en escenarios *one-hop* y *multi-hop*, permiten la detección de colisiones y terminación. Ambos pueden descubrir todos los vecinos con probabilidad 1 incluso en redes compuestas por nodos con una gran cantidad de vecinos, y son basados en *handshake*. Además, a medida que los *rounds* van pasando, hay menos nodos compitiendo. Por lo tanto, la probabilidad de descubrimiento se incrementa, la probabilidad de colisión se reduce, y el tiempo de descubrimiento también se reduce. En el caso particular de CDH, permite a los nodos iniciar la transmisión en diferentes instantes de tiempo, y sigue premisas más realistas. Entre ellas, los nodos no necesiten conocer el número de nodos de la red.

4.5 Escenario de simulación

Para la obtención de los resultados de simulación, utilizamos el mismo escenario de simulación para los 4 protocolos a comparar, esto es, Hello [22], PRR [21], CDH y CDPRR. El simulador usado para la comparación es Castalia versión 3.2 [88]. Este simulador está basado en el bien conocido OMNET++, y se usa básicamente para simular WSN y BAN. El principal motivo de su elección es que reúne los requisitos para validar protocolos de descubrimiento de vecinos. Es adecuado tanto para redes estáticas como móviles, y tanto en escenarios *one-hop* como *multi-hop*. Para comprobar la escalabilidad, variamos el número de nodos de la red, donde N nodos se organizan en mallas MxM. Para su simulación, tanto para Hello como para CDH, fijamos un tamaño de *round* de

$\omega = N \cdot \tau$. N es el número de nodos de la red y τ el tiempo que un nodo está transmitiendo. Fijamos $\tau = 0.07s$ para todos los protocolos simulados.

En cuanto al área de despliegue, tenemos dos casos diferenciados. En primer lugar, 10mx10m para escenarios *one-hop*, un caso simple pero útil para redes que cuentan con transceptores con muy alto rango de transmisión, y que puede llegar a 500m. En él todos los nodos están en rango de trasnmisión de todos los demás. En segundo lugar, 100mx100m, para el caso *multi-hop*, un escenario más realista. En él sólo algunos nodos están en rango de transmisión de los demás.

Para las simulaciones, en presencia de colisiones, se ha fijado el modelo de colision, haciendo uso del parámetro *collisionModel* de Castalia 3.2. Permite tomar los valores 0 (sin colisiones), 1 (modelo simplista para colisiones), o 2 (modelo de interferencia aditiva). En este caso, fijamos el valor de este parámetro a 2 (modelo de interferencia aditiva), el modelo más realista, para todas las simulaciones.

Para PRR y Hello, fijamos un número de *rounds* dado que tras un número finito de *rounds* ambos protocolos finalizan. Para simular PRR el número de *rounds* en escenario *one-hop* se ha fijado a 10N mientras que para *multi-hop* a 6N. Para simular Hello el núnero de *rounds* en escenario *one-hop* se ha fijado a 0.5N mientras que para *multi-hop* a 0.25N.

El principal objetivo de los protocolos de descubrimiento de vecinos es descubrir todos los vecinos, o casi todos, en una cantidad de tiempo reducido. Por tanto, elegimos como métricas el Tiempo de descubrimiento de vecinos y el Número de vecinos descubiertos. El protocolo será mejor cuando el Tiempo de descubrimiento de vecinos sea menor y el Número de vecinos descubiertos sea mayor. Adicionalmente, se ha obtenido el Consumo energético, dado que los nodos están equipados con baterías que se agotarán en un tiempo determinado. Además, se ha obtenido el *Throughput* y el *ratio* Número de vecinos descubiertos por paquetes enviados.

El Tiempo de descubrimiento de vecinos está inversamente relacionado con el Número de vecinos descubiertos. Para fijar el número de *rounds* de Hello y PRR, hemos de averiguar un número de *rounds* determinado. Usando este número de *rounds* las propuestas mejoren a los protocolos de referencia en las dos métricas o empeoren las dos métricas. Tras una serie de simulaciones se ha descubierto que con el número de *rounds* fijado más arriba y en la Tabla 4.2, las 2 propuestas mejoran a los protocolos de referencia en las 2 métricas.

Tabla 4.2: Parámetros de simulación.

Parámetro	Valor
Static	True
Modelo de radio	CC2420
Modelo de colisión	2
Potencia de transmisión	-5dBm
Tasa de paquetes	5 paquetes/s
Tamaño de paquete	2500 bytes
Tamaño de *slot* Hello y CDH	$\omega = N \cdot \tau$
τ	0.07s
Tamaño *one-hop*	10mx10m
Tamaño *multi-hop*	100mx100m
Despliegue	Malla MxM
Número de *rounds* PRR *one-hop*	10N
Número de *rounds* Hello *one-hop*	0.5N
Número de *rounds* PRR *multi-hop*	6N
Número de *rounds* Hello *multi-hop*	0.25N

El modelo de radio usado es *ZigBee* (*CC2420*), la potencia de transmisión se ha fijado a -5dBm, la tasa de paquetes a 5 paquetes/s, y el tamaño de paquete a 2500 bytes.

Los parámetros de simulación se recogen en la Tabla 4.2.

4.6 Resultados de simulación

En esta sección, se muestran y discuten los resultados de simulación obtenidos para comparar las 2 propuestas y los 2 protocolos de referencia, en entornos estáticos *one-hop* y *multi-hop*.

4.6.1 Tiempo de descubrimiento de vecinos

El tiempo de descubrimiento de vecinos se refiere a la cantidad de tiempo que el protocolo tarda en finalizar. La finalización será debida bien sea porque se completó el número de *rounds* en Hello y PRR, o bien cuando CDH y CDPRR han logrado descubrir todos los vecinos.

En primer lugar, se presentan los resultados para el escenario *one-hop*. Es un caso simplista pero útil para aplicarlo a situaciones reales cuando los transceptores tienen un rango de transmisión elevado.

La Figura 4.9 nos muestra que CDPRR mejora los otros protocolos en cuanto a tiempo de descubrimiento. Además, sigue una tendencia creciente a medida que el número de nodos aumenta. A continuación, CDH presenta buenos resultados, seguido por PRR con 10N *rounds* y finalmente Hello con 0.5N *rounds* tiene las peores prestaciones.

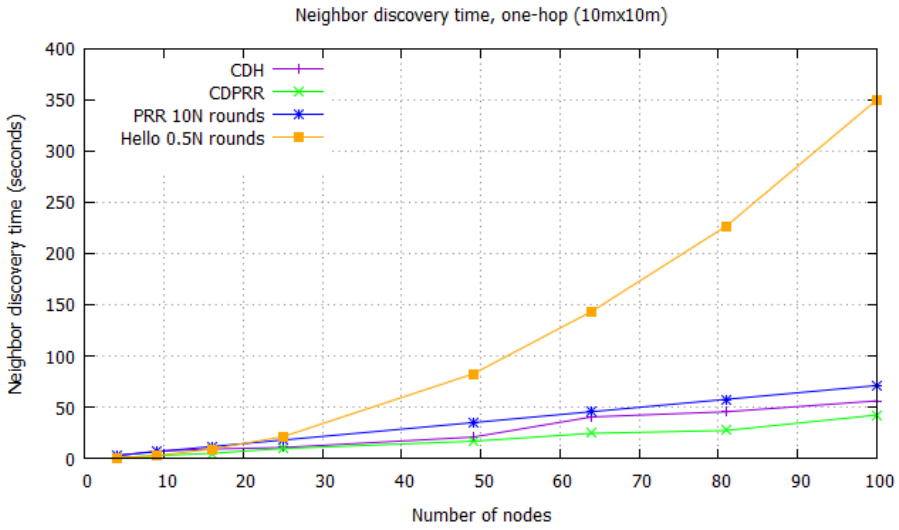

Figura 4.9: Tiempo de descubrimiento de vecinos (one-hop)

Para CDH y CDPRR, se puede concluir que, a medida que el número de nodos crece el tiempo de descubrimiento se incrementa. Esto se debe a que más nodos tienen que lograr transmitir con éxito. Para Hello y PRR el tiempo de descubrimiento se incrementa puesto que el número de *rounds* depende de N. Conforme pasan los *rounds*, en CDH y CDPRR, hay menos nodos compitiendo. Éste es el motivo por el cual el tiempo de descubrimiento es menor para las dos propuestas en comparación con Hello y PRR.

Añadir que, para obtener la Figura 4.9, se ha usado el modelo de interferencia aditiva, por ser el modelo de colisión más realístico. Para ello, se ha fijado el parámetro *collisionModel* de Castalia 3.2 a 2. Sin embargo, se puede demostrar que se obtienen idénticos resultados para los otros dos modelos de colisión.

A continuación, se presentan los resultados de simulación en un escenario *multi-hop*. Es un entorno más realista, en el cual sólo algunos nodos están en rango de transmisión de los demás. Para ello, se fija el parámetro *collisionModel* a 2, el modelo de colisión más realista. Sin embargo, se puede demostrar que se obtienen resultados idénticos para los otros dos modelos de colisión.

Según la Figura 4.10, CDPRR mejora los otros protocolos, seguido por CDH, que es mejor que PRR con 6N *rounds*, y finalmente Hello con 0.25N *rounds* presenta el peor comportamiento.

El tiempo de descubrimiento en CDH y CDPRR se incrementa a medida que el número de nodos crece ya que hay más nodos que tienen que lograr transmitir con éxito. Para Hello y PRR también se presenta este comportamiento dado que en ambos protocolos el número de *rounds* y por tanto el tiempo de descubrimiento depende de N.

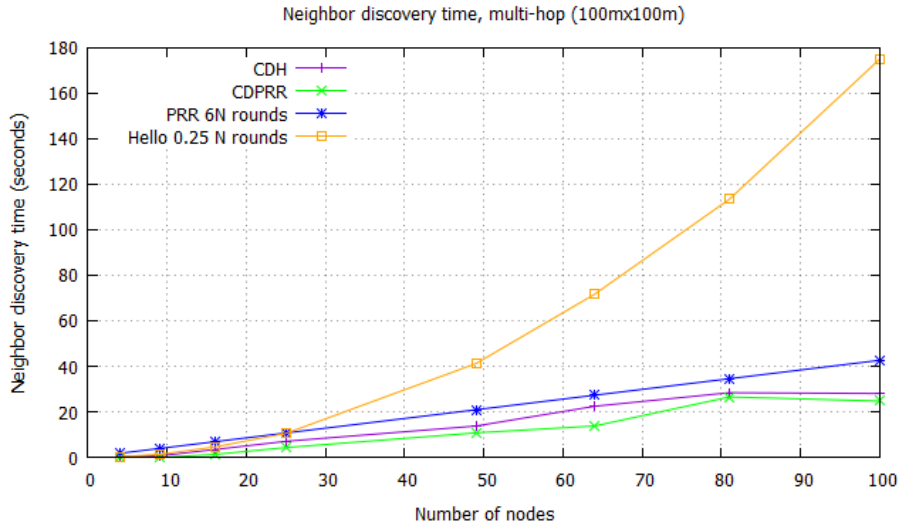

Figura 4.10: Tiempo de descubrimiento de vecinos (multi-hop)

4.6.2 Número de vecinos descubiertos

En primer lugar, se presentan los resultados en relación con el número de vecinos descubiertos en un entorno *one-hop*.

Variando el parámetro *collisionModel* de Castalia 3.2, obtenemos que para el caso sin colisiones todos los protocolos presentan el mismo comportamiento ideal. Todos los protocolos logran descubrir todos los vecinos ya que no hay colisiones. Utilizando el modelo de interferencia aditiva, modelo más realista, obtenemos los resultados que se muestran en la Figura 4.11. Según ella, CDH y CDPRR presentan resultados óptimos descubriendo todos los vecinos, y mejora los otros protocolos. Seguidamente PRR con 10N *rounds* y Hello con 0.5N *rounds* presentan resultados similares. Sin embargo, Hello con 0.5N *rounds* presenta los peores resultados para bajo número de nodos, ya que descubre casi todos los vecinos cuando el número de nodos es bajo, como se indicó en [22].

Se han validado los protocolos bajo diferentes modelos de colisión, "sin colisiones", "modelo simplista para colisiones" y "modelo de interferencia aditiva", que se presentaron en [22]. Concluimos, tras proceder a la simulación, que los resultados para los modelos de colisión 1 y 2 son similares, en relación con el número de vecinos descubiertos.

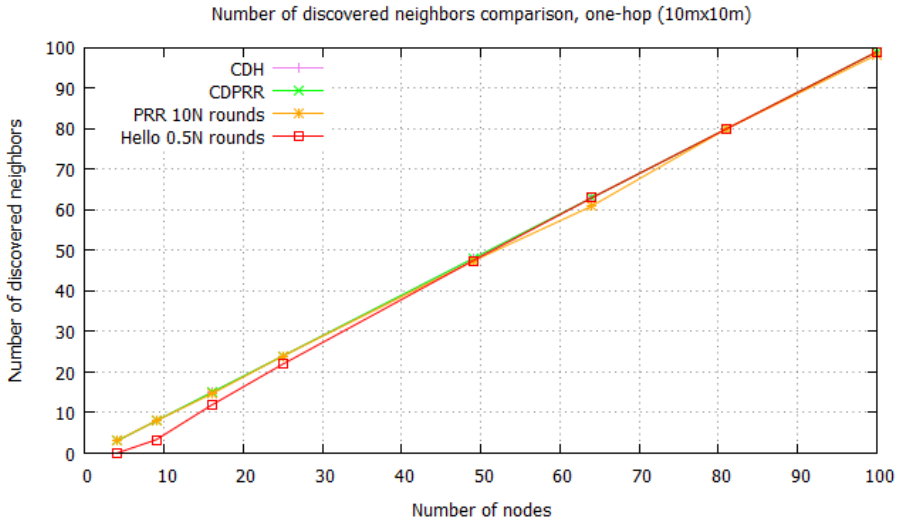

Figura 4.11: Número de vecinos descubiertos (one-hop)

A continuación, se muestran los resultados del número de vecinos descubiertos en un entorno *multi-hop*, obteniendo resultados para distintos modelos de colisión. Para el modelo sin colisiones, todos los protocolos presentan resultados óptimos, logrando descubrir todos los vecinos, ya que no hay colisiones. La Figura 4.12 muestra los resultados para el modelo de interferencia aditiva. El parámetro *collisionModel* de Castalia 3.2 ha sido fijado a 2, el modelo más realista. CDH y CDPRR logran descubrir todos los vecinos y mejoran las otras soluciones. Hello con 0.25N *rounds* presenta unas prestaciones ligeramente peores que CDH y CDPRR. Finalmente, PRR con 6N *rounds* es el peor. Un punto interesante es que PRR presenta peores prestaciones para redes extensas, como se indicó en [21]. Además, para un número de nodos menor que 10, ninguno de los nodos descubre ningún vecino dado que todos los nodos están fuera del rango de transmisión de todos los demás.

De nuevo, concluimos que los resultados de numero de vecinos descubiertos son similares para los modelos de colisión 1 y 2.

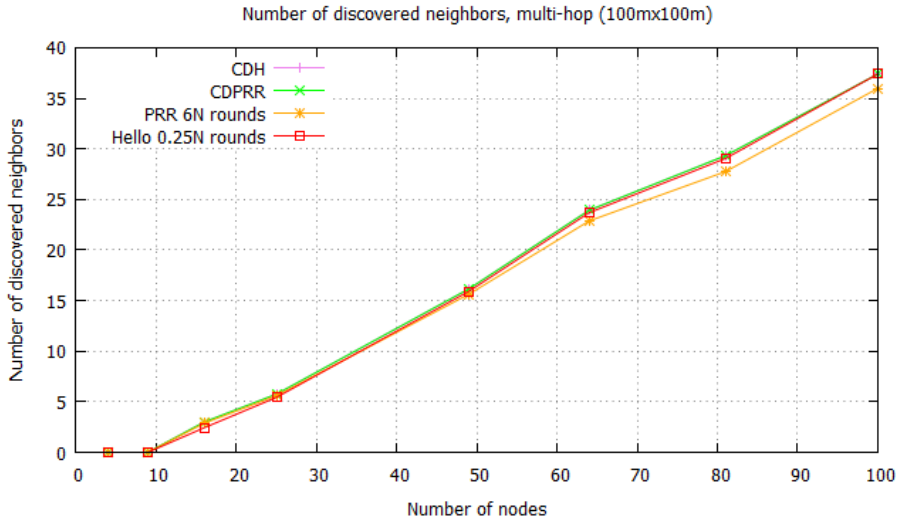

Figura 4.12: Número de vecinos descubiertos (multi-hop)

4.6.3 Consumo energético

En primer lugar, se muestran y discuten los resultados de simulación en relación con el consumo energético para el caso *one-hop*.

Según la Figura 4.13, CDPRR mejora las otras 3 soluciones, CDH tiene mejores prestaciones que PRR con 10N *rounds*, y Hello con 0.5N *rounds* es el que peores prestaciones presenta. Los 4 protocolos presentan una tendencia creciente con el número de nodos. Esta tendencia se debe al incremento del Tiempo de descubrimiento de vecinos que se muestra en la Figura 4.9.

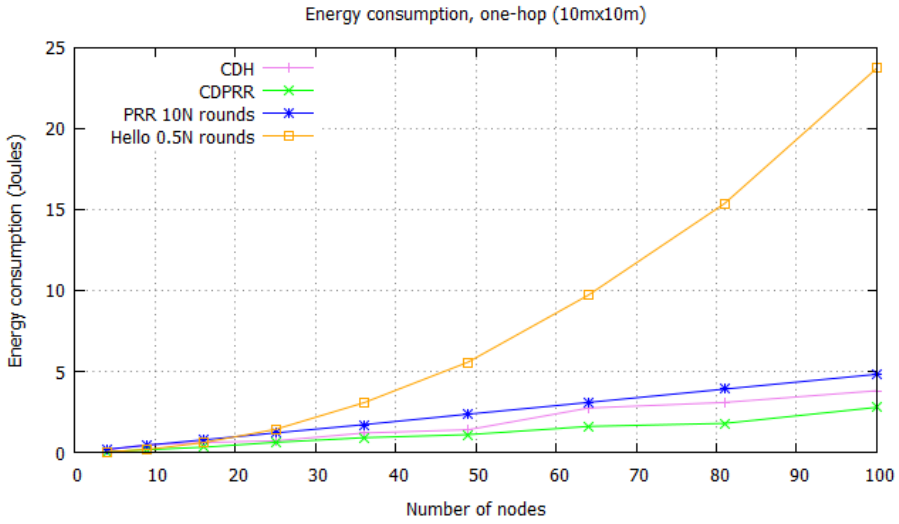

Figura 4.13: Consumo energético (one-hop)

A continuación, se presentan los resultados en relación al consumo energético para un escenario *multi-hop*. Como se muestra en la Figura 4.14, CDPRR presenta mejores prestaciones que los otros 3 protocolos. CDH mejora al PRR con 6N *rounds*, y Hello con 0.25N *rounds* es el peor. De nuevo, los 4 protocols siguen una tendencia creciente con el número de nodos en relación con el consumo energético. Esto es debido a que el Tiempo de descubrimiento se incrementa como se puede observar en la Figura 4.10.

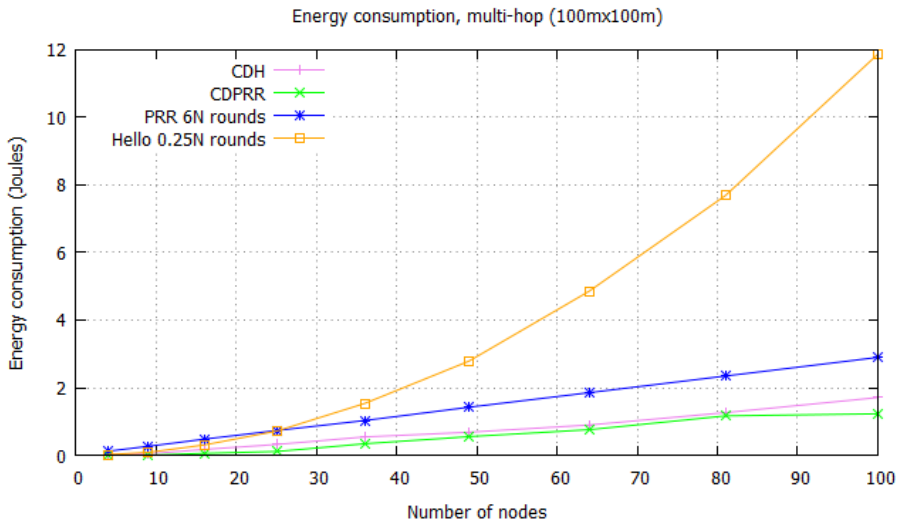

Figura 4.14: Consumo energético (multi-hop)

4.6.4 Throughput

En primer lugar, se presentan los resultados en relación con el *Throughput* para el caso *one-hop*. Según la Figura 4.15, CDPRR presenta los mejores resultados, con prestaciones similares al CDH cuando el número de nodos es superior a 50. CDH es mejor que Hello con 0.5N *rounds*, y PRR con 10N *rounds* presenta las peores prestaciones. Sin embargo, Hello con 0.5N *rounds* y PRR con 10N *rounds* presentan prestaciones similares cuando el número de nodos está por encima de 30. La tendencia es decreciente con el número de nodos para todos los protocolos. Esto se debe a un aumento en el número de colisiones dado que a medida que el número de nodos crece menos paquetes son recibidos por segundo. En el caso particular de CDH y CDPRR, el *throughput* es mayor que el de Hello y PRR porque, a medida que los *rounds* pasan, hay menos colisiones y por tanto más paquetes son recibidos.

Seguidamente, se muestran los resultados obtenidos en relación con el *throughput* en el caso *multi-hop*. Según la Figura 4.16, cuando el número de nodos es mayor que 15, CDH mejora las otras soluciones. Además, por encima de 35 nodos, CDPRR es mejor que Hello con 0.25N *rounds*, y PRR con 6N *rounds* presenta las peores prestaciones. Un punto interesante es que, en redes compuestas por menos de 9 nodos tienen un *throughput* de 0 byte/s ya que todos los nodos están fuera del rango de transmisión de los demás. La tendencia de-

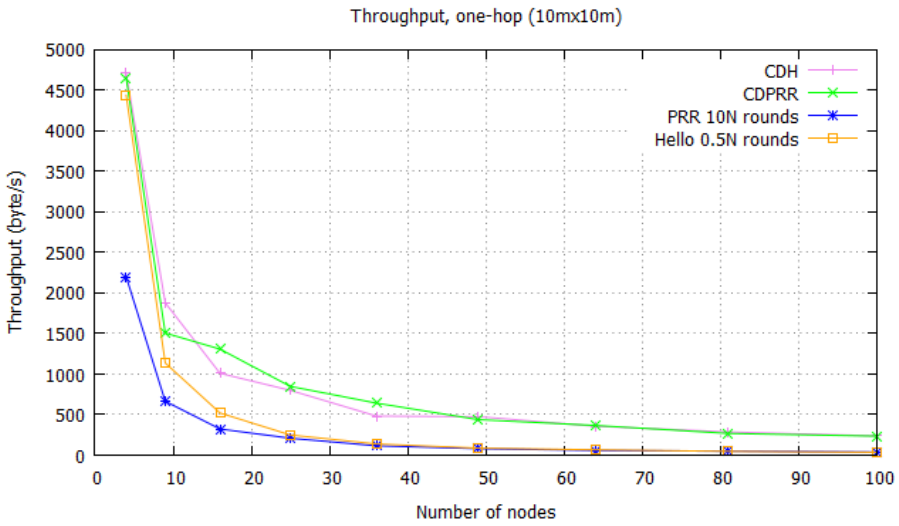

Figura 4.15: Throughput (one-hop)

creciente es debida al aumento de colisiones, provocando que menos paquetes lleguen por segundo. De nuevo, se concluye lo mismo que en el caso *one-hop*. En CDH y CDPRR el *throughput* es mayor que el de Hello y PRR debido a que, a medida que los *rounds* pasan hay menos colisiones y por tanto más paquetes son recibidos.

4.6.5 Número de vecinos descubiertos por paquetes enviados

En primer lugar, presentamos los resultados en relación con el *ratio* número de vecinos descubiertos por paquetes enviados para el caso *one-hop*.

Según la Figura 4.17, CDPRR mejora los demás protocolos, mientras que CDH presenta mejores prestaciones que PRR con 10N *rounds* a partir de 15 nodos, y Hello con 0.5N *rounds* es el peor.

Este comportamiento es debido a que a medida que el tiempo de descubrimiento se reduce en CDH y CDPRR, hay menos paquetes enviados para un número similar de vecinos descubiertos. Por lo tanto, el *ratio* número de vecinos descubiertos por paquetes enviados es mayor que el de los protocolos de referencia.

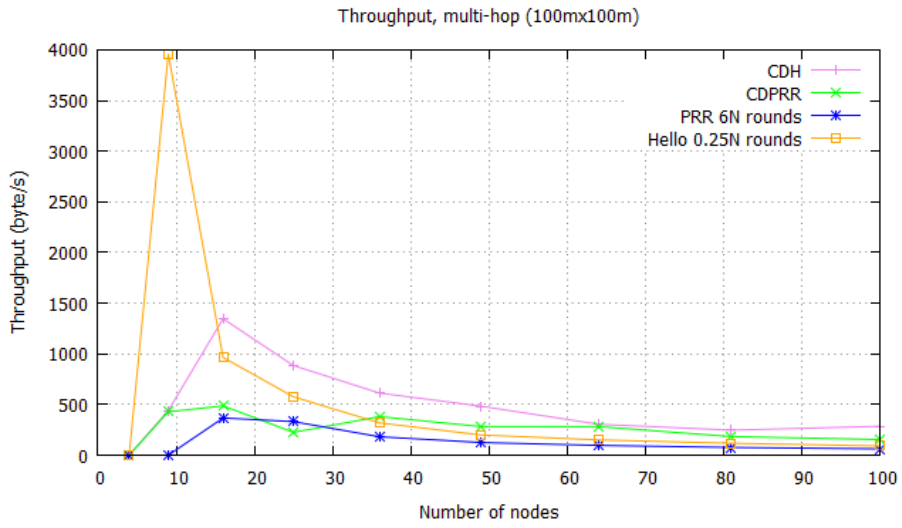

Figura 4.16: Throughput (multi-hop)

Seguidamente, se presentan los resultados en relación con el *ratio* número de vecinos descubiertos por paquetes enviados en el caso *multi-hop*. Según la Figura 4.18, CDPRR supera las otras 3 soluciones, mientras que CDH es mejor que PRR con 6N *rounds* y Hello con 0.25N *rounds* es el peor.

De nuevo, para un número de nodos menor que 10 el resultado es 0 byte/s dado que todos los nodos están fuera del rango de transmisión de todos los demás. Como en el caso *one-hop*, en CDH y CDPRR debido a que el tiempo de descubrimiento es menor, menos paquetes son enviados para similar número de vecinos descubiertos. Por lo tanto, el *ratio* número de vecinos descubiertos por paquetes enviados es mayor que el de los protocolos de referencia.

4.7 Discusión

Las dos propuestas operan siguiendo premisas más realistas, permitiendo la detección de colisiones y terminación, descubren todos los vecinos con probabilidad 1. En el caso particular de CDH, permite a los nodos iniciar la transmisión en cualquier instante de tiempo, y que el número de nodos sea desconocido.

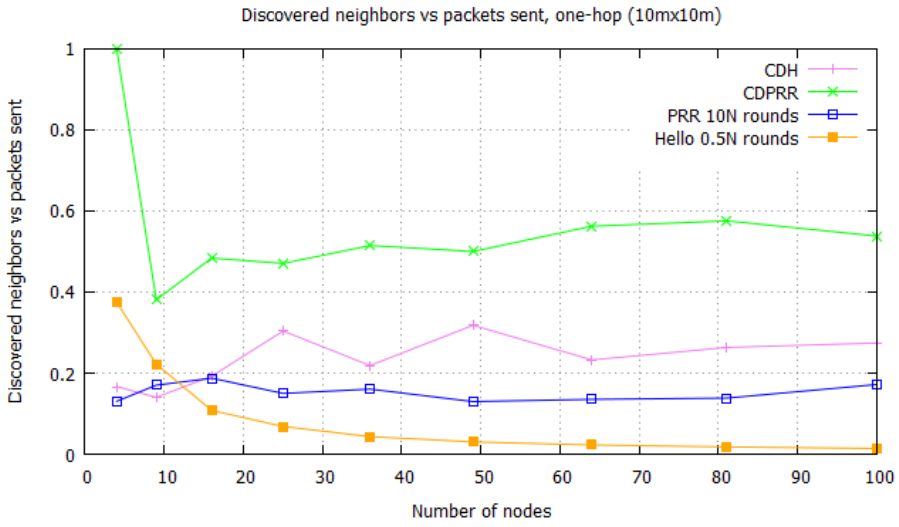

Figura 4.17: Ratio vecinos descubiertos por paquetes enviados (one-hop)

En CDH y CDPRR, el tiempo de descubrimiento tanto en *one-hop* como en *multi-hop*, es lineal $O(N)$, donde N es el número de nodos de la red.

Las principales limitaciones del protocolo CDPRR son que necesita conocer el número de nodos de la red y la transmisión no puede iniciar en diferentes instantes de tiempo. Para resolver esas limitaciones, contamos con el protocolo CDH, que permite el desconocimiento del número de nodos de la red, e iniciar la transmisión en diferentes instantes de tiempo. Como desventajas, CDH y CDPRR requieren sincronización en los límites de la ranura, pueden ser usados únicamente en redes estáticas, esto es, no pueden ser usados en MANETs. El tiempo debe ser ranurado y no hay detección de pérdida de paquetes.

Como posibles formas de solucionar las limitaciones, tenemos que desarrollar o usar un mecanismo de sincronizacion existente antes del descubrimiento. También hay que adaptar los protocolos para permitir que nuevos vecinos lleguen y salgan de las MANETs.

Como aplicaciones prácticas, las propuestas permiten su uso en redes inalámbricas ad hoc estáticas en escenarios *multi-hop*. Ejemplo de ellas son las redes espontáneas, como una reunión de personas que se reúnen en un determinado lugar para intercambiar información.

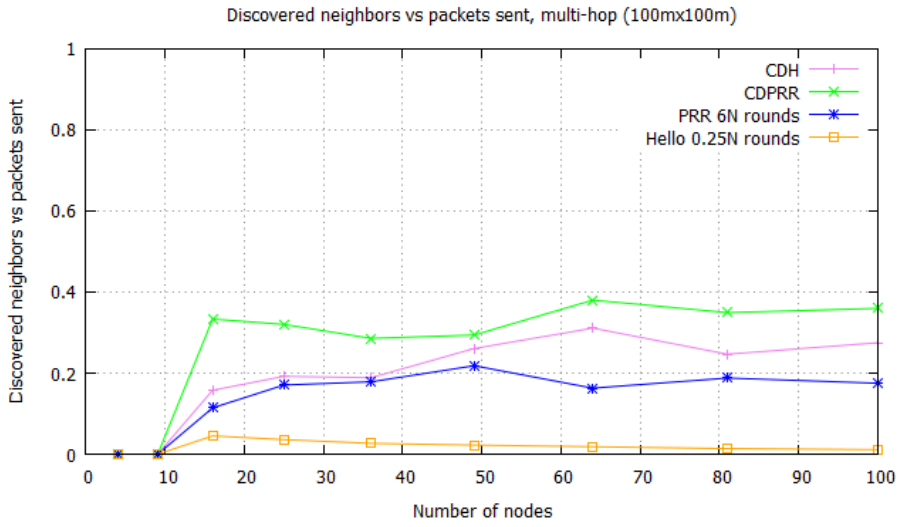

Figura 4.18: Ratio vecinos descubiertos por paquetes enviados (multi-hop)

4.8 Conclusiones

En este capítulo, se ha realizado un estudio de soluciones de descubrimiento de vecinos para redes estáticas *multi-hop* considerando la presencia de colisiones. Se han propuesto 2 soluciones aleatorias que toman las ventajas de la detección de colisiones, CDH y CDPRR. Se ha elegido 2 protocolos de la literatura y se han usado como referencia para el estudio. Estos protocolos son el Hello y el PRR. Los 4 protocolos se han implementado en Castalia 3.2 para su comparación.

Las simulaciones se han centrado en escenarios *one-hop* y *multi-hop*, y se han elegido varias métricas. Éstas son el Tiempo de descubrimiento de vecinos, el número de vecinos descubiertos, el Consumo energético, el *Throughput*, y el *Ratio* número de vecinos descubiertos por paquetes enviados.

Se concluye que CDPRR presenta mejores resultados en relación con el tiempo de descubrimiento, consumo enegético, y el *ratio* número de vecinos descubiertos por paquetes enviados que el CDH, Hello, y PRR. Esto se da tanto en *one-hop* como en *multi-hop*. CDH y CDPRR mejoran los protocolos de referencia en términos de número de vecinos descubiertos. En cuanto al *throughput*, CDPRR mejora los otros 3 protocolos en escenarios *one-hop*. CDH es mejor que las otras soluciones en un entorno *multi-hop*. En conclusión, las dos pro-

puestas mejoran los protocolos de referencia en relación con las 5 métricas en escenarios *one-hop* y *multi-hop*.

Como posibles trabajos futuros, se tiene pensado resolver las limitaciones enumeradas en la sección 4.7 de los protocolos CDH y CDPRR. Para ello, se requiere desarrollar un algoritmo que permita la sincronización en los límites de ranura. Además, se requiere adaptar los protocolos para su uso en MANETs permitiendo que nuevos nodos entren y salgan de la red. También se desea proponer y evaluar protocolos eficientes para redes móviles.

En el capítulo 6 se propone y evalúa un protocolo de bajo consumo energético.

Capítulo 5

Protocolo de descubrimiento de vecinos asíncrono basado en líder con detección de colisiones

En este capítulo se presenta un novedoso protocolo aleatorio basado en líder para redes estáticas one-hop. El protocolo logra el descubrimiento de todos los vecinos con probabilidad 1, y utiliza detección de colisiones. También permite conocer cuando terminar el descubrimiento, esto es, cuando todos los nodos han sido descubiertos, sigue premisas más realistas. Para comparar el protocolo con el Hello de la literatura y un protocolo determinístico basado en líder elegidos como referencia, se utiliza el simulador Castalia 3.2. La propuesta presenta mejores resultados que el Hello según Tiempo de descubrimiento de vecinos, Número de vecinos descubiertos, Consumo energético y Throughput. Por otro lado, la propuesta presenta resultados razonables en comparación con el protocolo determinístico basado en líder en relación a tiempos, consumo energético y throughput. Además, se trata de un protocolo asíncrono en el que un líder lanza el descubrimiento.

5.1 Introducción

Este capítulo centra el estudio en el problema del descubrimiento de vecinos en redes estáticas *one-hop* inalámbricas ad hoc. Se presenta un nuevo protocolo aleatorio basado en líder. Este protocolo soluciona el problema del descubrimiento en la presencia de colisiones. Una colisión es un fenómeno que ocurre cuando dos o más nodos intentan transmitir simultáneamente. En caso contrario, decimos que tuvo lugar un descubrimiento.

La propuesta permite detección de colisiones y terminación, a través de un líder se inicia el descubrimiento y sincroniza el resto de nodos. Se utiliza un mecanismo aleatorio para gestionar el envío de *feedbacks* por parte de los vecinos. El protocolo propuesto permite el descubrimiento de todos los vecinos con probabilidad 1. El protocolo diseñado funciona sólo en redes estáticas *one-hop*, y logra el descubrimiento sin necesidad de sincronización y sin necesidad de conocer el número de nodos de la red. El problema es que en el caso de que el líder falle, el descubrimiento deja de funcionar.

Para resumir, la propuesta detecta colisiones y terminación, usa un líder y un mecanismo aleatorio para enviar los *feedbacks* de los vecinos. Funciona adecuadamente en entornos estáticos *one-hop* inalámbricos ad hoc. El principal objetivo es solucionar el problema que tiene un protocolo previo basado en líder [92]. Este protocolo necesita sincronización y conocimiento a priori del número de nodos de la red. También tenemos como objetivo el descubrir todos los vecinos con probabilidad 1 en una cantidad de tiempo reducida. Una ventaja de la propuesta es que el líder inicia el descubrimiento y sincroniza al resto de nodos. No necesita de un procedimiento de sincronización externo. Como desventajas, la propuesta sólo funciona en redes *one-hop*, y si el líder falla el descubrimiento no funciona adecuadamente. Sin embargo, si el líder falla, los nodos conocen los vecinos descubiertos hasta ese momento.

Las principales contribuciones de este capítulo son: (i) Propuesta aleatoria basada en líder, en la detección de colisiones y Hello. Logra descubrir todos los vecinos con probabilidad 1, y termina cuando todos los vecinos han sido descubiertos. Sigue premisas más realistas, y permite su uso de forma asíncrona, lanzando el descubrimiento el nodo líder. El número de nodos puede ser desconocido. No requiere una planificación en la transmisión. Sin embargo, sólo funciona en entornos estáticos *one-hop*. (ii) Comparación cualitativa de la propuesta, Hello, y un protocolo basado en líder determinístico. (iii) Implementación en Castalia 3.2 de la propuesta y los 2 protocolos de referencia.

Se concluye que la propuesta tiene mejores resultados que el Hello según el tiempo de descubrimiento de vecinos, y número de vecinos descubiertos. También presenta mejoría en cuanto a consumo energético y *throughput*. Por otro lado, la propuesta presenta resultados razonables en comparación con el protocolo determinístico basado en líder en relación a tiempos, consumo energético y *throughput*.

5.2 Protocolos de referencia

Elegimos dos protocolos de la literatura para ser usados como referencia, un protocolo determinístico basado en líder [92], y el protocolo Hello [22].

En el protocolo determinístico basado en líder, un nodo especial conocido como líder inicia el descubrimiento mediante el envío de un paquete *BROADCAST* que contiene el identificador. El paquete llega a los nodos potenciales vecinos en su rango de transmisión. A continuación, los vecinos responden enviando un *ACK* con sus identificadores uno tras otro. Los nodos siguen una planificación en la transmisión para evitar colisiones. Cuando cada *ACK* llega al destino, éste guarda el identificador del vecino en su tabla de vecinos. En el momento en que todos los vecinos han enviado sus reconocimientos, el nodo líder difunde un paquete *BROADCAST* con la tabla de vecinos y el protocolo finaliza.

En el Hello, el tiempo está ranurado (*rounds*) y cada nodo envía un solo *BROADCAST* que contiene su identificador en cada *round* en un tiempo elegido de forma aleatoria. Se produce una colisión cuando los paquetes de dos o más nodos se solapan en el tiempo. En caso contrario, se produce una transmisión con éxito y los nodos guardan el identificador recibido en sus tablas de vecinos. Como el protocolo es *one-way*, no se sabrá cuándo se han descubierto todos los vecinos, por lo que se fija un número de *rounds* tras el cual el protocolo finaliza.

El protocolo determinístico basado en líder requiere que los nodos estén sincronizados. Además, siguen una planificación en la transmisión predeterminada para el envío de los *feedbacks* por parte de los vecinos. En cuanto al Hello, se debe fijar un número de *rounds* que sea suficientemente alto para descubrir una gran cantidad de vecinos, esto es, que la probabilidad de descubrimiento sea alta.

A continuación, se incluye la Tabla 5.1 con información sobre los protocolos de referencia y la propuesta. De acuerdo con ella, el protocolo determinístico basado en líder sólo puede ser usado en redes estáticas, y es determinístico. El

tiempo no está ranurado en *rounds*, requiere que los nodos estén sincronizados y una planificación en la transmisión predeterminada. El protocolo permite su funcionamiento en redes *one-hop* pero no es posible un funcionamiento adecuado en redes *multi-hop*. El protocolo logra el descubrimiento de todos los vecinos con probabilidad 1, no permite la detección de colisiones ni terminación. Además, no permite que los nodos inicien su transmisión en diferentes instantes de tiempo, y en el caso de que el nodo líder caiga, el protocolo deja de funcionar.

Por otro lado, Hello es un protocolo aleatorio, require sincronización en los límites de la ranura, puede ser usado con un funcionamiento adecuado en redes *multi-hop*. Hello no necesita la existencia de un nodo líder, es un protocolo *one-way* por lo que no detecta colisiones ni terminación. El protocolo logra descubrir todos los vecinos con alta probabilidad (pero distinta de 1), y si algunos nodos fallan el protocolo sigue funcionando.

Con el objetivo de solucionar los problemas que presentan los protocolos de referencia, principalmente la propuesta: (i) Permite su funcionamiento en redes estáticas. (ii) Es aleatorio. (iii) Es asíncrono. (iv) El número de nodos de la red puede ser desconocido. (v) El tiempo sigue estando ranurado. (vi) Permite su uso *half-duplex*, esto es, que los nodos transmitan o escuchen pero no simultáneamente. (vii) Sólo permite un uso adecuado en redes *one-hop*. (viii) Hace uso de la detección de colisión y terminación para mejorar el protocolo. (ix) Permite que los nodos inicien su transmisión en diferentes instantes de tiempo. (x) Hace uso de un líder para iniciar el descubrimiento y es un protocolo *two-way*. (x) Permite el descubrimiento de todos los vecinos con probabilidad 1.

Un posible entorno donde estos protocolos son aplicables es cualquier red inalámbrica ad hoc estática, en la que los transceptores de radio tengan un rango de transmisión elevado. Típicamente ese rango de transmisión puede llegar hasta 500m. Además, todos los nodos deben estar en rango de transmisión de todos los demás. Un ejemplo de aplicación son las redes espontáneas, como un encuentro de personas en una determinada localización y durante un periodo de tiempo. En cuanto al Hello, se puede usar en el caso de un escenario *multi-hop*.

Tabla 5.1: Comparación cualitativa de los 2 protocolos de referencia y la propuesta.

	[92]	Líder (aleatorio)	[22]
Red estática	✓	✓	✓
Red móvil			
Aleatorio		✓	✓
Tiempo ranurado		✓	✓
N conocido	✓		
Requiere sincronización	✓		
Sensores con batería	✓	✓	✓
Nodos coordinan acciones	✓	✓	
Transmitiendo o escuchando (pero no simultáneamente)	✓	✓	✓
One-hop	✓	✓	✓
Multi-hop			✓
Modo *Sleep* disponible			
Considera colisiones e interferencias	✓	✓	✓
Colisiones pierden transmisión			✓
Detección de pérdida de paquetes			
Líder necesario	✓	✓	
Detección de colisiones		✓	
Detección de terminación		✓	
Inicia transmisión en diferentes instantes de tiempo		✓	✓
Descubre todos los vecinos	✓	✓	
Con *feedback*	✓	✓	
Requiere gran número de ranuras			
Requiere N grande			
Protegido contra pérdida de paquetes		✓	✓
Si el líder falla, sigue funcionando			✓

5.3 Protocolo asíncrono basado en líder

En esta sección se presenta un protocolo de descubrimiento asíncrono basado en líder, cuyo uso se restringe a redes estáticas *one-hop*.

5.3.1 Premisas

En relación a la propuesta tendremos en cuenta las siguientes premisas.

- Aleatorización en los nodos, esto es, cada nodo puede transmitir en un tiempo elegido de forma aleatoria.
- Los nodos son estáticos, no permitiendo su movimiento en el área de despliegue.

- Cada nodo tiene un identificador único que le distingue de los demás, pudiendo ser la dirección MAC o el número de serie del fabricante.
- Los nodos se despliegan de forma aleatoria en un área dada.
- El tiempo está ranurado (*rounds*) en todos los nodos.
- Los nodos son asíncronos, el líder los sincroniza.
- El número de nodos de la red es desconocido por todos los nodos.
- Los nodos están equipados con transceptores de radio de alcance limitado.
- Todos los nodos tienen el mismo rango de transmisión.
- *Half-duplex*, esto es, los nodos pueden transmitir o recibir pero no simultáneamente.
- Cada nodo tiene una memoria interna para guardar la tabla de vecinos.
- Pueden existir colisiones.
- Se necesita un líder.
- Uso único en redes *one-hop*.
- Los nodos pueden detectar colisiones.
- Los nodos pueden detectar terminación.

5.3.2 Modelo

Según la Figura 5.1, el modelo de la propuesta consta de 4 pasos.

1. Un nodo elegido aleatoriamente como líder difunde un paquete *BROADCAST* que contiene su identificador hacia los nodos en su rango de transmisión, e inicia así el descubrimiento.
2. Cada vecino envía un paquete *UNICAST*, que lo llamaremos como *feedback* y que contiene su identificador con destino el nodo líder.
3. Tras la llegada de todos los *feedbacks* por parte de los vecinos, el nodo líder ya ha realizado la detección de colisiones. Si no se detectó colisión en uno o más vecinos, es decir, los *feedbacks* llegaron con éxito, el líder guarda el identificador de esos nodos en su tabla de vecinos. Además guarda

el identificador en un array de éxito *success*. Al finalizar el proceso de detección de colisiones el líder comienza un mecanismo de detección de terminación. El líder comprueba si todos los nodos enviaron sus *feedback* con éxito en el presente o en *rounds* previos.

4. Si no se detectó terminación, el nodo líder difunde un paquete *BROADCAST* con el array *success* indicando qué nodos emitieron con éxito su *feedback* en este *round*, y salta al paso 2. En caso contario, el líder envía un *BROADCAST* que contiene la tabla de vecinos y el protocolo finaliza.

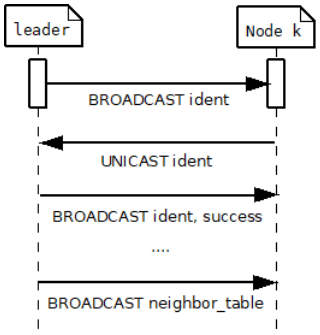

Figura 5.1: Propuesta asíncrona basada en líder

En primer lugar, el líder difunde un paquete *BROADCAST* que contiene su identificador. A continuación, los nodos en su rango de transmisión (vecinos) que lo reciben enviarán un solo paquete *UNICAST* como *feedback* en cada *round*, dado que el tiempo está ranurado en *rounds*.

Como se observa en la Figura 5.2, el tiempo está ranurado en *rounds* de tamaño ω, y cada vecino puede estar en uno de 2 posibles modos, que llamamos *sent* o *not sent*. En el primero de ellos, el vecino ya ha logrado emitir con éxito su identificador hacia el líder, esto es, sin producir una colisión en *rounds* previos. Este hecho se indica mediante una marca X roja en la Figura 5.2, de ahora en adelante no se emitirá ningún *feedback* y el nodo permanecerá en estado de escucha. En caso contrario, esto es, en el modo *not sent*, el nodo envía un solo paquete *UNICAST* hacia el líder en cada *round* que contiene su identificador. Este envío se realiza en un tiempo elegido de forma aleatoria t_i tal que $0 \leq t_i \leq \omega - \tau$ durante un tiempo τ. El nodo escucha el resto de la ranura durante un tiempo total $\omega - \tau$.

El nodo líder recibirá los *feedbacks* emitidos por el resto de nodos que no colisionaron en su rango de transmisión. Cuando el *round* finalice, el líder decidirá

si hubo colisión o no y qué paquetes la provocaron. Decimos que una colisión fue detectada si los paquetes *UNICAST* de dos o más vecinos se solapan en el tiempo. En caso contario, el paquete *UNICAST* de un vecino no colisiona con el del resto de los vecinos, decimos que ese vecino ha emitido un *feedback* con éxito. En ese caso, el líder guardará los identificadores de esos nodos vecinos que no provocaron colisión en un array *success*. Además, guardará los identificadores de esos vecinos en su tabla de vecinos local. Como se muestra en la Figura 5.2 finalmente el líder envía un paquete *BROADCAST* que contiene su identificador y el *array success* hacia el resto de nodos en su rango de transmisión. Ese *array* contendrá los identificadores de todos los nodos que emitieron *feedback* con éxito en ese *round*. En la Figura 5.2 ese *BROADCAST* se representa con un cuadrado gris tras el periodo ω. Cuando llegue ese *BROADCAST* cada vecino consultará en el *array* si está su identificador y así sabrá si ha logrado emitir el *feedback* con éxito o no en ese *round*. Si emitió con éxito, su estado cambiará a *sent* y a partir de ese momento en los *rounds* siguientes permanecerá escuchando. Si se produjo una colisión los nodos implicados en esa colisión continúan compitiendo en el siguiente *round*.

El protocolo incluye un mecanismo para detectar terminación, es decir, determinar cuando el protocolo concluye cuando todos los vecinos han sido descubiertos. Según este mecanismo, el líder es el que determina que todos los nodos en rango de transmisión tienen el estado *sent*. El líder comprueba si todos los vecinos han logrado transmitir con éxito sin que el *feedback* provoque colisiones. Los *feedbacks* contienen el identificador del vecino y es enviado hacia el líder. En el caso de que todos los vecinos hayan logrado enviar con éxito, el mecanismo de envío de los *feedbacks* finaliza. Luego, el líder procede a difundir la tabla de vecinos que tiene almacenada en local. En la Figura 5.2, este hecho se representa con un cuadrado gris oscuro al finalizar la línea de tiempos. En el momento en que cada vecino recibe ese paquete *BROADCAST* con la tabla de vecinos, la guarda en su tabla de vecinos local y finaliza el protocolo. Después de esto, el líder sabrá que el protocolo ha finalizado, dado que, en el *round*, escucha el canal y descubre que no hay señal en el canal. Esto indica que ningún nodo ha emitido el *BROADCAST* porque ya están todos descubiertos, es decir, todos los nodos están en estado *state*.

La Figura 5.2 también es útil para describir un ejemplo del funcionamiento de los *feedbacks* en el protocolo. En el primer *round*, los paquetes de los 3 vecinos colisionan y por tanto todos los nodos continúan en el siguiente *round*. En el *round* 2 el paquete del nodo 1 no colisiona y el líder emite un *BROADCAST* con el *array success* conteniendo el identificador del nodo 1. Cuando lo recibe el nodo 1 cambia su estado a *sent* ya que logró enviar el *feedback* y desde

este momento se puede observar una marca *X* roja en la Figura 5.2. En el *round* 3 los paquetes de los nodos 2 y 3 colisionan. En el *round* 4 ambos nodos transmiten con éxito y el líder envía el paquete *BROADCAST* con el *array success* que contiene el identificador de los vecinos 2 y 3. En el *round* 5 el líder detecta terminación dado que todos los vecinos enviaron los *feedbacks* con éxito en *rounds* previos, y al escuchar el canal detecta que no hay señal. En la Figura 5.2 aparece una marca *X* roja para todos los vecinos. El líder procede a emitir un *BROADCAST* con la tabla de vecinos y finaliza. Tras recibir ese *BROADCAST* el resto de nodos guardan la tabla de vecinos en local y finalizan.

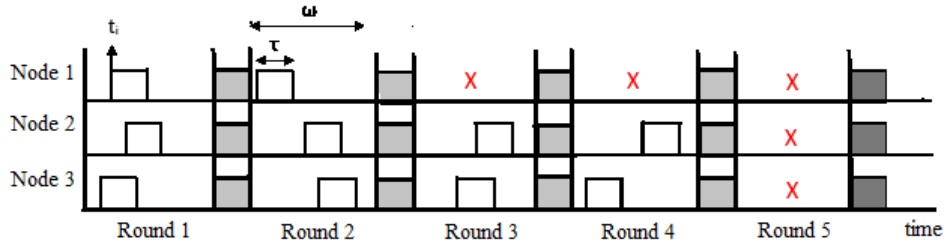

Figura 5.2: Feedbacks de los vecinos y respuestas del líder

5.4 Resultados de simulación

Esta sección tiene como objetivo mostrar y discutir las prestaciones de la propuesta asíncrona basada en líder. También se compara la propuesta con 2 protocolos elegidos de la literatura para ser usados como referencia. Esos 2 protocolos son: el protocolo determinístico basado en líder [92] y el protocolo Hello [22].

Para implementar y simular los 3 protocolos se ha usado Castalia 3.2 [88]. El simulador está basado en OMNET++ y se usa generalmente para simular WSN y BAN. Además, se ha demostrado que el simulador es útil permitiendo validar protocolos de descubrimiento de vecinos tanto en entornos estáticos como móviles (MANETs).

5.4.1 Escenario de simulación

El escenario de simulación usado para obtener los resultados es el mismo para los 3 protocolos a validar. Dado que una de las premisas es que pueden existir colisiones, se variarán los modelos de colisión. Para validar el comportamiento en relación con la escalabilidad se variarán el número de nodos que compone la red.

Para el Hello, se fija un número de *rounds*, dado que tras ese número de *rounds* el protocolo finalizará. Ese valor se ha fijado a 0.5N *rounds*. El tamaño de ranura (*round*) es fijado a $\omega = N \cdot \tau$, donde N es el número de nodos de la red, y el tiempo que transmite un nodo se ha fijado a $\tau = 0.07s$. En cuanto al modelo de radio usado es *ZigBee*, y Castalia 3.2 permite utilizar una librería de ese modelo, *ZigBee* (*CC2420*), para su simulación.

En cuanto al área de despliegue, se ha fijado a 10mx10m, correspondiendo a un entorno *one-hop*, i.e., todos los nodos están en rango de transmisión de todos los demás. Se organizan los N nodos en una malla MxM.

Dado que Castalia 3.2 permite el uso de diferentes modelos de colisión a través del parámetro *collisionModel*, para obtener resultados elegimos un modelo de colisión diferente. El parámetro *collisionModel* puede tomar los valores 0 (sin colisiones), 1 (modelo simplista para colisiones) o 2 (modelo de interferencia aditiva), siendo este último el más realista.

El objetivo principal de todo protocolo de descubrimiento de vecinos es descubrir todos los vecinos, o casi todos, en un tiempo reducido. Por ello, para las simulaciones se han elegido 2 métricas: *El Tiempo de descubrimiento de vecinos* y el *Número de vecinos descubiertos*. Además, dado que los dispositivos suelen estar alimentados con baterías que se agotarán en un tiempo determinado, se ha obtenido el *Consumo energético*. Finalmente, el *Throughput* también se ha obtenido.

Como se indicaba anteriormente, se ha usado el modelo de radio *ZigBee* (*CC2420*), se ha fijado la potencia de transmisión a 0dBm, la tasa de paquetes a 5 paquetes/s, y el tamaño de paquete a 2500 bytes.

Para fijar la duración de Hello, queremos averiguar si el Hello de referencia es peor que la propuesta según el *Tiempo de descubrimiento de vecinos* y el *Número de vecinos descubiertos* o es mejor en ambas métricas. Tras algunas simulaciones, se ha demostrado que el Hello es peor que la propuesta según ambas métricas si se fija la duración a 0.5N *rounds*.

Tabla 5.2: Parámetros de simulación.

Parámetro	Valor
Static	True
Modelo de radio	CC2420
Modelo de colisión	2
Potencia de transmisión	0dBm
Tasa de paquetes	5 paquetes/s
Tamaño de paquete	2500 bytes
Tamaño de ranura protocolo asíncrono y Hello	$\omega = N \cdot \tau$
τ	0.07s
Tamaño *one-hop*	10mx10m
Despliegue	Malla MxM
Número de *rounds* Hello *one-hop*	0.5N

En la Tabla 5.2 se resumen los parámetros de simulación fijados para llevar a cabo las simulaciones.

5.4.2 Resultados

Esta sección tiene como objetivo presentar los resultados de simulación obtenidos para los 3 protocolos a evaluar y comparar, en un escenario estático *one-hop*.

Tiempo de descubrimiento de vecinos

Cuando usamos Tiempo de descubrimiento de vecinos, nos referimos al tiempo que tarda un protocolo en finalizar.

La Figura 5.3, muestra los resultados obtenidos según esta métrica. Permite concluir que la propuesta aleatoria basada en líder es mejor que el protocolo Hello con 0.5N *rounds*. El tiempo de descubrimiento de vecinos sigue una tendencia creciente con el número de nodos. Además, la propuesta presenta peores resultados que el protocolo determinístico basado en líder. Sin embargo, la propuesta no requiere síncronización dado que el líder lanza el descubrimiento. Así, el comportamiento del resto de nodos es asíncrono. Resaltar que para una red de 100 nodos, el tiempo resultante para la propuesta es de alrededor de 60 segundos. Éste es un resultado razonable. El protocolo determinístico basado en líder en una red de 100 nodos emplea solo 7 segundos. En cuanto al modelo de colisión usado para obtener la Figura 5.3 es el más realista, esto es,

el parámetro *collisionModel* se ha fijado a 2. Sin embargo, se puede demostrar que idénticos resultados se obtienen para los otros dos modelos de colisión.

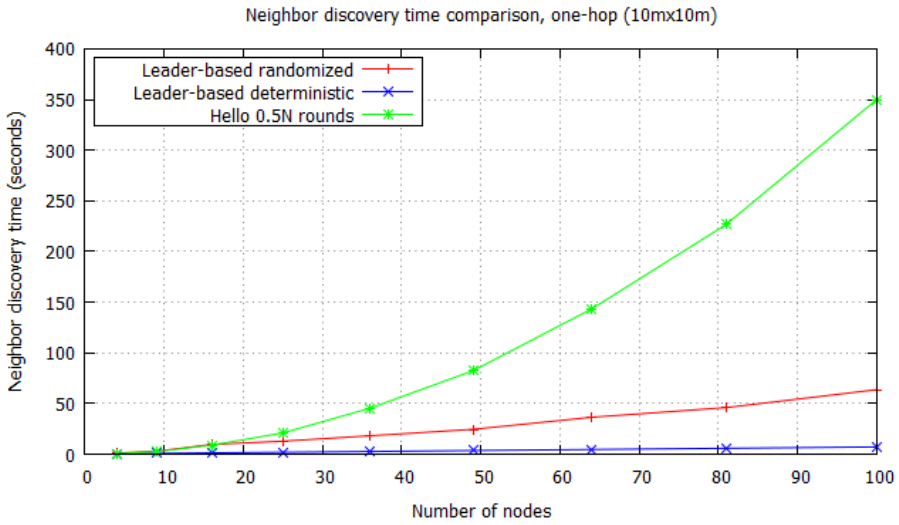

Figura 5.3: Tiempo de descubrimiento de vecinos

Número de vecinos descubiertos

Según la Figura 5.4, para el modelo sin colisiones (parámetro *collisionModel* 0), todos los protocolos presentan el comportamiento ideal, esto es, todos los nodos logran descubrir todos sus vecinos. Sin embargo, este modelo de colisión no es demasiado realista, por tanto procedemos a presentar los resultados usando modelos de colisión más realistas.

En la Figura 5.5 se muestran los resultados para el modelo simplista para colisiones (*collisionModel* fijado a 1). Ambos protocolos basados en líder logran descubrir todos los vecinos, y presentan mejor comportamiento que el Hello con 0.5N *rounds*. Este último no logra descubrir todos los vecinos cuando la red se compone de menos de 50 nodos.

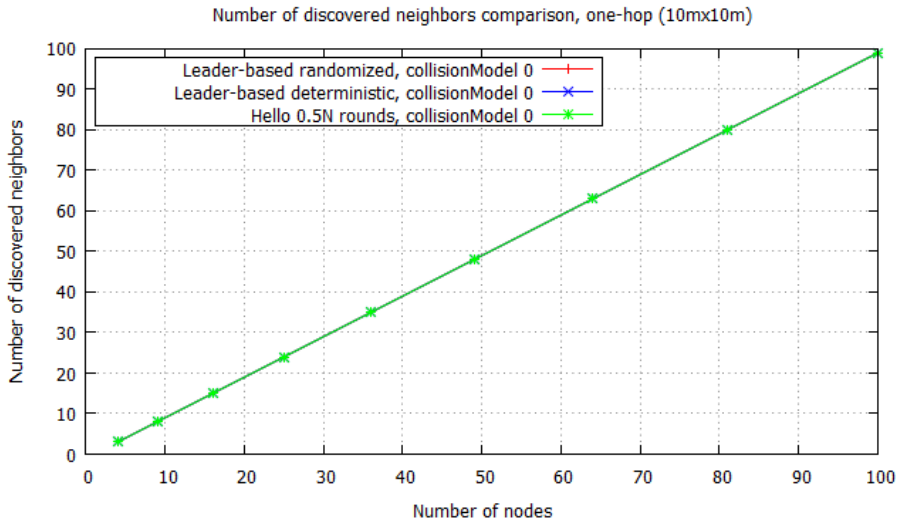

Figura 5.4: Número de vecinos descubiertos (collisionModel 0)

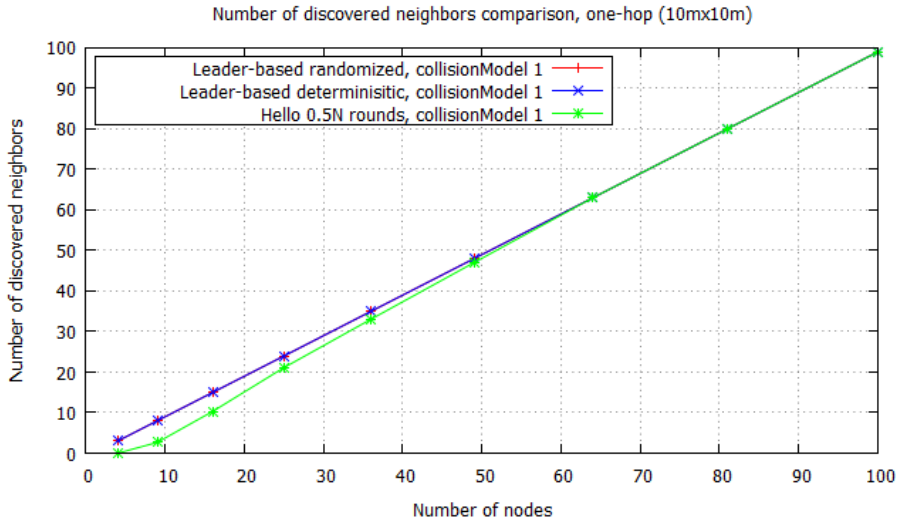

Figura 5.5: Número de vecinos descubiertos (collisionModel 1)

Por último, la Figura 5.6, se ha obtenido para el modelo de interferencia aditiva (*collisionModel* 2), el modelo de colisión más realista. Permite concluir que el comportamiento es el mismo que para el modelo simplista para colisiones (collisionModel 1). Ambos protocolos basados en líder presentan un comportamiento ideal mejor que el de Hello con 0.5N *rounds*, que no logra descubrir todos los vecinos cuando la red tiene menos de 40 nodos.

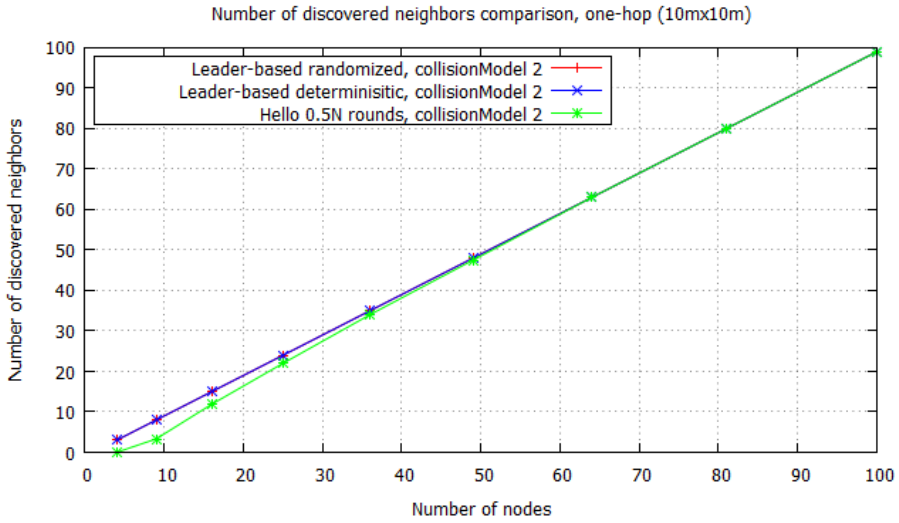

Figura 5.6: Número de vecinos descubiertos (collisionModel 2)

Consumo energético

Según la Figura 5.7, el protocolo determinístico basado en líder es mejor que las otras dos soluciones de acuerdo al consumo energético. La propuesta presenta resultados razonables y Hello es el peor. Los 3 protocolos siguen una tendencia creciente con el número de nodos, que es similar a la del Tiempo de descubrimiento de vecinos en la Figura 5.3.

Resaltar que el consumo energético de la propuesta en una red de 100 nodos es de aproximadamente 4 Julios. Para el protocolo determinístico basado en líder en una red de 100 nodos es de alrededor de 0.5 Julios. En este caso también se ha usado el modelo de colisión más realista (*collisionModel* 2).

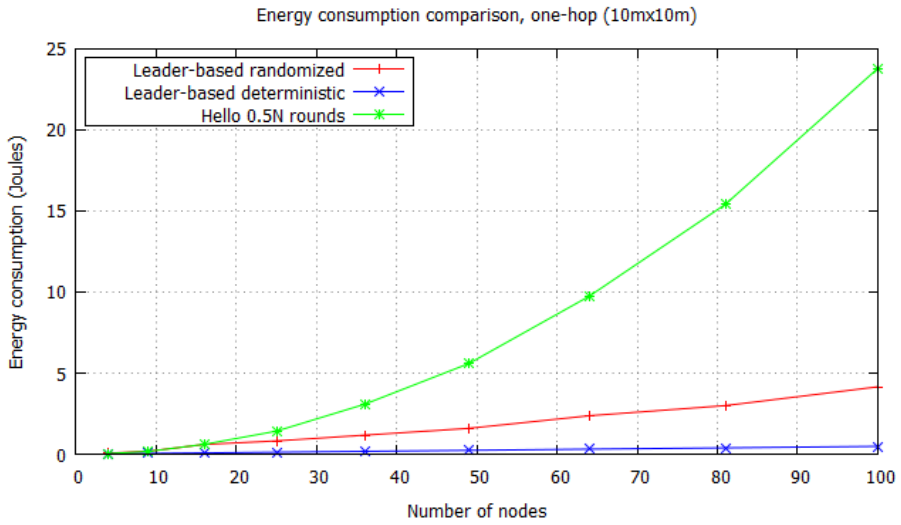

Figura 5.7: Consumo energético

Throughput

En relación al *throughput*, según se muestra en la Figura 5.8, el protocolo determínistico es mejor que la propuesta, que presenta resultados razonables, y el Hello es el peor. Sin embargo, la propuesta funciona de forma asíncrona. Las 3 soluciones siguen una tendencia decreciente con el número de nodos dado que cuando la red crece menos paquetes llegan a su destino.

Más en concreto, nuestra propuesta en una red de 100 nodos presenta un *throughput* de aproximadamente 664 byte/s. Para el protocolo determinístico en una red de 100 nodos el *throughput* es de aproximadamente 1036 byte/s.

De nuevo, se ha usado el modelo de interferencia aditiva (*collisionModel* 2).

5.5 Conclusiones

Este capítulo se ha centrado en redes *one-hop* estáticas en presencia de colisiones, y se ha presentado una solución asíncrona basada en líder.

Se han implementado en Castalia 3.2 para su comparación, la propuesta, el protocolo Hello y el protocolo determinístico basado en líder. Ambos protocolos

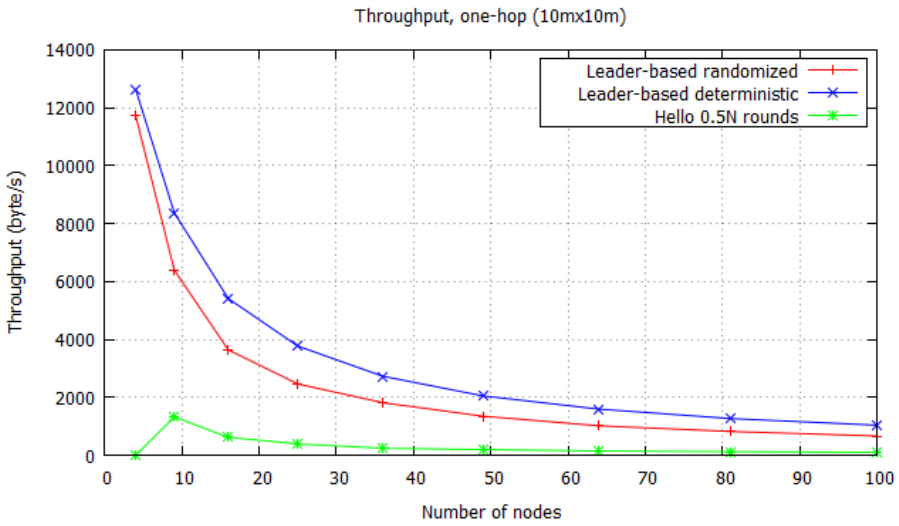

Figura 5.8: Throughput

han sido elegidos de la literatura y con el objetivo de usarlos como referencia. Se han elegido 4 métricas para evaluar los protocolos, esto es, tiempo de descubrimiento de vecinos, número de vecinos descubiertos, consumo energético y *throughput*.

Según los resultados de simulación obtenidos, la propuesta logra descubrir todos los vecinos y es mejor que el Hello según las 4 métricas. La propuesta presenta peores resultados que el protocolo determinístico basado en líder según tiempo de descubrimiento de vecinos, consumo energético y *throughput*. Sin embargo, la propuesta se puede usar de forma asíncrona.

Los protocolos presentan ventajas e inconvenientes. Ambos protocolos basados en líder logran descubrir todos los vecinos, pueden ser usados en redes *one-hop*, aunque su uso no es viable en escenarios *multi-hop*. El líder permite lanzar el descubrimiento del resto de nodos. La propuesta sigue premisas más realistas, permite detectar colisiones y terminación y su uso de forma asíncrona. El protocolo determinístico basado en líder requiere sincronización y una planificación en la transmisión predeterminada. La principal desventaja de los protocolos basados en líder es que si el líder cae en un momento determinado el protocolo de descubrimiento deja de funcionar. Además, si el *BROADCAST* con la tabla de vecinos se pierde los vecinos no lograrán haber descubierto los vecinos. Una mejora sería que el líder emitiera la tabla de vecinos con varios

paquetes de *UNICAST*, uno para cada vecino como destino. En cuanto al protocolo Hello se ha demostrado que es más lento y no logra descubrir todos los vecinos con probabilidad 1. Sin embargo, es completamente asíncrono, si un nodo cae el protocolo de descubrimiento sigue funcionando, y permite su uso en escenarios *multi-hop*.

Como posibles mejoras, el protocolo se puede mejorar con un mecanismo de bajo consumo para redes inalámbricas ad hoc como el propuesto en el capítulo 6. También se puede incluir ese protocolo en un modelo para la creación de redes espontáneas basadas en la confianza.

Capítulo 6

Protocolo de descubrimiento de vecinos aleatorio consciente de la energía basado en detección de colisiones

En este capítulo se presenta LECDH (Low Energy Collision Detection Hello), un protocolo de descubrimiento de vecinos aleatorio. Es consciente de la energía y está basado en la detección de colisiones. Está diseñado para su uso en entornos ad hoc estáticos one-hop y multi-hop. Se valida y compara el protocolo con una solución de la literatura EAH (Energy Aware Hello) usada como referencia. Para ello se realizaron simulaciones con Castalia 3.2. Se tienen en cuenta cinco métricas: consumo energético, tiempo de descubrimiento, número de vecinos descubiertos, throughput, y número de descubrimientos por paquetes enviados. Concluimos que la propuesta mejora al protocolo de referencia en las 5 métricas tanto en entornos one-hop como multi-hop para altos duty cycles. Además, para bajo número de nodos en LECDH a medida que el duty cycle se reduce las prestaciones son mejores según las 5 métricas en ambos entornos one-hop y multi-hop. En general, la propuesta sigue premisas más realistas, tales como no necesitar conocer el número de nodos y aún así proporciona un comportamiento adecuado. La propuesta permite a los nodos descubrir con éxito todos los vecinos con probabilidad casi 1. Es basado en handshake, esto es, basado en la detección de colisiones, y sabe cuándo terminar el descubrimiento. Además, este capítulo incluye una comparación cualitativa de la propuesta con la solución de referencia.

6.1 Introducción

La eficiencia energética es también un punto importante a tener en cuenta, dado que los dispositivos se alimentan de baterías que pueden durar una cantidad de tiempo determinada. Por este motivo, el protocolo presentado en este capítulo principalmente tiene como objetivo reducir el consumo energético.

Como novedad, la propuesta se enfrenta a las desventajas introducidas por trabajos anteriores y presenta un protocolo aleatorio consciente de la energía basado en *handshake*. En este protocolo no se usa ninguna planificación, trata con colisiones, funciona bajo premisas más realistas. También permite desconocer algunos parámetros de la red, y el protocolo está diseñado para entornos estáticos.

La propuesta LECDH se ha comparado con un protocolo existente: el EAH [22]. Para su comparación, la propuesta LECDH y el protocolo de referencia EAH han sido simulados con Castalia 3.2 [88].

El principal problema de EAH es que no logra descubrir todos los vecinos con probabilidad 1. Además los nodos desconocen cuándo terminar el descubrimiento, esto es, finaliza tras un número de *rounds* finito.

La principal motivación de este trabajo, aunque muchos trabajos previos se han centrado en mejorar el consumo energético, es extender el protocolo Hello. Para ello se usan mecanismos de detección de colisión y de terminación, y se tiene en cuenta la eficiencia energética. A continuación se comparan sus prestaciones con las de un protocolo existente, esto es, el protocolo EAH.

Este capítulo se centra en el descubrimiento de vecinos en el contexto de entornos inalámbricos ad hoc estáticos *one-hop* y *multi-hop*. Se presenta una propuesta LECDH aleatoria consciente de la energía en la presencia de colisiones.

Este protocolo tiene en cuenta la existencia de colisiones, detecta colisiones y terminación, requiere sincronización en los límites de ranura. Además, no se usa planificación en la transmisión, y el número de nodos es desconocido. Logra solucionar los problemas encontrados en otros protocolos aleatorios existentes. También tiene como objetivo reducir el consumo energético en comparación con soluciones existentes.

Las principales contribuciones de este capítulo son: (i) Propuesta LECDH, un protocolo aleatorio consciente de la energía basado en detección de colisiones que extiende el protocolo Hello. Presenta un tamaño fijo de ranura, descubre

todos los vecinos casi con probabilidad 1, y conoce cuándo terminar. Además no sigue una planificación en la transmisión, permite desconocer el número de nodos, y es apropiado para entornos tanto *one-hop* como *multi-hop*. (ii) Una comparación cualitativa del protocolo EAH y la propuesta. (iii) Una implementación en Castalia 3.2 y comparación de prestaciones de la propuesta y el protocolo EAH, para diferentes *duty cycles*.

6.2 LECDH

En esta sección se presenta LECDH, un protocolo aleatorio consciente de la energía.

6.2.1 Premisas

Las premisas que hemos de considerar al proponer LECDH son las siguientes:

- El tiempo está dividido en ranuras.
- Todos los nodos conocen el tamaño de la ranura.
- Los nodos no se pueden mover en el área de despliegue.
- Cada nodo tiene un identificador único, que le permite distinguirse de otros.
- Los identificadores no necesitan ser números consecutivos.
- Cada nodo debe conocer un identificador mínimo (ident_min) y un identificador máximo (ident_max). Ambos tienen el mismo valor para todos los nodos. Representan el mínimo y máximo identificador de todos los nodos posibles en la red. Los identificadores de los nodos deben estar en ese rango (entre ident_min e ident_max). Además, no es necesario que estén en la red desplegada todos los nodos ni los nodos con identificador ident_min ni ident_max.
- Los nodos son desplegados de forma aleatoria en un área.
- Se requiere sincronización en los límites de ranura.
- El número de nodos no es conocido por ningún nodo.

- Cada nodo tiene un transceptor de radio con un rango de transmisión limitado, que permite a los nodos transmitir o recibir pero no al mismo tiempo (*half-duplex*).
- Los transceptores de todos los nodos tienen idéntico rango de transmisión.
- Cada nodo incluye una memoria para guardar información topológica local (tabla de vecinos).
- Pueden aparecer colisiones.
- Los nodos pueden detectar colisiones y terminación.
- Los nodos pueden detectar energía.
- El protocolo debe ser basado en *handshake*.
- Los nodos deben poder iniciar la transmisión en diferentes instantes de tiempo.
- Cada nodo está alimentado por baterías.
- No se usa planificación en la transmisión.

El protocolo LECDH tiene como objetivo reducir el consumo energético en comparación con soluciones existentes.

6.2.2 Modelo

En LECDH el tiempo está ranurado en *rounds* de tamaño ω_t como se muestra en la Figura 6.1. Cada nodo puede estar en estados *Transmit*, *Listen*, *Sleep* o *Success*, como se muestra en la Figura 6.2. Esta Figura representa la máquina de estados de LECDH, que muestra como funciona el protocolo. El estado *Success* significa que el nodo ya transmitió con éxito y permanece en este estado hasta que el protocolo finaliza.

LECDH considera la existencia de colisiones, que se producen cuando los mensajes enviados por al menos dos nodos se solapan en el tiempo. Cuando sus mensajes no se solapan, decimos que el nodo logró transmitir con éxito. El tamaño de ranura ω_t tiene un valor fijo, no dependiendo del conocimiento del número de nodos N. Este valor se debe elegir adecuadamente ya que esta decisión afectará a las prestaciones del protocolo. Un valor bajo de ω_t puede producir una mejora en las prestaciones de redes pequeñas mientras que puede empeorar las prestaciones de redes grandes. En caso contrario, un valor grande

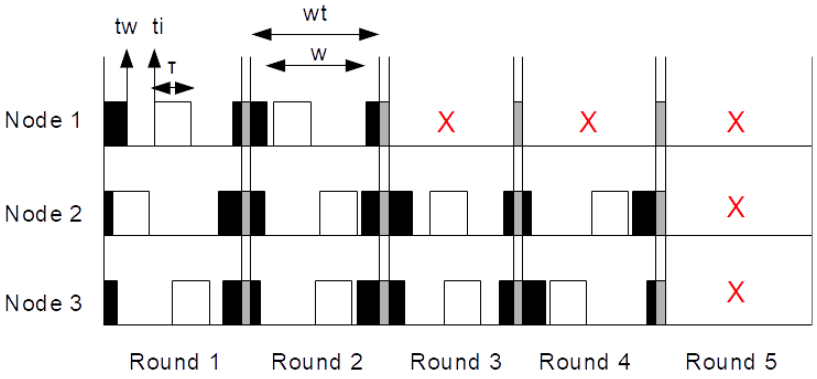

Figura 6.1: Protocolo LECDH (línea de tiempos).

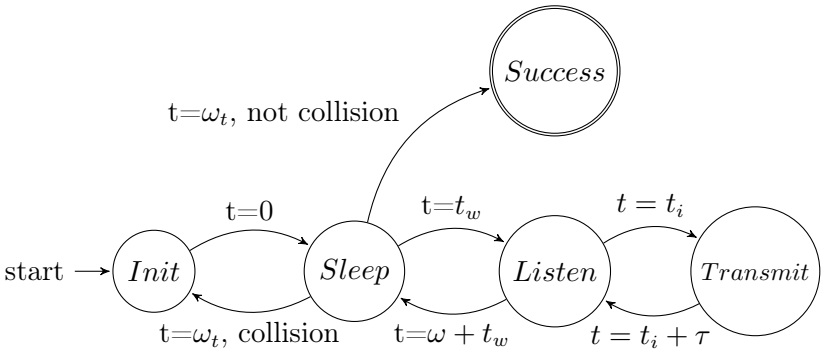

Figura 6.2: LECDH máquina de estados.

de ω_t puede producir una mejora en las prestaciones de redes grandes. En ese caso, las prestaciones pueden empeorar en redes pequeñas.

En primer lugar, como se muestra en la Figura 6.2, Figura 6.3 y Algoritmo 3, empieza el estado *Sleep*. El nodo se mantiene en este estado durante un tiempo elegido de forma aleatoria e independiente t_w de forma que $0 \leq t_w \leq s$. s es el tiempo total de *sleep* en un *round* y depende de ω_t y el DC (*Duty Cycle*), esto es, el porcentaje de tiempo que el nodo está activo. Durante t_w no se reciben mensajes dado que el nodo está dormido, y t_w es independiente entre nodos en un *round* dado e independiente entre *rounds* en un nodo dado. Esta situación se puede ver en la Figura 6.1 con cuadrados negros al principio y al final del *round*. La duración del *round* es ω_t mientras que el periodo activo que incluye escucha y transmisión tiene un tamaño de $\omega = \omega_t \cdot DC$. DC es el *duty cycle*,

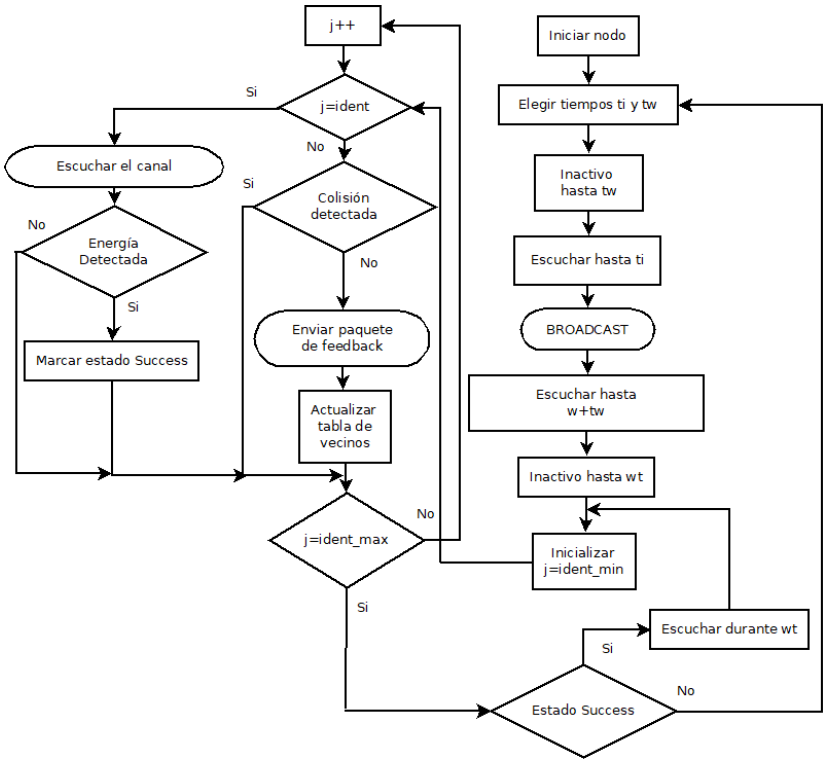

Figura 6.3: Diagrama de flujo LECDH.

y el tiempo total de *sleep* en un *round* es $s = \omega_t - \omega$. Un nodo se mantiene en el estado *Sleep* al principio del *round* durante un tiempo $t_w \in [0, s]$ y al final del *round* durante un tiempo $s_2 = s - t_w$.

Tras este periodo de *sleep* t_w, si un nodo no está en estado *Success* en ese *round*, el nodo se mantiene en estado *Listen* y elige de forma aleatoria un tiempo t_i, de forma que $t_w \leq t_i \leq t_w + \omega - \tau$.

A continuación el nodo envía un solo paquete *BROADCAST* en ese *round* iniciando en t_i durante un tiempo τ. Esto significa que este nodo está en estado *Transmit*, y luego permanece en estado *Listen*, escuchando mensajes durante $\omega - (t_i - t_w) - \tau$. Los paquetes *BROADCAST* deben incluir el identificador del emisor y será transmitido con éxito si durante la duración de transmisión todos los destinos están en estado *Listen*.

Algoritmo 3 LECDH

Entrada τ tiempo en que un nodo está transmitiendo, *DC duty cycle*, ω_t (duración de *round* fija), *ident* identificador

1: $\omega = \omega_t \times DC$
2: $s = \omega_t - \omega$
3: terminacion = false
4: **mientras** no terminacion **hacer**
5: Elige aleatoriamente $t_w \in [0, s]$ y comenzar estado *Sleep*.
6: Mantenerse en estado *Sleep* durante t_w segundos.
7: Elige aleatoriamente $ti \in [t_w, t_w + \omega - \tau]$
8: Mantenerse en estado *Listen* hasta t_i.
9: Enviar *BROADCAST(i)* iniciando en t_i durante τ, estado *Transmit*.
10: Mantenerse en estado *Listen* durante $\omega - (t_i - t_w) - \tau$.
11: Mantenerse en estado *Sleep* hasta el final del *round* (durante $s - t_w$)
12: **para** todos los j **hacer**
13: **si** j == ident **entonces**
14: Escucha el canal.
15: Realiza detección de energía.
16: **sino**
17: **si** nodo j transmitió con éxito **entonces**
18: Enviar paquete de *feedback*.
19: Actualiza tabla de vecinos con identificador j.
20: **fin si**
21: **fin si**
22: **si** j detectó energía **entonces**
23: Nodo j en estado *Success* a partir de este momento (cuando el proceso de *feedbacks* finaliza) y se mantiene escuchando hasta el final del protocolo, aunque enviará paquetes de *feedback* cuando sea necesario en los siguientes *rounds*.
24: **sino**
25: Nuevo *round* (cuando el proceso de *feedbacks* finaliza). Nodo j se mantiene compitiendo en el siguiente *round*.
26: **fin si**
27: **fin para**
28: **si** ningún *BROADCAST* fue recibido en un número de *rounds* consecutivos fijo **entonces**
29: terminacion = true
30: **fin si**
31: **fin mientras**

Finalmente, el nodo vuelve al estado *Sleep* y se mantiene en ese estado durante $s - t_w$.

En caso contrario, el nodo está en estado *Success* en ese *round*, circunstancia señalada con una marca roja X en la Figura 6.1. El nodo permanecerá en

este estado invariable a partir de ahora y se mantendrá escuchando paquetes *BROADCAST* de otros.

Además, se realiza detección de colisiones por todos los nodos cuando están escuchando. Si un nodo determinado logra transmitir con éxito, lo que significa que no hubo colisión para el nodo, el resto de nodos en su rango de transmisión actualizan sus tablas de vecinos con el identificador de este nodo. Una serie de paquetes de *feedback* serán enviados, en un segundo *sub-slot* de tamaño fijo ω_f, para indicar qué nodos transmitieron con éxito. El tiempo que un nodo está transmitiendo un paquete de *feedback* es τ_f. Los *feedbacks* se transmiten desde $ident\_min$ hasta $ident\_max$ (identificadores mínimo y máximo de los posibles nodos desplegados). Así, se permite ignorar el número de nodos que conforma la red. Esta situación se muestra en la Figura 6.1 con cuadrados gris claro tras el final del *round*.

Como se muestra en la Figura 6.3 los paquetes de *feedback* se enviarán uno tras otro desde el identificador $ident\_min$ hasta el identificador $ident\_max$. De acuerdo con la Figura 6.3 y el Algoritmo 3, los nodos enviarán el j_{th} paquete de *feedback* si sus identificadores no son iguales a j y el nodo j transmitió con éxito. Los nodos permanecen escuchando el canal si el identificador es igual a j. Un paquete de *feedback* en la j_{th} posición indica que el nodo cuyo identificador es j transmitió con éxito. La ausencia de paquete de *feedback* indica que el paquete de *BROADCAST* enviado por el nodo cuyo identificador es j colisionó.

Cuando el nodo con identificador j escucha el canal, procede a realizar detección de energía. Si se detecta energía, el nodo con identificador j, cambiará al estado *Success* al inicio del siguiente *round*. Ese nodo permanece en ese estado, y se mantiene escuchando hasta que el algoritmo finaliza. Esto significa que no competirá a partir de ese momento, aunque sigue enviando los paquetes de *feedback* cuando sea necesario. En caso contrario, tuvo lugar una colisión por el *BROADCAST* enviado por el nodo j, esto es, la j_{th} posición no incluye un paquete de *feedback*. El nodo j continuará compitiendo en el siguiente *round*.

El proceso de envío de paquetes de *feedback* también requiere que los nodos estén sincronizados en los límites de ranura. Los paquetes de *feedback* son enviados uno tras otro cuando sea necesario. Se requiere que todos los nodos envíen el j_{th} paquete de *feedback* simultaneamente. Así, se garantiza que los paquetes de *feedback* producirán detección de energía. Además, los paquetes de *feedback* son mucho más pequeños que los *BROADCASTs*.

Este protocolo también presenta un mecanismo de detección de terminación. Cada nodo averigua si todos los nodos en un escenario *one-hop* han logrado transmitir con éxito, lo que significa que todos los nodos están en el estado *Success*. Para este propósito, cada nodo debe descubrir la falta de existencia de señal en el canal durante un número de *rounds* consecutivos fijo. Esto significa que no se recibió ningún *BROADCAST*, dado que todos los nodos que están en estado *Success* no envían un *BROADCAST* en ese *round*. En este caso, se concluye que todos los nodos están en estado *Success* y el protocolo finaliza.

La Figura 6.1 también representa un ejemplo de operación del protocolo para una red compuesta por 3 nodos y todos los nodos están escenario *one-hop*. En el *round* 1, el mensaje del nodo 2 es enviado durante el periodo de *sleep* de los nodos 1 y 3. Los mensajes enviados por los nodos 1 y 3 se solapan en el tiempo provocando una colisión, por tanto los 3 nodos continúan compitiendo en el siguiente *round*. En el *round* 2, sólo el nodo 1 logra transmitir con éxito, por tanto no competirá a partir de este momento. Esto se indica con una marca X roja que aparece en los siguientes *rounds*. Los mensajes de los nodos 2 y 3 provocan una colisión, por tanto continúan compitiendo en el siguiente *round*. En el *round* 3 tiene lugar una colisión entre los nodos 2 y 3. En el *round* 4 ambos nodos restantes logran transmitir con éxito. En el *round* 5, todos los nodos ya han logrado transmitir con éxito, por tanto el protocolo finaliza.

El Algoritmo 3 muestra cómo funciona el protocolo LECDH, incluyendo las ecuaciones para obtener ω y s. Resalta las diferentes operaciones en cada estado, el proceso de detección de energía y los pasos a seguir cuando hay una transmisión con éxito o una colisión. Además, en el Algoritmo 3 se usa un valor fijo ω_t, que no depende de N, esto es, N permanece desconocido. En la línea 28, se comprueba la condición de terminación. La comprobación de la condición tiene lugar antes del proceso de detección de energía cuando no se recibe ningún *BROADCAST* en un fijo número de rounds consecutivos.

6.3 Comparación de prestaciones

En esta sección se presenta una comparación cualitativa de la propuesta con un protocolo de referencia, y el escenario de simulación.

6.3.1 Comparación cualitativa

En LECDH el tiempo está ranurado en *rounds* de tamaño ω_t como se muestra en la Figura 6.1, y cada nodo puede estar en estados *Transmit*, *Listen*, *Sleep* o *Success* como se muestra en la Figura 6.2. El estado *Success* significa que el nodo ya transmitió con éxito y permanece en este estado hasta que el protocolo finaliza. Esto se muestra en la Figura 6.2, la máquina de estados de LECDH que muestra cómo funciona el protocolo.

Se ha seleccionado un algoritmo aleatorio y se ha usado como referencia para su comparación con el protocolo EAH [22], dado que es similar a la propuesta LECDH.

EAH inicia con el estado *Sleep* y cada nodo permanece en él durante un tiempo aleatorio t_w tal que $0 \leq t_w \leq s$, donde es s el tiempo total de *sleep* en cada *round*. Durante este tiempo t_w los nodos no reciben mensajes. La duración del *round* es fija (ω_t) y el periodo activo que incluye escucha y transmisión presenta un tamaño de $\omega = \omega_t \cdot DC$. DC es el *duty cycle*, y $s = \omega_t - \omega$. Una vez concluido t_w, el nodo pasa al estado *Listen* (escucha). Se procede a elegir un tiempo aleatorio t_i, tal que $t_w \leq t_i \leq t_w + \omega - \tau$. Cuando llega el tiempo t_i, el nodo envía un paquete $BROADCAST$ en ese *round* durante un tiempo τ (estado *Transmit*). Luego vuelve al estado *Listen* en el que permanece durante $\omega - (t_i - t_w) - \tau$. Los paquetes $BROADCAST$ incluyen el identificador del emisor y se considera transmitido con éxito si durante la duración de transmisión todos los destinos están en estado *Listen*. Por último, el nodo vuelve al estado *Sleep* y se mantiene en él durante $s_2 = s - t_w$. Resaltar que EAH es un protocolo *one-way*, esto es, no incluye mecanismo de *feedback*.

La Tabla 6.1 resalta las principales características del protocolo de referencia y la propuesta. La Tabla 6.1 muestra que EAH es aleatorio, el tiempo está ranurado, sólo requiere que los nodos estén sincronizados en los límites de ranura. Puede ser usado tanto en entornos *one-hop* como *multi-hop*, y es un protocolo *one-way*, aunque no logra descubrir todos los vecinos con probabilidad 1. Por otra parte, la propuesta LECDH es aleatoria, el tiempo también está ranurado, también requiere sincronización en los límites de ranura. LECDH logra una probabilidad de descubrimiento casi 1, el número de nodos permanece desconocido por todos los nodos de la red. Su uso es posible tanto en entornos *one-hop* como *multi-hop*, y el modo *sleep* está disponible. LECDH permite detectar colisiones y terminación, permite a los nodos iniciar la transmisión en diferentes instantes de tiempo. Es un protocolo basado en *handshake*, esto es, incluye un mecanismo de *feedback*.

Tabla 6.1: Comparación cualitativa de EAH y LECDH.

	[22]	LECDH
Entorno estático	✓	✓
Entorno móvil		
Protocolo aleatorio	✓	✓
Tiempo ranurado	✓	✓
N permanece desconocido	✓	✓
Requiere sincronización en los límites de ranura	✓	✓
No sigue una planificación	✓	✓
Transmitiendo/escuchando (no simultaneamente)	✓	✓
Escenario *one-hop*	✓	✓
Escenario *multi-hop*	✓	✓
Modo *sleep* disponible	✓	✓
Se consideran colisiones	✓	✓
Detección pérdida de paquetes		
Líder necesario		
Detección de colisiones		✓
Detección de terminación		✓
Inicia transmisión en diferentes instantes de tiempo	✓	✓
Descubre todos los vecinos con probabilidad casi 1		✓
Con mecanismo de *feedback*		✓

6.3.2 Escenario de simulación

El mismo escenario se ha usado para ambos protocolos LECDH y EAH, variando el número de nodos N (escalabilidad), y también se ha variado el *DC*.

Hay muchos simuladores disponibles para comprobar las prestaciones de protocolos. Sin embargo, para llevar a cabo las simulaciones elegimos Castalia 3.2 [88], que está basado en OMNET++, ya que es apropiado para validar la propuesta.

Los resultados se han obtenido usando el modelo de interferencia aditiva para colisiones (el más realista), esto es, el parámetro *collisionModel* se ha fijado a 2 en Castalia 3.2. Se podrían haber usado otros valores en este simulador, esto es, el valor 0 (sin colisiones) o 1 (modelo simplista para colisiones).

Para comparar ambos protocolos consideramos un caso particular fijando el tamaño de ranura a $\omega_t = N \cdot \tau$, un periodo activo de $\omega = \omega_t \cdot DC$ y un periodo de *sleep* de $s = \omega_t - \omega$. Además, el tiempo que un nodo está transmitiendo se ha fijado a $\tau = 0.07s$, esto es, usando *ZigBee* como modelo de radio. Este valor es idéntico para ambos protocolos. Además, el EAH es *one-way*, mientras que el LECDH es basado en *handshake*.

Se han considerado dos áreas de despliegue. La primera, de 10mx10m, escenario *one-hop*. Es un escenario simple pero útil especialmente en aquellos casos en los cuales la potencia de transmisión de los transceptores es elevado. La segunda, de 100mx100m, escenario *multi-hop*, más realista, para dispositivos que tienen transceptores de radio con rango de transmisión limitado. En este caso, se han organizado N nodos en mallas M $\times$ M.

El estudio se centra en las siguientes métricas: el tiempo de descubrimiento, el número de vecinos descubiertos, el consumo energético (dado que LECDH es un protocolo consciente de la energía). También se considera el *throughput*, y el número de descubrimientos por paquetes enviados. El tiempo de descubrimiento se ha medido en LECDH cuando el protocolo termina, es decir, cuando los nodos han descubierto todos los vecinos.

El tiempo de descubrimiento y el número de vecinos descubiertos están inversamente relacionados. Llevando a cabo varios experimentos, descubrimos que LECDH es mejor que EAH con respecto a ambas métricas cuando fijamos $0.5 \cdot N$ *rounds* en el escenario *one-hop*. También es mejor cuando se fija $0.25 \cdot N$ *rounds* en el escenario *multi-hop*. Por lo tanto, para su comparación con EAH fijamos estos números de *rounds*. En cuanto a la propuesta LECDH, no necesita fijar un número de *rounds*, ya que finaliza cuando todos los vecinos han sido descubiertos.

Para los experimentos llevados a cabo, se ha usado el modelo de radio *ZigBee* (*CC2420*). Se ha fijado la potencia de transmisión a -5dBm, la tasa de paquetes a 5 paquetes/s y el tamaño de paquete a 2500 bytes.

En cuanto a los *feedbacks*, se ha fijado un tamaño de paquete de $\tau_f = 14bytes$, tamaño de ranura de $\omega_f = N \cdot \tau_f$, y el tiempo que un nodo está transmitiendo $\tau_f = 0.000392s$.

En la Tabla 6.2, se resumen los parámetros fijados en las simulaciones.

6.4 Resultados de simulación

Esta sección presenta una comparación cuantitativa de las prestaciones de la propuesta y el protocolo de referencia EAH.

Tabla 6.2: Parámetros de simulación.

Parámetro	Valor
Static	True
Modelo de radio	CC2420
Modelo de colisiones	2
Potencia de transmisión	-5dBm
Tasa de paquetes	5 paquetes/s
Tamaño de paquete	2500 bytes
Tamaño de paquete *feedback*	14 bytes
Tamaño de ranura	$\omega_t = N \cdot \tau$
Tamaño de ranura *feedbacks*	$\omega_f = N \cdot \tau_f$
Tiempo que un nodo está tranmitiendo *BROADCAST* (τ)	0.07 s
Tiempo que un nodo está transmitiendo *feedback* (τ_f)	0.000392s
Tamaño *one-hop*	10mx10m
Tamaño *multi-hop*	100mx100m
Despliegue	Malla MxM
EAH número de *rounds* (*one-hop*)	0.5 $\cdot N$
EAH número de *rounds* (*multi-hop*)	0.25 $\cdot N$
DC	40%-100%

6.4.1 Consumo energético

La Figura 6.4 muestra que LECDH, en el caso *one-hop*, mejora al protocolo de referencia EAH para *duty cycles* 90% y 100%. LECDH obtiene peores prestaciones a medida que el *duty cycle* se reduce ya que se necesita más tiempo para descubrir todos los vecinos, por tanto el consumo energético se incrementa. Cuando el *duty cycle* para EAH baja el consumo energético se mejora ya que EAH tiene un número fijo de *rounds*, por tanto el periodo activo se decrementará y el consumo energético es mejor. Además, el consumo energético para ambos protocolos se incrementa con el número de nodos. LECDH sigue esta tendencia ya que a medida que la red se hace más densa aparecen más colisiones. Por tanto para descubrir todos los vecinos el tiempo de descubrimiento y el consumo energético crecen. EAH sigue esta tendencia ya que el tiempo de descubrimiento depende de N, por tanto el consumo energético crece.

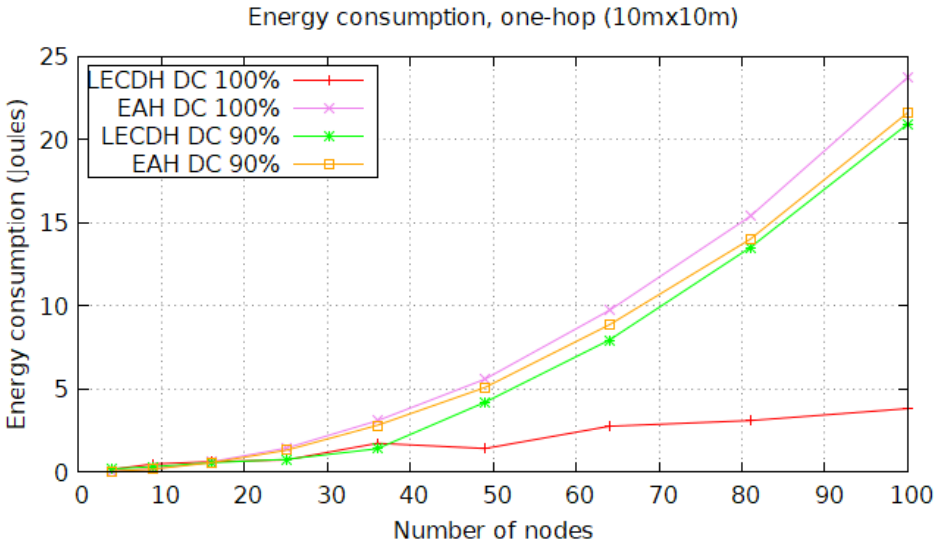

Figura 6.4: Consumo energético (one-hop)

La Figura 6.5 muestra que, en el caso *one-hop*, EAH supera al LECDH para *duty cycles* 60%, 70% y 80%. Por lo tanto hay un *duty cycle* en el cual LECDH deja de superar al EAH. De nuevo, a medida que el *duty cycle* para LECDH se reduce, el consumo energético se incrementa. Esto es debido a que el protocolo necesita más tiempo para descubrir todos los vecinos por tanto necesita gastar más energía. De nuevo, el consumo energético para ambos protocolos se incrementa a medida que la red se hace más densa, por el mismo motivo indicado antes.

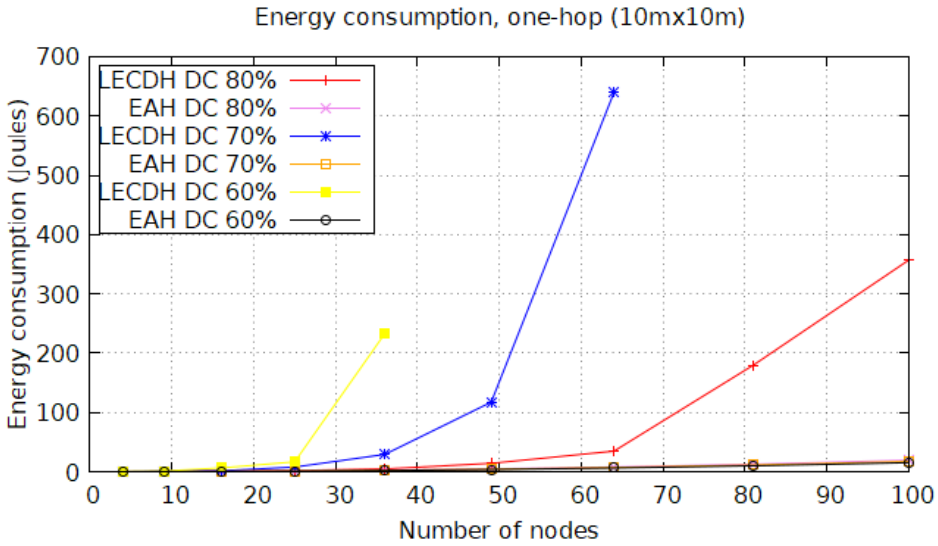

Figura 6.5: Consumo energético (one-hop)

Como se muestra en la Figura 6.6, en el caso *one-hop*, cuando el *duty cycle* es mayor o igual a 80%, LECDH funciona peor que el EAH para un número de nodos por debajo de 16. LECDH funciona mejor para un número de nodos por encima de 16 para *duty cycles* 90% y 100%. Cuando el *duty cycle* para EAH es menor el consumo energético es mejor dado que EAH tiene un número de *rounds* fijo. Por tanto esto es cierto pero el número de vecinos descubiertos disminuirá. Además, para LECDH cuando el número de nodos está por debajo de 9, el consumo energético se reduce a medida que el *DC* se decrementa, como era de esperar. Además, el EAH también podría tener un consumo energético y tiempo de descubrimiento mayores si el número de *rounds* se hubiera fijado a otro valor.

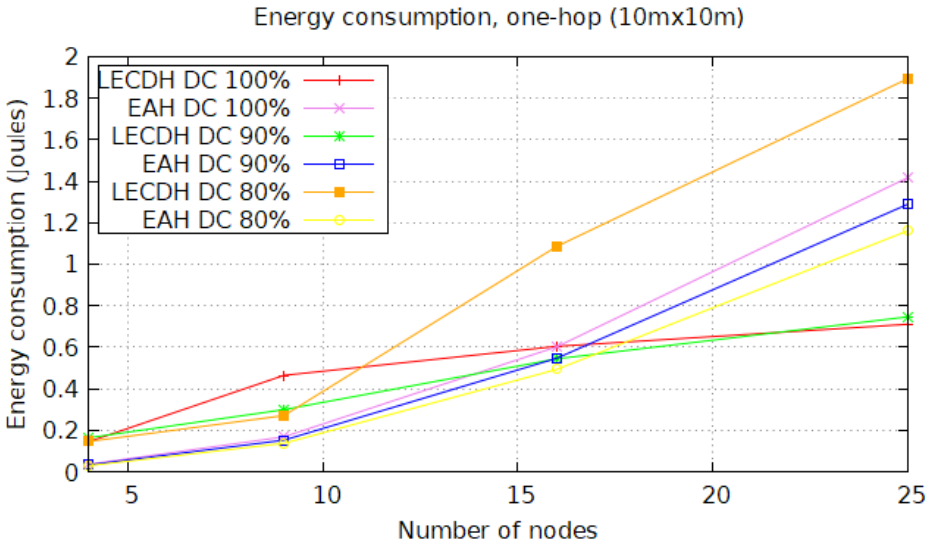

Figura 6.6: Consumo energético (one-hop). 25 nodos

La Figura 6.7, en el caso *multi-hop*, muestra que la propuesta supera al protocolo de referencia para *duty cycles* 80%, 90% y 100%. Ambos protocolos siguen una tendencia creciente por la misma razón indicada antes. Además, cuando el *duty cycle* se reduce la propuesta funciona peor ya que el tiempo necesario para descubrir todos los vecinos se incrementa, por tanto se gasta más energía. En cuanto al EAH, a medida que el *duty cycle* se reduce el consumo energético se mejora. Esto es debido a que este protocolo tiene un número de *rounds* fijo por tanto el periodo activo se reduce y el consumo energético también se reduce.

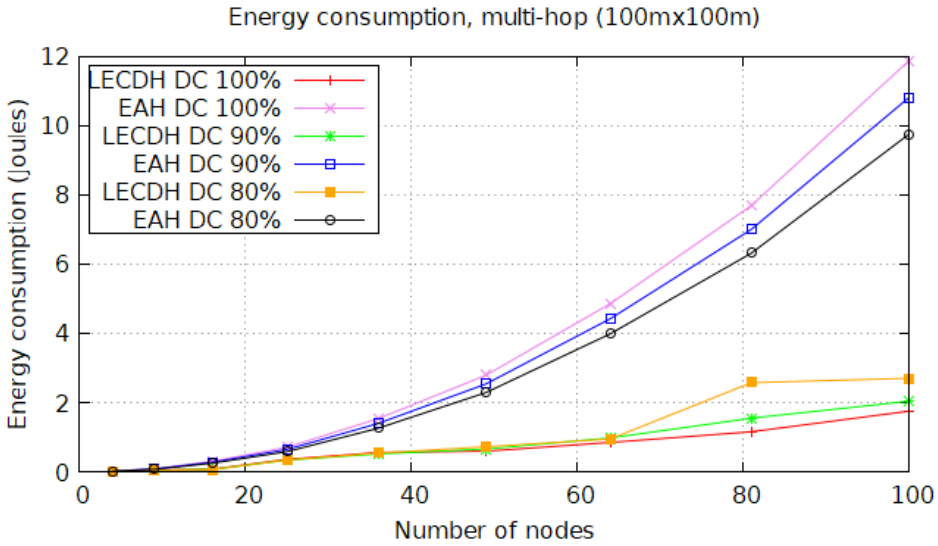

Figura 6.7: Consumo energético (multi-hop).

La Figura 6.8, en el caso *multi-hop*, muestra que LECDH supera al EAH cuando el *duty cycle* es 70%, mientras que presenta peores resultados que EAH para *duty cycles* 50% y 60%. Esto es debido a que el número de *rounds* para EAH es fijo. LECDH necesita más tiempo para descubrir todos los vecinos, por tanto se gasta más energía. Resaltar que hay un *DC* en el cual LECDH deja de superar al EAH. Además, para el EAH, a medida que el *DC* se decrementa el consumo energético se mejora (para un número de *rounds* fijo). En cuanto a LECDH, a medida que el *DC* decrece, el consumo energético crece, dado que requiere más tiempo para descubrir todos los vecinos, por tanto se gasta más energía. Para ambos protocolos, el consumo energético se incrementa a medida que el número de nodos crece por el mismo motivo indicado antes.

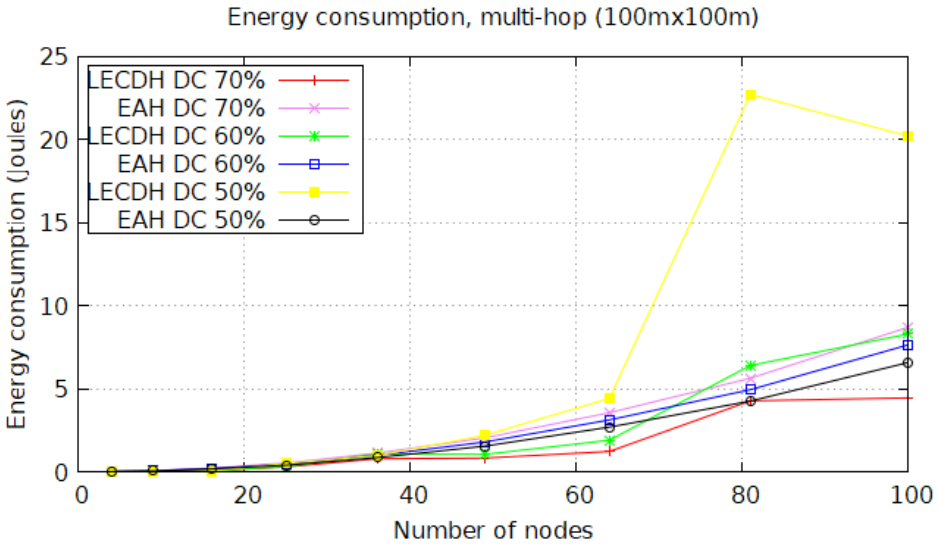

Figura 6.8: Consumo energético (multi-hop).

El caso *multi-hop*, en LECDH, para *duty cycles* 40%-100%, se muestra en la Figura 6.9. Según ella, para un número de nodos por debajo de 16, cuando el *DC* se decrementa el consumo energético se mejora, como era de esperar. Sin embargo, este comportamiento no tiene lugar para número de nodos por encima de 16.

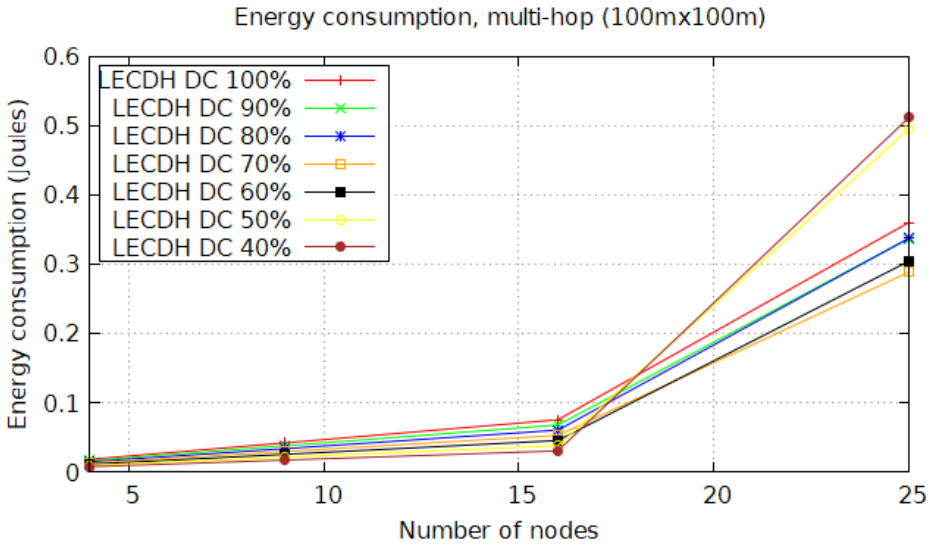

Figura 6.9: Consumo energético (multi-hop). 25 nodos

6.4.2 Tiempo de descubrimiento

Para EAH, fijamos el mismo número de *rounds* para todos los *duty cycles*, por tanto el tiempo de descubrimiento será el mismo no importa qué *duty cycle* fijemos.

La Figura 6.10 muestra que en el caso *one-hop*, LECDH supera al protocolo de referencia cuando el *DC* es 90% y 100%. Cuando el *DC* se reduce para LECDH el tiempo de descubrimiento se incrementa, dado que se necesita más tiempo para descubrir todos los vecinos. Sin embargo, el tiempo de descubrimiento es el mismo en EAH para ambos *duty cycles*, ya que el número de *rounds* es fijo. Además, el tiempo de descubrimiento para ambos protocolos se incrementa a medida que el número de nodos crece. Esto se debe al mismo motivo indicado antes para el consumo energético.

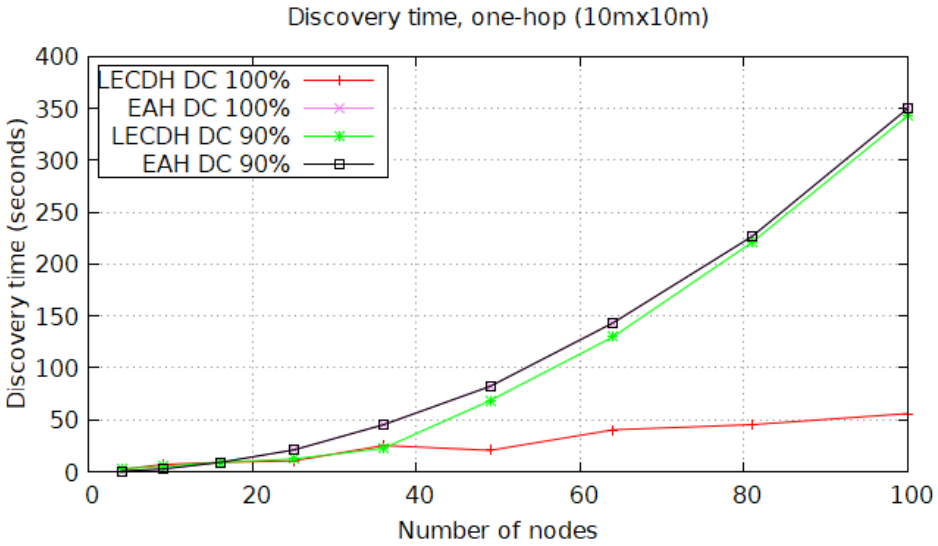

Figura 6.10: Tiempo de descubrimiento (one-hop).

Según la Figura 6.11, en el caso *one-hop*, para LECDH cuando el *duty cycle* se reduce mejor tiempo de descubrimiento se obtiene para un número de nodos por debajo de 9, como era de esperar. EAH es mejor que la propuesta cuando el número de nodos está por debajo de 16. LECDH supera al protocolo de referencia cuando el número de nodos está por encima de 16, excepto para el caso en que DC es 80%. Además, el tiempo de descubrimiento en el protocolo de referencia es el mismo para cualquier *duty cycle*, ya que el número de *rounds* es fijo.

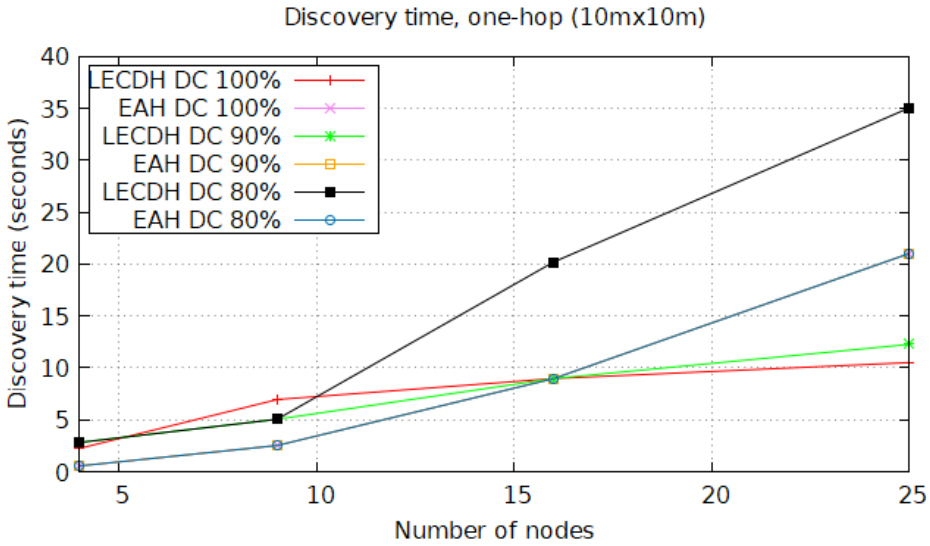

Figura 6.11: Tiempo de descubrimiento (one-hop). 25 nodos

La Figura 6.12 muestra que, en el caso *multi-hop*, para la propuesta a medida que el *duty cycle* se decrementa el tiempo de descubrimiento crece. Esto es debido a que se necesita más tiempo para descubrir todos los vecinos. LECDH es mejor que EAH cuando el *duty cycle* es de 70%, 80%, 90% y 100%, mientras que EAH es mejor cuando el *DC* está por debajo de 60%. Por lo tanto, hay un *DC* dado en el cual LECDH deja de superar al EAH. El tiempo de descubrimiento es el mismo para EAH no importa qué *DC* fijemos ya que el número de *rounds* toma un valor fijo. El tiempo de descubrimiento también presenta una tendencia creciente con el número de nodos para ambos protocolos. Esto es debido a que en LECDH a medida que el número de nodos se incrementa más colisiones tienen lugar. Por tanto el tiempo para descubrir todos los vecinos crece. En cuanto al EAH el tiempo de descubrimiento depende de N por tanto el tiempo también crece.

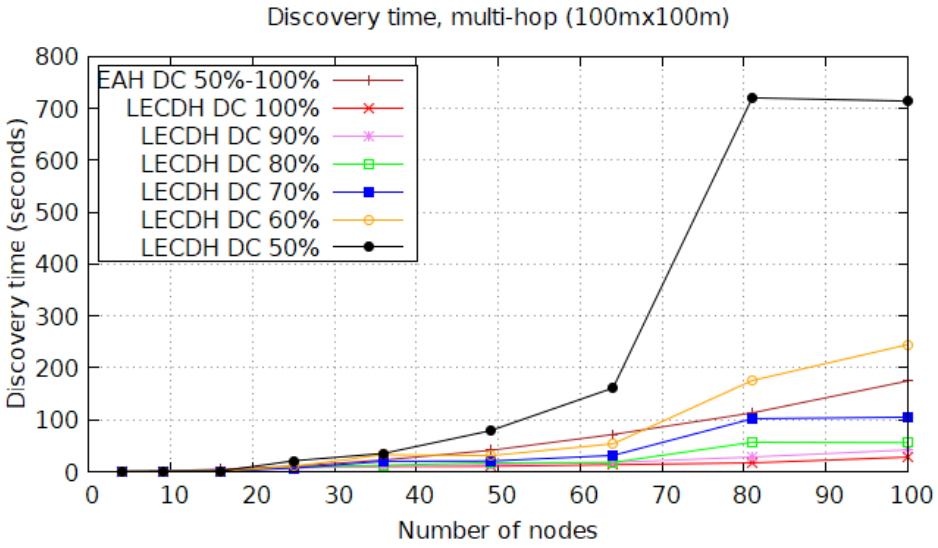

Figura 6.12: Tiempo de descubrimiento (multi-hop).

En la Figura 6.13, en el caso *multi-hop*, se presentan los resultados de simulación para redes compuestas por menos de 25 nodos. LECDH supera EAH para número de nodos por debajo de 17, mientras que EAH presenta los mismos resultados para todos los *duty cycles* ya que el número de *rounds* es fijo. Sin embargo, el tiempo de descubrimiento para LECDH crece cuando el *duty cycle* se reduce para un número de nodos por encima de 16. Esto es debido a que el protocolo requiere de más tiempo para descubrir todos los vecinos.

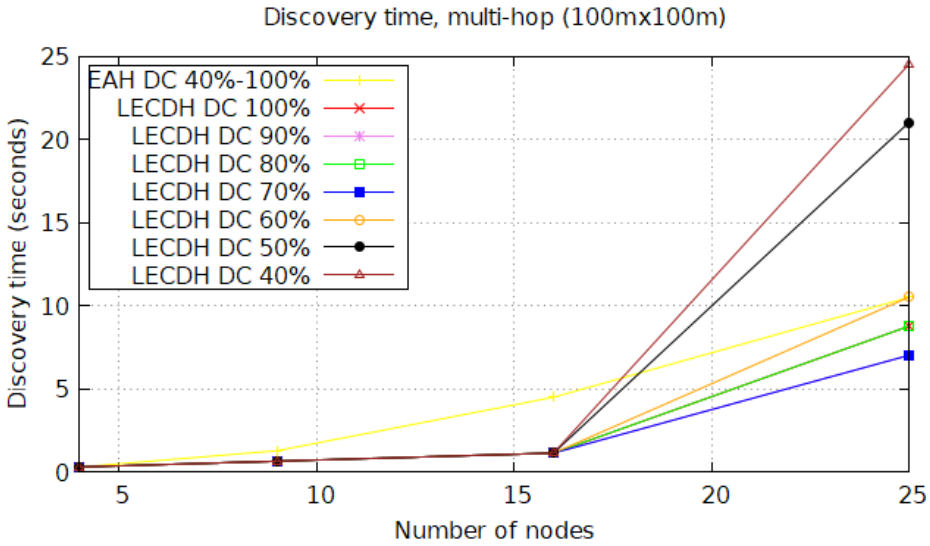

Figura 6.13: Tiempo de descubrimiento (multi-hop). 25 nodos

6.4.3 Número de vecinos descubiertos

En cuanto al escenario *one-hop*, como se muestra en la Figura 6.14, la propuesta supera al protocolo de referencia. Además, los resultados de LECDH para diferentes *duty cycles* es el mismo, ya que LECDH logra descubrir todos los vecinos no importa qué *duty cycle* fijemos. Además, para el EAH, fijamos el mismo número de *rounds* para todos los *duty cycles*. En EAH a medida que el *DC* se decrementa el número de vecinos descubiertos también se decrementa, como era de esperar. Sin embargo, el EAH no logra descubrir todos los vecinos.

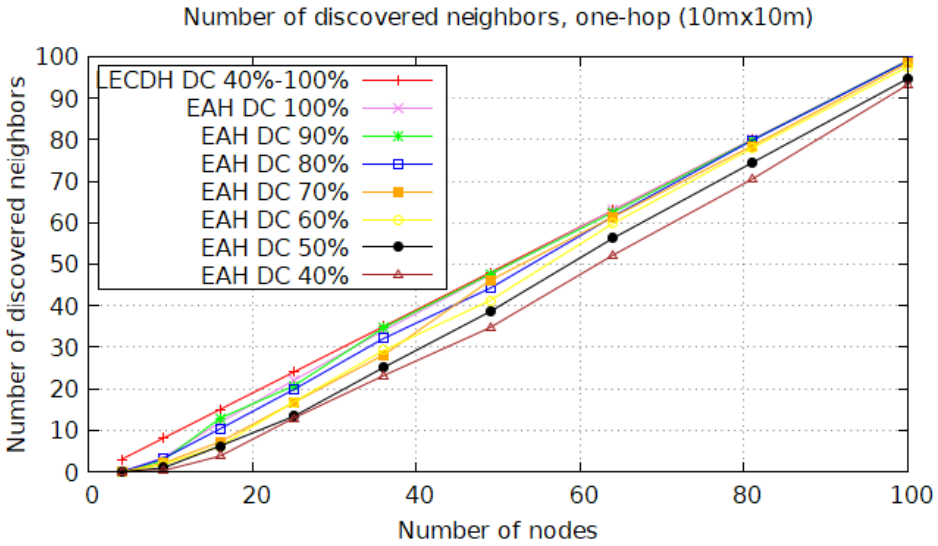

Figura 6.14: Número de vecinos descubiertos (one-hop).

De acuerdo con la Figura 6.15, en un escenario *multi-hop*, LECDH supera al EAH para todos los *duty cycles*, y logra descubrir todos los vecinos. Además el número de vecinos descubiertos para el EAH decrece a medida que el *duty cycle* se decrementa, como era de esperar. Esto es debido a que se ha fijado el mismo número de *rounds* para EAH. Sin embargo, de nuevo el EAH no logra descubrir todos los vecinos.

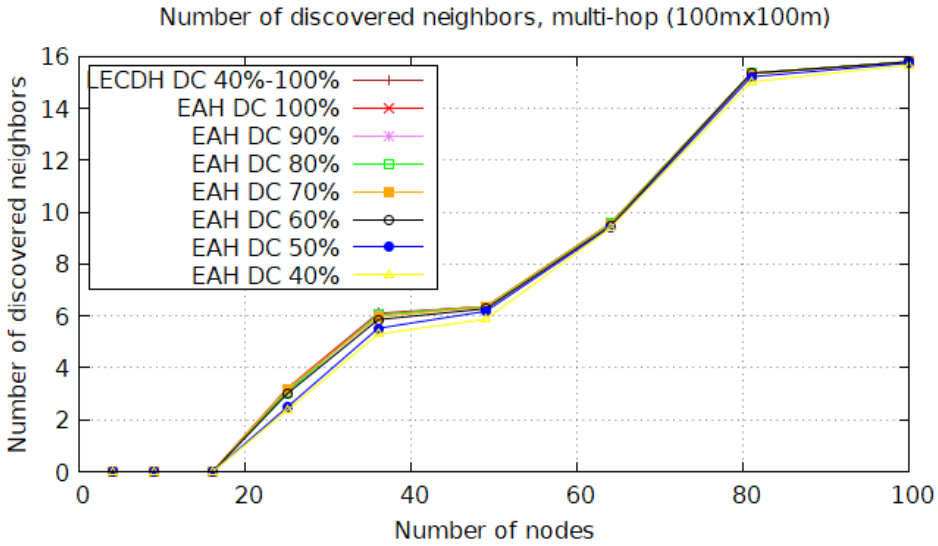

Figura 6.15: Número de vecinos descubiertos (multi-hop).

De acuerdo con la Figura 6.16, en el caso *multi-hop*, para bajo número de nodos, LECDH supera al EAH, y logra descubrir todos los vecinos. Además, para el EAH, habiendo fijado un número de *rounds* fijo para los diferentes *duty cycles*, a medida que el *DC* se decrementa, el número de vecinos descubiertos se reduce, como era de esperar. Además, el protocolo de referencia no logra descubrir todos los vecinos.

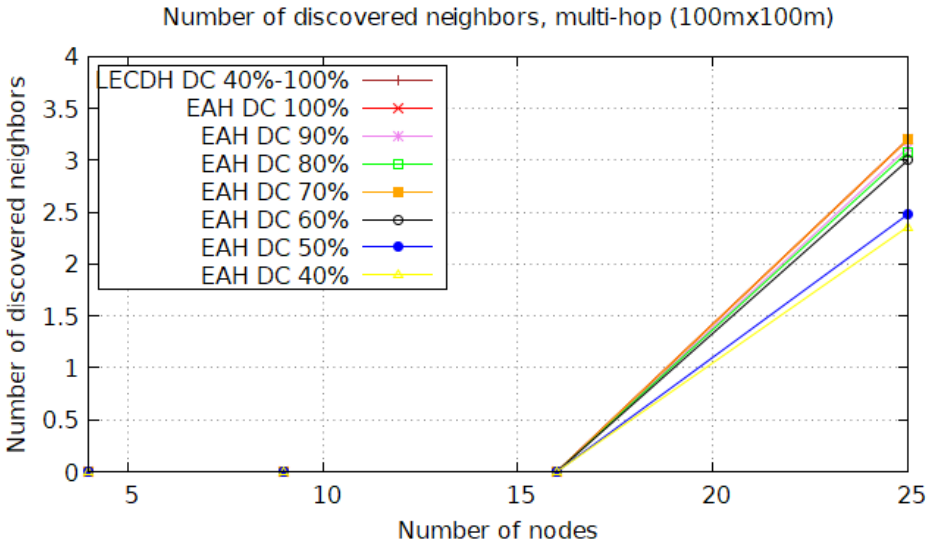

Figura 6.16: Número de vecinos descubiertos (multi-hop). 25 nodos

6.4.4 Throughput

Con respecto al *throughput*, en un escenario *one-hop*, y como se muestra en la Figura 6.17, LECDH supera al protocolo de referencia para *duty cycles* 70%, 80%, 90 % y 100%. Además, el *throughput* sigue una tendencia decreciente con el número de nodos para LECDH. Esto es debido a que se dan más colisiones. Por tanto el número de paquetes recibidos es menor y se requiere más tiempo para descubrir todos los vecinos. Además, el *throughput* sigue una tendencia decreciente para EAH ya que el tiempo depende de N y el tiempo crece. Por tanto el *throughput* decrecerá a medida que el número de nodos crece. Sin embargo, para LECDH cuando el *duty cycle* se reduce el *throughput* es peor. Esto es debido a que el tiempo está inversamente relacionado con el *throughput* y el protocolo necesita más tiempo para descubrir todos los vecinos.

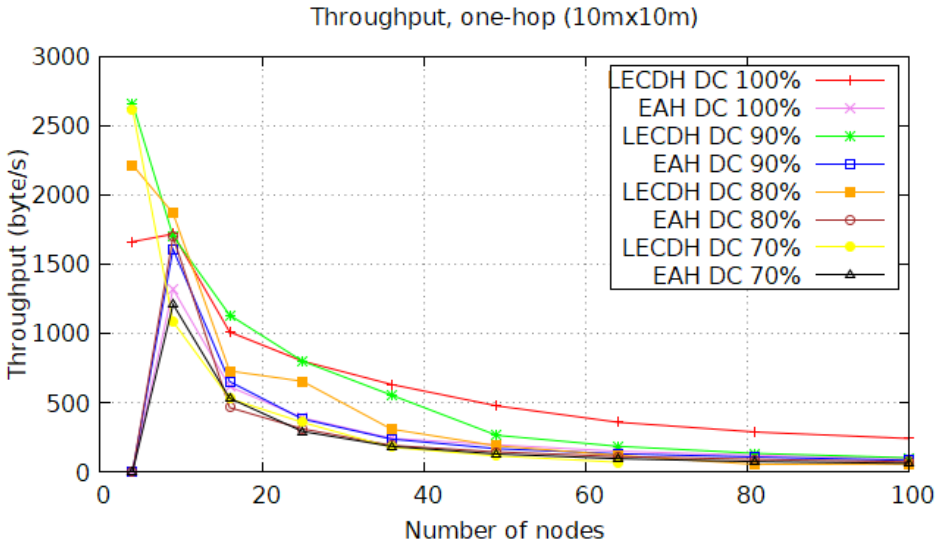

Figura 6.17: Throughput (one-hop).

Como se muestra en la Figura 6.18, en el caso *one-hop*, para *DC* 70%, 80%, 90% y 100%, y un número de nodos por debajo de 25, la propuesta también supera al protocolo de referencia con respecto al *throughput*.

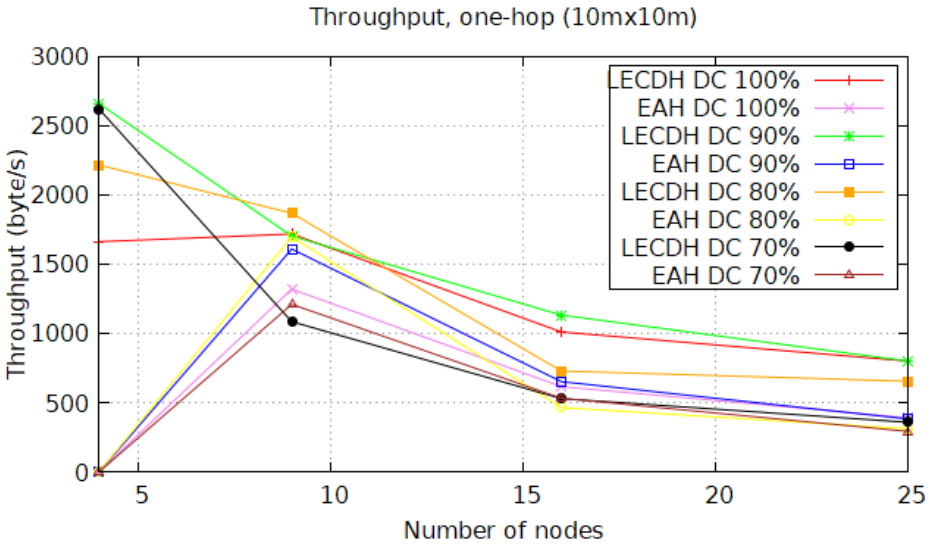

Figura 6.18: Throughput (one-hop). 25 nodos

De acuerdo con la Figura 6.19, en el caso *multi-hop*, LECDH supera al EAH en términos de *throughput* para *duty cycles* 70%, 80%, 90% y 100%. Ambos protocolos siguen una tendencia decreciente por el mismo motivo indicado antes. Para LECDH, a medida que el *duty cycle* se reduce se obtienen peores resultados. Esto es debido a que se necesita más tiempo para descubrir todos los vecinos, por tanto el *throughput* se reduce.

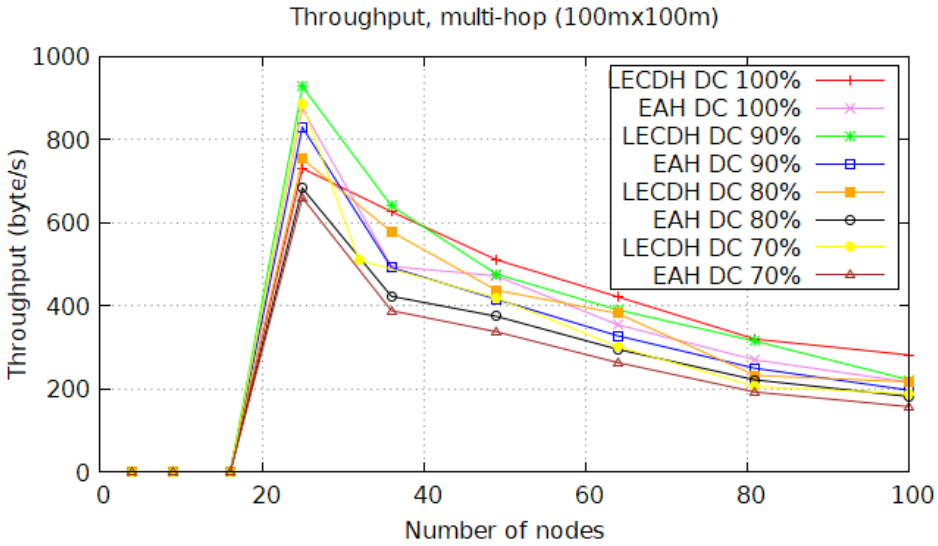

Figura 6.19: Throughput (multi-hop).

La Figura 6.20 muestra que, en el caso *multi-hop*, LECDH de nuevo supera al EAH con respecto al *throughput* para *duty cycles* 40%, 50% y 60%. De nuevo, ambos protocolos siguen una tendencia decreciente por el mismo motivo indicado antes. Sin embargo, para LECDH a medida que el *duty cycle* se reduce el *throughput* se reduce. Esto es debido a que es necesario más tiempo para descubrir todos los vecinos y el tiempo está inversamente relacionado con el *throughput*.

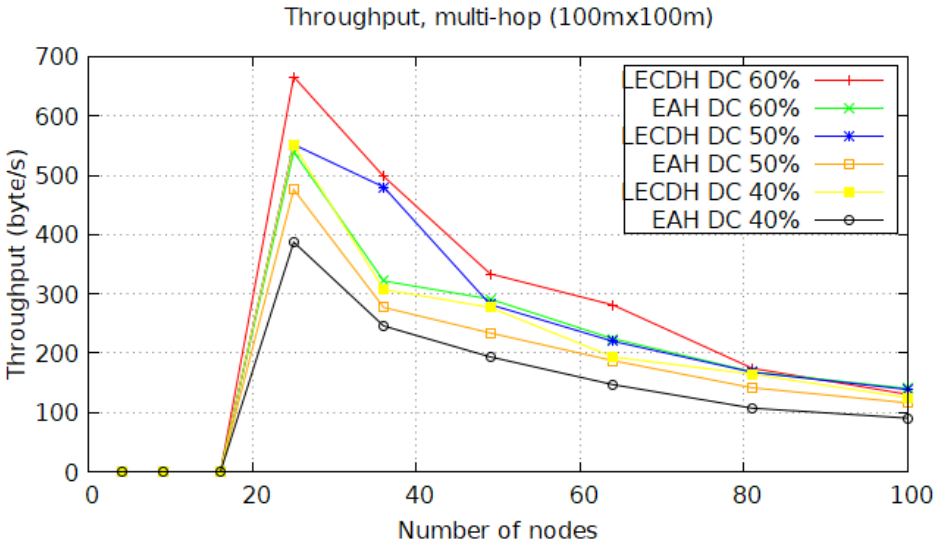

Figura 6.20: Throughput (multi-hop).

6.4.5 Número de descubrimientos por paquetes enviados

Según la Figura 6.21, en el caso *one-hop*, LECDH supera al EAH para *duty cycles* 90% y 100%, aunque EAH con *DC* 80% supera al LECDH para redes compuestas de más de 36 nodos. Por lo tanto, hay un valor *DC* en el cual LECDH deja de superar al EAH. Sin embargo, a medida que el *duty cycle* se reduce LECDH funciona mejor para un número de nodos por debajo de 10, como era de esperar.

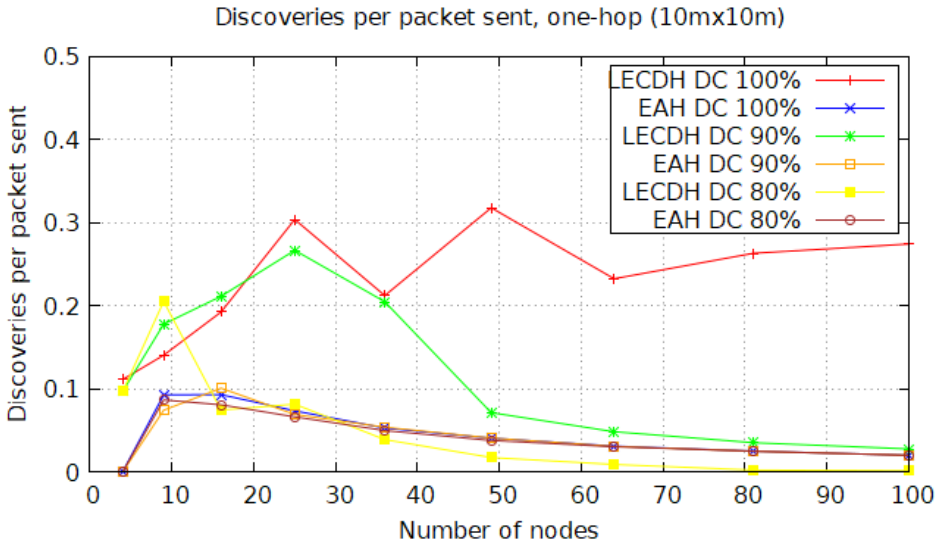

Figura 6.21: Número de descubrimientos por paquetes enviados (one-hop).

De acuerdo con la Figura 6.22, en el caso *multi-hop*, con respecto al número de descubrimientos por paquetes enviados, LECDH supera al EAH para *duty cycles* 80%, 90% y 100%. Sin embargo, a medida que el *duty cycle* se reduce, LECDH presenta peores resultados. Esto es debido a que se necesita más tiempo para descubrir todos los vecinos, por tanto se envían más paquetes para descubrir la misma cantidad de vecinos.

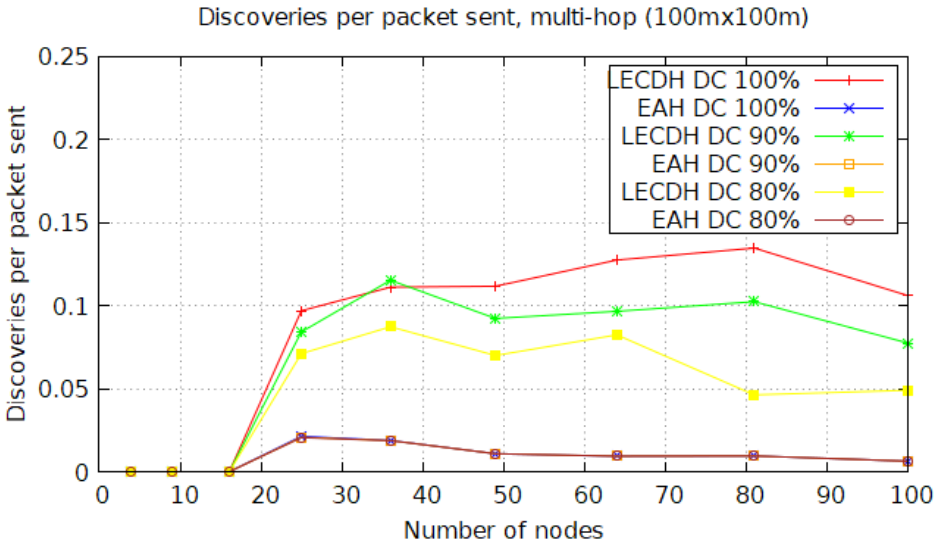

Figura 6.22: Número de descubrimientos por paquetes enviados (multi-hop).

6.5 Conclusiones

En este capítulo se aborda el descubrimiento de vecinos en el contexto de redes inalámbricas ad hoc estáticas *multi-hop* considerando la existencia de colisiones y centrándose en mejorar el consumo energético.

Se ha propuesto un nuevo protocolo aleatorio consciente de la energía LECDH, que toma las ventajas de la detección de colisiones.

Además, se ha elegido un protocolo de la literatura, para compararlo con la propuesta, y usado como referencia: el EAH. Ambos protocolos se han simulado con Castalia 3.2 para poder compararlos, centrándose tanto en entornos *one-hop* como *multi-hop*. Las métricas usadas son: el consumo energético, el tiempo de descubrimiento, el número de vecinos descubiertos, el *throughput*, y el número de descubrimientos por paquetes enviados, y variando el *duty cycle*.

De acuerdo con los resultados de simulación, se concluye que LECDH supera al protocolo de referencia según las cinco métricas tanto en entornos *one-hop* como *multi-hop* para altos *duty cycles*. Además, para LECDH en redes

pequeñas, cuando el DC se reduce mejores resultados se obtienen para las 5 métricas en ambos entornos.

En general, LECDH funciona siguiendo premisas más realistas, detecta colisiones y terminación, logra descubrir todos los vecinos casi con probabilidad 1. Además permite a los nodos iniciar la transmisión en cualquier instante de tiempo, y el número de nodos puede permanecer desconocido.

Entre sus limitaciones prácticas, LECDH requiere que los nodos estén sincronizados en los límites de ranura, y no se permite su uso en MANETs. Sólo es apropiado para altos *duty cycles*, esto es, por debajo de 40% el tiempo de descubrimiento y el consumo energético crecen desorbitadamente. Posibles formas de solucionar estas limitaciones podrían consistir en usar un mecanismo de sincronización conocido antes del proceso de descubrimiento. Además hay que tener en cuenta que los nodos entren o salgan de la red y nodos entrando y saliendo en el rango de transmisión de otros nodos para su uso en MANETs. También hay que centrarse en el problema de mejorar las prestaciones en bajos *duty cycles*. En el capítulo 5 se propone un protocolo asíncrono, esto es, no requiere de sincronización, pero que solo funciona en entornos *one-hop*.

En cuanto a la propuesta, puede ser aplicada en entornos del mundo real, tales como redes estáticas inalámbricas ad hoc con altos *duty cycles* y compuesta de una baja cantidad de nodos. Se puede usar tanto en entornos *one-hop* como *multi-hop*, tales como una red de sensores inalámbricos. En ellas el bajo consumo energético es un principal objetivo a lograr. Además, resaltar que para las simulaciones el modelo de radio elegido es *ZigBee* (para WSN).

Como futuras direcciones se podría extender el protocolo para solucionar las limitaciones indicadas. Además sería interesante proponer un nuevo protocolo de bajo consumo energético para la creación de redes espontáneas basadas en la confianza. También se podría proponer protocolos de descubrimiento adecuados para entornos móviles.

Capítulo 7

Modelo analítico para protocolos de descubrimiento de vecinos aleatorios basados en la detección de colisiones

En este capítulo se presentan modelos analíticos de protocolos de descubrimiento de vecinos aleatorios para entornos estáticos one-hop presentados en el capítulo 4. Para CDPRR se ha asumido una distribución geométrica y una distribución uniforme para CDH. Para su comparación se han elegido dos protocolos de la literatura usados como referencia: Hello y PRR. Se han obtenido prestaciones a través ocho métricas. Se presentan resultados gráficos mediante la representación de las ecuaciones obtenidas. De acuerdo con los resultados analíticos, CDH supera a las otras soluciones con respecto al tiempo de descubrimiento de vecinos, y consumo energético. También las supera según número de paquetes enviados, el packet delivery ratio y el CDF de descubrimientos. CDPRR logra buenos resultados y es mejor que el Hello y PRR en tiempo de descubrimiento de vecinos, throughput, consumo energético, CDF de descubrimientos y número de paquetes enviados. Además, se ha descubierto que CDPRR presenta mayor porcentaje de idle slots que PRR. Esto representa una clara ventaja en cuanto a energía consumida y número de paquetes enviados. También se ha centrado el estudio en el protocolo CDH variando el tamaño de ranura. Se ha demostrado que el número de nodos de la red puede ser desconocido, y aún así proporcionar resultados razonables.

7.1 Introducción

Como novedad con relación a los protocolos de referencia, ambas propuestas CDH y CDPRR logran descubrir todos los vecinos. Ambos conocen cuando terminar el proceso de descubrimiento y logran funcionar bajo premisas más realistas.

El principal objetivo de este capítulo es proponer un modelo analítico de protocolos aleatorios que no requieren de una planificación en la transmisión. También tiene en cuenta la existencia de colisiones, sigue premisas más realistas y tienen como objetivo obtener mejores prestaciones que soluciones existentes.

Este capítulo se centra en la presentación de modelos analíticos de dos protocolos aleatorios de descubrimiento de vecinos basados en la detección de colisión. Se permite su uso en entornos estáticos *one-hop* inalámbricos ad hoc en la presencia de colisiones. Las colisiones tienen lugar cuando dos o más nodos transmiten al mismo tiempo. Además ambos protocolos conocen cuando terminar el proceso de descubrimiento, y el número de nodos debe ser desconocido.

Para ambas propuestas se tiene como objetivo descubrir todos los vecinos con probabilidad 1, por tanto mejorando protocolos aleatorios existentes. También se centra el estudio en mejorar varias métricas. Estas métricas son el tiempo de descubrimiento de vecinos, el consumo energético, el *throughput*, y el *overhead* (número de paquetes enviados). También, el *packet delivery ratio*, el porcentaje de descubrimientos por *round*, el CDF de descubrimientos, y el porcentaje de *idle slots*. Los protocolos siguen premisas más realistas.

CDPRR sabe cuándo terminar el proceso de descubrimiento, logra descubrir todos los vecinos con probabilidad 1. Sin embargo, requiere conocer el número de nodos. En cuanto al CDH, incluye un mecanismo de detección de terminación, también logra descubrir todos los vecinos con probabilidad 1. Además, no requiere conocer el número de nodos de la red.

Las principales contribuciones de este capítulo son: (i) Un modelo analítico para ambas propuestas CDPRR y CDH. Las métricas son: tiempo de descubrimiento de vecinos, consumo energético, *throughput*, y *overhead* (número de paquetes enviados). También se modela el *packet delivery ratio*, porcentaje de descubrimientos por *round*, CDF de descubrimientos y el porcentaje de *idle slots*. El modelo se ha desarrollado para un escenario *one-hop*. (ii) Un modelo analítico para dos protocolos de referencia (Hello y PRR). Las métricas evaluadas son: tiempo de descubrimiento de vecinos, consumo energético, *throughput*, y *overhead* (número de paquetes enviados). También se modela el

packet delivery ratio, porcentaje de descubrimientos por *round*, CDF de descubrimientos y el porcentaje de *idle slots*. El modelo también se ha desarrollado para un escenario *one-hop*. (iii) Resultados gráficos usando las ecuaciones obtenidas, comparando los cuatro protocolos en cuanto a las ocho métricas. (iv) Resultados gráficos de CDH variando el tamaño de ranura, que depende tanto del número de nodos como con tamaño fijo de ranura.

Las propuestas aleatorias difieren de soluciones previas ya que las propuestas logran descubrir todos los vecinos con probabilidad 1, incluso en redes densas. Permiten solucionar el problema de protocolos aleatorios previos, que descubren todos los vecinos con alta probabilidad (distinta de 1). Además, CDH no necesita conocer el número de nodos de la red, y ambas propuestas conocen cuándo terminar el proceso de descubrimiento y son basadas en *handshake*. Además se tiene como objetivo mejorar en cuanto a todas las métricas. Esas métricas son el tiempo de descubrimiento de vecinos, consumo energético, *throughput*, y número de paquetes enviados. También se desea mejorar en cuanto a *packet delivery ratio*, porcentaje de descubrimientos por *round*, CDF de descubrimientos y porcentaje de *idle slots*. Además, las propuestas son adecuadas para escenarios estáticos *multi-hop*. Las dos propuestas son probabilísticas, lo que significa que no son determinísticas, no se incluye mecanismo de *wakeup*, no pueden ser usadas en MANETs. No se usa ni antena ni radar, no se centran en el problema de la seguridad, y se usa un control de acceso relacionado con el uso del canal para lograr descubrimiento de vecinos.

7.2 Modelo analítico

En este capítulo se presentan ecuaciones para obtener las prestaciones de las dos propuestas de descubrimiento de vecinos aleatorias en entorno estático *one-hop*.

7.2.1 CDPRR

De acuerdo con el protocolo, las variables usadas en el análisis se definen en la Tabla 7.1.

Para el análisis, asumimos que el protocolo consiste en diferentes fases, cada una finalizando cuando un único nodo logra transmitir con éxito. Por lo tanto, hay N fases, cada fase i cumpliendo $1 \leq i \leq N$ y consistiendo en un número de *rounds*. El número de *rounds* necesario para descubrir todos los vecinos es el número total de *rounds*.

Tabla 7.1: Definición de variables de CDPRR.

Variable	Definición
τ	El tiempo que un nodo está transmitiendo en segundos, y coincide con la duración del *round*.
τ_f	El tiempo que un nodo está transmitiendo un *feedback* en segundos
N	El número de nodos de la red.
$\frac{1}{N}$	La probabilidad de que un nodo transmita.
$1-\frac{1}{N}$	La probabilidad de que un nodo escuche.
p_i	La probabilidad de que un nodo transmita con éxito.
W_i	El número de *rounds* hasta que un nodo transmite con éxito en la fase i.
W	El número total de *rounds* hasta que todos los vecinos han sido descubiertos.
X_i	El valor esperado del número de *rounds* hasta que un nodo transmita con éxito en la fase i.
X	El valor esperado del número total de *rounds* cuando todos los vecinos han sido descubiertos.
Tt	El tiempo de descubrimiento de vecinos en segundos.
T_{hr}	El *throughput* en paquetes/s.
NT_i	El número total de nodos transmitiendo en la fase i, teniendo en cuenta solo los nodos que están compitiendo.
NL_i	El número total de nodos escuchando en fase i, teniendo en cuenta sólo los nodos que están compitiendo.
$E(NT_i)$	El valor esperado de NT_i.
$E(NL_i)$	El valor esperado de NL_i.
n_i	El número de experimentos de los nodos que aún están compitiendo en la fase i.
$E(P_i)$	El valor esperado del consumo energético en la fase i.
$E(P)$	El promedio del consumo energético por cada nodo en las N fases.
$E(P_t)$	El promedio del consumo energético por cada nodo en las N fases sumado al de los 2 rounds de terminación.
$E(P_f)$	El promedio del consumo energético de los *feedbacks*.
E_T	El total del consumo energético.
E_{tx}	La cantidad de energía consumida por un nodo transmitiendo por segundo.
E_l	La cantidad de energía consumida por un nodo escuchando por segundo.
P_{sent}	El número total de paquetes enviados en las N fases.
P_{rec}	El número total de paquetes recibidos por cada nodo en las N fases.
P_{sentf}	El número total de paquetes enviados en los *feedbacks*.
P_{recf}	El número total de paquetes recibidos por cada nodo en las N fases en los *feedbacks*.
s_b	El tamaño del paquete de *BROADCAST* en bytes.
s_f	El tamaño del paquete de *feedback* en bytes.
PDR	El *packet delivery ratio*.
PDR_f	El *packet delivery ratio* para los *feedbacks*.
PND	Porcentaje de descubrimientos por *round*.
$Fx(k)$	CDF de los descubrimientos.
$E(NIS_i)$	Número total de *rounds* en la fase i en la que todos los nodos restantes nodos están escuchando.
$E(NIS)$	Número total de *idle slots*.
PIS	Porcentaje de *idle slots* vs número total de *rounds*.

Tiempo de descubrimiento de vecinos

Cuando hablamos de tiempo de descubrimiento de vecinos nos referimos al tiempo que tarda el algoritmo en descubrir todos los vecinos.

La probabilidad de que un único nodo transmita con éxito en un *round* en la fase 1 (p_1) es:

$$p_1 = \frac{1}{N} \cdot \left(1 - \frac{1}{N}\right)^{N-1} \tag{7.1}$$

siendo N el número de nodos de la red, $\frac{1}{N}$ la probabilidad de que un nodo transmita, y $1 - \frac{1}{N}$ la probabilidad de que un nodo escuche.

En la fase i, $i-1$ nodos ya han transmitido con éxito, por tanto no están compitiendo en la fase i, y permanecen escuchando con probabilidad 1. En la fase i solo $N-i+1$ nodos están compitiendo. Por lo tanto, para los nodos que están compitiendo, la probabilidad de que un único nodo transmita en el *round* i mientras que el resto de los nodos $(N-i)$ escuchan es p_i.

$$p_i = \frac{1}{N} \cdot \left(1 - \frac{1}{N}\right)^{N-i} \tag{7.2}$$

Asumimos que W_i (el número de *rounds* en la fase i hasta que un nodo transmite con éxito) sigue una distribución geométrica, $W_i \sim Geo((N-i+1) \cdot p_i)$, siendo p_i obtenida de la ecuación 7.2. Obtenemos X_i como el valor esperado del número de *rounds* en la fase i hasta que un nodo transmite con éxito. Se asume distribución geométrica ya que este tipo de distribución de probabilidad modela el número de ensayos Bernoulli necesarios para tener un éxito. En este caso, el número de *rounds* hasta un éxito, es decir, un nodo transmite con éxito.

El valor esperado del número de rounds en la fase i viene dada por la ecuación 7.3 ya que en la fase i hay $N-i+1$ nodos que no han transmitido con éxito en rounds previos.

$$X_i = E(W_i) = \frac{1}{(N-i+1) \cdot p_i} = \frac{N}{(N-i+1) \cdot (1-\frac{1}{N})^{N-i}} \tag{7.3}$$

El número de *rounds* total resultante es $W = W_1 + W_2 + \cdots W_N$, y el valor esperado viene dado por la ecuación 7.4.

$$X = E(W) = \sum_{i=1}^{N} X_i = \sum_{j=1}^{N} \frac{N}{j \cdot (1-\frac{1}{N})^{j-1}} \tag{7.4}$$

Debemos añadir al número total de *rounds* 2 *rounds* más para incluir el *handshake* final para terminar el descubrimiento como se muestra en la sección 4.3.2.

En conclusión, el promedio del tiempo de descubrimiento de vecinos es T_t. Este es el valor esperado del número total de *rounds* $X+2$ multiplicado por $\tau + \tau_f$, es decir, la duración total de *round* y es dada por la ecuación 7.5.

$$T_t = (X + 2) \cdot (\tau + \tau_f) \tag{7.5}$$

Consumo energético

A continuación, se calcula el consumo energético P_i en cada fase i y luego el promedio del consumo energético para todas las fases $E(P)$ en Julios.

Se asume una distribución binomial $NT_i \sim B(n_i, \frac{1}{N})$ para el número total de transmisiones de todos los nodos que están compitiendo (n_i) en la fase i con una probabilidad de transmisión fija $\frac{1}{N}$. Hay dos posibles resultados, esto es, transmitir o escuchar. También se asume una distribución binomial $NL_i \sim B(n_i, 1 - \frac{1}{N})$ para el número total de escuchas de todos los nodos que están compitiendo (n_i) en la fase i. La probabilidad de escucha es fija $1 - \frac{1}{N}$. También hay dos posibles resultados, transmitir o escuchar.

Mostraremos más adelante que $n_i = (N - i + 1) \cdot X_i$ para la fase i.

El número total de veces que los nodos transmiten en la fase 1 es NT_1, teniendo en cuenta que la fase 1 consiste en X_1 *rounds*. Además, en la fase 1 hay N nodes compitiendo y pueden o transmitir o escuchar, durante X_1 *rounds*, por tanto hay $n_1 = N \cdot X_1$ experimentos. De forma similar, el número total de veces que los nodos escuchan en la fase 1 viene dado por NL_1.

En cuanto a la fase i, $i-1$ nodos transmitieron con éxito en fases previas, por tanto estos nodos no tomarán ni estado T ni L.

El número total de veces que los nodos transmiten en X_i *rounds* (en fase i) viene dado por la ecuación 7.6, siendo $n_i = (N-i+1) \cdot X_i$. La ecuación para n_i se obtiene dado que en la fase i hay aún $N-i+1$ nodos que están compitiendo y hay X_i *rounds*. Y para el número de veces que los nodos escuchan en X_i *rounds* (en la fase i) viene dado por la ecuación 7.7.

$$E(NT_i) = \sum_{k=1}^{n_i} k \cdot \binom{n_i}{k} \cdot \left(\frac{1}{N}\right)^k \cdot \left(1 - \frac{1}{N}\right)^{n_i - k} \tag{7.6}$$

$$E(NL_i) = \sum_{k=1}^{n_i} k \cdot \binom{n_i}{k} \cdot \left(\frac{1}{N}\right)^{n_i - k} \cdot \left(1 - \frac{1}{N}\right)^k \tag{7.7}$$

$E(P_1)$, es el valor esperado de energía consumida durante la fase 1 en Julios y viene dado por la ecuación 7.8.

$$E(P_1) = \tau[E_{tx} \cdot E(NT_1) + E_l \cdot E(NL_1)] \tag{7.8}$$

Considerando que sólo un nodo transmitió con éxito en la fase 1, es decir, se mantiene escuchando, en la fase 2 se debe añadir un término $\tau \cdot X_2 \cdot E_l$ a $E(P_2)$ para este nodo. En la fase 3, dos nodos transmitieron con éxito por tanto X_3 *rounds* escuchando, por tanto un total de energía consumida de $\tau \cdot 2 \cdot X_3 \cdot E_l$ se debe añadir a $E(P_3)$. En general, para la fase i, la energía consumida por los nodos que transmitieron con éxito en las $i-1$ fases previas es $\tau \cdot (i-1) \cdot X_i \cdot E_l$. Este término se debe añadir a $E(P_i)$ obteniendo la siguiente ecuación para $E(P_i)$.

$$E(P_i) = \tau \cdot [E_{tx} \cdot E(NT_i) + E_l \cdot E(NL_i) + E_l \cdot (i-1) \cdot X_i] \tag{7.9}$$

siendo $\tau \cdot E_l \cdot (i-1) \cdot X_i$ la energía consumida en la fase i por los nodos que transmitieron con éxito en fases previas.

El promedio del consumo energético por cada nodo ($E(P)$) en Julios viene dado por la ecuación 7.10, usando el sumatorio de la ecuación 7.11. Recordar que se deben añadir 2 *rounds* más para el *handshake* de terminación, por tanto el total de consumo viene dado por la ecuación 7.12.

$$E(P) = \frac{1}{N} \cdot \sum_{i=1}^{N} E(P_i) = \frac{\tau}{N} \cdot \left[E_{tx} + E_l \cdot N \cdot \left(1 - \frac{1}{N}\right)\right] \cdot X + \frac{\tau}{N^2} \cdot [E_l - E_{tx}] \cdot \sum_{k=1}^{N} (k-1) \cdot X_k \tag{7.10}$$

$$\sum_{k=1}^{N} (k-1) \cdot X_k = \sum_{k=1}^{N} \frac{N \cdot (k-1)}{(N-k+1) \cdot (1 - \frac{1}{N})^{N-k}} \tag{7.11}$$

$$E(P_t) = E(P) + \tau \cdot \frac{2 \cdot N - 1}{N} \cdot E_l \tag{7.12}$$

A continuación, se deben añadir los *feedbacks*. Para ello, se debe tener en cuenta que en la fase i, un número de $\tau_f \cdot E_l \cdot N \cdot (X_i - 1)$ están escuchando

cuando no se descubre ningún vecino. Además $\tau_f \cdot E_{tx} \cdot (N-1)$ están enviando paquetes de *feedback* en el último *round* de la fase i, es decir, cuando el vecino es descubierto. $\tau_f \cdot E_l$ están escuchando en el último *round* de la fase i cuando el vecino es descubierto. Sin embargo, en los dos últimos *rounds* (comprobación de terminación), 1 nodo está transmitiendo *feedback*. Los otros nodos escuchan $(\tau_f \cdot E_{tx} \cdot 1 + \tau_f \cdot E_l \cdot (N-1))$ en el primer *round* para comprobación de terminación. En el segundo *round* para comprobación de terminación $\tau_f \cdot E_{tx} \cdot (N-1)$. Por lo tanto, el promedio del consumo energético para los *feedbacks* es:

$$E(P_f) = \frac{1}{N} \sum_{i=1}^{N} [\tau_f \cdot E_l \cdot N \cdot (X_i - 1) + \tau_f \cdot E_l + \tau_f \cdot (N-1) \cdot E_{tx}] \quad (7.13)$$

$$+ \frac{1}{N} \cdot [\tau_f \cdot (N-1) \cdot E_l + \tau_f \cdot E_{tx} + \tau_f \cdot (N-1) \cdot E_{tx}] \quad (7.14)$$

$$E(P_f) = \tau_f \cdot E_l \cdot X + \frac{2 \cdot N - 1}{N} \cdot \tau_f \cdot E_l + N \cdot \tau_f \cdot E_{tx} \quad (7.15)$$

Finalmente, el total de consumo energético viene dado por la ecuación 7.16.

$$E_T = E(P_t) + E(P_f) \quad (7.16)$$

Overhead

De las ecuaciones 7.6 obtenemos el número de nodos que transmiten en la fase i de longitud X_i *rounds*.

$$E(NT_i) = (N \cdot X_i - (i-1) \cdot X_i) \cdot \frac{1}{N} \quad (7.17)$$

En cuanto al total de paquetes enviados (P_{sent}) viene dado por la ecuación 7.19, usando el sumatorio de la ecuación 7.11.

$$P_{sent} = \sum_{i=1}^{N} \frac{1}{N} \cdot (N \cdot X_i - (i-1) \cdot X_i) = \sum_{i=1}^{N} X_i - \frac{1}{N} \cdot \sum_{i=1}^{N} (i-1) \cdot X_i \quad (7.18)$$

$$P_{sent} = X - \frac{1}{N} \cdot \sum_{i=1}^{N} (i-1) \cdot X_i = \frac{1 - (1 - \frac{1}{N})^{-N}}{1 - (1 - \frac{1}{N})^{-1}} \quad (7.19)$$

Se debe añadir el número total de paquetes enviados en los 2 *rounds* de terminación. En el primer y segundo *round* de terminación, ninguno de los nodos envía (están escuchando) por tanto se envían 0 paquetes. Por lo tanto, P_{sent} permanece como en la ecuación 7.19.

En cuanto a los paquetes enviados en los *feedbacks*, se envían $(N-1)$ paquetes en el último *round* de la fase i, mientras que no se envía ningún *feedback* en los $X_i - 1$ primeros *rounds* de la fase i. Por tanto, un total de $N \cdot (N-1)$ paquetes son enviados en los N *rounds*. En cuanto al primer *round* de terminación, se envía 1 paquete, mientras que en el segundo *round* de terminación, se envían $(N-1)$ paquetes. Por tanto, para los *feedbacks* se obtiene la ecuación 7.21.

$$P_{sentf} = N \cdot (N-1) + 1 + (N-1) \tag{7.20}$$

$$P_{sentf} = N^2 \tag{7.21}$$

Throughput

En cuanto al *throughput* se calcula contando el promedio del número de paquetes recibidos por nodo, que es $P_{rec} = N - 1$. Esto se obtiene ya que hay N fases para descubrir todos los N-1 vecinos y en cada fase se recibe con éxito 1 paquete. Por tanto un total de $N - 1$ paquetes son recibidos, y el tiempo de descubrimiento de vecinos T_t se obtiene de la ecuación 7.5.

$$P_{rec} = N - 1 \tag{7.22}$$

Se debe añadir el número total de paquetes enviados en los 2 *rounds* de terminación. En el primer y segundo *round* de terminación, ninguno de los nodos envía (están escuchando) por tanto se envían 0 paquetes. Por tanto, P_{rec} permanece como en la ecuación 7.22.

En cuanto a los paquetes recibidos en los *feedbacks*, $N-1$ paquetes son recibidos en el último *round* de la fase i. No se recibe ningún *feedback* en los primeros $X_i - 1$ *rounds* de la fase i. Por lo tanto, un total de $N \cdot (N-1)$ paquetes son recibidos en los N *rounds*. En el primer *round* de terminación, se reciben $(N-1)$ paquetes. En el segundo *round* de terminación, se reciben 0 paquetes. Por lo tanto, para los *feedbacks* se obtiene la ecuación 7.24.

$$P_{recf} = N \cdot (N-1) + (N-1) \tag{7.23}$$

$$P_{recf} = (N+1) \cdot (N-1) = N^2 - 1 \tag{7.24}$$

Se puede obtener el *throughput* de la ecuación 7.25, usando la ecuación 7.22 y la ecuación 7.24. T_{hr} es el *throughput* en paquetes/s, s_b el tamaño del paquete *BROADCAST*, y s_f el tamaño del paquete de *feedback*.

$$T_{hr} = \frac{P_{rec} \cdot s_b + P_{recf} \cdot s_f}{T_t} \tag{7.25}$$

Packet delivery ratio

Para obtener el *packet delivery ratio* en la ecuación 7.26 y el *packet delivery ratio* para los *feedbacks* en la ecuación 7.27, usamos la ecuación 7.19 para P_{sent}, y ecuación 7.22 para P_{rec}. Se usa la ecuación 7.21 para P_{sentf} y ecuación 7.24 para P_{recf}.

$$PDR = \frac{P_{rec}}{P_{sent}} \tag{7.26}$$

$$PDR_f = \frac{P_{recf}}{P_{sentf}} \tag{7.27}$$

Porcentaje de descubrimientos por round

El porcentaje de nodos descubiertos por número de *rounds* se define como el número total de nodos descubiertos (todos ellos) dividido por el número de *rounds* donde se logra el 100% de convergencia. Se muestra en la ecuación 7.28, siendo X obtenido de la ecuación 7.4.

$$PND = \frac{1}{N-1} \cdot \frac{N-1}{X+2} = \frac{1}{X+2} \tag{7.28}$$

CDF de descubrimientos

Asumimos que CDPRR sigue una distribución geométrica. Por tanto, la CDF de descubrimientos para un *round* k se obtiene como sigue:

$$Fx(k) = \begin{cases} 1-(1-p_1)^k \ si \ 1 \leq k \leq X_1 \\ Fx(X_1) + (1-p_2)^{X_1} - (1-p_2)^k \ si \ X_1 < k \leq X_1 + X_2 \\ Fx(X_1+X_2) + (1-p_3)^{X_1+X_2} - (1-p_3)^k \ si \ X_1 + X_2 < k \leq X_1 + X_2 + X_3 \\ ... \end{cases} \tag{7.29}$$

siendo $p_i = (N-i+1) \cdot \frac{1}{N} \cdot (1-\frac{1}{N})^{N-i}$.

Porcentaje de idle slots

De acuerdo con la ecuación 7.30, se muestra el número total de *rounds* en la fase i en el cual todos los restantes nodos están escuchando, siendo $pl_i = (1-\frac{1}{N})^{N-i+1}$, y $X_i = \frac{N}{(N-i+1)\cdot(1-\frac{1}{N})^{N-i}}$. El número total de *idle slots* $E(NIS)$ viene dado en la ecuación 7.31.

$$\begin{aligned} E(NIS_i) &= \sum_{k=0}^{X_i} k \cdot \binom{X_i}{k} \cdot pl_i^k \cdot (1-pl_i)^{X_i-k} = X_i \cdot pl_i \\ &= \frac{N}{(N-i+1)\cdot(1-\frac{1}{N})^{N-i}} \cdot (1-\frac{1}{N})^{N-i+1} \\ &= \frac{N \cdot (1-\frac{1}{N})}{N-i+1} \end{aligned} \tag{7.30}$$

$$\begin{aligned} E(NIS) &= \sum_{i=1}^{N} E(NIS_i) + 2 = \sum_{i=1}^{N} [\frac{N(1-\frac{1}{N})}{N-i+1}] + 2 \\ &= N \cdot (1-\frac{1}{N}) \cdot [\sum_{j=1}^{N} \frac{1}{j}] + 2 = N \cdot (1-\frac{1}{N}) \cdot H_N + 2 \end{aligned} \tag{7.31}$$

siendo H_N el número Harmónico.

Y el porcentaje de *idle slots* se puede encontrar en la ecuación 7.32, siendo X dado en la ecuación 7.4.

Tabla 7.2: Definición de variables para CDH.

Variable	Definición
τ	El tiempo que un nodo está transmitiendo en segundos.
ω	El tamaño del primer *sub-slot* en segundos.
ω_f	La duración del segundo *sub-slot* en segundos.
τ_f	Duración de un paquete de *feedback*.
$\omega - \tau$	El tiempo que un nodo está escuchando en segundos en el primer *sub-slot*.
r	El número total de *rounds* para descubrir todos los vecinos.
N	El número de nodos de la red.
a	Probabilidad de que dos nodos colisionen.
t_i	Un tiempo aleatorio en el cual el nodo transmitirá un *BROADCAST*.
n_i	El valor esperado del número de nodos que transmiten con éxito en el *round* i.
$P(C_i)$	La probabilidad de que un nodo i colisione.
$P(S_i)$	La probabilidad de transmisión con éxito.
p_k	La probabilidad de que un nodo transmita con éxito en el *round* k.
Y_i	Número de transmisiones con éxito en el *round* i.
T_t	El promedio del tiempo de descubrimiento de vecinos en segundos.
Pr	El total número de paquetes recibidos en los r *rounds*.
T_{hr}	El *throughput* en paquetes/s.
s_b	El tamaño del paquete *BROADCAST* en bytes.
s_f	El tamaño de los paquetes de *feedback* en bytes.
$E(P_i)$	El valor esperado del consumo de energía en el *round* i.
$E(P_f)$	Promedio de energía consumida por cada nodo durante los *feedbacks*.
$E(P)$	El promedio del consumo energético por cada nodo en Julios.
$E(P_t)$	El total del consumo energético en Julios, que incluye los *feedbacks*.
P_{rec}	Número total de paquetes recibidos por cada nodo en los r *rounds*.
P_{recf}	Número total de paquetes de *feedback* recibidos.
P_{sent}	El número total de paquetes enviados en el primer *sub-slot*.
P_{sentf}	El número total de paquetes de *feedback* enviados.
P_{sentT}	El número total de paquetes enviados.
E_{tx}	La cantidad de energía consumida por un nodo transmitiendo por segundo.
E_l	La cantidad de energía consumida por un nodo escuchando por segundo.
PDR	*Packet delivery ratio*.
PDR_f	*Packet delivery ratio* para los *feedbacks*.
PND	Porcentaje de descubrimientos por *round*.
$Fx(k)$	CDF de los descubrimientos.
PIS	Porcentaje de *idle slots*.

$$PIS = \frac{E(NIS)}{X+2} = \frac{N \cdot (1 - \frac{1}{N}) \cdot [\sum_{j=1}^{N} \frac{1}{j}] + 2}{X+2} = \frac{N \cdot (1 - \frac{1}{N}) \cdot H_N + 2}{X+2} \tag{7.32}$$

7.2.2 CDH

Para resumir, las variables usadas en el análisis para CDH se definen en la Tabla 7.2.

Para el análisis, se asume que el protocolo consiste en diferentes *rounds*, y un número de 0 o más nodos pueden ser descubiertos en un *round*. Llamamos n_i al número de nodos que logra transmitir con éxito en el *round* i. Tras r *rounds* todos los vecinos han sido descubiertos en $r \cdot (\omega + \omega_f)$ segundos, es decir, el tiempo que el algoritmo tarda en descubrir todos los vecinos.

Se asume una distribución uniforme $t_i \sim U(0, \omega - \tau)$, ya que para todos los intervalos de igual longitud (τ) en la distribución en su rango ($[0, \omega - \tau]$) son igualmente probables. Resaltar que el tiempo está ranurado, con tamaño del primer *sub-slot* ω, en el cual los nodos pueden transmitir. Lo hacen iniciando en un tiempo elegido de forma aleatoria t_i de acuerdo con una distribución $U(0, \omega - \tau)$ en este *sub-slot*, y un segundo *sub-slot* se usa para los *feedbacks*.

Tiempo de descubrimiento de vecinos

Cuando hablamos del tiempo de descubrimiento de vecinos, nos referimos al tiempo que tarda el algoritmo en descubrir todos los vecinos.

Asumiendo una distribución uniforme, la siguiente propiedad puede ser aplicada para obtener la probabilidad de que para dos nodos i y j sus mensajes de *BROADCAST* se solapan:

$$P((t_i \leq t_j \leq t_i + \tau) \cup (t_i - \tau \leq t_j \leq t_i)) = \frac{\tau}{\omega - \tau} \tag{7.33}$$

$P(C_i)$ es la probabilidad de que un nodo i colisione es la unión de colisiones con los otros nodos en el *round* 1. En el *round* 1, hay N nodos que no transmitieron con éxito, siendo $P(C_{i,j})$ la probabilidad de que un nodo i colisione con un nodo j.

$$P(C_i) = P(C_{i,1} \cup C_{i,2} \cup \cdots \cup C_{i,N-1}) \tag{7.34}$$

Aplicando la ecuación para la unión de probabilidades, y definiendo $a = \frac{\tau}{\omega - \tau}$ para simplificar las ecuaciones, obtenemos la siguiente ecuación.

$$P(C_i) = 1 - (1 - a)^{N-1} \tag{7.35}$$

En cuanto a la probabilidad de transmisión con éxito $P(S_i)$ en el *round* 1, siendo S_i el evento de que un nodo i transmita con éxito.

$$p_1 = P(S_i) = 1 - P(C_i) = (1-a)^{N-1} \tag{7.36}$$

Para el *round* 2, obtenemos las siguientes probabilidades, siendo n_1 el número de nodos que transmitieron con éxito en el *round* 1.

$$P(C_i) = P(C_{i,1} \cup C_{i,2} \cup \cdots \cup C_{i,N-n_1-1}) \tag{7.37}$$

Y la probabilidad para el *round* k, viene dada por la siguiente ecuación.

$$P(C_i) = P(C_{i,1} \cup C_{i,2} \cup \cdots \cup C_{i,N-n_1-\cdots-n_{k-1}-1}) \tag{7.38}$$

Por lo tanto, obtenemos la ecuación general, para el *round* k (p_k). Todos los nodos en el *round* k tienen la misma probabilidad de éxito $P(S_i)$. A partir de ahora, llamaremos p_k a la probabilidad de que un nodo transmita con éxito en el *round* k.

$$p_k = P(S_i) = 1 - P(C_i) = (1-a)^{N-n_i\cdots-n_{k-1}-1} \tag{7.39}$$

A continuación averiguamos el número de nodos que transmiten con éxito en el *round* 1 (n_1). En el *round* 1 se cumple la siguiente ecuación, ya que n_1 nodos transmiten con éxito, siendo $p_1 = (1-a)^{N-1}$, Y_1 el número de transmisiones con éxito en el *round* 1, y $Y_1 \sim B(N, p_1)$ sigue una distribución binomial. Una vez que se ha calculado p_i usando una distribución uniforme, usamos una distribución binomial para contar el número de nodos que transmite con éxito.

$$n_1 = E(Y_1) = \sum_{x=0}^{N} x \cdot \binom{N}{x} p_1^x \cdot (1-p_1)^{N-x} \tag{7.40}$$

$$n_1 = N \cdot (1-a)^{N-1} \tag{7.41}$$

Y para el *round* k, el valor esperado del número de nodos que transmiten con éxito es n_k, que viene dado en la ecuación 7.42, siendo $p_k = (1-a)^{N-\sum_{i=1}^{k-1} n_i - 1}$.

$$n_k = (N - \sum_{i=1}^{k-1} n_i) \cdot p_k = (N - \sum_{i=1}^{k-1} n_i) \cdot (1-a)^{N-\sum_{i=1}^{k-1} n_i - 1} \tag{7.42}$$

A continuación, se calcula el número total de *rounds* r después del cual el algoritmo finaliza, es decir, los N nodos han sido descubiertos, de la ecuación 7.43. Dado que la expresión es difícil de derivar, solo se muestran los resultados obtenidos en la sección donde se muestran las gráficas 7.4.

$$n_1 + n_2 + \cdots n_r = N \tag{7.43}$$

Los *rounds* incluyen un segundo *sub-slot*, es decir, un mecanismo de *feedback*, de tamaño ω_f. Para obtener el tiempo de descubrimiento de vecinos, usamos el valor del número total de *rounds* r obtenido de la ecuación 7.43. Tenemos en cuenta que la duración de *round* es $\omega + \omega_f$ y ω_f es la duración del segundo *sub-slot* (*feedbacks*). Así, el tiempo de descubrimiento de vecinos en segundos puede ser obtenido de la ecuación 7.44. Resaltar que el protocolo incluye un *round* de terminación, por tanto el número de *rounds* debe ser $(r + 1)$.

$$T_t = (r + 1) \cdot (\omega + \omega_f) \tag{7.44}$$

Consumo energético

En el *round* 1, N nodos transmiten en una duración τ y N escuchan en una duración $\omega - \tau$. Por lo tanto el valor esperado de la energía consumida en el *round* 1 ($E(P_1)$) en Julios viene dada por la ecuación 7.45. E_{tx} es la cantidad de energía consumida por un nodo transmitiendo por segundo. E_l es la cantidad de energía consumida por un nodo escuchando por segundo.

$$E(P_1) = \tau \cdot E_{tx} \cdot N + (\omega - \tau) \cdot E_l \cdot N \tag{7.45}$$

Asumiendo que n_1 nodos transmitieron con éxito en el *round* 1, en el *round* 2 los nodos restantes que no han sido descubiertos en el *round* 1, es decir, $N - n_1$ nodos transmiten en una duración τ. Los mismos $N - n_1$ nodos escuchan en una duración $\omega - \tau$. Los nodos que transmitieron con éxito en el *round* 1, es decir, n_1 permanecen escuchando durante toda la ranura ω. Por lo tanto, el valor esperado de la energía consumida en el *round* 2 ($E(P_2)$). El valor esperado de la energía consumida en el *round* 3 ($E(P_3)$). Ambos vienen dados a continuación. El promedio del consumo enerético $E(P)$ por cada nodo en Julios viene dado por la ecuación 7.49. Añadimos el promedio de energía consumida durante el *round* de terminación, esto es, $\omega \cdot E_l$.

$$E(P_2) = \tau \cdot E_{tx} \cdot (N - n_1) + (\omega - \tau) \cdot E_l \cdot (N - n_1) + \omega \cdot E_l \cdot n_1 \tag{7.46}$$

$$E(P_3) = \tau \cdot E_{tx} \cdot (N - n_1 - n_2) + (\omega - \tau) \cdot E_l \cdot (N - n_1 - n_2) + \omega \cdot E_l \cdot (n_1 + n_2) \tag{7.47}$$

$$E(P) = \frac{1}{N} \cdot \sum_{k=1}^{r} E(P_k) + \omega \cdot E_l \tag{7.48}$$

$$E(P) = \frac{1}{N} \cdot [\tau E_{tx} \cdot N \cdot r + (\omega - \tau) \cdot E_l \cdot N \cdot r - (\tau \cdot E_{tx} + (\omega - \tau) \cdot E_l - \omega \cdot E_l) \cdot \sum_{k=1}^{r-1} \sum_{i=1}^{k} n_i] + \omega \cdot E_l \tag{7.49}$$

A continuación, añadimos el promedio de la energía consumida por cada nodo durante los *feedbacks*, esto es, $E(P_f)$.

El número de *rounds* r se obtiene de la ecuación 7.43.

El promedio del consumo energético por cada nodo ($E(Pf)$) en Julios para los *feedbacks* viene dado por la ecuación 7.50. Se ha añadido el promedio de la energía consumida en el *round* de terminación $\tau_f \cdot N \cdot E_l$. El total de consumo energético viene dado por la ecuación 7.51.

$$E(P_f) = \frac{\tau_f}{N} \cdot \sum_{i=1}^{r} [n_i \cdot [(n_i - 1) \cdot E_{tx} + (N - n_i + 1) \cdot E_l] + [(N - n_i) \cdot [n_i \cdot E_{tx} + (N - n_i) \cdot E_l]] + \tau_f \cdot N \cdot E_l \tag{7.50}$$

$$E(P_t) = E(P) + E(P_f) \tag{7.51}$$

Overhead

Para obtener el número total de paquetes enviados (P_{sent}) en el primer *sub-slot*, añadimos el número de paquetes que son enviados en el primer *sub-slot* de cada *round*. En el *round* 1 todos los N nodos transmiten 1 paquete, en el *round* 2 sólo los restantes nodos que no transmitieron con éxito en el *round* 1, es decir, $(N - n_1)$ nodos transmiten 1 paquete. Concluimos que en el *round* 3 sólo $N - n_1 - n_2$ nodos no han logrado transmitir con éxito en *rounds* previos, por tanto $N - n_1 - n_2$ nodos transmiten 1 paquete.

$$P_{sent} = N + (N - n_1) + (N - n_1 - n_2) + \cdots + (N - n_1 - \cdots - n_{r-1}) \quad (7.52)$$

$$P_{sent} = N \cdot r - \sum_{k=1}^{r-1} \sum_{i=1}^{k} n_i \quad (7.53)$$

Podemos solucionar la ecuación 7.53 usando r de la ecuación 7.43 y ecuación 7.42. A P_{sent} se debe añadir los paquetes enviados en el *round* de terminación, en el cual se envían 0 nodos, por tanto el P_{sent} no es modificado.

Debemos obtener los paquetes de *feedback* enviados (segundo *sub-slot*). En el *feedback* de cada *round* i, $(n_i - 1) \cdot n_i + n_i \cdot (N - n_i)$ paquetes son enviados. Así el número total de paquetes de *feedback* enviados en r *rounds* viene dado por la ecuación 7.54.

$$P_{sentf} = \sum_{k=1}^{r} [(n_i - 1) \cdot n_i + n_i \cdot (N - n_i)] = (N - 1) \cdot N \quad (7.54)$$

En el *round* de terminación, se envían 0 paquetes de *feedback*, por tanto P_{sentf} no es modificada. El total de paquetes enviados viene dado por la ecuación 7.55 usando la ecuación 7.53 y la ecuación 7.54.

$$P_{sentT} = P_{sent} + P_{sentf} \quad (7.55)$$

Throughput

En el *round* 1, $N-1$ nodos reciben n_1 paquetes cada uno, mientras que en el *round* 2, $N-1$ nodos reciben n_2 paquetes cada uno. Así, obtenemos P_{rec} como el número total de paquetes recibidos en la ecuación 7.56.

$$P_{rec} = \sum_{k=1}^{r}[(N-1)\cdot n_k] = (N-1)\cdot N \tag{7.56}$$

En el *round* de terminación, 0 paquetes son recibidos, por tanto el P_{rec} no es modificado. En cuanto a los *feedbacks*, en el *round* i, $(N-1)\cdot n_i$ total paquetes son recibidos. P_{recf} es obtenida de la ecuación 7.57.

$$P_{recf} = \sum_{i=1}^{r}[n_i\cdot(N-1)] = N\cdot(N-1) \tag{7.57}$$

En el *round* de terminación, 0 paquetes de *feedback* son recibidos, por tanto P_{recf} no es modificada.

Para el *throughput* T_{hr} en paquetes/s, mostrado en la ecuación 7.58, usamos T_t de la ecuación 7.44, ecuación 7.56 para los P_{rec} y ecuación 7.57 para los P_{recf}. s_b es el tamaño de paquete de los *BROADCAST*, y s_f el tamaño de los paquetes de *feedback*.

$$T_{hr} = \frac{P_{rec}\cdot s_b + P_{recf}\cdot s_f}{T_t} \tag{7.58}$$

Packet delivery ratio

El *packet delivery ratio* se obtiene de la ecuación 7.59, usando P_{sent} de la ecuación 7.53 y P_{rec} de la ecuación 7.56.

$$PDR = \frac{P_{rec}}{P_{sent}} \tag{7.59}$$

En cuanto al *packet delivery ratio* para los *feedbacks*, se obtiene de la ecuación 7.60, usando P_{sentf} de la ecuación 7.54 y P_{recf} de la ecuación 7.57.

$$PDR_f = \frac{P_{recf}}{P_{sentf}} \tag{7.60}$$

Porcentaje de descubrimientos por round

El porcentaje de nodos descubiertos por número de *rounds* se define como el número total de nodos descubiertos (todos ellos) dividido por el número de *rounds* donde el 100% de convergencia es lograda. Se muestra en la ecuación 7.61, siendo r obtenido de la ecuación 7.43.

$$PND = \frac{1}{N-1} \cdot \frac{N-1}{r+1} = \frac{1}{r+1} \tag{7.61}$$

CDF de descubrimientos

La CDF de descubrimientos para un *round* k representa una CDF para indicar cuánto tarda para la convergencia a llegar al 100%.

La probabilidad de descubrimiento en el *round* k (p_k) viene dada en la ecuación 7.39 y el número de vecinos descubiertos en el *round* k (n_k) viene dado en la ecuación 7.42. Teniendo en cuenta ambas ecuaciones, la CDF de los descubrimientos para un *round* k puede ser obtenida como sigue:

$$Fx(k) = P(X \leq k) = \sum_{x=0}^{k} \binom{r}{x} \cdot p_x^x \cdot (1-p_x)^{r-x} \tag{7.62}$$

siendo $p_x = (1-a)^{N-\sum_{i=1}^{x-1} n_i - 1}$, $n_x = (N - \sum_{i=1}^{x-1} n_i) \cdot p_x$, y r obtenida de la ecuación 7.43.

Porcentaje de idle slots

CDH sólo genera 1 *idle slot*, es decir, el *round* de terminación. Por tanto, la probabilidad de generar *idle slots* viene dada por la ecuación 7.63

$$PIS = \frac{1}{r+1} \tag{7.63}$$

Dado que el porcentaje de *idle slots* es muy bajo, no lo incluimos en la sección 7.4 donde figuran las gráficas de prestaciones.

7.3 Análisis de protocolos de referencia

Con el propósito de comparación, incluimos el análisis de dos protocolos aleatorios elegidos de la literatura: Hello [22] y PRR [21].

CDPRR es similar al PRR mientras que CDH es similar al Hello, pero en el caso de PRR y Hello son *one-way* (no basados en *handshake*) y todos los nodos compiten durante todos los *rounds*. Ningún nodo deja de competir cuando transmite con éxito. Por lo tanto, PRR y Hello son apropiados para ser comparados y se han elegido como referencia. Dado que no hay un modelo analítico completo disponible para PRR y Hello, se ha desarrollado un modelo analítico para esos protocolos de referencia. Además, el modelo analítico para PRR se basa en el de CDPRR pero ningún nodo deja de competir y no hay mecanismo de detección de colisiones. El modelo analítico para Hello se basa en el modelo obtenido para CDH pero ningún nodo deja de competir y no se usa ningún mecanismo de detección de colisión. Los modelos analíticos para CDPRR y CDH se incluyen en la sección 7.2.

7.3.1 PRR

En PRR, el tiempo está ranurado en *rounds*, y en todos los *rounds* cada nodo elige transmitir con probabilidad $\frac{1}{N}$ o escuchar con probabilidad $1 - \frac{1}{N}$. Además, el protocolo no está basado en *handshake*, ningún nodo deja de competir. Por lo tanto el número de *rounds* (Nr), después del cual el protocolo finaliza, debe ser fijado cuidadosamente.

Las variables usadas en el analísis de PRR se definen en la Tabla 7.3.

El tiempo de descubrimiento de vecinos es el número de *rounds* multiplicado por la duración del *round*, en segundos. Obtenemos el tiempo de descubrimiento de vecinos en segundos de la ecuación 7.64.

$$T_t = Nr \cdot \tau \tag{7.64}$$

En primer lugar, para obtener el consumo energético, calculamos NT como el número de nodos transmitiendo en Nr *rounds* y NL como el número de nodos escuchando en Nr *rounds*. Siguen una distribución binomial: $NT \sim B(n, p)$

Tabla 7.3: Definición de variables para PRR.

Variable	Definición
Nr	Número de *rounds* (valor fijo).
τ	El tiempo que un nodo está transmitiendo en segundos.
N	El número de nodos de la red.
$\frac{1}{N}$	Probabilidad de que un nodo transmita.
$1-\frac{1}{N}$	Probabilidad de que un nodo escuche.
T_t	El tiempo de descubrimiento de vecinos en segundos.
NT	El número total de nodos transmitiendo en los Nr *rounds*.
NL	El número total de nodos escuchando en los Nr *rounds*.
p	Probabilidad de que un nodo transmita en un *round* ($\frac{1}{N}$).
q	Probabilidad de que un nodo escuche en un *round* ($1-\frac{1}{N}$).
n	El número de experimentos en el cual los nodos pueden transmitir o escuchar.
$E(NT)$	El valor esperado del número de transmisiones en Nr *rounds*.
$E(NL)$	El valor esperado del número de escuchas en Nr *rounds*.
$E(P_1)$	El valor esperado del consumo energético en Julios en el *round* 1.
$E(P)$	El promedio del consumo energético por cada nodo en Julios en los Nr *rounds*.
Y	El número de *rounds* en los cuales 1 solo nodo transmite con éxito en Nr *rounds*.
p_s	La probabilidad de que un nodo transmita con éxito en un *round*.
P_{rec}	El número total de paquetes recibidos con éxito por cada nodo en Nr *rounds*.
P_{sent}	El número total de paquetes enviados.
T_{hr}	El *throughput* en paquetes/s.
E_{tx}	La cantidad de energía consumida por un nodo transmitiendo por segundo.
E_l	La cantidad de energía consumida por un nodo escuchando por segundo.
PDR	El *packet delivery ratio*.
PND	Porcentaje de descubrimientos por *round*.
$F(k;Nr,p)$	CDF de descubrimientos.
$E(NIS)$	Número total de *rounds* en los cuales todos los nodos están escuchando.
PIS	Porcentaje de *idle slots*.

para contar el número de transmisiones en varios experimentos n y $p = \frac{1}{N}$. $NL \sim B(n,q)$ para contar el número de escuchas en varios experimentos n siendo $q = 1 - \frac{1}{N}$.

De la ecuación 7.6, obtenemos $E(NT)$ (el valor esperado del número de nodos transmitiendo), siendo $i = 1$, y $n = N \cdot Nr$ ya que hay N nodos que pueden transmitir o recibir y Nr *rounds*. Por tanto el número total de transmisiones posibles se cuentan en n experimentos. De la ecuación 7.7 obtenemos $E(NL)$ (el valor esperado del número de nodos escuchando) siendo $i = 1$ y $n = N \cdot Nr$ (por el mismo motivo explicado para las transmisiones).

El consumo energético en Julios $E(P_1)$ se obtiene de la ecuación 7.8, usando Nr en vez de X. El promedio del consumo energético por cada nodo ($E(P)$) en Julios se obtiene de la ecuación 7.10. Se ha tenido en cuenta que ningún nodo deja de competir.

$$E(P) = \frac{\tau \cdot Nr}{N} \cdot \left[E_{tx} + E_l \cdot N \cdot \left(1 - \frac{1}{N} \right) \right] \tag{7.65}$$

A continuación se obtiene el número total de paquetes enviados (P_{sent}) en Nr *rounds*. Se tiene en cuenta que NT sigue una distribución binomial como se indicaba antes $NT \sim B(n, \frac{1}{N})$ y $n = N \cdot Nr$, es decir, N nodos en Nr *rounds*. Por tanto se tienen $N \cdot Nr$ experimentos. El *overhead*, es decir, P_{sent} se obtiene de la ecuación 7.67.

$$P_{sent} = E(NT) = \sum_{k=1}^{n} \binom{n}{k} k \cdot \left(\frac{1}{N} \right)^k \cdot \left(1 - \frac{1}{N} \right)^{n-k} = N \cdot Nr \cdot \frac{1}{N} = Nr \tag{7.66}$$

$$P_{sent} = Nr \tag{7.67}$$

Para obtener el *throughput* en paquetes/s, debemos tener en cuenta que en Nr *rounds* el número total de rounds en el cual 1 solo nodo transmite. Por tanto P_{rec} el número total de paquetes que son recibidos por cada nodo viene dado en la ecuación 7.69. Se tiene que $p_s = \frac{1}{N} \cdot (1 - \frac{1}{N})^{N-1}$ y $Y \sim B(Nr, p_s)$ es una distribución binomial.

$$P_{rec} = E(Y) = \sum_{k=0}^{Nr} \binom{Nr}{k} \cdot k \cdot p_s^k \cdot (1 - p_s)^{Nr-k} = Nr \cdot p_s \tag{7.68}$$

$$P_{rec} = \frac{Nr}{N} \cdot \left(1 - \frac{1}{N} \right)^{N-1} \tag{7.69}$$

El *throughput* T_{hr} en paquetes/s se obtiene de la ecuación 7.70. El tiempo de descubrimiento de vecinos T_t es obtenido de la ecuación 7.64 y P_{rec} de la ecuación 7.69.

$$T_{hr} = \frac{P_{rec}}{T_t} = \frac{1}{N \cdot \tau} \cdot \left(1 - \frac{1}{N} \right)^{N-1} \tag{7.70}$$

El *packet delivery ratio* se obtiene de la ecuación 7.71 usando P_{sent} de la ecuación 7.67 y P_{rec} de la ecuación 7.69.

$$PDR = \frac{P_{rec}}{P_{sent}} = \frac{1}{N} \cdot \left(1 - \frac{1}{N}\right)^{N-1} \tag{7.71}$$

En cuanto al porcentaje de descubrimientos por número de *rounds*, se muestra en la ecuación 7.72. En este caso, en diferentes *rounds* el mismo nodo puede ser descubierto 0, 1 o más veces. Se incluye aquí el porcentaje de descubrimientos por *round*.

$$PND = \frac{1}{N-1} \cdot (1 - \frac{1}{N})^{N-1} \tag{7.72}$$

La CDF de los descubrimientos para *round* k viene dada a continuación. Se asume que PRR sigue una distribución binomial $B(n, p)$, y se tiene en cuenta que la probabilidad de transmisión con éxito en un *round* es $p = \frac{1}{N} \cdot (1 - \frac{1}{N})^{N-1}$ y $n = Nr$ el número de *rounds* fijo.

$$F(k; Nr, p) = P(X \leq k) = \sum_{x=0}^{k} \binom{Nr}{x} \cdot p^x \cdot (1-p)^{Nr-x} \tag{7.73}$$

Por lo tanto, obtenemos la siguiente ecuación:

$$F(k; Nr, p) = I_{1-p}(Nr - k, k + 1) \tag{7.74}$$

siendo $I_x(c, d)$ la función beta regularizada.

Se concluye que para PRR, el *throughput* y el *packet delivery ratio* no dependen del número de *rounds* Nr, aunque depende de N. Sin embargo, la CDF de descubrimientos para PRR depende de Nr, esto es, si Nr se incrementa el número de *rounds* para llegar al 100% de convergencia crece. Resaltar que en PRR no es posible saber si todos los vecinos han sido descubiertos cuando se usa un Nr fijo. Si fijamos un Nr mayor, esto resultará en más tiempo de descubrimiento de vecinos y consumo energético. Sin embargo, el número de vecinos descubiertos probablemente crecerá. Además, si fijamos mayor Nr, los paquetes enviados se incrementarán, mientras que el porcentaje de descubrimientos por *round* no variará.

El número total de *rounds* en los que todos los nodos están escuchando en Nr *rounds* viene dado por la ecuación 7.75. La probabilidad de un *idle slot* (todos los nodos están escuchando) en un *round* es $p = (1 - \frac{1}{N})^N$.

$$E(NIS) = \sum_{k=0}^{Nr} k \cdot \binom{Nr}{k} \cdot p^k \cdot (1-p)^{Nr-k} = Nr \cdot p = Nr \cdot (1 - \frac{1}{N})^N \quad (7.75)$$

El porcentaje de *idle slots* viene dado en la ecuación 7.77.

$$PIS = \frac{E(NIS)}{Nr} = \frac{Nr \cdot (1 - \frac{1}{N})^N}{Nr} \quad (7.76)$$

$$PIS = (1 - \frac{1}{N})^N \quad (7.77)$$

Además, si se incrementa Nr el porcentaje de *idle slots* no variará.

7.3.2 Hello

En Hello, el tiempo también está ranurado en *rounds* (de duración ω). En todos los *rounds* cada nodo transmite un solo paquete iniciando en instante de tiempo elegido de forma aleatoria t_i de duración τ. Cada nodo escucha durante el resto de la ranura $\omega - \tau$. Además, el protocolo no está basado en *handshake*, ningún nodo deja de competir. Por tanto el número de *rounds* (Nr), tras el cual el protocolo finaliza, debe ser fijado cuidadosamente.

Las variables usadas para el análisis de Hello se definen en la Tabla 7.4.

El protocolo sigue una distribución uniforme $U(0, t_w)$ para los tiempos t_i. Esto es debido a que para todos los intervalos de igual longitud (τ) en la distribución en su rango ($[0, \omega - \tau]$) son igualmente probables. Esta distribución es la misma que para CDH. Sin embargo, en este caso tenemos en cuenta que ningún nodo deja de competir, por tanto hay siempre N nodos compitiendo.

El tiempo de descubrimiento de vecinos (T_t) en segundos se obtiene de la ecuación 7.78, teniendo en cuenta que hay Nr *rounds* y la duración del *round* es ω.

$$T_t = Nr \cdot \omega \quad (7.78)$$

Tabla 7.4: Definición de variables para Hello.

Variable	Definición
Nr	El número de *rounds* (valor fijo).
ω	El tamaño de la ranura en segundos.
τ	El tiempo que un nodo está transmitiendo en segundos.
$\omega - \tau$	El tiempo que un nodo está escuchando en un *round*.
N	El número de nodos de la red.
t_i	El tiempo en el cual un nodo inicia la transmisión.
T_t	El tiempo de descubrimiento de vecinos en segundos.
$E(P_1)$	El valor esperado del consumo energético en un *round*.
$E(P)$	El promedio del consumo energético por cada nodo en los Nr *rounds*.
n_1	El número de paquetes recibidos con éxito en un *round*.
P_{rec}	El número total de paquetes recibidos por cada nodo en Nr *rounds*.
T_{hr}	El *throughput* en paquetes/s.
P_{sent}	El número total de paquetes enviados en Nr *rounds*.
E_{tx}	La cantidad de energía consumida por un nodo transmitiendo por segundo.
E_l	La cantidad de energía consumida por un nodo escuchando por segundo.
PDR	*Packet delivery ratio*.
PND	Porcentaje de descubrimientos por *round*.
$F(k; Nr, p)$	CDF de los descubrimientos.

Obtenemos el valor esperado del consumo energético $E(P_1)$ en el *round* 1 de la ecuación 7.45. En cuanto al promedio del consumo energético por cada nodo en Julios, es decir, el promedio del consumo energético por cada nodo en Nr *rounds* ($E(P)$) se da en la ecuación 7.79.

$$E(P) = \frac{1}{N} \cdot \sum_{i=1}^{Nr} E(P_1) = Nr \cdot [E_{tx} \cdot \tau + E_l \cdot (\omega - \tau)] \tag{7.79}$$

En cuanto al número total de paquetes enviados (P_{sent}), teniendo en cuenta que N nodos transmiten por *round*, se encuentra en la ecuación 7.80.

$$P_{sent} = Nr \cdot N \tag{7.80}$$

A continuación, obtenemos el *throughput* en paquetes/s. En primer lugar, en 1 *round* (todos los nodos están compitiendo), usamos la ecuación 7.41 de CDH: $n_1 = N \cdot (1-a)^{N-1}$, siendo $a = \frac{\tau}{\omega - \tau}$. En Nr *rounds*, el número total de paquetes recibidos es $Nr \cdot (N-1) \cdot n_1$, y el número total de paquetes recibidos por cada nodo es (P_{rec}):

$$P_{rec} = Nr \cdot (N-1) \cdot N \cdot (1-a)^{N-1} \tag{7.81}$$

Finalmente, el *throughput* T_{hr} en paquetes/s se obtiene de la ecuación 7.82 usando P_{rec} de la ecuación 7.81 y el tiempo de descubrimiento de vecinos T_t obtenido de la ecuación 7.78.

$$T_{hr} = \frac{P_{rec}}{T_t} = \frac{Nr \cdot (N-1) \cdot N \cdot (1-a)^{N-1}}{Nr \cdot \omega} = \frac{1}{\omega} \cdot (N-1) \cdot N \cdot (1-a)^{N-1} \tag{7.82}$$

El *packet delivery ratio* se obtiene de la ecuación 7.83 usando P_{rec} de la ecuación 7.81 y P_{sent} de la ecuación 7.80.

$$PDR = \frac{Nr \cdot (N-1) \cdot N \cdot (1-a)^{N-1}}{Nr \cdot N} = (N-1) \cdot (1-a)^{N-1} \tag{7.83}$$

En cuanto al porcentaje de nodos descubiertos por número de *rounds*, se muestra en la ecuación 7.84. En cada *round*, n_1 nodos son descubiertos obtenidos de la ecuación 7.41.

$$PND = \frac{1}{N-1} \cdot N \cdot (1-a)^{N-1} \tag{7.84}$$

De nuevo, en diferentes *rounds* el mismo nodo puede ser descubierto 0, 1 o más veces.

En cuanto al CDF de descubrimientos para un *round* k, se tiene en cuenta que $p_1 = (1-a)^{N-1}$ la probabilidad de que un nodo transmita con éxito en un *round*, y Nr el número de *rounds* fijo.

$$F(k; Nr, p_1) = P(X \leq k) = \sum_{x=0}^{k} \binom{Nr}{x} \cdot p_1^x \cdot (1-p_1)^{Nr-x} \tag{7.85}$$

Usando $I_x(c,d)$, la función beta regularizada.

$$F(k; Nr, p_1) = I_{1-p_1}(Nr-k, k+1) \tag{7.86}$$

Para Hello, no hay *idle slots*, ya que en todos los *rounds* hay N nodos transmitiendo. Por tanto, no se representará este valor en la sección 7.4.

Se concluye que para Hello, el *throughput* y el *packet delivery ratio* no dependen del número de *rounds* Nr, aunque dependen de N. Sin embargo, la CDF de descubrimientos depende del número de *rounds* Nr fijo, esto es, si Nr se incrementa, el número de *rounds* para llegar al 100% de convergencia crecerá. Resaltar que en Hello no es posible saber si todos los vecinos han sido descubiertos cuando fijamos Nr. De nuevo, si fijamos un mayor Nr, esto resultará en más tiempo de descubrimiento de vecinos y consumo energético. Sin embargo, el número de vecinos descubiertos probablemente se incrementará. Además, si Nr incrementa el número de paquetes enviados crece, mientras que el porcentaje de descubrimientos por *round* no variará.

7.4 Resultados gráficos

A continuación, se procede a presentar los resultados gráficos obtenidos de las ecuaciones mostradas en este capítulo.

Hello [22] y PRR [21] han sido elegidos de la literatura con el objetivo de compararlos con las dos propuestas ya que se han usado para comparar en otros trabajos. Se ha usado el modelo de radio *ZigBee* (*CC2420*). $E_{tx} = 0.0522J$ es la energía consumida por un nodo transmitiendo por segundo. $E_l = 0.068J$ es la energía consumida por un nodo escuchando por segundo. También se ha fijado $\tau = 0.07$ s, tiempo que un nodo está transmitiendo un *BROADCAST*. Para los *feedbacks* $\tau_f = 0.000392s$ es el tiempo que un nodo está transmitiendo un paquete de *feedback*. El tamaño de paquete de *BROADCAST* será 2500 bytes y el tamaño de paquete de *feedback* será de 14 bytes. Para CDH y Hello fijamos el tamaño de ranura $\omega = N \cdot \tau$. Asumimos que todos los nodos tienen el mismo rango de transmisión, y los nodos se despliegan en un escenario *one-hop*. Esto significa que todos los nodos están en el rango de transmisión de todos los demás. El modelo de CDPRR mostrado anteriormente sigue una distribución geométrica $Geo(p_i)$. En cuanto al CDH se usa una distribución uniforme $U(0, \omega - \tau)$. Para su comparación fijamos la duración de Hello Nr a $0.5 \cdot N$ *rounds*, y para PRR Nr a $10 \cdot N$ *rounds*. Esto es debido a que ambos protocolos de referencia logran descubrir todos los vecinos con alta probabilidad (distinta de 1) fijando esos números de *rounds*. La selección de este parámetro debe ser llevada a cabo adecuadamente. El motivo es que a medida que el número de *rounds* crece el número de vecinos descubiertos, el tiempo de descubrimiento de vecinos y el consumo energético también crece. Sin embargo, a medida que el número de *rounds* varía, el *throughput* y el *packet delivery ratio* no varían. Sin embargo, a medida que el número de *rounds* varía el CDF de descubrimientos también variará. No podemos averiguar el número

Tabla 7.5: Parámetros.

Parámetro	Valor
Modelo de radio	CC2420
E_{tx}	0.0522J
E_l	0.068J
Tamaño de paquete *BROADCAST* s_b	2500 bytes
Tamaño de paquete *feedback* s_f	14 bytes
Tamaño de ranura CDH y Hello	$\omega = N \cdot \tau$
Tamaño de ranura CDPRR y PRR	τ
Tamaño de ranura de *feedback* CDH	$\omega_f = N \cdot \tau_f$
Tamaño de ranura de *feedback* CDH (para $25\tau and 50\tau$)	$\omega_f = 100 \cdot \tau_f$
Tamaño de ranura de *feedback* CDPRR	τ_f
τ	0.07s
τ_f	0.000392s
Número de *rounds* Hello	$0.5 \cdot N$
Número de *rounds* PRR	$10 \cdot N$
Variación de ω para CDH	$(N-1) \cdot \tau$, $N \cdot \tau$, $2N \cdot \tau$, $3N \cdot \tau$, $50 \cdot \tau$, $25 \cdot \tau$

de vecinos descubiertos ni para Hello ni para PRR en un modelo analítico. Para CDH con un tamaño de ranura fijo, esto es, $50 \cdot \tau$ y $25 \cdot \tau$, fijamos una duración de *feedback* de $100 \cdot \tau_f$. Para los otros tamaños de ranura, esto es, $\omega = (N-1) \cdot \tau$, $\omega = N \cdot \tau$, $\omega = 2 \cdot N \cdot \tau$ y $\omega = 3 \cdot N \cdot \tau$ la duración del *feedback* depende de N ($N \cdot \tau_f$).

La Tabla 7.5 resume los principales parámetros fijados para obtener los resultados gráficos.

7.4.1 Tiempo de descubrimiento de vecinos

Para obtener el tiempo de descubrimiento de vecinos en segundos representado en la Figura 7.1 usamos las ecuaciones para L_t.

De acuerdo con la Figura 7.1, se muestra que todos los protocolos presentan una tendencia creciente con el número de nodos, y CDH supera a las otras soluciones con respecto a esta métrica. Sin embargo, las prestaciones de CDPRR y CDH son similares y superan al PRR con 10N *rounds*, y Hello con 0.5N *rounds* es el peor.

Por tanto, concluimos que el CDH es más rápido que las otras soluciones ya que logra descubrir todos los vecinos en un tiempo reducido.

Como se muestra en la Figura 7.2, variando el tamaño de ranura (ω), las mejores prestaciones se obtienen cuando fijamos $\omega = 50 \cdot \tau$ para un número de nodos por encima de 50. CDH con $\omega = 25 \cdot \tau$, $\omega = (N-1) \cdot \tau$ y $\omega = N \cdot \tau$

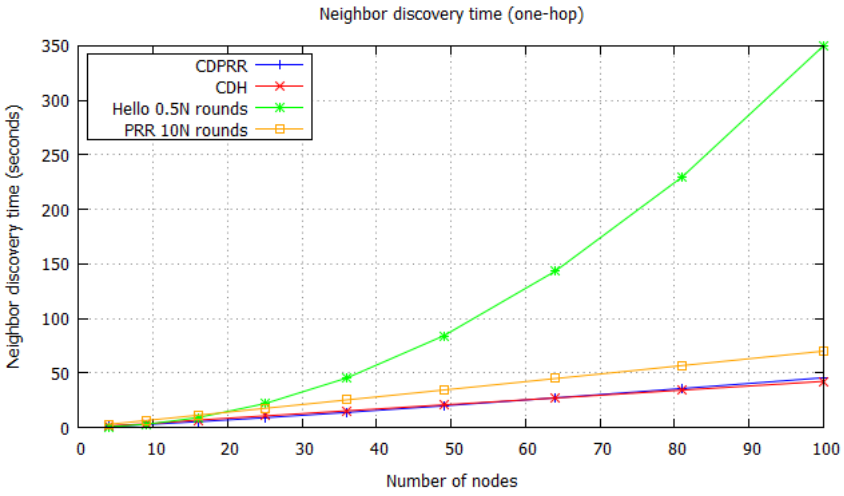

Figura 7.1: Tiempo de descubrimiento de vecinos, comparación (one-hop).

presentan resultados intermedios, seguido por CDH con $\omega = 2 \cdot N$ y CDH $\omega = 3 \cdot N$ es el peor.

CDH con $\omega = 50 \cdot \tau$ para un número de nodos por encima de 50 tiene suficiente tiempo en cada *round* para descubrir los vecinos y no desperdicia tiempo. Sin embargo, CDH con $\omega = 2N \cdot \tau$ y CDH con $\omega = 3N \cdot \tau$ desperdician mucho tiempo presentando *rounds* grandes.

7.4.2 Throughput

Para obtener el *throughput* en byte/s mostrado en la Figura 7.3 se usan las ecuaciones obtenidas de T_{hr} para CDH y CDPRR multiplicando el número de paquetes recibidos por s_b (2500 bytes). A continuación se añade el número de paquetes de *feedback* recibidos multiplicados por s_f (14 bytes). El resultado se divide por el tiempo de descubrimiento de vecinos. En cuanto a Hello y PRR se multiplica el número de paquetes recibidos por s_b (2500 bytes) y se divide por el tiempo de descubrimiento de vecinos.

Todos los protocolos bajo prueba presentan una tendencia decreciente en cuanto al *throughput*.

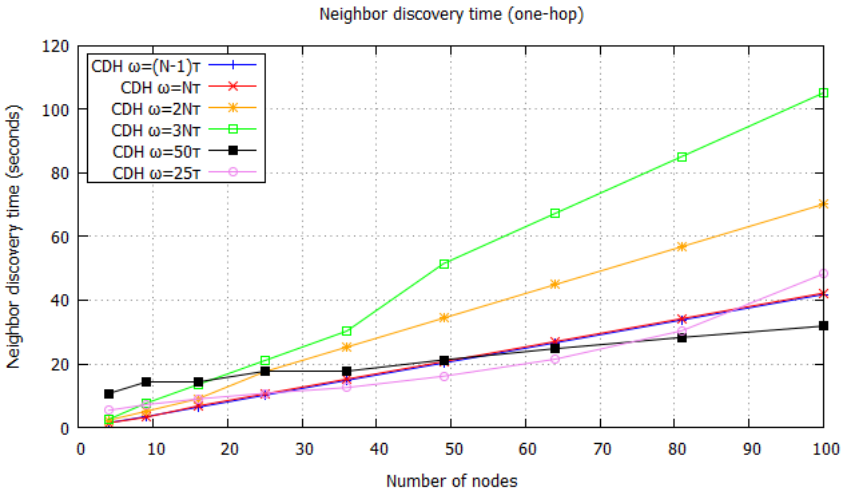

Figura 7.2: Tiempo de descubrimiento de vecinos, comparación (one-hop).

De acuerdo con la Figura 7.3, CDPRR supera a las otras soluciones. CDH presenta mejores prestaciones que PRR con 10N *rounds*, y Hello con 0.5N *rounds* presenta los peores resultados.

Como se muestra en la Figura 7.4, las mejores prestaciones se obtienen cuando fijamos $\omega = (N-1) \cdot \tau$ y $\omega = N \cdot \tau$. Luego CDH con $\omega = 2 \cdot N \cdot \tau$ es mejor que CDH con $\omega = 3 \cdot N \cdot \tau$, CDH con $\omega = 50 \cdot \tau$ es el peor para un número de nodos por debajo de 16. CDH con $\omega = 25 \cdot \tau$ presenta resultados intermedios para número de nodos por debajo de 25.

7.4.3 Consumo energético

La Figura 7.5 presenta el consumo energético en Julios obtenido usando las ecuaciones para el consumo energético.

De acuerdo con la Figura 7.5, se concluye que todos los protocolos presentan una tendencia creciente con el número de nodos. Los resultados siguen la misma tendencia que para los tiempos de descubrimiento en la Figura 7.1. CDH supera a los otros protocolos con respecto al consumo energético. Sin embargo, CDPRR supera al PRR con 10N *rounds*, y el Hello con 0.5N *rounds* es el peor.

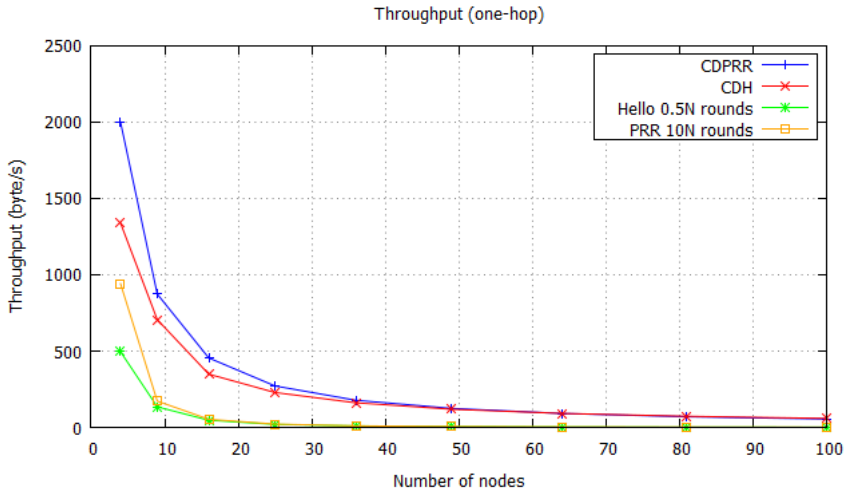

Figura 7.3: Throughput, comparación (one-hop).

De nuevo, CDH consume menos energía dado que el tiempo de descubrimiento de vecinos es menor. CDPRR consume menos energía que PRR con 10N *rounds* y Hello con 0.5N *rounds* dado que tiene un menor tiempo de descubrimiento de vecinos.

En cuanto al consumo energético en la Figura 7.6, variando el tamaño de ranura (ω), los mejores resultados se obtienen cuando se fija $\omega = 25 \cdot \tau$ para número de nodos por debajo de 75. CDH con $\omega = 50 \cdot \tau$, $\omega = (N-1) \cdot \tau$ y $\omega = N \cdot \tau$ presentan resultados intermedios. Luego CDH con $\omega = 2 \cdot N \cdot \tau$ es mejor que CDH con $\omega = 3 \cdot N \cdot \tau$, que es el peor.

La misma conclusión que para la métrica del tiempo de descubrimiento de vecinos es válida para el consumo energético.

Además, concluimos que ambos protocolos CDPRR y CDH logran descubrir todos los vecinos en un escenario *one-hop*.

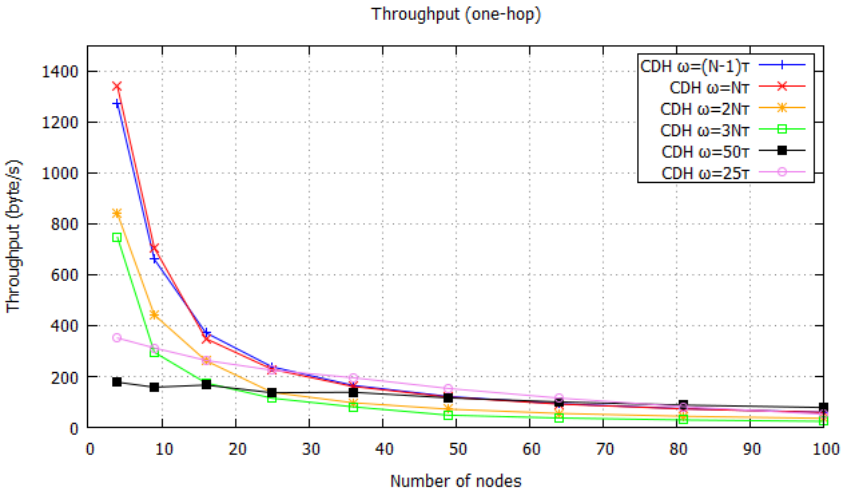

Figura 7.4: Throughput comparación (one-hop).

7.4.4 Número de paquetes enviados

Un menor número de paquetes enviados (*overhead*) significa una ventaja para el protocolo considerado.

De acuerdo con la Figura 7.7, tanto CDH como CDPRR superan al PRR con 10N *rounds* y Hello 0.5N *rounds* es el peor, con respecto a número de paquetes enviados. Además, todos los protocolos siguen una tendencia creciente con el número de nodos.

Resaltar que CDH y CDPRR envían menos paquetes ya que finalizan el proceso de descubrimiento en una menor cantidad de tiempo. Además, a medida que el tiempo pasa, hay más nodos que tienen que ser descubiertos por tanto están escuchando y no envían paquetes.

Como se muestra en la Figura 7.8, CDH con $\omega = 2 \cdot N \cdot \tau$ y $\omega = 3 \cdot N \cdot \tau$ presentan los mejores resultados, seguido por CDH con $\omega = (N-1) \cdot \tau$ y $\omega = N \cdot \tau$. CDH con $\omega = 50 \cdot \tau$ presentan resultados intermedios, mientras que CDH con $\omega = 25 \cdot \tau$ es el peor.

Resaltar que los resultados para $\omega = 50 \cdot \tau$ son mejores que los de $\omega = 25 \cdot \tau$. Esto es debido a que más paquetes enviados son recibidos y más vecinos son descubiertos en un *round*. Por tanto menos nodos están compitiendo en los siguientes *rounds* y el descubrimiento de vecinos finaliza antes. Por tanto

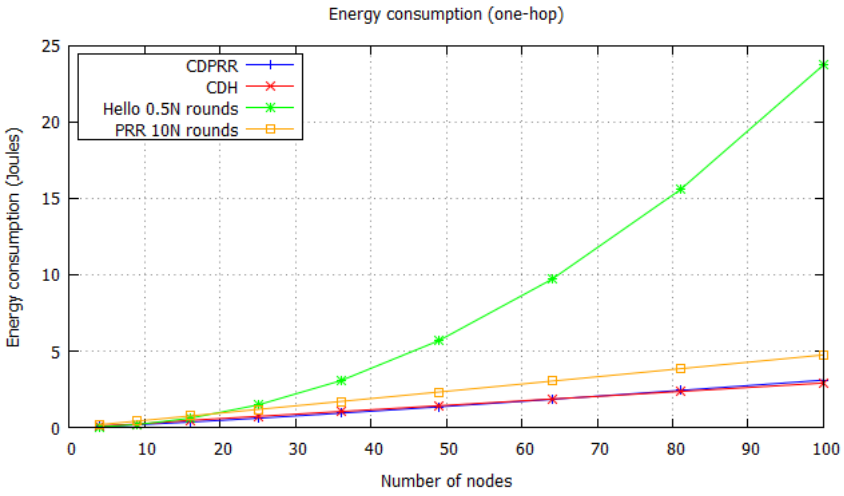

Figura 7.5: Consumo energético, comparación (one-hop).

el número de paquetes enviados es menor. CDH con $\omega = 2N \cdot \tau$ y CDH con $\omega = 3N \cdot \tau$ envían menor número de paquetes. Esto es debido a que su duración de *round* es mayor resultando en más descubrimientos de vecinos por *round*. Por consiguiente, el descubrimiento de vecinos finaliza antes y por tanto menos paquetes son enviados.

La Figura 7.9 muestra el número de paquetes de *feedback* enviados y permite concluir que presenta una tendencia creciente con el número de nodos. El resultado es casi el mismo para CDPRR y CDH para cualquier ω, esto es, aproximadamente N^2 como se puede observar en la ecuación 7.21 y ecuación 7.54. Resaltar que el número de paquetes de *feedback* enviados es fijo y son enviados de forma determinística.

7.4.5 Packet delivery ratio

En esta sección se procede a presentar los resultados obtenidos con respecto a los paquetes recibidos por paquetes enviados, de las ecuaciones obtenidas.

Un alto *packet delivery ratio* significa una ventaja para el protocolo considerado.

De acuerdo con la Figura 7.10, se concluye que CDH supera a las otras soluciones con respecto al *packet delivery ratio*. Hello 0.5N *rounds* también presenta

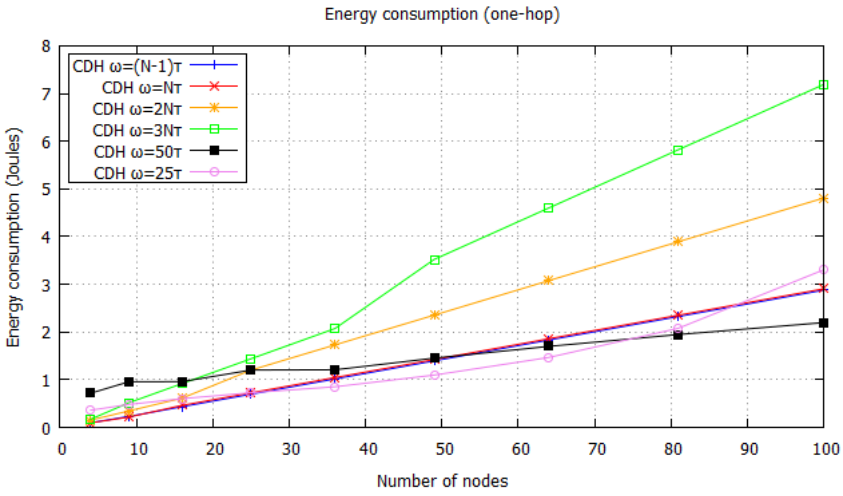

Figura 7.6: Consumo energético, comparación (one-hop).

buenos resultados, mientras que CDPRR presenta mejores resultados que PRR con 10N *rounds*, que es el peor.

Una mayor cantidad de paquetes enviados son recibidos en CDH. Esto es debido a que a medida que el tiempo pasa más nodos han sido descubiertos por tanto menos nodos están enviando paquetes. Por tanto las colisiones se reducen y más paquetes son recibidos.

Como se muestra en la Figura 7.11, CDH con $\omega = 50 \cdot \tau$ es el mejor para número de nodos por debajo de 16, con respecto al *packet delivery ratio*. CDH con $\omega = 25 \cdot \tau$ es el peor para número de nodos por encima de 25. CDH con $\omega = 3 \cdot N \cdot \tau$ es el mejor para número de nodos por encima de 16, seguido por $\omega = 2 \cdot N \cdot \tau$. CDH con $\omega = N \cdot \tau$ presenta resultados intermedios. CDH con $\omega = (N-1) \cdot \tau$ también presenta resultados intermedios, aunque es el peor para número de nodos por debajo de 25.

CDH con $\omega = 50 \cdot \tau$ presenta un mayor *packet delivery ratio* para bajo número de nodos dado que la duración del *round* es mayor por tanto más paquetes enviados son recibidos. Para $\omega = 25 \cdot \tau$ el *packet delivery ratio* es el peor para un número de nodos por encima de 25. Esto es debido a que la duración del *round* es baja y produce más colisiones por tanto el número de paquetes recibidos es menor. CDH con $\omega = 3N \cdot \tau$ es el mejor porque la duración de *round* es mayor por tanto se reciben más paquetes. Resaltar que el *packet*

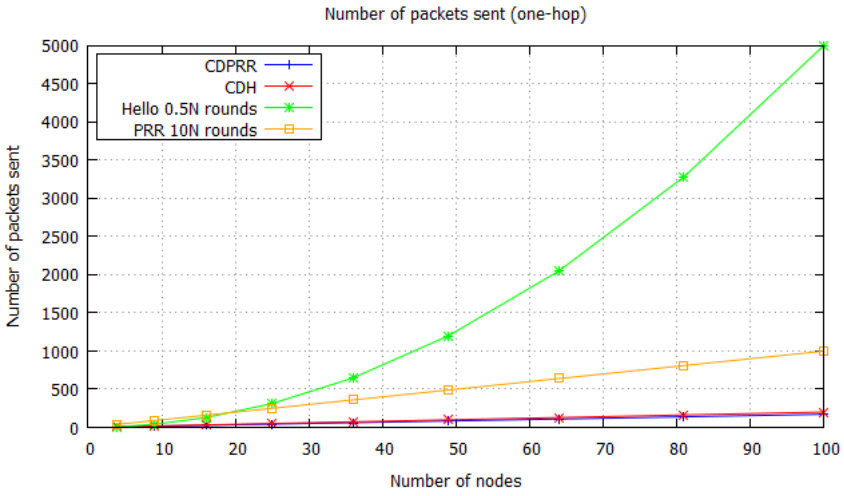

Figura 7.7: Número de paquetes enviados, comparación (one-hop).

delivery ratio cae con un número de nodos creciente para $\omega = 25 \cdot \tau$ y $\omega = 50 \cdot \tau$, y este es el comportamiento esperado.

La Figura 7.12 incluye los resultados del *packet delivery ratio* para los *feedbacks*. CDH supera al CDPRR no importa qué ω fijemos para CDH. Sin embargo, las prestaciones de CDH y CDPRR son casi las mismas para un número de nodos por encima de 9. Este resultado no significa que CDPRR pierda *feedbacks*. La única pérdida de paquetes de *feedback* corresponde al último *round* de terminación ya que 1 nodo para y no recibe los *feedbacks* de los otros N-1 nodos.

Resaltar que en CDH el *packet delivery ratio* para los *feedbacks* es 100% dado que todos los paquetes de *feedback* son recibidos. El comportamiento para el *packet delivery ratio* de ambos protocolos es el esperado ya que los *feedbacks* se envían de forma determinística y todos los paquetes enviados son recibidos.

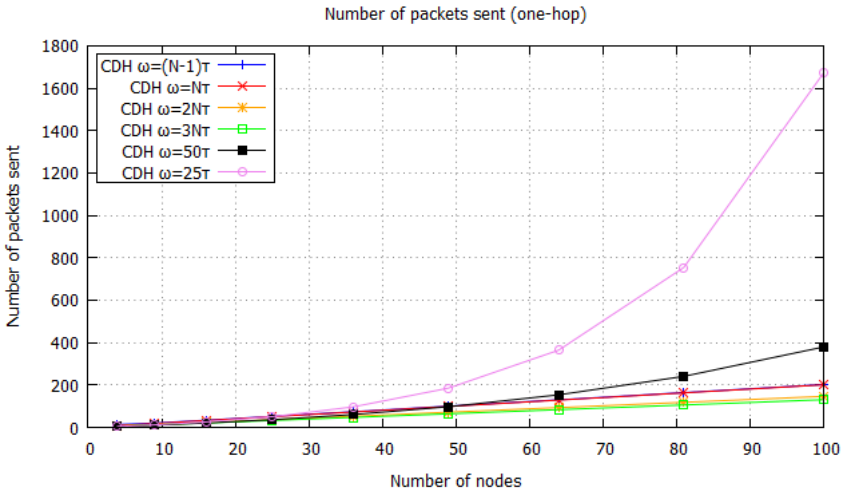

Figura 7.8: Número de paquetes enviados, comparación (one-hop).

7.4.6 Porcentaje de descubrimientos por round

De acuerdo con la Figura 7.13, Hello con 0.5N *rounds* logra más descubrimientos que las otras soluciones, seguida por CDH que supera al PRR 10N *rounds* y finalmente CDPRR es el peor. Sin embargo, en Hello y PRR un nodo puede ser descubierto 0, 1 o más veces en diferentes *rounds*.

Como se muestra en la Figura 7.14, CDH $\omega = 3N \cdot \tau$ presenta las mejores prestaciones, CDH $\omega = (N-1)$ es el peor para número de nodos por debajo de 25. En cuanto al CDH con $\omega = N \cdot \tau$, $\omega = 2N \cdot \tau$ y $\omega = 50 \cdot \tau$, CDH presenta resultados intermedios, y CDH con $\omega = 25 \cdot \tau$ es el peor para número de nodos por encima de 25.

7.4.7 CDF de descubrimientos

Para obtener la CDF de los descubrimientos, fijamos una red de $N = 4$ nodos. Se obtienen resultados similares para redes más grandes.

Cuanto antes llegue la convergencia a 100% mejor será el protocolo.

De acuerdo con la Figura 7.15, CDH necesita menos *rounds* para llegar a la convergencia al 100% que las otras soluciones. Además, CDPRR mejora al PRR 10N *rounds* según la CDF de descubrimientos, y Hello 4N *rounds* es el

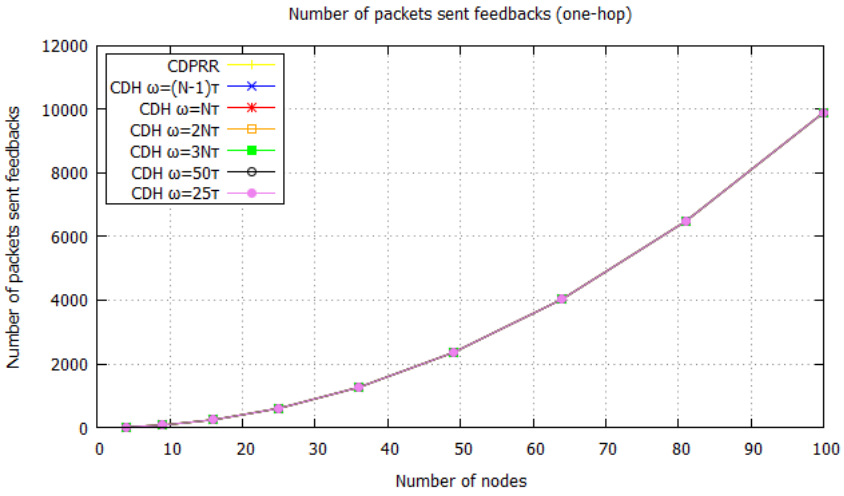

Figura 7.9: Número de paquetes de feedback enviados, comparación (one-hop).

peor. Si se incrementa el número de *rounds* fijado para Hello, el número de vecinos descubiertos crecerá pero el número de *rounds* para llevar a convergencia 100% de descubrimientos y el consumo temporal también crecerá. Además, la duración de los *rounds* fijado para CDH y Hello es mayor ($\omega = N \cdot \tau$) que para CDPRR y PRR.

Como se muestra en la Figura 7.16, CDH $\omega = 3N \cdot \tau$, $\omega = 50 \cdot \tau$ y $\omega = 25 \cdot \tau$ logran descubrir todos los vecinos en 2 rounds presentando la mejor CDF de descubrimientos. Esto es debido a que el tamaño de ranura es mayor que en las otras soluciones y son descubiertos más vecinos en cada *round*. A continuación, CDH $\omega = 2N \cdot \tau$ logra el descubrimiento de todos los vecinos en 3 *rounds*. CDH con $\omega = N \cdot \tau$ logra descubrir todos los vecinos en 4 *rounds*. Finalmente CDH con $\omega = (N-1) \cdot \tau$ es el peor, completando el descubrimiento de todos los vecinos en 6 *rounds*. Esto se debe a que el tamaño de ranura es menor por tanto se descubren menos vecinos en cada *round*.

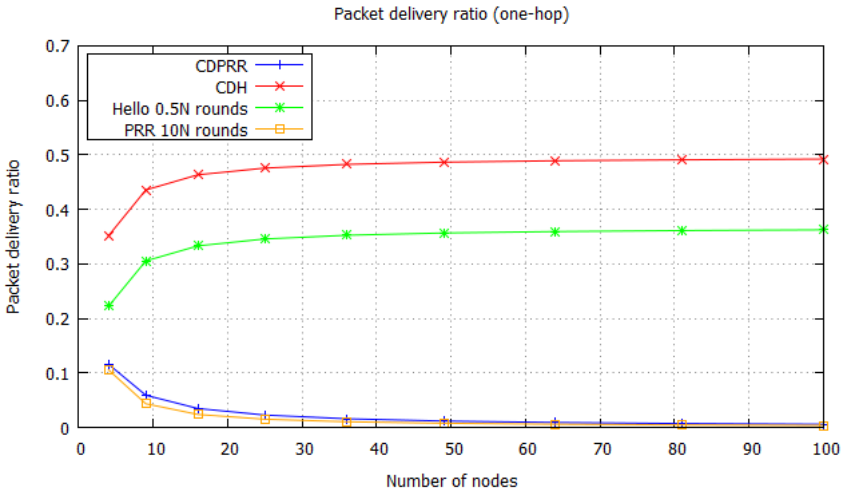

Figura 7.10: Packet delivery ratio, comparación (one-hop).

7.4.8 Porcentaje de idle slots

En cuanto al porcentaje de *idle slots*, tener más *idle slots* sería considerado como una ventaja dado que se consume menos energía y el número de paquetes enviados se reduce. Sin embargo, más *idle slots* producen un incremento en el tiempo de descubrimiento de vecinos.

De acuerdo con la Figura 7.17, CDPRR incluye más porcentaje de *idle slots* que PRR con 10N *rounds*. Además, ambos protocolos siguen una tendencia creciente asintótica con el número de nodos.

CDPRR presenta más *idle slots* dado que a medida que el tiempo pasa hay menos nodos compitiendo por tanto esos nodos están escuchando y la probabilidad de generar *idle slots* crece.

7.5 Discusión

De acuerdo con el modelo analítico, CDH y CDPRR presentan un tiempo de descubrimiento de vecinos lineal $O(N)$, siendo N el número de nodos de la red.

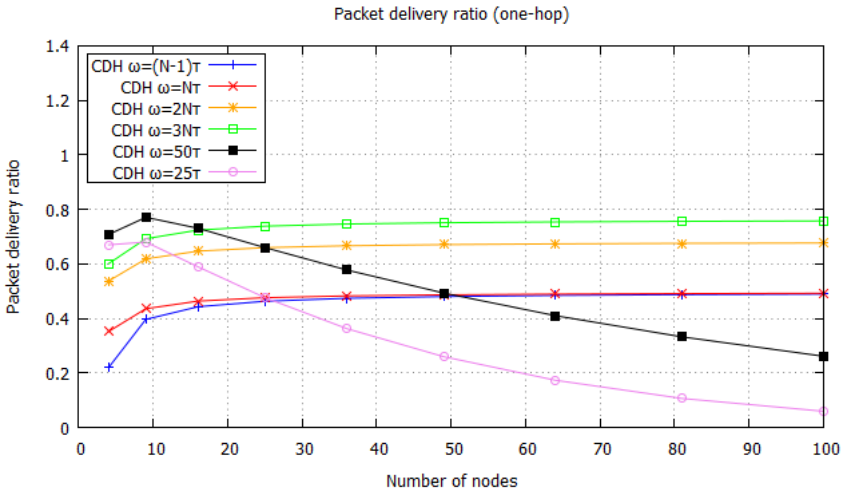

Figura 7.11: Packet delivery ratio, comparación (one-hop).

Además, tanto CDH como CDPRR logran descubrir todos los vecinos con probabilidad 1 en un escenario *one-hop*, mientras que ni Hello ni PRR logran descubrir todos los vecinos con probabilidad 1.

En general, CDPRR y CDH siguen premisas más realistas que protocolos aleatorios existentes.

En cuanto al porcentaje de *idle slots*, tener más *idle slots* sería preferible ya que se consume menos energía. Además, cuando ocurren más *idle slots* el número de paquetes enviados se reduce. Sin embargo, tener más *idle slots* incrementa el tiempo de descubrimiento de vecinos. Además, los nodos que llegan antes al estado S (transmitió con éxito) en CDPRR están siempre escuchando a partir de entonces por tanto se producen más *idle slots*. El *packet delivery ratio* es un fenómeno positivo y un alto *packet delivery ratio* se considera como deseable.

Entre sus limitaciones prácticas, CDPRR presenta un bajo *packet delivery ratio*, necesita conocer el número de nodos, y los nodos no inician la transmisión en diferentes instantes de tiempo. Sin embargo, CDH soluciona estas limitaciones. Tanto CDPRR como CDH necesitan sincronización en los límites de ranura y no pueden se usados en redes móviles, esto es, MANETs, y el tiempo debe estar ranurado. Como posibles formas de solucionar estas limitaciones, un mecanismo de sincronización podría ser usado antes de comenzar el proceso de descubrimiento de vecinos. También se requiere mejorar los protocolos para

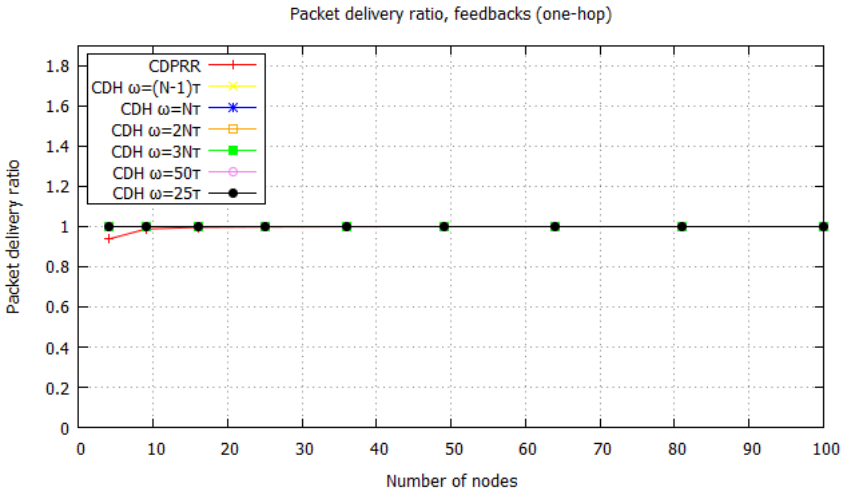

Figura 7.12: Packet delivery ratio para los feedbacks, comparación (one-hop).

permitir a los nodos entrar o salir de la red detectando cuando un nuevo nodo entra en el rango de transmisión de los otros nodos o deja la red, en MANETs.

Como aplicaciones prácticas, CDH es rápido y gasta poca energía, tiene poco *overhead* (paquetes enviados), y un alto *packet delivery ratio*, lo que significa una ventaja. Por tanto es adecuado para ser usado en escenarios prácticos en los cuales las baterías no se pueden recargar con frecuencia. También es adecuado cuando el número de nodos que componen la red es desconocido. Tanto CDH como CDPRR pueden ser usados en entornos estáticos inalámbricos ad hoc o redes espontáneas basadas en la confianza. En esta última, la gente se junta por ejemplo en una reunión para intercambiar información durante un periodo de tiempo.

En cuanto al CDH con tamaño de ranura fijo, presenta resultados similares para $\omega = 50 \cdot \tau$ y $\omega = 25 \cdot \tau$ con respecto al tiempo de descubrimiento de vecinos. También presenta resultados similares en cuanto al consumo energético y el número de *rounds* para llegar a una convergencia 100% de descubrimientos. Las prestaciones para $\omega = 25 \cdot \tau$ son peores con respecto al número de paquetes enviados y *packet delivery ratio*.

De acuerdo con el modelo analítico y los resultados gráficos, concluimos que no hay valor óptimo para todas las métricas variando el tamaño de ranura ω en CDH. Sin embargo, CDH con $\omega = 50 \cdot \tau$ y $\omega = 25 \cdot \tau$ presentan los mejo-

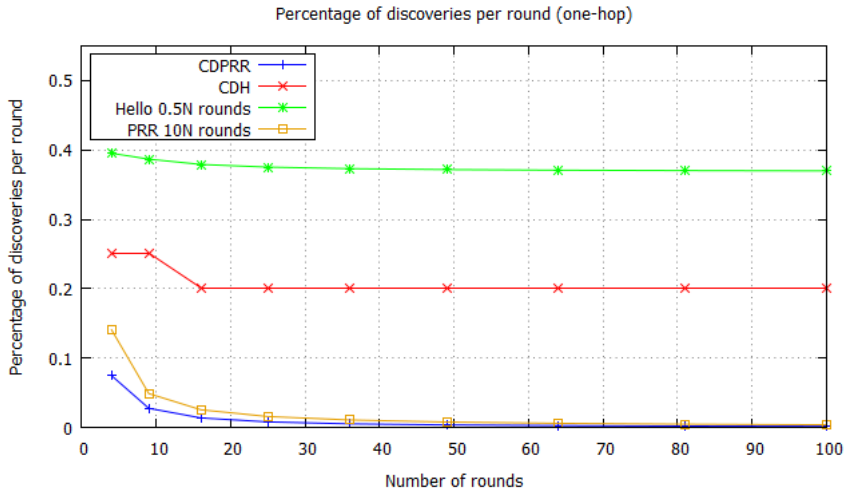

Figura 7.13: Porcentaje de descubrimientos por round, comparación (one-hop).

res resultados con respecto al tiempo de descubrimiento de vecinos y consumo energético. Ambos presentan los peores resultados con respecto al número de paquetes enviados, *packet delivery ratio*, y presentan resultados intermedios con respecto a la CDF de descubrimientos. En cuanto al CDH con $\omega = 2N \cdot \tau$ y $\omega = 3N \cdot \tau$, presentan los peores resultados con respecto al tiempo de descubrimiento de vecinos, consumo energético. Ambos presentan los mejores resultados con respecto al número de paquetes enviados y *packet delivery ratio*. CDH con $\omega = 3N \cdot \tau$ presenta los mejores resultados con respecto a la CDF de descubrimientos. Sin embargo, CDH con $\omega = (N-1) \cdot \tau$ y $\omega = N \cdot \tau$ presenta resultados intermedios en tiempo de descubrimiento de vecinos, consumo energético, paquetes enviados y *packet delivery ratio*. Ambos presentan los peores resultados con respecto a la CDF de descubrimientos.

En caso de que se quiera añadir seguridad a las propuestas, una posible solución puede ser crear una firma de los identificadores usando la clave privada. A continuación, se debe enviar en el paquete de *BROADCAST* el identificador, la clave pública y la firma. El receptor puede comprobar la firma (usando la clave pública) y si hay un error cuando se comprueba la firma esto significa que el mensaje ha sido interceptado y manipulado.

Para mejorar las propuestas para ser usadas en MANETs, deberíamos tener en cuenta que los nodos entren y salgan del rango de transmisión de otros. Estos

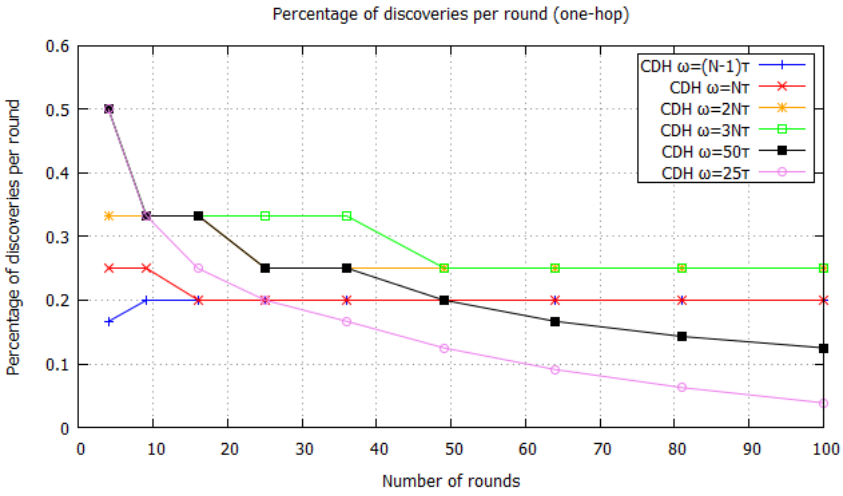

Figura 7.14: Porcentaje de descubrimientos por round, comparación (one-hop).

nodos deben permitir el intercambio de notificaciones de nodos uniéndose y saliendo.

Un escenario realista se da cuando el número real de nodos en la red (N') es menor que el número de nodos conocido (N). En ese caso, en el primer *sub-slot* la probabilidad de colisión será reducida en comparación con cuando $N' = N$. Por tanto la probabilidad de descubrimiento se incrementará. El número de *rounds* tras el cual el protocolo finaliza será reducido por tanto el tiempo de descubrimiento de vecinos también se reducirá. Esto se debe a que el protocolo finaliza cuando los N' nodos han sido descubiertos (y $N' < N$). El total de energía consumida será reducido, y el número de paquetes enviados también se reducirá. El *packet delivery ratio* se incrementará dado que menos nodos están compitiendo por tanto la probabilidad de colisión se reduce y el número de paquetes recibidos se incrementa. El porcentaje de *idle slots* también se incrementará dado que hay menos nodos en la red. Además, en el segundo *sub-slot* se envían menos paquetes de *feedback* dado que hay menos nodos en la red. En ese caso, el consumo energético se decrementa y el *packet delivery ratio* no cambiará. Sin embargo, el protocolo no logrará descubrir todos los vecinos con probabilidad 1.

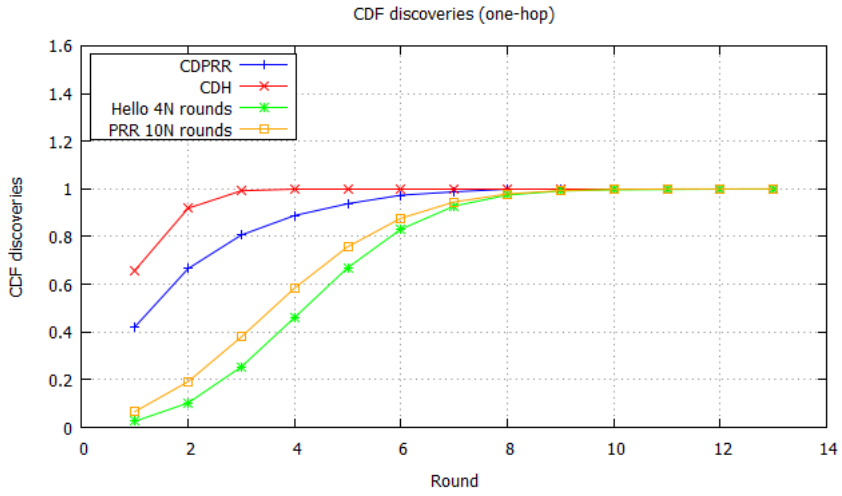

Figura 7.15: CDF de descubrimientos (one-hop).

7.6 Conclusiones

En este capítulo se ha llevado a cabo un estudio analítico sobre dos protocolos de descubrimiento de vecinos aleatorios basados en la detección de colisiones. Están diseñados para entornos *one-hop* estáticos inalámbricos ad hoc.

Para el modelo analítico, usamos una distribución geométrica $Geo(p_i)$ para el CDPRR y una distribución uniforme $U(0, t_u)$ para el CDH.

Para validar y comparar los protocolos se obtuvo un modelo matemático para CDH, CDPRR y dos protocolos de referencia, esto es, Hello y PRR de la literatura. Se han representado los resultados en varias gráficas usando las ecuaciones obtenidas, con respecto a 8 métricas. Se han obtenido gráficas en cuanto a tiempo de descubrimiento de vecinos, *throughput*, consumo energético, y *overhead* (número de paquetes enviados). También se ha representado el *packet delivery ratio*, porcentaje de descubrimientos por *round*, CDF de descubrimientos, y el porcentaje de *idle slots*.

De acuerdo con los resultados analíticos obtenidos en un escenario *one-hop*, CDH supera a las otras soluciones en cuanto al tiempo de descubrimiento de vecinos, y consumo energético. CDH también supera a las otras soluciones en cuanto a número de paquetes enviados, *packet delivery ratio* y CDF de descubrimientos. CDPRR es mejor que CDH en cuanto al *throughput*, y logra

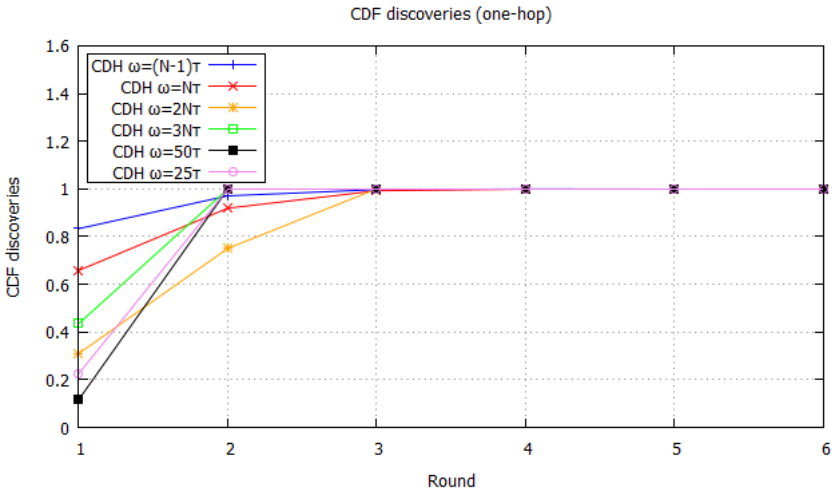

Figura 7.16: CDF de descubrimientos (one-hop).

buenos resultados. CDPRR es mejor que el Hello y PRR en cuanto al tiempo de descubrimiento de vecinos, *throughput*, consumo energético, CDF de descubrimientos y paquetes enviados. En cuanto al porcentaje de descubrimientos por *round*, Hello logra mejores prestaciones que las otras soluciones, seguida por CDH que supera al PRR y finalmente CDPRR es el peor. Sin embargo, en Hello y PRR un nodo puede ser descubierto 0, 1 o más veces en diferentes *rounds*.

Además, descubrimos que CDPRR presenta un mayor porcentaje de *idle slots* que PRR, lo cual es una clara ventaja en consumo energético y número de paquetes enviados.

También centramos el estudio en CDH cuando el tamaño de ranura (ω) se varía, y se demuestra que para CDH el número de nodos de la red puede ser desconocido. En CDH se puede fijar un tamaño de ranura ω que no dependa del número de nodos, y aún así proporcionar resultados razonables.

Como trabajo futuro, se prevee modelar otros protocolos de descubrimientos aleatorios, modelar protocolos para la creación de red espontánea ad hoc basada en la confianza. También es interesante modelar protocolos de descubrimiento de vecinos para MANETs.

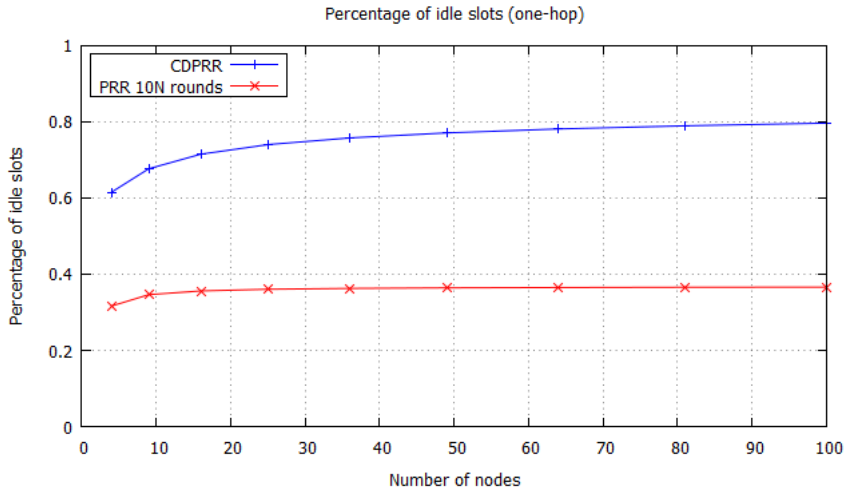

Figura 7.17: Porcentaje de idle slots, comparación (one-hop).

Capítulo 8

Protocolo para la creación de redes espontáneas inalámbricas ad hoc basadas en la confianza

En este capítulo se presenta un nuevo modelo aleatorio de creación de red espontánea inalámbrica ad hoc basada en vecinos de confianza. La idea utilizada es combinar el descubrimiento de vecinos con el intercambio de tarjetas de identidad. Mediante la comprobación de una firma se establece una relación de vecinos basada en la confianza. Para evaluar las prestaciones de la propuesta se compara con un protocolo existente que se usa como referencia, utilizando Castalia 3.2. Las métricas utilizadas son el tiempo de descubrimiento, consumo energético, throughput, y el número de descubrimientos por paquetes enviados. Tras la simulación, comprobamos que la propuesta mejora al protocolo de referencia según el tiempo de descubrimiento, y el consumo energético, en un entorno one-hop. También mejora al protocolo de referencia en cuanto a los descubrimientos por paquetes enviados, también en escenarios one-hop. La propuesta mejora al protocolo de referencia según las 4 métricas anteriores en entornos multi-hop. También se ha evaluado la propuesta con Castalia 3.2 al variar la probabilidad de transmisión. Por tanto, la propuesta no requiere del conocimiento del número de nodos al fijar una probabilidad de transmisión, mostrando buenos resultados. Además la propuesta se basa en la detección de colisiones, conociendo cuándo terminar el proceso de descubrimiento, y no sigue una planificación en la transmisión. También se muestra una comparación cualitativa, de la propuesta con protocolos de la literatura.

8.1 Introducción

Las redes espontáneas se caracterizan por no proporcionar una infraestructura de comunicaciones, y los vecinos no son conocidos justo tras el despliegue. Se usan en un periodo de tiempo en un lugar determinado. Además, en este tipo de redes no hay un servidor central, por tanto no hay disponible ninguna AC centralizada, y cada nodo debe actuar como una AC.

El concepto de redes espontáneas ad hoc fue introducido en [10]. Los dispositivos que la forman (también conocidos como nodos) son autónomos y equipados con transceptores de radio de alcance limitado. Por tanto, sólo algunos nodos pueden comunicar directamente con los nodos en su rango de transmisión, esto es, los vecinos. Otros nodos requiren de múltiples nodos intermedios que reenvían la información no destintada a ellos de una manera *multi-hop*. Para cumplir ese objetivo, cada nodo debe poder actuar como un *router* [11, 12].

Dado que los nodos no conocen a priori a sus vecinos, en la creación de una red espontánea ad hoc, el descubrimiento de vecinos [5, 6] se hace necesario. El objetivo es descubrir los nodos en su rango de transmisión.

Las redes espontáneas son un tipo de redes ad hoc que presenta las siguientes características [13]:

- Los nuevos servicios estarán disponibles sin intervención de usuarios.
- Los nodos se pueden encontrar en una localización física en una cantidad de tiempo determinada.
- Los nodos colaboran en todo momento para proporcionar servicios como comunicación de grupo, seguridad, etc.
- Los nodos pueden unirse o dejar la red a voluntad en cualquier momento y los dispositivos pueden venir de cualquier sitio.
- Estas redes se componen de nodos móviles, por tanto no hay topología fija.
- Estas redes emulan las relaciones humanas para lograr la creación y funcionaminento.
- Estas redes están compuestas de un conjunto de nodos que no se conocen.
- Estas redes deben tener un nivel de seguridad similar al de las redes cableadas tradicionales.

- Cada nodo actúa como *router*.
- Los nodos tienen un rango de transmisión limitado hacia los otros nodos.
- Los nodos tienen recursos limitados, en cuanto a CPU, memoria, y energía (baterías).
- Los nodos móviles se pueden mover libremente en el área donde fueron desplegados, incluso fuera del rango de transmisión de otros nodos.
- El medio de transmisión físico es compartido.
- Las identidades vienen dadas por direcciones IP obtenidas dinámicamente.
- No hay administración central.

En resumen, una red espontánea ad hoc es diferente de una red ad hoc, dado que su uso está restringido a una determinada localización durante un periodo de tiempo. Una red espontánea no depende de un servidor central, y no se requiere que el usuario sea un experto. Además, este tipo de redes imita las relaciones humanas para funcionar conjuntamente en grupos, con intervención mínima del usuario. Generalmente, este tipo de redes usa relaciones de confianza [14], imitando cómo interactúan los humanos, en la creación y administración. Permiten la construcción de cadenas de confianza, también conocidas como *trust net*. Además, más adelante, cuando la red espontánea ya está creada y es necesario encaminamiento ad hoc si un nodo confía en un segundo nodo, puede enviar mensajes directamente a él. En el caso de que un segundo nodo no confíe en el primero, la comunicación no está permitida. Cuando un nodo quiere enviar un mensaje a un nodo que no es de confianza debe hacerlo a través de un nodo de confianza.

Cuando se desea construir una red espontánea ad hoc basada en la confianza, se tiene como idea el usar las relaciones humanas como modelo. Ese modelo sigue un escenario como por ejemplo cuando un grupo de humanos se unen para comunicar, intercambiar información o funcionar conjuntamente por un periodo de tiempo en un lugar determinado.

En la literatura, los protocolos determinísticos necesitan una planificación en la transmisión. Algunos protocolos aleatorios requieren premisas poco realistas tales como no utilizar la detección de colisiones y no saber cuándo terminar.

Por tanto, el principal objetivo de este trabajo es proponer y evaluar protocolos que no utilizan una planificación en la transmisión. Además deben tratar

con las colisiones, funcionar bajo premisas más realistas, y obtener mejores prestaciones que las soluciones existentes. El principal objetivo de crear una red espontánea ad hoc es establecer un servicio de gestión de clave distribuido a través del uso de una red de confianza. Por tanto, las claves públicas sólo será necesario obtenerlas cuando sea necesario en futuras operaciones.

Hay muchas áreas de aplicación [9] de las redes espontáneas ad hoc, que incluyen: la industrial (e.g. comunicación entre sensores, robots y redes digitales), negocio (e.g., reunión, control de existencias). También tienen aplicaciones militares (e.g., entornos duros y hostiles), médicas (e.g. monitorización de pacientes), y en la enseñanza.

Entre posibles ejemplos de este tipo de redes tenemos una red de sensores inalámbricos en un bosque para detectar fuego por un periodo de tiempo determinado. Además se puede incluir sensores en un puente que cuentan el número de vehículos y su velocidad. Los sensores pueden ser desplegados en un lago para estudiar la calidad del agua en un periodo determinado de tiempo.

En este capítulo se presenta un protocolo para la creación de redes espontáneas inalámbricas ad hoc estáticas basadas en la confianza. Se ha implementado en Castalia 3.2 [88] para validarla y compararla con un protocolo de referencia. Se ha tenido en cuenta 4 métricas: tiempo de descubrimiento, consumo energético, *throughput* y número de descubrimientos por paquetes enviados. Así, el protocolo se centra en redes estáticas, esto es, los nodos no se pueden mover en el área de despliegue. Para su uso en redes móviles (MANETs), la propuesta debe ser mejorada si se tiene en cuenta los nodos que entran y salen de la red y los nodos que entran y salen del rango de transmisión de otros.

La propuesta combina el descubrimiento de vecinos con el intercambio de tarjetas de identidad y comprobación de firma para establecer una red basada en la confianza. El intercambio de tarjetas de identidad permite diseminar las claves públicas en toda la red. Tras este intercambio de tarjetas de identidad, la firma se comprueba usando la clave pública y si es correcta, el vecino se considera de confianza. Este tipo de vecinos crea así una red de confianza.

El problema al cual se enfrenta nuestra propuesta es que tiene que tratar con colisiones y lograr descubrir todos los vecinos. También debe intercambiar con éxito las tarjetas de identidad y descubrir todos los vecinos de confianza. No debe seguir una planificación en la transmisión. La propuesta debe mejorar el tiempo de descubrimiento, consumo energético, *throughput*, y número de descubrimientos por paquetes enviados, comparado con trabajos previos. No debe necesitar conocer el número de nodos. Otros protocolos aleatorios no lograban

enfrentarse a este problema dado que no tienen en cuenta las colisiones, o no tratan con ellas, y no saben cuándo terminar el descubrimiento. Un mecanismo probabilístico se introduce, el cual trata con colisiones y descubre todos los vecinos.

Al introducir la propuesta, el problema a que nos enfrentamos es que: (i) Los nodos deben operar en escenarios estáticos. (ii) Los dispositivos están equipados con transceptores de radio de alcance limitado. (iii) Los dispositivos usan solo *half-duplex*. (iv) Los nodos se despliegan de forma aleatoria en un área dada. (v) Los nodos deben poder operar de forma asíncrona. (vi) Los nodos deben tratar con colisiones y deben poder detectarlas. (vii) El número de nodos debe ser desconocido para todos los nodos. (viii) Los nodos deben poder descubrir todos los vecinos con probabilidad 1 (o casi 1). (ix) Los nodos deben saber cuándo terminar el descubrimiento. (x) La propuesta no debe seguir una planificación en la transmisión, y (xi) la propuesta debe poder obtener mejores prestaciones que soluciones existentes.

Más adelante en este capítulo se muestran las diferencias de la propuesta con trabajos relacionados existentes de la literatura, en la Tabla 2.5 presentada en el capítulo 2.

Básicamente, la novedad de este trabajo en relación con trabajos previos es que no usa planificación, no se requiere conocer el número de nodos, y el protocolo está diseñado para entornos estáticos.

Las principales contribuciones de este capítulo son: (i) Proponer un protocolo aleatorio en 2 fases, que logre tratar con y detectar las colisiones, permite detectar cuándo terminar el descubrimiento. El protocolo puede usar una probabilidad de transmisión fija y no requiere del conocimiento del número de nodos. No depende de seguir una planificación en la transmisión. Permite descubrir todos los vecinos, intercambia con éxito las tarjetas de identidad y descubra todos los vecinos de confianza con probabilidad casi 1. Además, sigue premisas más realistas, y es adecuada para ser usada tanto en entornos *one-hop* como *multi-hop*. (ii) Una comparación cualitativa de trabajos relacionados y la propuesta. (iii) Implementación en Castalia 3.2 y comparación de la propuesta con un protocolo de referencia. (iv) Un estudio del comportamiento de la propuesta variando la probabilidad de transmisión. Además, se ha comprobado que la propuesta es más rápida y consume menos energía que soluciones existentes.

De acuerdo con los resultados de simulación, concluimos que la propuesta mejora al protocolo de referencia en tiempo de descubrimiento, y consumo energé-

tico, en un entorno *one-hop*. También mejora en cuanto a descubrimientos por paquete enviado en un escenario *one-hop*. Por otro lado, la propuesta mejora al de referencia en tiempo de descubrimiento, consumo energético, *throughput*, y descubrimientos por paquetes enviados, en entornos *multi-hop*. También se centró la evaluación de la propuesta variando la probabilidad de transmisión. Se ha demostrado que la propuesta no requiere conocer el número de nodos cuando se fija la probabilidad de transmisión, y proporciona buenos resultados.

8.2 Modelo aleatorio de creación de red de confianza en dos fases

A continuación se presenta un nuevo modelo para la creación de redes espontáneas inalámbricas ad hoc basadas en la confianza. Primero se enumeran las premisas que se tienen en cuenta y a continuación se presenta el modelo.

8.2.1 Premisas

Las premisas que siguen el modelo son las siguientes:

- Cada nodo puede estar en un estado elegido de forma aleatoria, o transmitiendo o escuchando.
- Los nodos son estáticos.
- Cada nodo tiene un identificador único que le distingue de los demás; este identificador puede ser por ejemplo la dirección MAC o el número de serie del fabricante.
- Los nodos son desplegados de forma aleatoria en un área dada.
- El tiempo está ranurado en *rounds*.
- Los nodos requieren sincronización en los límites de ranura.
- El número de nodos N puede ser desconocido por todos los nodos de la red.
- Cada nodo está equipado con un transceptor de radio de alcance limitado.
- Todos los nodos tienen idéntico rango de transmisión.
- Cada nodo puede transmitir o recibir pero no simultáneamente, es decir, usan *half-duplex*.

- Cada nodo tiene una memoria para guardar información topológica local, como identificadores de vecinos, tarjetas de identidad y valores de confianza.
- Pueden existir colisiones.
- Uso adecuado en entornos *one-hop* y *multi-hop*.
- Los nodos pueden detectar colisiones y terminación.
- Los nodos pueden detectar energía en el canal.
- Cada nodo tiene un par de claves pública-privada.

En la Tabla 2.5 se puede encontrar más información acerca del protocolo presentado en esta sección.

8.2.2 Modelo

El protocolo aleatorio propuesto para la creación de redes espontáneas basadas en la confianza consiste en 2 fases:

- Cada nodo envía un paquete *BROADCAST* hacia los nodos en su rango de transmisión, conteniendo su tarjeta de identidad.
- Cada vecino que recibió el paquete *BROADCAST* reconoce mediante el envío de un paquete *UNICAST*, llamado *ACK*. Este paquete contiene su tarjeta de identidad, y es enviado hacia el nodo que envió el *BROADCAST*.

Como se muestra en la Figura 8.1, teniendo en cuenta la existencia de colisiones, el tiempo está ranurado en *rounds*, y el tamaño de la ranura es τ. Al principio de cada *round* todos los nodos eligen aleatoriamente un estado T (transmitiendo) con probabilidad p o L (escuchando) con probabilidad $1-p$. El estado elegido puede ser diferente entre nodos de un mismo *round* y diferente entre *rounds* de un mismo nodo.

En primer lugar, se muestra un ejemplo de operación del protocolo, en un escenario *one-hop*, como se muestra en la Figura 8.1. En el primer *round*, los nodos 1 y 3 transmiten un paquete *BROADCAST*, donde cada paquete está representado por un rectángulo de color rojo, y el otro nodo escucha. Por tanto se produce una colisión y todos los nodos continúan compitiendo en el siguiente *round*. En el *round* 2, sólo el nodo 1 transmite un paquete *BROADCAST* y

por tanto una transmisión con éxito tiene lugar. El nodo 1 deja de competir a partir de ahora, esto es, ningún rectángulo rojo aparece en los siguientes *rounds* para el nodo 1. En el *round* 3, el proceso para los *ACKs* inicia para los nodos 2 y 3, representado con rectángulos azules. Los nodos 2 y 3 transmiten un *ACK* por tanto provocan colisión y ambos continúan compitiendo en el siguiente *round*. En el *round* 4, el nodo 3 transmite con éxito el *ACK* por tanto el nodo 3 deja de competir los *ACKs*. En el *round* 5 todos los nodos están escuchando por tanto el nodo 2 continúa compitiendo en los *ACK*. En el *round* 6, sólo el nodo 2 envía el *ACK* con éxito, por tanto dejará de competir en los *ACKs*. En este momento, los *ACKs* para el nodo 1 terminan. En el *round* 7, el nodo 3 transmite con éxito el *BROADCAST* y por tanto dejará de competir a partir de ahora. En el *round* 8, los *ACKs* para el nodo 3 empiezan, y el nodo 1 transmite con éxito el *ACK*, y por tanto deja de competir en los *ACKs*. En el *round* 9, el nodo 2 transmite con éxito el *ACK* y deja de competir en los *ACKs* y finaliza los *ACKs* para el nodo 3. En el *round* 10, todos los nodos están escuchando, por tanto el nodo 2 continúa en el siguiente *round*. En el *round* 11, el nodo 2 logra transmitir con éxito el *BROADCAST* por tanto no competirá a partir de ahora. En el *round* 12, los nodos 1 y 3 provocan una colisión por tanto continúan en el siguiente *round*. En el *round* 13, el nodo 1 transmite con éxito el *ACK* y deja de competir. En el *round* 14, el nodo 3 transmite con éxito el *ACK*. El algoritmo finaliza dado que todos los vecinos de cada nodo han logrado transmitir con éxito su *ACK* y todos los nodos han logrado transmitir con éxito.

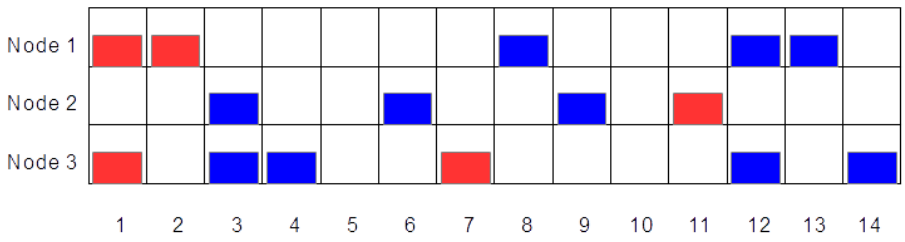

Figura 8.1: Ejemplo de operación de la propuesta

La Figura 8.2 muestra un diagrama de flujo del funcionamiento del prototocolo. Según ella, en un *round* tras elegir el estado, cada nodo en estado T envía un mensaje $BROADCAST(identitycard)$ en ese *round*. Eso corresponde a la fase 1, y permanece escuchando si el estado es L o S. El estado S significa que el nodo logró transmitir con éxito en *rounds* previos. La tarjeta de identidad de cada nodo contiene el identificador, la clave pública y la firma (usando la

clave privada). Al final del *round*, los receptores han llevado a cabo detección de colisiones. Una colisión se produce cuando dos o más nodos intentan transmitir simultáneamente. En caso contrario, decimos que el nodo transmitió con éxito, esto es, no se produce ni colisión ni *idle slot*.

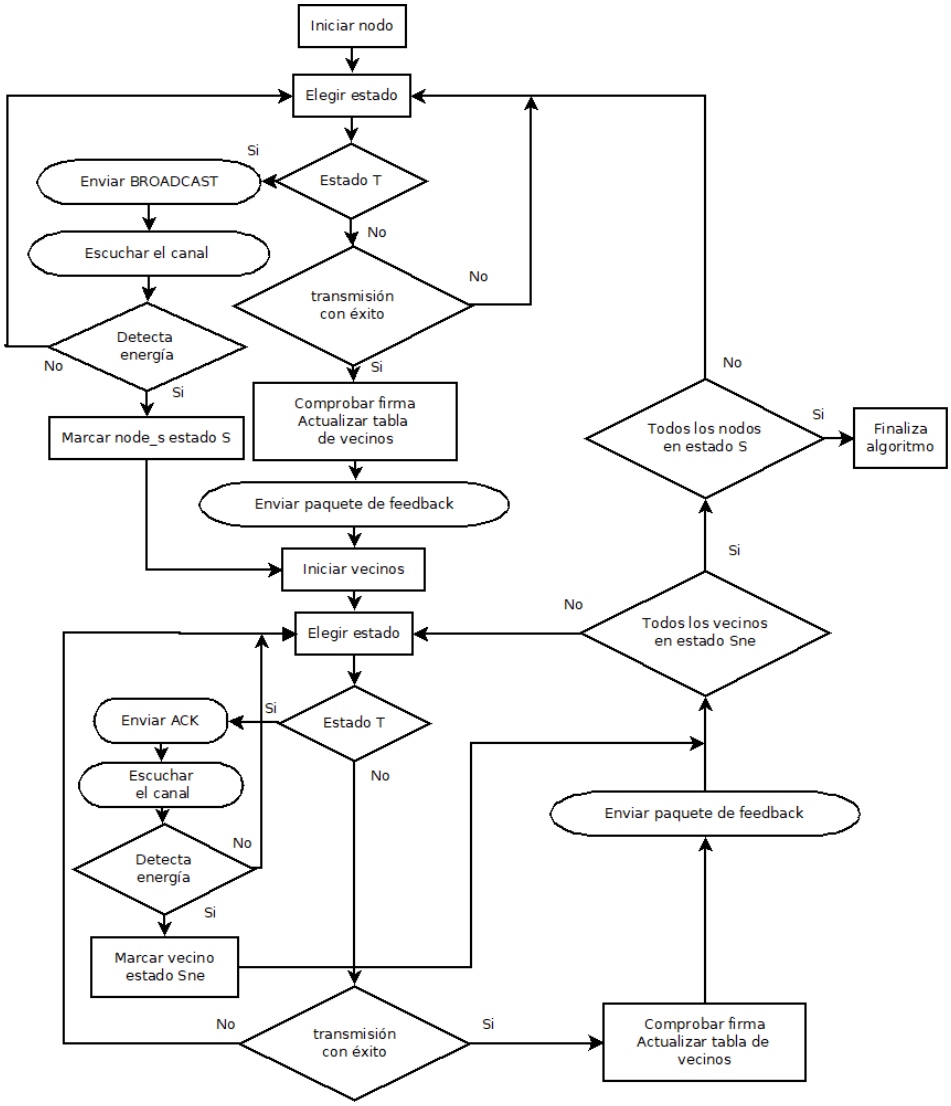

Figura 8.2: Diagrama de flujo de la propuesta

Si los receptores detectan que un nodo logró transmitir con éxito el *BROADCAST* al final del *round*, estos receptores difunden un paquete de *feedback* simultáneamente a los nodos en rango de transmisión. En caso contrario, los receptores no envían ningún paquete de *feedback*, lo cual indica que se produjo colisión o no hay señal en el canal. Resaltar que la transmisión de esos paquetes de *feedback* no provocan colisión. Al mismo tiempo, los nodos que transmitieron el *BROADCAST* escuchan el canal y cuando detectan energía, el estado del nodo cambia a S. Esto significa que el nodo transmitió con éxito, y a partir de ese momento el nodo en estado S permanecerá escuchando. El estado S no cambiará para este nodo en los siguientes *rounds* hasta que el algoritmo finalice. Resaltar que el nodo en estado S permanecerá escuchando *BROADCASTs* de otros nodos de forma que podrá descubrir los otros nodos en su rango de trasnmisión que no colisionen. Sin embargo, el nodo en estado S enviará paquetes de *feedback* cuando sea necesario. También resaltar que el paquete de *feedback* es mucho más pequeño que el paquete *BROADCAST* y que el paquete *ACK*. Hay que tener en cuenta que por ejemplo el 802.11 *ACK* tiene un tamaño de 14 bytes.

Si no se detectó energía, esto es, o se produjo colisión o no hubo señal en el canal. En ese caso, todos los nodos en estado diferente de S continúan compitiendo en el siguiente *round*, y todos los nodos con estado diferente de S continuarán compitiendo en el siguiente *round*. Si no hay señal en el canal en un *round*, significando que ningún *BROADCAST* se recibió en un *round*. En ese caso, todos los nodos están en estado L o S, los nodos en estado L continúan compitiendo en el siguiente *round*. Tan pronto como sólo un nodo transmite con éxito un *BROADCAST* en un *round*, which we call $node_s$, un nuevo proceso inicia para los *ACKs*, esto es, fase 2. En ese nuevo proceso, todos los vecinos que ya estén en estado L o S, el resto de nodos en rango de transmisión de $node_s$, seguirán el mismo mecanismo enviando paquetes $ACK(identitycard)$ al $node_s$. En este caso, se usa el estado S_{ne} en lugar de S, y al principio todos los vecinos tienen un estado inicial I diferente de S_{ne}.

Cuando sólo un nodo envió con éxito el *BROADCAST*, decimos que ese nodo transmitió con éxito y los vecinos guardan su identificador, y la tarjeta de identidad del nodo en sus tablas de vecinos. Además, tras la recepción del *BROADCAST*, cada vecino comprueba la firma de la tarjeta de identidad (usando la clave pública). Si es correcta el emisor se marca como de confianza por los receptores. En caso contrario, se marca como válido. Este valor de confianza también se almacena en la tabla de vecinos. La clave privada debe mantenerse en el nodo. Si hay un error al comprobar la firma, esto significa que el mensaje ha sido interceptado y manipulado de alguna forma. Además

cuando un vecino envía el ACK con éxito, el mismo procedimiento se lleva a cabo en el otro sentido de la comunicación. En ese caso, los vecinos serán de confianza por el $node_s$. Por tanto, la confianza mutua se podría establecer si es necesario.

El protocolo también incluye un mecanismo de detección de terminación. Cuando todos los vecinos enviaron reconocimientos con éxito, esto es, todos los vecinos están en estado S_{ne}, el proceso actual para los $ACKs$ finaliza. En ese caso, todos los nodos inician un nuevo *round* (pasan a la fase 1) y este proceso recurre. Sin embargo, en ese nuevo *round*, los nodos que transmitieron con éxito antes de iniciar el proceso de las $ACKs$ mantienen el estado S.

Los nodos saben que todos los vecinos están en el estado S_{ne}, esto es, los nodos saben que el algoritmo finaliza para los $ACKs$, cuando en varios *rounds* consecutivos no se detectan transmisiones. Esto indica que todos los vecinos están en estados L o S_{ne}. El número de *rounds* consecutivos es un parámetro que debe ser fijado cuidadosamente. Este procedimiento es válido dado que la probabilidad de que todos los restantes nodos estén en estado L en un número de *rounds* consecutivos es muy baja. Por lo tanto concluimos que todos los nodos están en estado S_{ne}. El mismo procedimiento es usado para saber que todos los nodos están en estado S, y por tanto los nodos saben cuándo finaliza el protocolo.

Si todos los vecinos enviaron reconocimientos con éxito para cada $node_s$ y no todos los nodos transmitieron con éxito el $BROADCAST$, un nuevo *round* inicia para los nodos (fase 1). Si no todos los vecinos enviaron reconocimientos con éxito, un nuevo *round* para los $ACKs$ inicia (fase 2). En otro caso, esto es, todos los vecinos enviaron reconocimientos con éxito para cada nodo $node_s$ y todos los nodos transmitieron con éxito su $BROADCAST$, el algoritmo finaliza.

De esta forma, cada nodo tiene una lista de vecinos de confianza, en base a sus firma para conformar la red de confianza. Este protocolo soluciona la intercepción y el problema *man in the middle* pero no soluciona el problema de nodos *sybil*.

8.3 Simulación y resultados

A continuación se procede a presentar las prestaciones obtenidas por la propuesta en comparación con un modelo existente para redes de confianza [67].

Para los experimientos se ha usado Castalia 3.2 [88]. Se ha elegido el protocolo en [67] porque es el más apropiado para la comparación con la propuesta.

8.3.1 Protocolo de referencia

Como referencia se ha decidido usar un protocolo existente para la creación de redes basadas en la confianza [67]. El protocolo consiste en 3 fases: (i) descubrimiento de vecinos, (ii) envío de tarjeta de identidad, y (iii) respuesta de tarjeta de identidad. Para comparar con la propuesta, se ha decidido implementar el protocolo de forma determinística sin *backoffs*.

El protocolo de referencia permite un servicio de gestión de clave pública distribuida a través de su propuesta de modelo de creación de redes espontáneas basada en la confianza.

La fase de descubrimiento de vecinos, se ha implementado de forma que cada nodo envíe uno tras otro para evitar colisiones de acuerdo con una planificación en la transmisión predeterminada. Se envían 100 paquetes *BROADCAST*. Tras la llegada de todos los paquetes *BROADCAST* a los nodos en rango de transmisión, se ha fijado un umbral de 95%. Éste indica el porcentaje de mensajes recibidos por encima del cual un vecino se considera como descubierto.

En una segunda fase, cada nodo envía un mensaje *PUBLICKEY* conteniendo su tarjeta de identidad hacia los nodos vecinos, uno tras otro para evitar colisiones.

En una tercera fase, tras la recepción de un *PUBLICKEY* cada vecino que recibió el *PUBLICKEY* reconoce enviando un mensaje *PUBLICKEYRETURN*. Ese paquete contiene su tarjeta de identidad y se envía uno tras otro para evitar colisiones. Tan pronto como se recibe un reconocimiento, el nodo guarda el identificador del vecino, y la tarjeta de identidad del vecino en su tabla de vecinos. Además, tras recibir el *PUBLICKEYRETURN*, el nodo que envió el *PUBLICKEY* calcula el *hash* de la tarjeta de identidad. Si el *hash* calculado es igual al *hash* recibido el vecino se marca como de confianza, lo que significa que el nodo que envió el *PUBLICKEY* confía en el vecino. En caso contrario, se marca como válido. Este valor de confianza también se guarda en la tabla de vecinos.

Cuando todos los reconocimientos se han recibido, el siguiente nodo envía su *PUBLICKEY* y el proceso recurre hasta que el algoritmo finaliza. Al final, todas las tarjetas de identidad han sido intercambiadas y todos los vecinos de confianza se han descubierto.

Además, todos los vecinos de la red envían reconocimientos de acuerdo con una planifiación incluso en escenarios *multi-hop*.

En la Tabla 8.1, se incluye una comparación cualitativa del protocolo de referencia y la propuesta aleatoria para la creación de redes de confianza. Entre las características más importantes mostradas, se encuentran las siguientes. El protocolo de referencia consiste en 3 fases, sólo funciona en entornos estáticos, esto es, no funciona en MANETs, requiere sincronización y sigue una planificación en la transmisión. Aunque considera colisiones, no las trata ya que el protocolo está libre de colisiones, y no detecta ni colisiones ni terminación. El protocolo finaliza cuando todos los nodos y vecinos han transmitido con éxito de acuerdo con la planificación. Además, logra descubrir todos los vecinos en el caso ideal, logra el intercambio de tarjetas de identidad y descubre todos los vecinos de confianza. Llegamos a esta conclusión dado que el protocolo evita colisiones. El protocolo funciona adecuadamente tanto en entornos *one-hop* como *multi-hop*, aunque el número de nodos debe ser conocido para aplicar la planificación. Sin embargo, el protocolo se basa en la comprobación de un *hash* para descubrir los vecinos de confianza.

Con el objetivo de solucionar los problemas encontrados en el protocolo de referencia y otros protocolos existentes, se propone un protocolo aleatorio. Como se muestra en la Tabla 8.1, la propuesta consiste en 2 fases, esto es, envío de *BROADCAST* y envío de *ACK*. La propuesta sólo requiere sincronización en los límites de ranura, no necesita seguir una planificación, y el número de nodos puede ser desconocido. El protocolo se puede usar adecuadamente tanto en entornos *one-hop* como *multi-hop*. Se asume que los nodos permiten detección de colisiones y terminación por tanto los nodos saben cuándo termina el algoritmo. El protocolo logra descubrir todos los vecinos, intercambiar con éxito las tarjetas de identidad y descubrir todos los vecinos de confianza. El protocolo es basado en *handshake*, y la comprobación de firma se usa para descubrir los vecinos de confianza. Al presentar la propuesta también se tiene como objetivo mejorar las prestaciones en comparación con el protocolo de referencia.

Tabla 8.1: Comparación cualitativa del protocolo de referencia y la propuesta.

	[67]	Propuesta
Número de fases	3	2
Entorno estático	✓	✓
Entorno móvil		
Aleatorio		✓
Tiempo ranurado		✓
N desconocido		✓
Requiere sincronización	✓	✓
Planificación en la transmisión no requerida		✓
Transmitir o escuchar (pero no simultáneamente)	✓	✓
Uso en *one-hop*	✓	✓
Uso en *multi-hop*	✓	✓
Sleep disponible		
Considera colisiones		✓
Colisiones pierden transmisión	✓	✓
Detección de pérdida de paquetes		
Detección de colisiones		✓
Detección de terminación		✓
Inicia transmisión en diferentes momentos de tiempo		
Usa comprobación de *hash*	✓	
Usa comprobación de firma		✓
Descubre todos los vecinos	✓	✓
Intercambia tarjetas de identidad con éxito	✓	✓
Descubre todos los vecinos de confianza	✓	✓

8.3.2 Escenario de simulación

Con el objetivo de obtener los resultados para el protocolo de referencia y la propuesta usamos el mismo escenario, variando el número de nodos (escalabilidad). Aunque hay disponibles muchas herramientas de simulación en la literatura, se ha elegido Castalia 3.2 [88]. Este simulador está basado en OMNET++ y se usa principalmente para testear WSNs y BANs. En nuestro caso, concluimos que reúne los requerimientos para validar protocolos para redes de confianza en entornos espontáneos estáticos *multi-hop*.

Para ambos protocolos se ha fijado idéntico tiempo que un nodo está transmitiendo $\tau = 0.07s$ usando *ZigBee* como modelo de radio.

Se ha definido un área de despliegue de 10mx10m (*one-hop*) en la cual todos los nodos están en rango de transmisión de todos los demás. También se han

desplegado los nodos en un área de 100mx100m (*multi-hop*) en la cual sólo algunos nodos están en rango de transmisión de los demás. Se han organizado N nodos en mallas MxM.

Para los experimientos, teniendo en cuenta la existencia de colisiones, se ha usado el parámetro *collisionModel* de Castalia 3.2. Este parámetro puede tomar el valor 0 (sin colisiones), 1 (modelo simplista para colisiones) o 2 (modelo de interferencia aditiva). En este caso, se ha fijado el parámetro *collisionModel* a 2 (el modelo de colisiones más realista).

Dado que los modelos de red basada en la confianza usan técnicas de descubrimiento de vecinos, para las simulaciones elegimos una métrica: el consumo de tiempo. Además, dado que ambos protocolos logran descubrir todos los vecinos, no se presentan los resultados para la métrica número de vecinos descubiertos. Sin embargo, también se presentan los resultados para el consumo energético, dado que los dispositivos usan baterías que se pueden descargar en un tiempo dado. También se presenta el *throughput*, y el número de vecinos descubiertos por paquetes enviados.

Definimos el consumo energético como el promedio del consumo energético de todos los nodos. *ZigBee* tiene en cuenta el consumo por nodo cuando el radio está transmitiendo (0.05742 Joules por segundo) o escuchando (0.062 Joules por segundo). En relación con el *throughput*, calculamos el número de paquetes recibidos por todos los nodos y lo multiplicamos por el tamaño de paquete y lo dividimos por el consumo de tiempo. Finalmente, para obtener el número de descubrimientos por paquetes enviados, dividimos el número de vecinos descubiertos por el total de número de paquetes enviados por los nodos. Castalia 3.2 tiene una opción que permite mostrar el consumo de tiempo, el promedio de consumo energético y una opción para mostrar el número de paquetes enviados y recibidos.

Para los experimentos, usamos el modelo de radio *ZigBee* (*CC2420*), fijando una potencia de transmisión a 0dBm, una tasa de paquetes de 5 paquetes/s y el tamaño de paquete a 2500 bytes. Para comparación de prestaciones fijamos para la propuesta diferentes probabilidades de transmisión: $\frac{1}{N}$, $\frac{1}{2N}$, $\frac{2}{N}$ y una probabilidad fija de 0.25.

En la Tabla 8.2, se resumen los parámetros de simulación.

Tabla 8.2: Parámetros de simulación.

Parámetro	Valor
Entorno estático	*True*
Modelo de radio	CC2420
Modelo de colisión	2
Potencia de transmisión	0dBm
Tasa de paquetes	5 paquetes/s
Tamaño de paquete	2500 bytes
Tamaño de ranura	τ
τ	0.07s
Tamaño *one-hop*	10mx10m
Tamaño *multi-hop*	100mx100m
Despliegue	malla MxM
Prob. transm. 1	$\frac{1}{N}$
Prob. transm. 2	$\frac{1}{2N}$
Prob. transm. 3	$\frac{2}{N}$
Prob. transm. 4	0.25

8.3.3 Resultados

A continuación se presentan los resultados de simulación comparando las prestaciones de ambos protocolos, el protocolo de referencia y la propuesta, en entornos *one-hop* y *multi-hop*.

Tiempos

En primer lugar, se presentan los resultados para un escenario *one-hop* en relación con la cantidad de tiempo que los algoritmos tardan en crear la red espontánea basada en la confianza. Este es un caso simple aunque aplicable a situaciones reales en concreto cuando el transceptor de radio tiene un rango de transmisión muy elevado.

Como se observa en la Figura 8.3, la propuesta aleatoria con probabilidad de transmisión $\frac{2}{N}$ mejora al protocolo de referencia en un entorno *one-hop* para un número de nodos por debajo de 40. La propuesta con probabilidad de transmisión $\frac{1}{N}$ también mejora al protocolo de referencia en una red compuesta de menos de 30 nodos. Esta mejora también tiene lugar cuando la probabilidad de transmisión es $\frac{1}{2N}$ para redes compuestas de menos de 17 nodos. En general, la propuesta con probabilidad $\frac{2}{N}$ presenta las mejores prestaciones

mientras que la propuesta con una probabilidad fija 0.25 es el peor. El protocolo determinístico de referencia presenta resultados intermedios. A medida que la red crece hay más vecinos que deben ser descubiertos y más tarjetas de identidad que intercambiar, por tanto el consumo temporal crece. Principalmente, el tiempo es peor en el protocolo determinístico cuando el número de nodos es bajo. Esto es debido a que la fase de descubrimiento de vecinos separada introduce mucho tiempo (cada nodo envía 100 paquetes uno tras otro para evitar colisiones). Resaltar que la propuesta y el protocolo de referencia siguen una tendencia creciente a medida que el número de nodos se incrementan. Además, se ha comprobado que ambos protocolos (propuesta y protocolo de referencia) logran descubrir todos los vecinos, intercambian con éxito las tarjetas de identidad. Además, descubren todos los vecinos de confianza.

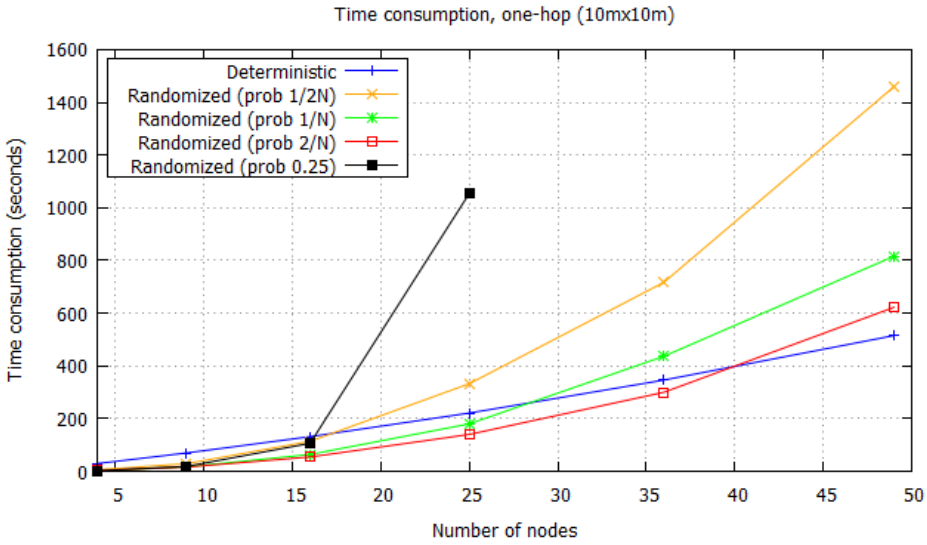

Figura 8.3: Comparación de tiempos (one-hop)

A continuación, se presentan los resultados de simulación obtenidos en escenarios más realistas: un entorno *multi-hop* de tamaño 100mx100m. De acuerdo con la Figura 8.4, la propuesta con probabilidad de transmisión 0.25 mejora a los otros, seguida por la propuesta con probabilidades $\frac{2}{N}$, $\frac{1}{N}$, y $\frac{1}{2N}$. Finalmente, el protocolo determinístico de referencia tiene claramente las peores prestaciones. Los protocolos siguen una tendencia creciente a medida que el número de nodos crece, por la misma razón que se expuso para escenarios *one-hop*. De nuevo, el tiempo es peor para el protocolo determinístico debido principalmente

a la fase adicional de descubrimiento de vecinos que desperdicia mucho tiempo. La planificación es el motivo de este desperdicio ya que los nodos transmiten uno tras otro para evitar colisiones y la mayoría de reconocimientos no llegan a su destino.

De nuevo, concluimos que ambos protocolos logran descubrir todos los vecinos, intercambian con éxito las tarjetas de identidad, y descubren todos los vecinos de confianza.

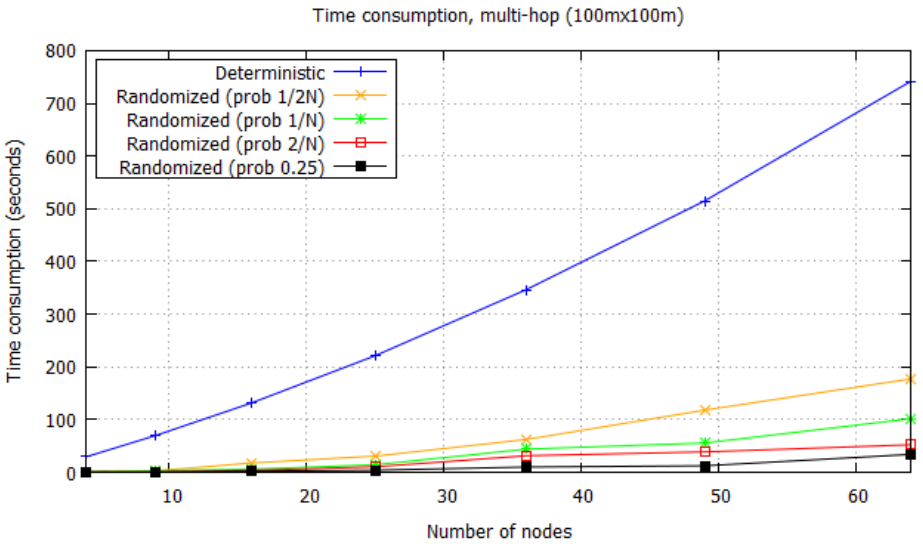

Figura 8.4: Comparación de tiempos (multi-hop)

Consumo energético

En relación con el consumo energético, como se muestra en la Figura 8.5, los protocolos presentan el mismo comportamiento que en consumo de tiempo para el caso *one-hop*. Para resumir, la propuesta con probabilidad $\frac{2}{N}$ presenta los mejores resultados mientras que la propuesta con una probabilidad fija 0.25 es el peor en consumo energético. El protocolo de referencia determinístico presenta resultados intermedios. Todos los protocolos siguen una tendencia creciente con el número de nodos dado que a medida que el consumo temporal se incrementa el consumo energético también se incrementa.

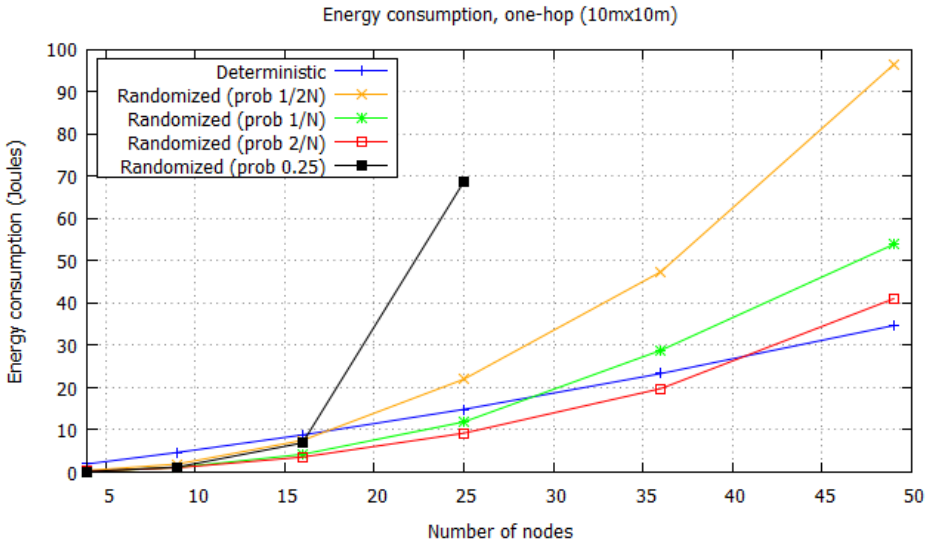

Figura 8.5: Comparación de consumo energético (one-hop)

Como se muestra en la Figura 8.6, para el caso *multi-hop* en relación con el consumo energético, un comportamiento similar al del consumo temporal tiene lugar. La propuesta con probabilidad 0.25 mejora al resto, seguido por la propuesta con probabilidades $\frac{2}{N}$, $\frac{1}{N}$, y $\frac{1}{2N}$. Finalmente, el protocolo determinístico consume más energía que los otros protocolos. Todos los protocolos siguen una tendencia creciente a medida que el número de nodos crece, por la misma razón que en el escenario *one-hop*. El protocolo de referencia determinístico consume más energía que los otros debido a la fase de descubrimiento de vecinos adicional que desperdicia una gran cantidad de tiempo y energía. La planificación el la razón de este desperdicio ya que los nodos transmiten uno tras otro para evitar colisiones incluso en un escenario *multi-hop* y la mayoría de los reconocimientos no llegan a su destino.

Throughput

De acuerdo con la Figura 8.7, que representa el caso *one-hop*, el *throughput* es mejor para el protocolo de referencia que para la propuesta. Ambos siguen una tendencia decreciente a medida que el número de nodos crece. La propuesta con probabilidad de transmisión $\frac{1}{N}$ y 0.25 presentan mejores resultados que las otras probabilidades para un número de nodos por debajo de 9. En general,

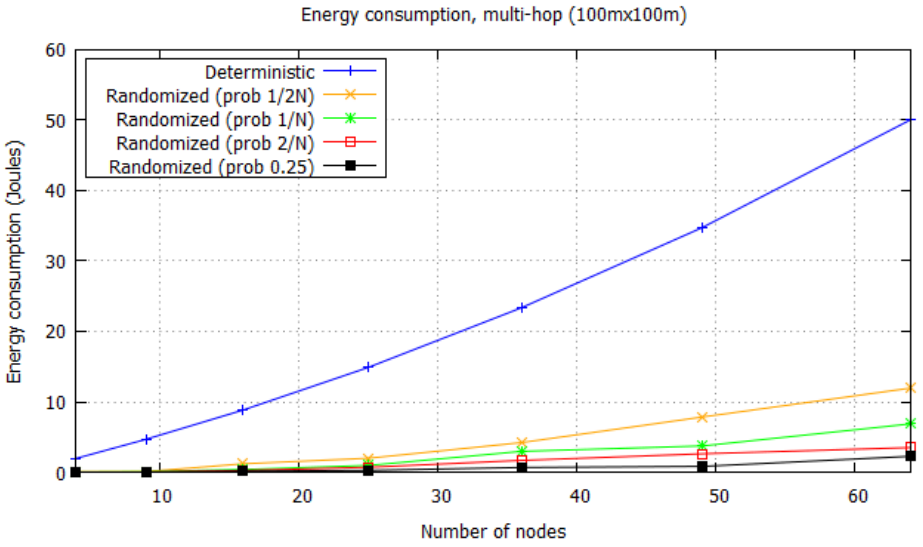

Figura 8.6: Comparación de consumo energético (multi-hop)

la propuesta con probabilidad $\frac{2}{N}$ mejora a la de probabilidad $\frac{1}{N}$, seguido por la de probabilidad $\frac{1}{2N}$. La propuesta con probabilidad 0.25 presenta las peores prestaciones en redes compuestas por más de 15 nodos. El comportamiento decreciente de la propuesta es debido al decrecimiento de paquetes recibidos por segundo ya que hay más colisiones a medida que el número de nodos crece. El protocolo determinístico se comporta mejor que las otras soluciones en un escenario *one-hop* ya que está libre de colisiones y todos los paquetes enviados son recibidos.

A continuación, la métrica *throughput* será evaluada en un entorno *multi-hop*, y se muestra en la Figura 8.8. La propuesta con una probabilidad de transmisión fija 0.25 mejora a las otras soluciones, seguida por la propuesta con $\frac{2}{N}$. La propuesta con probabilidad $\frac{1}{N}$ es mejor que el protocolo determinístico y la propuesta con probabilidad $\frac{1}{2N}$ en redes con menos de 32 nodos. La propuesta con probabilidad $\frac{1}{2N}$ es la peor para número de nodos por encima de 20. Para resumir, la propuesta con probabilidad 0.25 tiene las mejores prestaciones mientras que la propuesta con probabilidad $\frac{1}{2N}$ es la peor y el protocolo de referencia presenta resultados intermedios. Además, todos los protocolos presentan una tendencia decreciente a medida que el número de nodos crece. El *throughput* en el protocolo determinístico presenta malos resultados especialmente con un

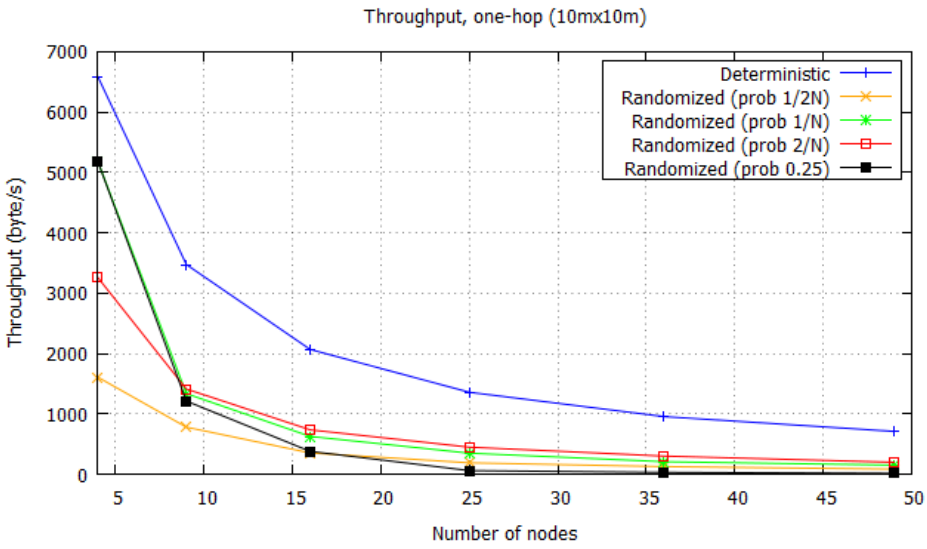

Figura 8.7: Comparación throughput (one-hop)

bajo número de nodos ya que el consumo temporal es mayor. El protocolo invierte mucho tiempo enviando reconocimientos en el caso *multi-hop* que no llegan a su destino debido a la planificación. Además para redes con menos de 10 nodos el *throughput* es 0 byte/s ya que todos los nodos están fuera del rango de transmisión de todos los demás y ningún paquete es recibido.

Descubrimientos por paquetes enviados

Se presentan los resultados del número de descubrimientos por paquetes enviados. En primer lugar, se muestran los resultados en un escenario *one-hop*.

De acuerdo con la Figura 8.9, la propuesta con probabilidad de transmisión $\frac{1}{2N}$ mejora a las otras soluciones para un número de nodos por encima de 9. A continuación, la propuesta con probabilidad $\frac{1}{N}$ presenta resultados intermedios, luego la propuesta con probabilidad $\frac{2}{N}$. Finalmente la propuesta con probabilidad 0.25 para un número de nodos por debajo de 15. En general, el protocolo determinístico (de referencia) presenta los peores resultados con un *ratio* constante de aproximadamente 0.008. Este mal resultado es debido al número de paquetes enviados en la fase de descubrimiento de vecinos, que es por encima de 100. Resaltar que en la fase de descubrimiento de vecinos cada

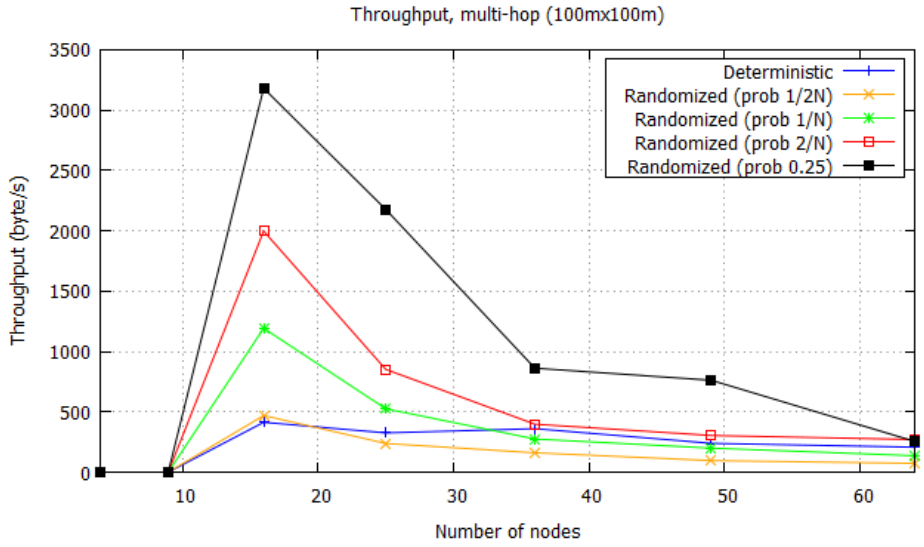

Figura 8.8: Comparación throughput (multi-hop)

nodo transmite 100 paquetes uno tras otro para evitar colisiones. La propuesta presenta una tendencia decreciente a medida que el número de nodos crece.

Para el caso *multi-hop*, como se muestra en la Figura 8.10, la propuesta con probabilidad de transmisión $\frac{1}{2N}$ mejora las otras soluciones para un número de nodos por encima de 25. A continuación, la propuesta con probabilidad $\frac{1}{N}$ y $\frac{2}{N}$ presenta resultados intermedios. Finalmente, la propuesta con probabilidad 0.25 es la peor. De nuevo, el protocolo de referencia determinístico presenta los peores resultados, con un *ratio* constante de 0.002. Este mal comportamiento es de nuevo principalmente debido al número de paquetes enviados (por encima de 100) en la fase de descubrimiento de vecinos separada. La propuesta también presenta una tendencia decreciente. El número de descubrimientos por paquetes enviados para redes compuestas de menos de 10 nodos presentan un valor de 0. Esto es debido a que todos los nodos están fuera del rango de transmisión de los otros y por tanto el número de vecinos descubiertos es 0.

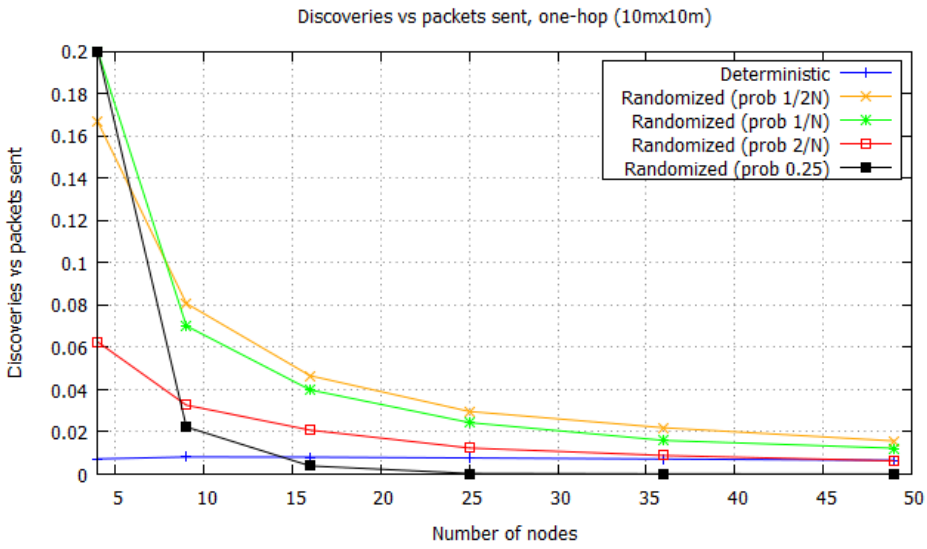

Figura 8.9: Comparación descubrimientos por paquetes enviados (one-hop)

8.4 Comparación cualitativa de protocolos

En esta sección se incluye una comparación cualitativa de protocolos de la literatura y la propuesta, mostrada en la Tabla 2.5.

De acuerdo con la tabla, se presentan protocolos para diferentes tipo de redes. Existen protocolos para *IoTs* [66] para comunidades con bajos recursos, y redes de computación en la nube ad hoc móviles [67]. También hay protocolos para redes espontáneas en [68, 69, 70, 71, 72, 73, 74] y la propuesta.

Sin embargo, todos los protocolos pueden ser usados en redes inalámbricas ad hoc más generales. Los protocolos en [67, 69, 70, 71, 72, 73, 74], son apropiados para ser usados en redes móviles, mientras que la propuesta puede ser sólo usada en entornos estáticos. Todos los protocolos presentan algunas características comunes, tales como crear con éxito la red espontánea, los dispositivos tienen identidades únicas. Los protocolos usan tarjetas de identidad, par de claves pública-privada, infraestructura de clave pública, y están diseñados para una interacción mínima del usuario. Además, forman una cadena de confianza, el valor de confianza está basado en las relaciones humanas, e incluyen una AC distribuida. Sin embargo, todos los protocolos excepto la propuesta permiten a los nodos entrar y salir de la red a voluntad. En la propuesta la confianza se

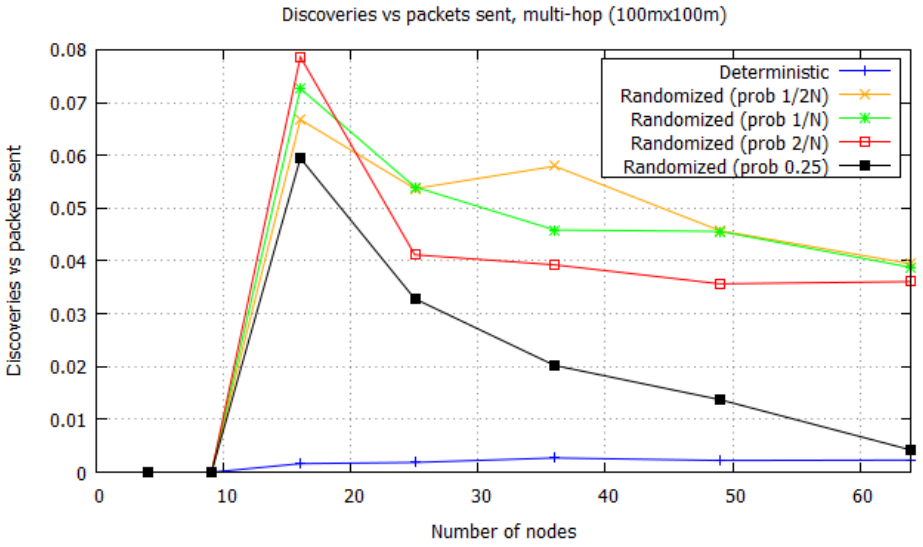

Figura 8.10: Comparación descubrimientos por paquetes enviados (multi-hop)

establece automáticamente y la autenticación se lleva a cabo intercambiando claves a través de *ZigBee*. Además, la propuesta está diseñada para permitir la creación de la red, mientras que para considerar los nuevos vecinos que se unen a la red el protocolo debe ser modificado. Aunque la mayoría de los protocolos en la Tabla 2.5 incluyen un mecanismo de encriptación, la propuesta no considera ningún procedimiento de encriptación.

En [68] se desarrolla un prototipo en *Java* (*J2ME*) para *Mobile Nokia E65*. La mayoría de los protocolos son implementados para su simulación usando Castalia, algunos permiten cambiar el nivel de confianza. Algunos protocolos usan *hash SHA-1*, y la mayoría usan un certificado. Los protocolos presentados en [66, 67, 70, 71] permiten acceso a Internet. El cifrado simétrico *AES* y cifrado asimétrico *RSA/ECC* son usados ampliamente, y la autenticación a través de *Bluetooth* o *ZigBee* también es común. Una técnica de detección de intrusiones se usa en [69, 70, 73], mientras que [70] usa una técnica de almacenamiento en caché.

8.5 Conclusiones

En este capítulo, se ha llevado a cabo un estudio de estrategias de creación de redes basadas en la confianza. El estudio se centra en entornos espontáneos inalámbricos ad hoc estáticos *multi-hop* considerando la existencia de colisiones. Un modelo aleatorio de red basada en la confianza se ha presentado. El modelo se beneficia de las ventajas de la detección de colisión y no requiere de planificación.

Este modelo y un modelo de red basada en la confianza existente usado como referencia han sido implementados en Castalia 3.2 para su comparación.

Los experimentos se han centrado tanto en entornos *one-hop* como *multi-hop*. Se han evaluado cuatro métricas: consumo de tiempo, consumo energético, *throughput*, y el número de descubrimientos por paquetes enviados. De los resultados de simulación concluimos que la propuesta mejora al protocolo de referencia en tiempo y energía para número de nodos bajo en el caso *one-hop*. También mejora al protocolo de referencia en cuanto a número de descubrimientos por paquetes enviados para el caso *one-hop*. La propuesta mejora al protocolo de referencia según las cuatro métricas en entornos *multi-hop*. Además, se han evaluado las prestaciones de la propuesta fijando la probabilidad de transmisión a $\frac{1}{2N}$, $\frac{1}{N}$, $\frac{2}{N}$ y un valor fijo 0.25. También se ha concluido que la propuesta no requiere conocer el número de nodos, dado que se puede usar una probabilidad de transmisión fija. Aún así, el protocolo proporciona resultados razonables.

Además, la propuesta permite detección de colisiones y terminación, y no requiere de una planificación en la transmisión. La propuesta consiste sólo en dos fases y sigue premisas más realistas.

Se ha concluido que tanto el protocolo de referencia como la propuesta logran descubrir todos los vecinos, intercambian adecuadamente sus tarjetas de identidad, y descubren todos los vecinos de confianza. Esto se produce en un tiempo razonable y bajo consumo energético, tanto en entorno *one-hop* como *multi-hop*.

La estrategia usada en la propuesta para descubrir los vecinos de confianza es que cada nodo envía su tarjeta de identidad conteniendo la firma construida mediante la clave privada del nodo. Cuando el paquete llega a su vecino, éste comprueba la firma usando la clave pública del nodo y si es correcta el nodo es de confianza de los vecinos. El mismo procedimiento se lleva a cabo en el

otro sentido de la comunicación, esto es, los vecinos son de confianza del nodo. Por tanto se puede establecer confianza mutua si es necesario.

Además, la complejidad computacional de la propuesta, esto es, el consumo temporal tanto en *one-hop* como *multi-hop* es aproximadamente $O(N^2)$. En este caso, se ha fijado la probabilidad de transmisión $\frac{1}{N}$, y N es el número de nodos de la red.

Las principales limitaciones prácticas de la propuesta son que requiere sincronización en los límites de ranura y sólo se puede usar en entornos estáticos. Para solucionar estas limitaciones se debe usar algún mecanismo de sincronización conocido antes de que el protocolo comience. También se debe permitir al protocolo considerar nodos que entren y salgan de la red y nodos que entren y salgan en el rango de transmisión de otros para que se pueda usar en MANETs.

Entre las aplicaciones prácticas, la propuesta puede ser usada en entornos espontáneos estáticos *one-hop* o *multi-hop*. Dado que el número de nodos usados para evaluar la propuesta es bajo pero suficiente para su aplicación en entornos del mundo real. Un ejemplo son las reuniones esporádicas de estudiantes organizados en una clase. Otra aplicación puede ser una reunión de compañeros de trabajo (una localización dada) para intercambiar información durante un tiempo dado. También se permite su uso en una red de sensores inalámbricos desplegada en un campo para determinar varios parámetros para servicios de regadío en un periodo de tiempo. Finalmente, se puede usar en una red de robots que intercambian información con el objetivo de trabajar conjuntamente para cumplir una tarea dada.

Como posibles extensiones, se puede desarrollar y evaluar un nuevo modelo de bajo consumo para la creación de redes espontáneas basadas en la confianza. También se prevee mejorar la seguridad para construir redes espontáneas basadas en la confianza.

Capítulo 9

Descubrimiento y selección de vecinos basada en la gestión de prioridades

En determinadas situaciones prácticas sería conveniente elegir un vecino favorito para ser usado por ejemplo como un gateway. El objetivo es que la red tenga conectividad con el exterior. Por tanto la selección de vecinos es necesaria. En este capítulo, se presenta NDSP (Neighbor Discovery and Selection Protocol), una propuesta aleatoria. La propuesta utiliza detección de colisiones para el descubrimiento de vecinos en escenarios estáticos y utiliza prioridades para elegir los nodos favoritos. Se obtienen resultados mediante Castalia 3.2 para comparar la propuesta con dos protocolos existentes. Estos dos protocolos han sido convenientemente ampliados para incluir selección de vecinos: el NS-PRR (Neighbor Selection-PRR), basado en PRR, y el NS-Hello (Neighbor Selection-Hello), basado en Hello. Se concluye que NDSP presenta mejores prestaciones que los protocolos de referencia con respecto a tiempo y consumo energético en escenarios multi-hop. También mejora el throughput y el número de paquetes enviados tanto en entornos one-hop como multi-hop. Además, NDSP funciona siguiendo premisas más realistas.

9.1 Introducción

Este capítulo se centra en el descubrimiento de vecinos y selección de vecinos logrados usando gestión de prioridades en redes inalámbricas ad hoc estáticas.

Se presenta NDSP, un protocolo aleatorio basado en la detección de colisiones. Este protocolo logra tanto del descubrimiento como la selección de vecinos en entornos estáticos, y usa prioridades para determinar los vecinos favoritos.

También se proporcionan resultados de simulación de la propuesta en comparación con dos protocolos de la literatura ampliados para lograr la selección de vecinos: NS-Hello y NS-PRR.

La importancia de este trabajo se describe a continuación. La propuesta usa detección de colisión y energía, conoce cuándo terminar el descubrimiento de vecinos, y se descubren todos los vecinos con probabilidad 1. Además, no necesita conocer el número de nodos de la red, y permite que los nodos inicien la transmisión en diferentes instantes de tiempo. Por otra parte, tras finalizar el protocolo, todos los nodos conocen sus favoritos y todos los nodos saben si son favoritos o no. De acuerdo con resultados analíticos y de simulación, se concluye que la propuesta supera a ambos protocolos de referencia.

Por lo tanto, le propuesta mejora a los protocolos de referencia, especialmente en la fase de descubrimiento de vecinos. Esto es debido a que no incluyen un mecanismo de terminación, los vecinos no son descubiertos con probabilidad 1, y PRR necesita conocer el número de nodos. Sin embargo, el mecanismo de selección de vecinos usado en las extensiones es el mismo.

Las principales contribuciones de este capítulo son: (i) NDSP, una propuesta aleatoria basada en el protocolo Hello, que hace uso de la detección de colisiones. El protocolo logra descubrir todos los vecinos y gestiona prioridades para elegir nodos favoritos que serán usados en operaciones futuras. (ii) Implementación de la propuesta, NS-Hello (extensión de Hello [22] para selección de vecinos) y NS-PRR (extensión de PRR [21] para selección de vecinos). Se ha usado Castalia 3.2 [88] para compararlos con respecto a tiempos, consumo energético, *throughput*, y *overhead* (número de paquetes enviados).

9.2 Protocolo de descubrimiento y selección de vecinos

En esta sección se presenta la propuesta para el descubrimiento y selección de vecinos NDSP.

9.2.1 Premisas

Con el objetivo de desarrollar el protocolo las siguientes premisas se deben tener en cuenta:

- El tiempo se presenta ranurado en *rounds*.
- Los nodos no se pueden mover en el área de despliegue, son estáticos, por tanto el protocolo no se puede utilizar en redes móviles (MANET).
- Cada nodo tiene un identificador único.
- Cada nodo tiene una prioridad asignada (un valor numérico).
- Los nodos son deplegados de forma aleatoria en una determinada área.
- Se requiere que los nodos estén sincronizados en los límites de ranura.
- El número de nodos es desconocido para todos los nodos.
- Cada nodo tiene un transceptor de radio con un rango de transmisión limitado y que funciona en modo *half-duplex*.
- Cada nodo tiene una tabla de vecinos para guardar identificadores y prioridades.
- Cada nodo debe conocer un identificador mínimo (ident_min) y un identificador máximo (ident_max). Ambos tienen el mismo valor para todos los nodos. Representan el mínimo y máximo identificador de todos los nodos posibles en la red. Los identificadores de los nodos deben estar en ese rango (entre ident_min e ident_max). Además, no es necesario que estén en la red desplegada todos los nodos ni los nodos con identificador ident_min ni ident_max.
- Pueden existir colisiones.
- Los nodos pueden detectar colisiones cuando están escuchando.
- Los nodos pueden detectar energía.

- Los nodos pueden detectar terminación.
- Los nodos pueden iniciar la transmisión en diferentes instantes de tiempo.

9.2.2 Modelo

NDSP combina el descubrimiento de vecinos con la selección de vecinos y la gestión de prioridades. El objetivo es lograr el descubrimiento y marcar los nodos favoritos que pueden ser usados por ejemplo como *gateways* en futuras operaciones como el encaminamiento.

La Figura 9.1 muestra que el tiempo está ranurado en *rounds* y hay dos *sub-slots* en cada *round*. El primer *sub-slot* (de tamaño ω) se usa para intercambiar mensajes de descubrimiento. El segundo *sub-slot* (de tamaño ω_f) se usa para intercambiar los *feedbacks*. Ambos ω y ω_f son de tamaño fijo (y tienen el mismo valor para todos los *rounds*).

Cada nodo tiene una prioridad, cuyo valor inicial es predeterminado y fijado antes del despliegue.

La bondad de los nodos favoritos seleccionados por la propuesta depende de la prioridad predeterminada. Esto significa que cuanto mayor es la prioridad, mejores características presentará el nodo seleccionado como favorito.

Además, NDSP finaliza cuando todos los vecinos han sido descubiertos, todos los nodos conocen sus nodos favoritos, y cada nodo conoce si es favorito o no.

De esta forma, tienen lugar dos cosas, esto es, cada nodo conoce quién es su nodo favorito y cada nodo que es favorito sabe que lo es.

De acuerdo con la Figura 9.1 y el diagrama de flujo en la Figura 9.2, en el primer *sub-slot* cada nodo transmite un solo paquete de *BROADCAST*. El paquete contiene su identicador y prioridad iniciando su transmisión en un tiempo aleatorio $t_i \in [0, \omega - \tau]$ durante τ. El nodo permanece escuchando el resto de la ranura durante un tiempo total de $\omega - \tau$. En el primer *sub-slot*, todos los nodos llevan a cabo detección de colisiones durante los periodos en los cuales están escuchando.

En el segundo *sub-slot*, usado para el envío de *feedbacks*, todos los nodos son planificados para enviar una serie de paquetes de *feedback* con identificadores desde $ident\_min$ hasta $ident\_max$. Los *feedbacks* se envían uno tras otro indicando qué nodos transmitieron con éxito.

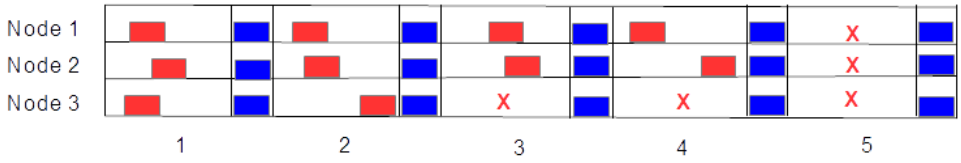

Figura 9.1: NDSP (línea de tiempos).

Tan pronto como el *jth* paquete de *feedback* es planificado para ser enviado en el segundo *sub-slot*, los nodos cuyo $identificador$ es distinto de ese j enviará un solo paquete de *feedback* si el nodo j transmitió con éxito. El nodo cuyo $identificador$ es igual a j escuchará el canal. En caso contrario, es decir, si el *BROADCAST* del nodo j colisionó el nodo $identificador$ distinto de j no enviará el paquete de *feedback*.

Además, si el nodo j transmitió con éxito, el resto de nodos que recibieron el *BROADCAST* almacenan j (el identificador del nodo j) y $prio_j$ (la prioridad de j) en sus tablas de vecinos. Estos dos valores están disponibles en el *BROADCAST*. Si el nodo j transmitió con éxito, esto significa que no fue detectada colisión para el nodo j. En caso contrario, esto es, si se detectó colisión, no se almacena ninguna información en las tablas de vecinos.

Los nodos con $identificador$ igual a j escuchan el canal en este momento en el segundo *sub-slot* y si se detecta energía por el nodo j, esto es, hay un paquete de *feedback* en el canal. Esto se indica con una marca X en la Figura 9.1. En el caso de que detecte energía, el nodo cambiará su estado a S, no competirá a partir de entonces en el primer *sub-slot* y permanecerá escuchando. Sin embargo, enviará paquetes de *feedback* cuando sea neceario en el segundo *sub-slot*. En caso contrario, seguirá compitiendo en los siguientes *rounds* eligiendo un nuevo t_i en el primer *sub-slot*. Cuando j llega a $ident_max$ un nuevo *round* inicia para los restantes nodos.

Hay que tener en cuenta que los paquetes de *feedback* son mucho más pequeños que los *BROADCASTs*.

NDSP proporciona un mecanismo de detección de terminación, según el cual el protocolo termina el descubrimiento cuando todos los nodos han logrado transmitir con éxito en *rounds* previos. Esto significa que en un *round* no hay señal en el canal durante el primer *sub-slot*, esto es, todos los nodos están en el estado S.

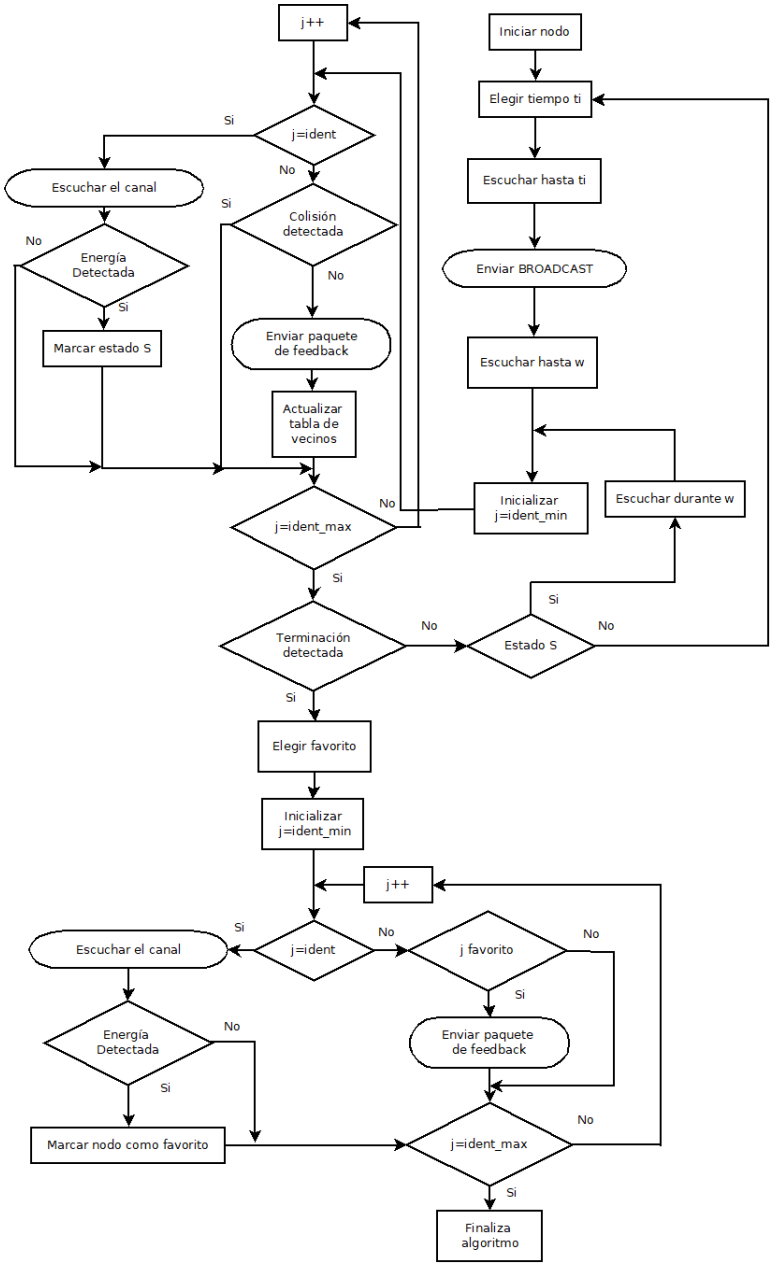

Figura 9.2: Diagrama de flujo de NDSP.

En el momento en que se detecta terminación, el proceso de selección de vecinos inicia. Cada nodo elige su nodo favorito teniendo en cuenta las prioridades almacenadas en su tabla de vecinos, eligiendo el nodo *ident* que presenta la prioridad *prio* más alta de entre todos los vecinos. En el caso de que en la tabla de vecinos haya varios vecinos con la misma más alta prioridad, el nodo elige el que tenga el menor identificador. Tras elegir los favoritos, se abre una ranura de tamaño w_f. En ella, los nodos indican a los otros nodos en rango de transmisión qué nodo es su favorito desde $ident\_min$ a $ident\_max$. Esto se indica mediante el envío de paquetes de *feedback*. Este proceso es similar al que se utiliza en el descubrimiento de vecinos para el envío de *feedbacks*.

En el caso de la selección de vecinos, cuando el jth paquete de *feedback* es planificado para ser enviado, el nodo con $identificador$ distinto de j envía el *feedback* sólo si el nodo j es un favorito del $identificador$. En cuanto al nodo cuyo $identificador$ es igual a j escucha el canal. En el caso de que el nodo j detecte energía, ese nodo sabrá que es un favorito y almacena internamente una marca como favorito. Este procedimiento continúa hasta que j llega a $ident\_max$. Además, los nodos saben qué nodos son favoritos al comprobar en la tabla de vecinos qué nodo tiene la prioridad más alta. En el caso de que haya más de un nodo con la prioridad más alta el favorito será el que tenga menor identificador.

El Algoritmo 4 permite explicar el funcionamiento del NDSP.

Algoritmo 4 NDSP

Entrada τ tiempo que un nodo está transmitiendo, ω (duración fija del primer *sub-slot*), ω_f (duración fija del segundo *sub-slot*), *ident* identificador, *prio* prioridad

```
1:  terminacion = false
2:  mientras no terminacion hacer
3:    Se elige aleatoriamente ti ∈ [0, ω − τ]
4:    Escucha hasta t_i.
5:    Envío del BROADCAST(ident,prio) en t_i durante τ.
6:    Escucha hasta ω.
7:    para cada j hacer
8:      si j == ident entonces
9:        Escucha el canal.
10:       Lleva a cabo detección de energía.
11:     sino
12:       si nodo j transmitió con éxito entonces
13:         Envía paquete de feedback.
14:         Guarda en la tabla de vecinos el identificador j y la prioridad
            prio_j, ambos valores contenidos en el BROADCAST del nodo j.
15:       fin si
16:     fin si
17:     si j detectó energía entonces
18:       El nodo j cambia al estado S y permanece en él a partir de este
          momento escuchando hasta que finaliza el protocolo, aunque enviará
          los feedbacks cuando sea necesario.
19:     sino
20:       El nodo j inicia nuevo round y continúa compitiendo.
21:     fin si
22:   fin para
23:   si no se recibe ningún BROADCAST entonces
24:     terminacion = true
25:   fin si
26: fin mientras
27: para cada j hacer
28:   El nodo j elige el favorito de la tabla de vecinos.
29: fin para
30: para cada j hacer
31:   si j == ident entonces
32:     Escucha el canal.
33:     Lleva a cabo detección de energía.
34:   sino
35:     si nodo j es favorito entonces
36:       Envía paquete de feedback.
37:     fin si
38:   fin si
39:   si j detectó energía entonces
40:     Nodo j se marca como favorito.
41:   fin si
42: fin para
```

Teniendo en cuenta un escenario *one-hop*, un caso simple en el cual todos los N nodos están en rango de transmisión de todos los demás, se obtienen las siguientes ecuaciones para el mecanismo de selección de vecinos. En este caso, asumimos que $N-1$ nodos seleccionan al nodo 0 como favorito, mientras que el nodo 0 selecciona al nodo 1 como favorito. Por tanto, la ecuación 9.1 muestra el tiempo que tarda el mecanismo de selección de vecinos en terminar en segundos, siendo τ_f el tiempo que un nodo está transmitiendo un paquete de *feedback*.

$$T = N \cdot \tau_f \tag{9.1}$$

Se concluye que el consumo temporal en el mecanismo de selección es lineal $O(N)$.

La ecuación 9.2 muestra el consumo energético por nodo en Julios, siendo E_{tx} la energía consumida por un nodo cuando transmite por segundo, y E_l la energía consumida por un nodo cuando escucha por segundo.

$$E = \frac{1}{N} \cdot \tau_f \cdot [N \cdot E_{tx} + N \cdot (N-1) \cdot E_l] = \tau_f \cdot [E_{tx} + (N-1) \cdot E_l] \tag{9.2}$$

En cuanto a los paquetes enviados, se muestra en la ecuación 9.3.

$$P_{sent} = N \tag{9.3}$$

El *throughput* viene dado por la ecuación 9.4.

$$Thr = \frac{N}{N \cdot \tau_f} = \frac{1}{\tau_f} \tag{9.4}$$

El *packet delivery ratio* se muestra en la ecuación 9.5.

$$PDR = \frac{P_{rec}}{P_{sent}} = 1 \tag{9.5}$$

Estas ecuaciones han sido obtenidas para el mecanismo de selección de vecinos.

En cuanto al mecanismo de descubrimiento de vecinos, las ecuaciones se presentan en el capítulo 7 para el CDH.

9.3 Simulación y resultados

En esta sección, se procede a presentar los resultados de simulación de NDSP y se compara con dos protocolos elegidos de la literatura que han sido extendidos para que solucionen la selección de vecinos. Estos protocolos son: NS-Hello (extensión de Hello [22] para incluir selección de vecinos) y NS-PRR (extensión de PRR [21] para incluir selección de vecinos).

9.3.1 Escenario de simulación

Para evaluar las prestaciones se ha usado Castalia 3.2 [88]. En este contexto, se ha fijado un área de despliegue *one-hop* (10mx10m) y *multi-hop* (100mx100m). Los nodos han sido desplegados en mallas MxM. Para tener en cuenta las colisiones, se ha utilizado el parámetro de modelo de colisiones de Castalia 3.2 (*collisionModel*). Este parámetro se ha fijado a 2, y sigue el *modelo de interferencia aditiva*. En el primer *sub-slot*, se ha fijado un tamaño de $\omega = N \cdot \tau$ tanto en NS-Hello como en NDSP. τ representa el tiempo que un nodo está transmitiendo y se ha fijado a $\tau = 0.07s$. Como se comentó con anterioridad, NDSP consta de dos *sub-slots* en el descubrimiento de vecinos, y el tamaño del segundo *sub-slot* se ha fijado a $\omega_f = N \cdot \tau_f$. τ_f es el tiempo que un nodo está transmitiendo el paquete de *feedback* que en este caso es de $\tau_f = 0.000392s$. En NS-PRR el tamaño de ranura se ha fijado a $\tau = 0.07s$. En cuanto al procedimiento de selección de vecinos, usa para su funcionamiento una ranura adicional (tras el descubrimiento de vecinos) de tamaño $N \cdot \tau_f$. Este procedimiento y el tamaño de ranura es idéntico para los 3 protocolos, y $\tau_f = 0.000392s$. El modelo de radio que se ha usado para las simulaciones es *ZigBee* (*CC2420*). La potencia de transmisión es de - $5dBm$, y la tasa de paquetes es de $5paquetes/s$. El tamaño del paquete *BROADCAST* es de $2500bytes$, mientras que el tamaño del paquete de *feedback* es de $14bytes$. Tanto para NS-PRR como para NS-Hello se ha de fijar un número de *rounds* tras el cual ambos protocolos finalizan. El número de *rounds* para NS-PRR se ha fijado a $10 \cdot N$ en escenarios *one-hop* y $6 \cdot N$ en escenarios *multi-hop*. En cuanto al NS-Hello se ha fijado este parámetro a $0.5 \cdot N$ *rounds* en entornos *one-hop* y $0.25 \cdot N$ *rounds* en entornos *multi-hop*.

En primer lugar, el parámetro N se fija con diferentes valores. Luego, para cada N, se han obtenido el número de vecinos descubiertos y otras métricas. Finalmente, se muestran los resultados de diferentes métricas en función del número de vecinos descubiertos.

Tabla 9.1: Parámetros de simulación.

Parámetro	Valor
Static	*True*
Modelo de radio	CC2420
Modelo de colisión	2
Potencia de transmisión	-5dBm
Tasa de paquetes	5 paquetes/s
Tamaño de paquete	2500 bytes
Tamaño de paquete *feedback*	14 bytes
Tamaño de primer *sub-slot* NS-Hello y NDSP	$\omega = N \cdot \tau$
τ	0.07s
Tamaño segundo *sub-slot* NDSP	$\omega_f = N \cdot \tau_f$
τ_f	0.000392s
Ranura selección de vecinos	$N \cdot \tau_f$
Tamaño *one-hop*	10mx10m
Tamaño *multi-hop*	100mx100m
Despliegue	Malla MxM
Número de *rounds* NS-PRR *one-hop*	10N
Número de *rounds* NS-Hello *one-hop*	0.5N
Número de *rounds* NS-PRR *multi-hop*	6N
Número de *rounds* NS-Hello *multi-hop*	0.25N

El número de vecinos descubiertos se define como la cantidad de vecinos que son descubiertos por un nodo en la fase de descubrimiento de vecinos.

En la Tabla 9.1 se resumen los parámetros usados para obtener los resultados de simulación.

9.3.2 Resultados de simulación

En esta sección se procede a presentar y discutir los resultados de simulación para NDSP, NS-Hello y NS-PRR, obtenidos mediante Castalia 3.2.

Consumo temporal

La Figura 9.3 representa el consumo temporal de los 3 protocolos en un entorno *one-hop*. Se concluye que NDSP mejora a ambos protocolos de referencia, seguido por NS-PRR, y finalmente NS-Hello presenta el peor consumo temporal. El consumo temporal presenta una tendencia creciente con el número de vecinos descubiertos en los 3 protocolos. Esto es debido a que en NDSP a medida

que el número de descubrimientos crece se necesita más tiempo para descubrir todos los vecinos. En cuanto a NS-Hello y NS-PRR el número de *rounds* depende de N (número de nodos), como se indicó en la sección 9.3.1. Por tanto el consumo temporal también depende del número de vecinos descubiertos con lo cual también se presenta una tendencia creciente. Resaltar que el procedimiento de selección de vecinos incluye muy poco *overhead* temporal, siendo de $0.0392s$ para redes compuestas por 100 nodos.

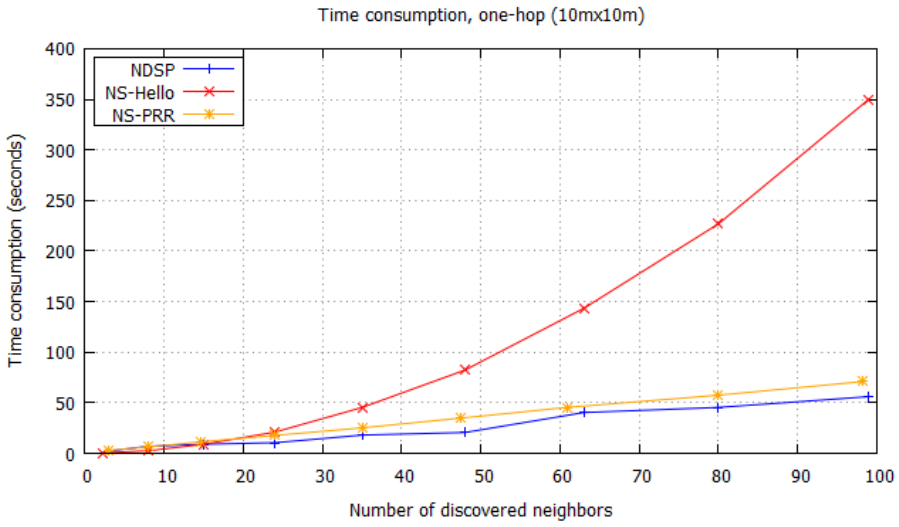

Figura 9.3: Consumo temporal (one-hop).

Según la Figura 9.4, NDSP de nuevo mejora a ambos protocolos de referencia en entornos *multi-hop*, seguido por NS-PRR y finalmente NS-Hello presenta el peor consumo temporal. De nuevo, el consumo temporal presenta una tendencia creciente con el número de vecinos descubiertos por el mismo motivo que en el caso *one-hop*. El *overhead* temporal introducido por la selección de vecinos también es muy bajo ($0.0392s$ para 100 nodos).

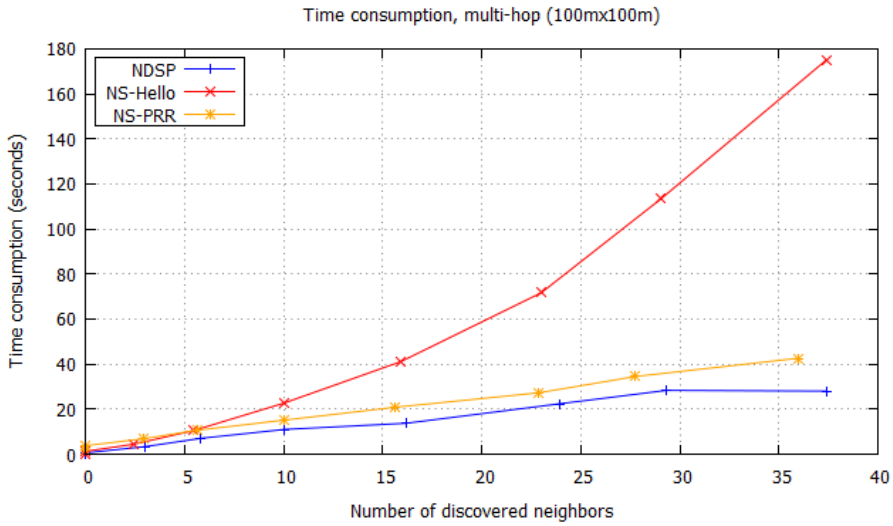

Figura 9.4: Consumo temporal (multi-hop).

Por lo tanto, el *overhead* en consumo temporal introducido por la gestión de prioridades es bajo. Esto significa una mejora con respecto a métodos previos.

Consumo energético

La Figura 9.5 representa el consumo energético de los 3 protocolos en un entorno *one-hop*. NDSP mejora ambos protocolos de referencia, seguido por NS-PRR, y NS-Hello es el peor. Ese orden es idéntico al obtenido para el consumo temporal, dado que a mayor consumo temporal mayor consumo energético. De nuevo, los 3 protocolos siguen una tendencia creciente con el nuúmero de vecinos descubiertos, principalmente debido a la tendencia creciente del consumo temporal. El consumo energético introducido por la selección de vecinos también es muy bajo.

De acuerdo con la Figura 9.6, de nuevo NDSP mejora a ambos protocolos de referencia en entornos *multi-hop*, seguido por NS-PRR, y NS-Hello es el peor. El consumo energético también sigue una tendencia creciente por el mismo motivo que en el caso *one-hop*. De nuevo, el consumo energético introducido por la selección de vecinos es buy bajo.

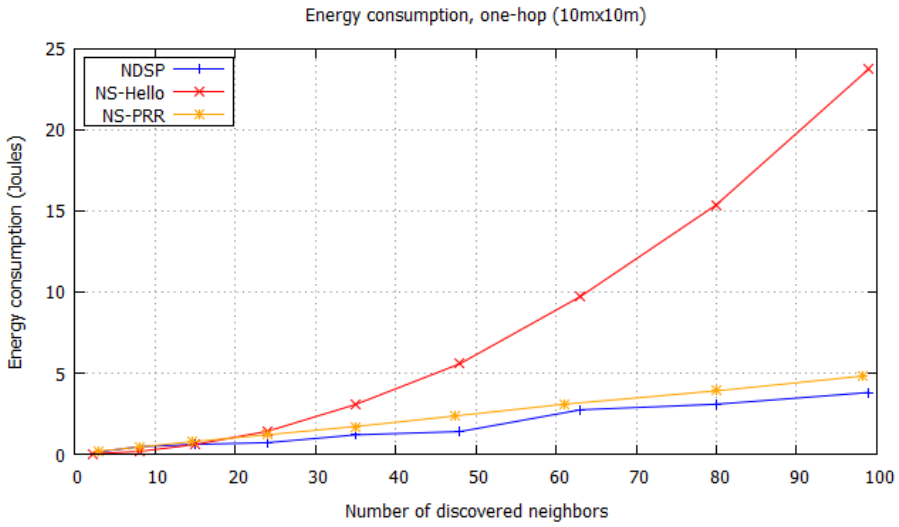

Figura 9.5: Consumo energético (one-hop).

Por lo tanto el *overhead* en consumo energético introducido por la gestión de prioridades es bajo. Esto significa una mejora con respecto a métodos previos.

Throughput

La Figura 9.7 muestra que NDSP mejora al NS-Hello con respecto al *throughput*, y NS-PRR es el peor, en entornos *one-hop*. El *throughput* decrece a medida que el número de vecinos descubiertos crece, principalmente debido a que a medida que el número de vecinos descubiertos crece hay más vecinos que descubrir. Por tanto se requiere más tiempo y el consumo temporal está inversamente relacionado con el *throughput*.

Según la Figura 9.8, NDSP mejora a ambos protocolos de referencia en entornos *multi-hop*, y NS-PRR presenta el peor *throughput*. De nuevo, el *throughput* decrece cuando el número de vecinos descubiertos crece, por el mismo motivo que para el caso *one-hop*. Sin embargo, en el caso *multi-hop*, NDSP y NS-Hello presentan resultados similares para número de vecinos descubiertos por encima de 16.

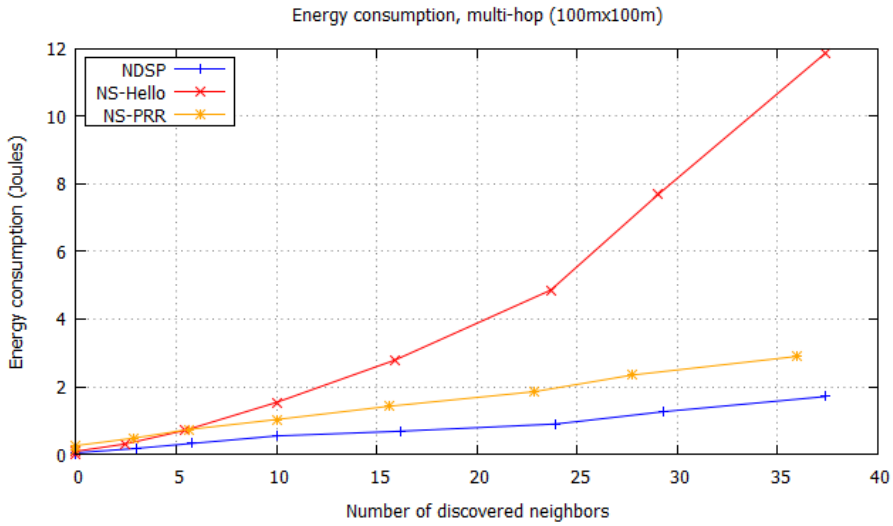

Figura 9.6: Consumo energético (multi-hop).

Número de paquetes enviados

Como se muestra en la Figura 9.9, NDSP mejora ambos protocolos de referencia en entornos *one-hop*, y NS-Hello presenta peores resultados. Los paquetes enviados presentan una tendencia creciente cuando el número de vecinos descubiertos se incrementa. Esto es debido a que en NDSP a medida que el número de vecinos se incrementa se necesita más tiempo para descubrir todos los vecinos por tanto se envían más paquetes. En cuento al NS-Hello y NS-PRR a medida que el número de vecinos descubiertos crece se tiene un mayor número de *rounds* por tanto la cantidad de paquetes enviados crece.

Según la Figura 9.10, NDSP mejora los protocolos de referencia en entornos *multi-hop*, y NS-Hello es el peor. El número de paquetes enviados crece a medida que el número de vecinos descubiertos se incrementa, por el mismo motivo que para entornos *one-hop*.

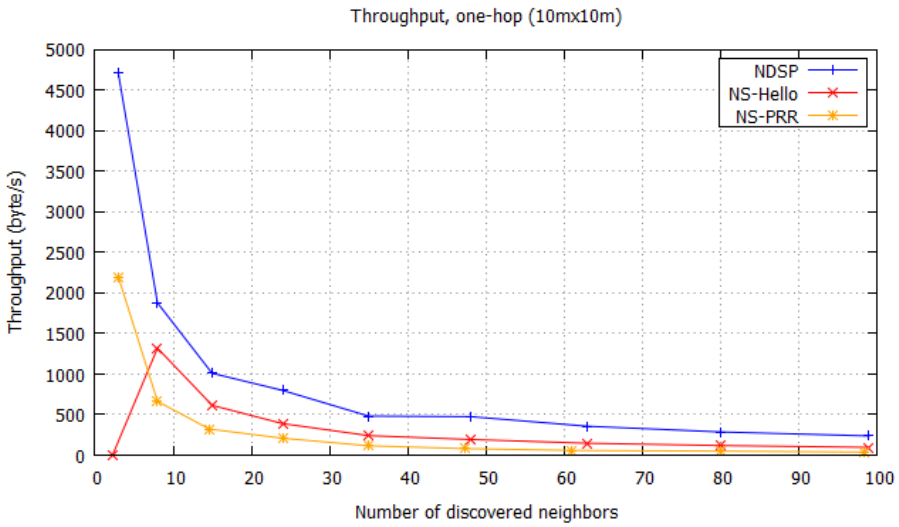

Figura 9.7: Throughput (one-hop).

9.4 Conclusiones

En este capítulo se aborda el estudio de procedimientos de descubrimiento y selección de vecinos para redes inalámbricas ad hoc estáticas teniendo en cuenta la existencia de colisiones.

Con este objetivo, se han elegido dos protocolos de la literatura, esto es, Hello y PRR, y han sido extendidos para llevar a cabo tanto el descubrimiento como la selección de vecinos: NS-Hello y NS-PRR. Ambos protocolos se usan como referencia para comparar con NDSP, una propuesta aleatoria basada en *handshake*, que combina el descubrimiento de vecinos y la selección de vecinos.

Además, mediante simulaciones con Castalia 3.2 descubrimos que NDSP logra mejores resultados que NS-PRR y NS-Hello. Esta mejoría se da con respecto a consumo temporal, consumo energético, *throughput*, y el número de paquetes enviados.

También concluimos que el *overhead* introducido por la seleción de vecinos es muy bajo, y que NDSP funciona siguiendo premisas más realistas.

Además, esta mejora es debida al esquema de descubrimiento de vecinos usado, esto es, la gestión de prioridades no afecta a las prestaciones dado que se usa

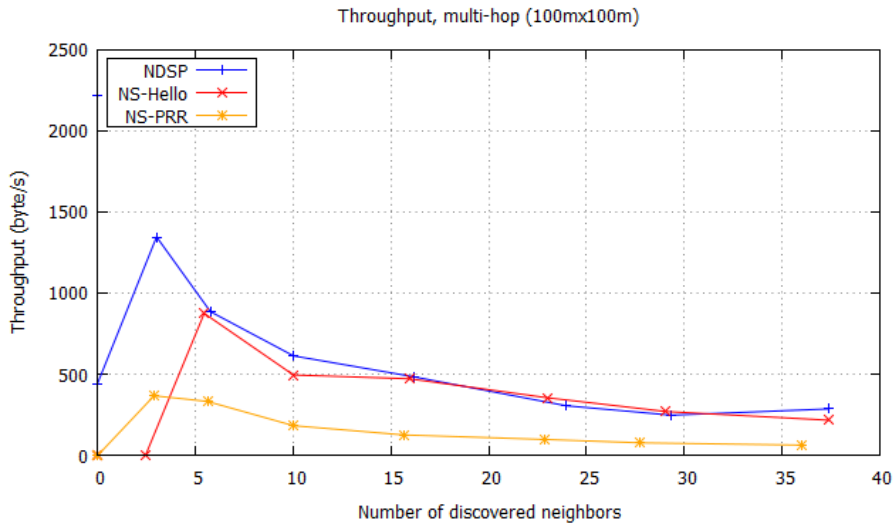

Figura 9.8: Throughput (multi-hop).

el mismo mecanismo en cuanto a prioridades en NDSP, NS-Hello y NS-PRR. Sin embargo, el *overhead* introducido por la gestión de prioridades es bajo, lo cual significa una mejora.

Como se indicó previamente, la propuesta elimina las dificultades de los protocolos de referencia, ya que sigue premisas más realistas. Entre ellas, sabe cuándo terminar, no necesita conocer el número de nodos, descubre todos los vecinos con probabilidad 1, y permite a los nodos iniciar la transmisión en diferentes instantes de tiempo.

NDSP se puede aplicar en redes de sensores inalámbricos. En ellas, se necesitan nodos favoritos para permitir que la información esté disponible para el exterior de la red (a través de un *gateway*). También puede ser útil para futuras operaciones como el encaminamiento.

Como posible trabajo futuro, se podría desarrollar y evaluar protocolos de descubrimiento y selección de vecinos conscientes de la energía para entornos estáticos con energía limitada. También sería interesante solucionar las limitaciones de la propuesta y así permitir su uso en redes móviles.

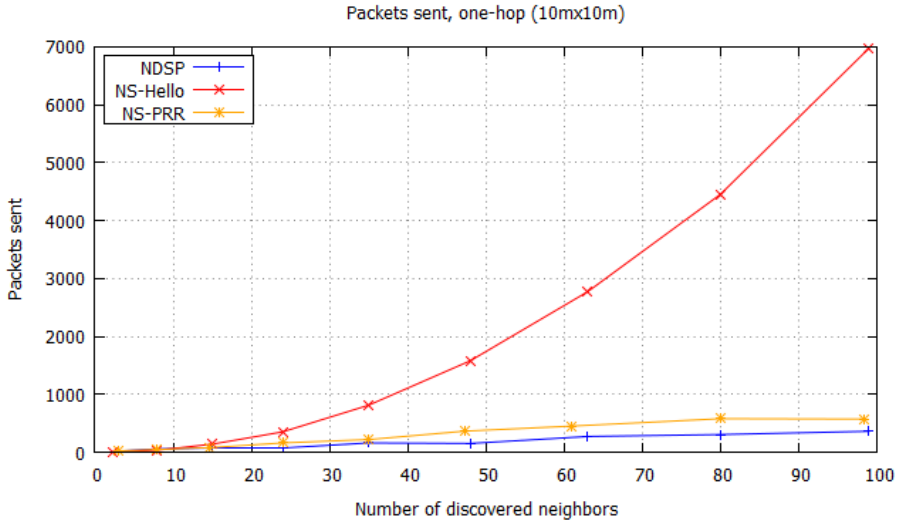

Figura 9.9: Paquetes enviados (one-hop).

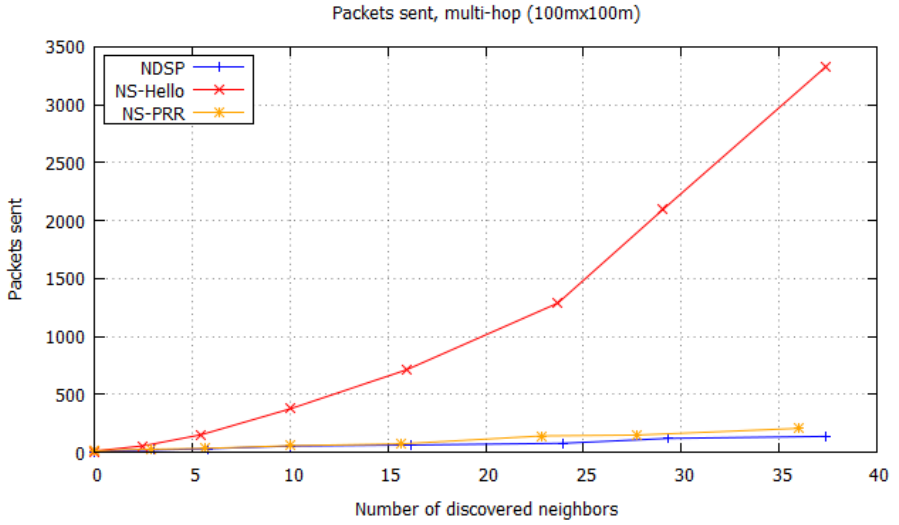

Figura 9.10: Paquetes enviados (multi-hop).

Capítulo 10

Comparación cualitativa de propuestas

En este capítulo se presenta una clasificación de las propuestas incluidas en esta tesis, así como una comparación cualitativa de las mismas.

10.1 Clasificación de las propuestas

De acuerdo con la Figura 10.1, la propuesta Leader-based es determinística, se basa en *handshake* lo cual significa que hay *ACKs*, y sólo funciona en un entorno *one-hop*. La propuesta TDMA-based es también determinística, se basa en *handshake* y funciona tanto en entornos *one-hop* como *multi-hop*. Ambas propuestas siguen una planificación en la transmisión predeterminada. Existe una propuesta aleatoria basada en la detección de colisiones (basada en *handshake*) usando un líder, lo que permite su uso de forma asíncrona, pero su uso se restringe a escenarios *one-hop*. En cuanto a CDH y CDPRR, son propuestas aleatorias basadas en detección de colisiones, que requieren sincronización en los límites de ranura, y permite su uso en *one-hop* y *multi-hop*. En cuanto a LECDH, se trata de una propuesta consciente de la energía. También es aleatoria basada en la detección de colisiones, requiere sincronización en los límites de ranura, permite su uso en *one-hop* y *multi-hop*. La propuesta para la creación de redes espontáneas basadas en la confianza, basada en vecinos de confianza, permite su uso tanto en escenarios *one-hop* como *multi-hop*. Final-

mente, NDSP es una propuesta aleatoria basada en la detección de colisiones que permite el descubrimiento y selección de vecinos. Utiliza un *handshake* tanto para determinar los nodos que son descubiertos con éxito como para determinar los nodos favoritos, y permite su uso tanto en entornos *one-hop* como en *multi-hop*.

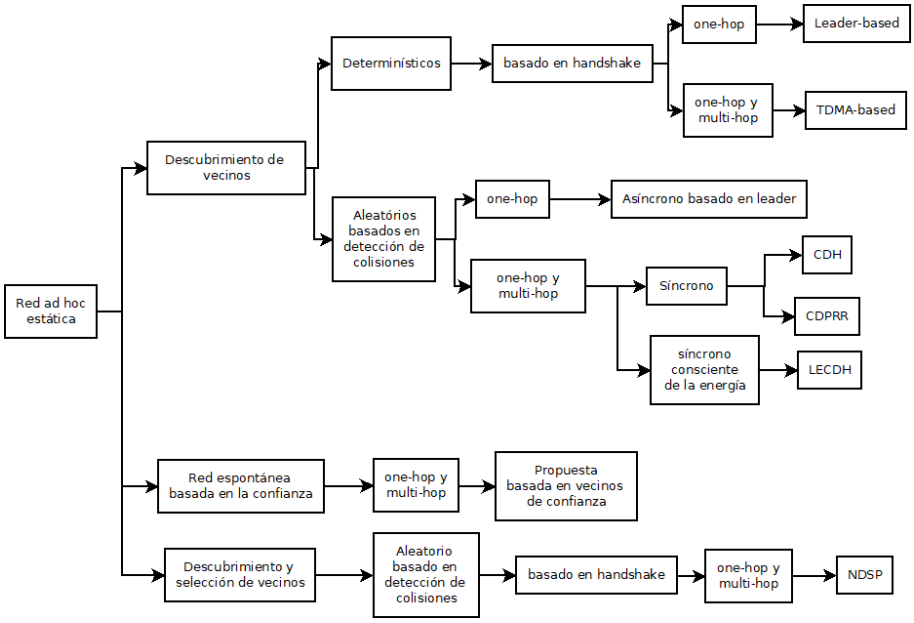

Figura 10.1: Clasificación de las propuestas.

10.2 Comparación cualitativa de las propuestas

En esta sección se presenta la Tabla 10.1, que incluye una comparación cualitativa de los protocolos de descubrimiento de vecinos desarrollados en esta tesis.

Según la Tabla 10.1, el número de nodos N puede ser desconocido en los protocolos CDH, asíncrono y LECDH. El protocolo asíncrono no requiere sincronización, mientras que TDMA y Leader son los únicos que requieren planificación en la transmisión predeterminada. Los protocolos basados en líder no funcionan en entornos *multi-hop*. LECDH es la única propuesta que tiene modo *sleep* y por tanto es consciente de la energía. El protocolo CDH, asíncrono y LECDH permiten a los nodos iniciar la transmisión en diferentes instantes de

tiempo. Además, todos los protocolos permiten el descubrimiento de todos los vecinos con probabilidad 1.

Tabla 10.1: Comparación cualitativa de las propuestas de descubrimiento de vecinos.

	TDMA	Leader	CDPRR	CDH	Asíncrono	LECDH
Entorno estático	✓	✓	✓	✓	✓	✓
Entorno móvil						
Protocolo aleatorio			✓	✓	✓	✓
Tiempo ranurado			✓	✓	✓	✓
N permanece desconocido				✓	✓	✓
No requiere sincronización					✓	
No sigue una planificación			✓	✓	✓	✓
Half-duplex	✓	✓	✓	✓	✓	✓
Escenario *one-hop*	✓	✓	✓	✓	✓	✓
Escenario *multi-hop*	✓		✓	✓		✓
Modo *sleep* disponible						✓
Se consideran colisiones	✓	✓	✓	✓	✓	✓
Detección pérdida de paquetes						
Líder necesario		✓			✓	
Detección de colisiones			✓	✓	✓	✓
Detección de terminación			✓	✓	✓	✓
Inicia transmisión en diferentes instantes de tiempo				✓	✓	✓
Descubre todos los vecinos con probabilidad 1	✓	✓	✓	✓	✓	✓
Con mecanismo de *feedback*	✓	✓	✓	✓	✓	✓

Capítulo 11

Conclusiones

En este capítulo se presentan las conclusiones del trabajo llevado a cabo, las contribuciones de esta tesis, publicaciones fruto del trabajo llevado a cabo. También se incluyen algunas líneas de trabajo futuro.

11.1 Conclusiones

En esta tesis se han presentado varios protocolos de descubrimiento para redes inalámbricas ad hoc, cada uno considerando unas premisas distintas que limitan su funcionamiento en entornos prácticos.

Además, se ha propuesto un protocolo para la creación de redes espontáneas basadas en la confianza, y un protocolo para la selección de vecinos.

También se ha procedido a su simulación a través de Castalia 3.2 para obtener resultados de sus prestaciones, según distintas métricas.

En el caso de los protocolos determinísticos, CDH, CDPRR y NDSP, se ha procedido a presentar un modelo analítico según varias métricas.

Se ha concluido que cada protocolo permite obtener unos resultados de simulación distintos a medida que se van relajando las premisas que sigue. Cada protocolo presenta unas ventajas en inconvenientes así como limitaciones prácticas y posibles aplicaciones prácticas.

Además, todos los protocolos de descubrimiento de vecinos elaborados en esta tesis pueden ser usados para la creación de redes espontáneas basadas en la confianza si se considera el emitir la tarjeta de identidad con cada paquete enviado y en el otro extremo se comprueba si la firma es correcta, con lo que detectará vecinos de confianza.

11.2 Problemas encontrados y cómo se han solucionado

El principal problema encontrado ha sido la existencia de colisiones, por tanto se ha utilizado un mecanismo de detección de colisiones.

Otro problema ha sido cómo indicar a los transmisores que emitieron con éxito para dejar de transmitir. El mecanismo utilizado ha sido la detección de energía que averigua si los receptores han enviado *feedbacks*. Además, se ha desarrollado un mecanismo de *feedback* a usar en la mayoría de protocolos de descubrimiento.

Además, han ido surgiendo problemas a abordar como el desconocimiento del número de nodos, saber cuándo terminar el descubrimiento de vecinos, o descubrir todos los vecinos con probabilidad 1. Además los protocolos deberían ser asíncronos, proporcionar bajo consumo energético. Para solucionar estos problemas se han propuesto distintos protocolos que los solucionan.

En cuanto al protocolo de creación de redes espontáneas basadas en la confianza, surgen los mismos problemas que para los protocolos de descubrimiento. Se han solucionado aplicando un protocolo de descubrimiento en dos fases. Además, se desea detectar los vecinos de confianza, lo que se soluciona intercambiando tarjetas de identidad y comprobando firma para conocer los vecinos de confianza.

En la selección de vecinos, el problema principal es determinar los vecinos favoritos de forma que sepan que lo son, afrontando los mismos problemas que los protocolos de descubrimiento de vecinos. Para ello se ha intercambiado la prioridad usando un protocolo de descubrimiento de vecinos, y luego se ha comunicado a los nodos si son favoritos usando detección de energía.

11.3 Aportaciones personales

En la realización de la tesis he aprendido muchas cosas, como el uso del simulador Castalia 3.2 para la implementación de protocolos, y el conocimiento de los protocolos más relevantes de la literatura. También, he desarrollado la habilidad para el diseño, evaluación y valoración de protocolos de distinto tipo. En cuanto a las publicaciones, he aprendido muchas cosas, incluso en la fase de revisión, que son útiles tanto para la confección de artículos como para la defensa del trabajo llevado a cabo.

11.4 Contribuciones

Las principales contribuciones de esta tesis son las siguientes:

- Leader-based, una propuesta determinística que logra descubrir todos los vecinos con probabilidad 1. Sigue una planificación en la transmisión predeterminada, incluye un nodo especial conocido como líder que inicia el descubrimiento. Termina el descubrimiento de acuerdo con la planificación, sólo puede usarse en entornos *one-hop*, aunque debe conocer el número total de nodos de la red.

- TDMA-based, una propuesta determinística que también logra descubrir todos los vecinos con probabilidad 1. Sigue una planificación en la transmisión predeterminada, termina el descubrimiento de acuerdo con la planificación. Se puede usar tanto en entornos *one-hop* como *multi-hop*, aunque debe conocer cuántos nodos hay en la red.

- Una comparación cualitativa de protocolos determinísticos de la literatura.

- Una comparación cualitativa de Hello, PRR y las propuestas Leader-based y TDMA-based.

- Un estudio analítico de las propuestas Leader-based y TDMA-based en cuanto a consumo temporal, consumo energético, *throughput*. También en cuanto a número de descubrimientos por paquetes enviados, y paquetes recibidos por paquetes enviados.

- Una implementación del Leader-based, TDMA-based y protocolos de referencia se ha llevado a cabo con Castalia 3.2. El objetivo es comparar las prestaciones de esos protocolos en cuanto al número de vecinos descubiertos y las otras cinco métricas usadas en el estudio analítico.

- CDH, una propuesta aleatoria basada en la detección de colisiones y en el protocolo Hello, con un tamaño de ranura fijo. Logra descubrir todos los vecinos con probabilidad 1. Permite terminar cuando todos los vecinos han sido descubiertos mediante un mecanismo de detección de terminación. Sigue premisas más realistas, tales como no requerir conocer el número de nodos de la red, y los nodos pueden transmitir en diferentes instantes de tiempo. Es adecuado para su uso tanto en entornos *one-hop* como *multi-hop*.

- CDPRR, una propuesta aleatoria basada en detección de colisiones y en el protocolo PRR. Se tiene una probabilidad de transmisión $\frac{1}{N}$ fija durante todo el proceso de descubrimiento de vecinos. Logra el descubrimiento de todos los vecinos con probabilidad 1. Incluye un mecanismo de detección de terminación, según el cual el protocolo termina cuando todos los vecinos han sido descubiertos. Sin embargo, requiere conocer el número de nodos de la red. Es adecuado para su uso tanto en entornos *one-hop* como *multi-hop*.

- Una comparación cualitativa de los protocolos del estado del arte y de las dos propuestas CDH y CDPRR.

- Implementación de ambas propuestas CDH y CDPRR, y los protocolos de referencia en el simulador Castalia 3.2. El objetivo es obtener resultados en cuanto a tiempos, número de vecinos descubiertos, consumo energético, y *throughput*. También se obtiene el *ratio* número de vecinos descubiertos por paquetes enviados. Además, se concluye que las propuestas son más rápidas y más eficientes en energía que las soluciones existentes.

- Propuesta aleatoria basada en líder, en la detección de colisiones y Hello. Logra descubrir todos los vecinos con probabilidad 1, y terminar cuando todos los vecinos han sido descubiertos. Sigue premisas más realistas, y permite su uso de forma asíncrona, lanzando el descubrimiento el nodo líder. El número de nodos puede ser desconocido. No requiere una planificación en la transmisión. Sin embargo, sólo funciona en entornos estáticos *one-hop*.

- Comparación cualitativa de la propuesta aleatoria basada en líder, Hello, y un protocolo basado en líder determinístico.

- Implementación en Castalia 3.2 de la propuesta aleatoria basada en líder y los 2 protocolos de referencia. Se concluye que la propuesta tiene mejores resultados que el Hello según el tiempo de descubrimiento de vecinos, y número de vecinos descubiertos. También presenta mejoría en cuanto

a consumo energético y *throughput*. Por otro lado, la propuesta presenta resultados razonables en comparación con el protocolo determinístico basado en líder en relación a tiempos, consumo energético y *throughput*.

- Propuesta LECDH, un protocolo aleatorio consciente de la energía basado en detección de colisiones. Presenta un tamaño fijo de ranura. Descubre todos los vecinos casi con probabilidad 1, conoce cuándo terminar, y no sigue una planificación en la transmisión. Además permite desconocer el número de nodos, y es apropiado para entornos tanto *one-hop* como *multi-hop*.

- Una comparación cualitativa del protocolo EAH y la propuesta LECDH.

- Una implementación en Castalia 3.2 y comparación de prestaciones de la propuesta LECDH y el protocolo EAH, para diferentes *duty cycles*.

- Un modelo analítico para ambas propuestas CDPRR y CDH en cuanto a tiempo de descubrimiento de vecinos, consumo energético, *throughput*, y *overhead* (número de paquetes enviados). También se ha modelado el *packet delivery ratio*, porcentaje de descubrimientos por *round*, la CDF de descubrimientos, y el porcentaje de *idle slots*. Todo ello en un escenario *one-hop*.

- Un modelo analítico para dos protocolos de referencia (Hello y PRR). Se modela el tiempo de descubrimiento de vecinos, consumo energético, *throughput*, y *overhead* (número de paquetes enviados). También se ha modelado el *packet delivery ratio*, porcentaje de descubrimientos por *round*, la CDF de descubrimientos y el porcentaje de *idle slots*. El modelo se ha realizado teniendo en cuenta un escenario *one-hop*.

- Resultados gráficos usando las ecuaciones obtenidas, comparando los cuatro protocolos (CDH, CDPRR, Hello y PRR) en cuanto a ocho métricas.

- Resultados gráficos de CDH variando el tamaño de ranura tanto dependiendo del número de nodos y con tamaño fijo de ranura.

- Un protocolo aleatorio en 2 fases para la creación de redes espontáneas basadas en la confianza. Logra tratar con y detectar las colisiones, permite detectar cuándo terminar el descubrimiento, puede usar una probabilidad de transmisión fija. Además, no requiere del conocimiento del número de nodos, no depende de seguir una planificación en la transmisión. Logra descubrir todos los vecinos, intercambiar con éxito las tarjetas de identidad y descubrir todos los vecinos de confianza con probabilidad casi

1. Sigue premisas más realistas, y es adecuado para ser usada tanto en entornos *one-hop* como *multi-hop*.

- Una comparación cualitativa de trabajos relacionados y la propuesta de creación de redes espontáneas basadas en la confianza.
- Implementación en Castalia 3.2 y comparación de la propuesta de creación de redes espontáneas basadas en la confianza con un protocolo de referencia.
- Un estudio del comportamiento de la propuesta de creación de redes espontáneas basadas en la confianza, variando la probabilidad de transmisión. Además, se ha comprobado que la propuesta es más rápida y consume menos energía que soluciones existentes.
- NDSP, una propuesta aleatoria basada en el protocolo Hello y en la detección de colisiones. Logra descubrir todos los vecinos y utiliza prioridades para seleccionar nodos favoritos que serán usados en operaciones futuras.
- NS-Hello y NS-PRR, propuestas para descubrimiento y selección de vecinos, ampliando Hello y PRR.
- Implementación en Castalia 3.2 de la propuesta NDSP, NS-Hello y NS-PRR. También se ha llevado a cabo su comparación con respecto a consumo temporal, consumo energético, *throughput*, y *overhead* (número de paquetes enviados).

11.5 Trabajo futuro

Como posible trabajo futuro, se podría realizar lo siguiente:

- Incluir un mecanismo que permita elegir un líder, mediante el envío de notificaciones, para mejorar los protocolos determinístico y aleatorio basados en líder.
- Proponer y evaluar un método que permita sincronizar los protocolos CDH y CDPRR en los límites de ranura.
- Proponer y evaluar un protocolo aleatorio asíncrono que permitiera su uso en entornos *multi-hop*, y así mejorar la propuesta asíncrona incluida en esta tesis.

- Mejorar el protocolo consciente de la energía LECDH para que pueda funcionar mejor con bajos *duty cycles* y gran cantidad de nodos.
- Extender los protocolos para su uso en redes móviles.
- Mejorar la seguridad del modelo de creación de redes espontáneas basada en la confianza.
- Implementar y evaluar la integración de otra propuesta de descubrimiento de vecinos de esta tesis con el intercambio de tarjetas de identidad. El objetivo será la creación de redes espontáneas basadas en la confianza siguiendo otro esquema de descubrimiento de vecinos.
- Implementación de los protocolos presentados por ejemplo en una red de sensores inalámbricos real, para obtener resultados experimentales.

11.6 Publicaciones

En esta sección se enumeran las publicaciones producto del trabajo llevado a cabo en esta tesis.

Revistas internacionales

1. Sorribes, J.V., Peñalver, L., Calafate, C.T., Lloret, J. (2021) Randomized neighbor discovery protocols with collision detection for static multihop wireless ad hoc networks. Telecommunication Systems, 77, 577-596 (2021). https://doi.org/10.1007/s11235-021-00763-4
2. Jose Vicente Sorribes, Lourdes Peñalver, Jaime Lloret. (2021) A Spontaneous Wireless Ad Hoc Trusted Neighbor Network Creation Protocol. Wireless Communications and Mobile Computing, vol. 2021, Article ID 5531923, 20 pages, 2021. https://doi.org/10.1155/2021/5531923
3. Sorribes, J.V., Peñalver, L., Lloret, J. et al. Collision Avoidance Based Neighbor Discovery in Ad Hoc Wireless Networks. Wireless Pers Commun (2021). https://doi.org/10.1007/s11277-021-09091-x
4. Jose Vicente Sorribes, Jaime Lloret, Lourdes Peñalver. (2021) Analytical Models for Randomized Neighbor Discovery Protocols based on Collision Detection in Wireless Ad Hoc Networks, Ad Hoc Networks, Vol. 126, 102739, 1-19 (2021). https://doi.org/10.1016/j.adhoc.2021.102739.

Congresos internacionales

1. Sorribes Díaz, José Vicente; Peñalver Herrero, Mª Lourdes; Tavares De Araujo Cesariny Calafate, Carlos Miguel; Lloret, Jaime (2020). Collision-aware Deterministic Neighbor Discovery in Static Ad Hoc Wireless Networks. EN Global Conference on Wireless and Optical Technologies (GC-WOT'20). (1 - 8). Online.

2. Sorribes Díaz, José Vicente; Peñalver Herrero, Mª Lourdes; Lloret, Jaime (2020). An Asynchronous Leader based Neighbor Discovery protocol in Static Wireless Ad hoc Networks. EN International Conference on Applied Soft Computing and Communication Networks (ACN'20). (145 - 161). Online

3. Sorribes Díaz, José Vicente; Lloret, Jaime; Peñalver Herrero, Mª Lourdes (2021) Neighbor Discovery and Selection based on the Management of Priorities in Wireless Ad Hoc Networks. EN 2021 Global Congress on Electrical Engineering (GC-ElecEng 2021), 10-12 December 2021, Valencia, Spain. Online.

Congresos nacionales

1. Sorribes Díaz, José Vicente; Peñalver Herrero, Mª Lourdes (2018). Protocolo de descubrimiento de vecinos asíncrono basado en leader para redes inalámbricas ad hoc. EN XXIX Jornadas de Paralelismo. Jornadas SARTECO 2018. (397 - 403). Teruel, Spain: Universidad de Zaragoza.

2. Sorribes Díaz, José Vicente; Peñalver Herrero, Mª Lourdes (2017). Estudio de Protocolos de Descubrimiento de Vecinos en Redes Inalámbricas Ad Hoc. EN XXVIII Jornadas de Paralelismo. Jornadas SARTECO 2017. (487 - 496). Málaga, Spain: Sociedad de Arquitectura y Tecnología de Computadores (SARTECO) .

3. Sorribes Díaz, José Vicente; Peñalver Herrero, Mª Lourdes (2016). TrustedNet: protocolo para la creación de redes espontáneas basadas en la confianza. EN XXVII Jornadas de Paralelismo. Jornadas SARTECO 2016. (369 - 377). Salamanca, Spain: Universidad de Salamanca.

Bibliografía

[1] Sun, G., Wu, F., Gao, X., Chen, G., Wang, W. (2013). Time-efficient protocols for neighbor discovery in wireless ad hoc networks. IEEE Transactions on Vehicular Technology, 62(6), 2780–2791. https://doi.org/10.1109/TVT.2013.2246204

[2] Vasudevan, S., Adler, M., Goeckel, D., Towsley, D. (2013). Efficient algorithms for neighbor discovery in wireless networks. IEEE/ACM Transactions on Networking, 21(1), 69–83. https://doi.org/10.1109/TNET.2012.2189892

[3] Ling, H., Yang, S. (2019). Passive neighbor discovery with social recognition for mobile ad hoc social networking applications. Wireless Networks, 25, 4247–4258. https://doi.org/10.1007/s11276-019-02087-3

[4] Chen, H., Qin, Y., Lin, K., Luan, Y., Wang, Z., Yu, J., Li, Y. (2020). PWEND: Proactive wakeup based energy-efcient neighbor discovery for mobile sensor networks. Ad Hoc Networks, 107, 102247. https://doi.org/10.1016/j.adhoc.2020.102247

[5] Vasudevan, S., Towsley, D., Goeckel, D., Khalili, R. (2009) Neighbor discovery in wireless networks and the coupon collector's problem. MobiCom '09: Proceedings of the 15th annual international conference on Mobile computing and networking. September 2009 Pages 181–192. doi: 10.1145/1614320.1614341.

[6] Han, G., Li, X., Jiang, J., Shu, L., Lloret, J. (2015). Intrusion detection algorithm based on neighbor information against sinkhole attack

in wireless sensor networks. The Computer Journal, 58(6), 1280–1292. https://doi.org/10.1093/comjnl/bxu036

[7] Mcglynn, M. J., Borbash, S. A. (2001). Birthday protocols for low energy deployment and fexible neighbor discovery in ad hoc wireless networks. In Proceedings of the 2nd ACM international symposium on mobile ad hoc networking computing (pp. 137–145). ACM Press. https://doi.org/10.1145/501431.501435

[8] Stoleru, R., Wu, H., Chenji, H. (2011). Secure neighbor discovery in mobile ad hoc networks. In Proceedings—8th IEEE international conference on mobile ad-hoc and sensor systems, MASS 2011 (pp. 35–42). https://doi.org/10.1109/MASS.2011.15

[9] Varghane, N., Kurade, B. (2014). Secure protocol and signature based intrusion detection for spontaneous wireless AD HOC network. International Journal of Computer Science and Mobile Computing (IJCSMC), 3(5), 758–768.

[10] Feeney, L.M., Ahlgren, B., Westerlund, A. (2001). Spontaneous networking: an application oriented approach to ad hoc networking. IEEE Communications Magazine, vol. 39, no. 6, pp. 176–181, 2001.

[11] Sun, G., Wu, F., Gao, X., Chen, G., Wang, W. (2013). Time-efficient protocols for neighbor discovery in wireless ad hoc networks. IEEE Transactions on Vehicular Technology, vol. 62, no. 6, pp. 2780–2791, 2013.

[12] Conti, M., Crowcroft, J., Maselli, G., Turi, G. (2005) A modular cross-layer architecture for ad hoc networks. In Handbook on Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless, and Peer-to-Peer Networks, J. Wu, Ed., pp. 1–12, Auerbach Publications, New York, NY, USA, 2005.

[13] Preuß, S., Cap, C.H., Rostock, U. (2000) Overview of spontaneous networking-evolving concepts and technologies. Rostocker Informatik-Berichte, vol. 24, pp. 113–123, 2000.

[14] Ye, Z., Wen, T., Liu, Z., Song, X., Fu, C. (2017) An Efficient Dynamic Trust Evaluation Model for Wireless Sensor Networks. Journal of Sensors, vol. 2017, Article ID 7864671, 16 pages, 2017.

[15] Sakthidasan, K., Vasudevan, N., Devabalaji, K.R., Sudhakar, T., Haes, H., Yuvara, T. (2021) A Recurrent Reward Based Learning Technique for Secure Neighbor Selection in Mobile AD-HOC Networks. IEEEAccess, 9:21735-21745, 2021. doi: 10.1109/ACCESS.2021.3055422.

[16] Zarifzadeh, S., Yazdani, N. (2013) Neighbor Selection Game in Wireless Ad Hoc Networks. Wireless Pers Commun (2013) 70:617–640. doi: 10.1007/s11277-012-0711-6.

[17] Călinescu, G., Măndoiu, I.I., Wan, PJ. et al. (2004) Selecting Forwarding Neighbors in Wireless Ad Hoc Networks. Mobile Networks and Applications 9, 101–111 (2004). doi: 10.1023/B:MONE.0000013622.63511.57.

[18] Liu, B.H, Gao, Y., Chou, C.T., Jha, S. (2004) An energy efficient select optimal neighbor protocol for wireless ad hoc networks. 29th Annual IEEE International Conference on Local Computer Networks, 2004, pp. 626-633. doi: 10.1109/LCN.2004.19.

[19] Uchiyama, A., Fujii, S., Umedu, T., Yamaguchi, H., Higashino, T. (2008) Neighbor Selection Algorithm for Ad Hoc Networks with Highly Dynamic Urban Mobility. 2008 International Wireless Communications and Mobile Computing Conference, 2008, pp. 165-170. doi: 10.1109/IWCMC.2008.29.

[20] R. Khalili, D. L. Goeckel, D. Towsley and A. Swami (2010) Neighbor Discovery with Reception Status Feedback to Transmitters. 2010 Proceedings IEEE INFOCOM, 2010, pp. 1-9. doi: 10.1109/INFCOM.2010.5462064.

[21] McGlynn, M.J., Borbash, S.A. (2001). Birthday protocols for low energy deployment and flexible neighbor discovery in ad hoc wireless networks. In Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking Computing, ACM Press, 2001, pp.137-145.

[22] Ben Hamida, E., Chelius, G., Busson, A., Fleury, E. (2008). Neighbor discovery in multi-hop wireless networks: Evaluation and dimensioning with interference considerations. Discrete Mathematics and Theoretical Computer Science DMTCS, 10(2):87-114, May 2008.

[23] Vasudevan, S., Adler, M., Goeckel, D., Towsley, D. (2013) Efficient algorithms for neighbor discovery in wireless networks. IEEE/ACM Transactions on Networking, vol. 21, Feb.2013, pp. 69-83. doi:10.1109/TNET.2012.2189892.

[24] Sun, G., Wu, F., Gao, X., Chen, G., Wang, W. (2013). Time-efficient protocols for neighbor discovery in wireless ad hoc networks. IEEE Transactions on Vehicular Technology, vol. 62, Jul.2013, pp. 2780-2791. doi:10.1109/TVT.2013.2246204.

[25] Muhammed Irfan, S., Ali, S., Mathew, J.A. (2014). Protocol Design for Neighbor Discovery in Ad-Hoc Network. International Journal of Electronic and Electrical Engineering, 7(9):915-922, 2014.

[26] Vasudevan, S., Adler, M., Goeckel, D., Towsley, D. (2013) Efficient algorithms for neighbor discovery in wireless networks. IEEE/ACM Transactions on Networking, vol. 21, Feb.2013, pp. 69-83. doi:10.1109/TNET.2012.2189892.

[27] Luo, J., Guo, D. (2008). Neighbor Discovery in Wireless Ad Hoc Networks Based on Group Testing. In 46th Annual Allerton Conference on Communication, Control, and Computing, pp. 791-797. ACM Press Dec. 2008.

[28] Chen, L., Li, Y., Chen, Y., Liu, K., Zhang, J., Cheng, Y., You, H., Luo, Q. (2015). Prime-set-based neighbor discovery algorithm for low duty-cycle dynamic WSNs. Electronics Letters, 51(6):534-536, 2015. doi: 10.1049/el.2014.3879.24.

[29] Dutta, P., Culler, D. (2008). Practical asynchronous neighbor discovery and rendezvous for mobile sensing applications. In SenSys, January 2008, pp. 71-84. doi:10.145/1460412.1460420.

[30] Bakht, M., Kravets, R. (2010). SearchLight: A systematic probing-based asynchronous neighbor discovery protocol. In Illinois Digital Environment for Access to Learning and Scholarship Repository, 2010, unpublished.

[31] Margolies, R., Grebla, Chen, G.T., Rubenstein, D. Zussman, G. (2016). Panda: Neighbor discovery on a power harvesting budget. IEEE Journal on Selected Areas in Communications, 34(12):3606-3619, 2016. doi: 10.1109/JSAC.2016.2611984.

[32] Qiu, Y., Li, S., Xu, X., Li, Z. (2016). Talk more listen less: Energy-efficient neighbor discovery in wireless sensor networks. In The 35th Annual IEEE International Conference on Computer Communications, IEEE INFOCOM 2016, pp. 1-9, 2016. doi: 10.1109/INFOCOM.2016.7524336.

[33] Zheng, R., Hou, J.C., Sha, L. (2003). Asynchronous wakeup for ad hoc networks. In Proc. of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc'03, pp. 35-45, 2003. doi: 10.1145/778415.778420.

[34] Kandhalu, A., Lakshmanan, K., Rajkumar, R. (2010). U-Connect: A low-latency energy-efficient asynchronous neighbor discovery protocol. In Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks, IPSN'10, January 2010, pp. 350-361. doi:10.14/1791212.1791253.

[35] Wang, K., Mao, X. Liu, Y. (2013). Blinddate: A neighbor discovery protocol. IEEE Transactions on Parallel and Distributed Systems, 26(4):120-129, 2013. doi: 10.1109/ICPP.2013.21.

[36] Astudillo, G., Kadoch, M. (2017). Neighbor discovery and routing schemes for mobile ad-hoc networks with beamwidth adaptive smart antennas. Telecommunication Systems, 66, 17–27. doi: 10.1007/s11235-016-0268-x.

[37] Yuan, Z., Lizhao, Y., Li, W., Chen, B., Xu, Z. (2011) History-Aware Adaptive Backoff for Neighbor Discovery in Wireless Networks. 2011 Seventh International Conference on Mobile Ad-hoc and Sensor Networks, 2011, pp. 174-181. doi: 10.1109/MSN.2011.38.

[38] Khatibi, S., Rohani, R. (2010). Quorum-based neighbor discovery in self-organized cognitive MANET. In 21st Annual IEEE International Symposium on Personal. Indoor and Mobile Radio Communications. (pp. 2239–2243). IEEE. doi: 10.1109/PIMRC.2010.5671683.

[39] Yang, S., Wang, C., Jiang, C. (2018). Centron: Cooperative neighbor discovery in mobile ad-hoc networks. Computer Networks, vol. 136, March 2018, pp. 128-136. doi: 10.1016/j.comnet.2018.03.003.

[40] Chen, L., Fan, R., Zhang, Y., Shi, S., Bian, K., Chen, L., et al. (2018). On heterogeneous duty cycles for neighbor discovery in wireless sensor networks. Ad Hoc Networks, Elsevier, 77, 54–68. doi: 10.1016/j.adhoc.2018.04.007.

[41] Wang, Q., He, X., Chen, N. (2019) A Cross-layer Neighbour Discovery Algorithm in Ad hoc Networks based on Hexagonal Clustering and GPS, IOP Conference Series: Earth and Environmental Science, 6th Annual 2018 International Conference on Geo-Spatial Knowledge and Intelligence, 14-16 December 2018, Hubei, China, vol. 234, 012050, pp. 1-6. doi: 10.1088/1755-1315/234/1/012050.

[42] Sravankumar, B., Moparthy, N. R. (2021). A survey on continuous neighbor discovery for mobile low duty cycle wireless sensor network. In Materials Today: Proceedings. ISSN 2214-7853. doi: 10.1016/j.matpr.2021.01.463.

[43] Hess, A., Hyytia, E., Ott, J. (2014). Efficient neighbor discovery in mobile opportunistic networking using mobility awareness. In Proc. 6th International Conference on Communication Systems and Networks (COMSNETS) (pp. 1–8).

[44] Gu, Z., Cao, Z., Tian, Z., Wang, Y., Du, X., Mohsen, G. (2020). A low-latency and energy-efficient neighbor discovery algorithm for wireless sensor networks. Sensors. doi: 10.3390/s20030657.

[45] Ling, H., Yang, S. (2019) Passive neighbor discovery with social recognition for mobile ad hoc social networking applications. Wireless Networks, 25:4247-4258. doi: 10.1007/s11276-019-02087-3.

[46] Chen, H., Qin, Y., Lin, K., Luan, Y., Wang, Z., Y, J., Li, Y. (2020). PWEND: Proactive wakeup based energy-efficient neighbor discovery for mobile sensor networks. Ad Hoc Networks, vol. 107, 102247, Oct. 2020. doi: 10.1016/j.adhoc.2020.102247.

[47] Chen, H., Lou, W., Wang, Z., Xia, F. (2018) On achieving asynchronous energy-efficient neighbor discovery for mobile sensor networks. IEEE Trans. Emerg. Top. Comput., 6, 553-565.

[48] Chunfeng, L., Gang, Z., Weisi, G., Ran, H. (2020). Kalman Prediction-Baed Neighbor Discovery and Its Effect on Routing Protocol in Vehicular Ad Hoc Networks. IEEE Transactions on Intelligent Transportation Systems, 21(1):159-169. doi: 10.1109/TITS.2018.2889923.

[49] Li, X., Mitton, N., Simplot-Ryl, D. (2011). Mobility prediction based neighborhood discovery in mobile ad hoc networks. In Proc. 10th Int. IFIP TC Netw. Conf., Valencia, Spain, May 2011, pp. 241-253.

[50] Taleb, T., Sakhaee, E., Jamalipour, A., Hashimoto, K., Kato, N., Nemoto, Y. (2007). A stable routing protocol to support ITS services in VANET networks. IE Trans. Veh. Technol, 56(6):3337-3347, Nov. 2007.

[51] Carty, J., Jayaweera, S.K. (2019) Distributed Network, Neighbor Discovery and Blind Routing for Mobile Wireless Ad-hoc Networks. 12th IFIP Wireless and Mobile Networking Conference (WMNC), Paris, France, pp. 131-135. doi: 10.23919/WMNC.2019.8881802.

[52] Wei, Z., Han, C., Qiu, C., Feng, Z., Wu, H. (2019) Radar Assisted Fast Neighbor Discovery for Wireless Ad Hoc Networks. IEEE Access, vol. 7, pp. 176514-176524. doi: 10.1109/ACCESS.2019.2950277.

[53] Li, J., Peng, L., Ye, Y., Xu, R., Zhao, W., Tian, C. (2014). A neighbor discovery algorithm in network of radar and communication integrated system. In Proc. IEEE 17th Int. Conf. Comput. Sci. Eng. (CSE), Chengdu, China, Dec. 2014, pp. 1142-1149.

[54] El Khamlichi, B., Nguyen, DHN., El Abbadi, J., Rowe, N.W., Kumar, S. (2019). Learning Automaton-Based Neighbor Discovery for Wireless Networks Using Directional Antennas. IEEE Wireless Communications Letters, 8(1):69-72, Feb. 2019. doi: 10.1109/LWC.2018.2855120.

[55] Zhang, Z., Li, B. (2008). Neighbor discovery in mobile ad hoc selfconfiguring networks with directional antennas: Algorithms and comparisons. IEEE Trans. Wireless Commun., 7(5):1540-1549, May 2008.

[56] Vasudevan, S., Kurose, J., Towsley, D. (2005). On neighbor discovery in wireless networks with directional antennas. In Proc. IEEE Int. Conf. Comput. Commun., Miami, FL, USA, Mar. 2005, pp. 2502-2512.

[57] Ji, D., Wei, Z., Chen, X., Han, C., Chen, Q., Feng, Z., Ning, F. (2019) Radar-Communication Integrated Neighbor Discovery for Wireless Ad Hoc Networks. 11th International Conference on Wireless Communications and Signal Processing (WCSP), Xi'an, China, pp. 1-5. doi: 10.1109/WCSP.2019.8927896.

[58] Chadha, R., Sistla, A.P., Viswanathan, M. (2017) Verification of randomized security protocols. 2017 32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), 2017, pp. 1-12. doi: 10.1109/LICS.2017.8005126.

[59] Qiu, J., Tian, Z., Du, C., Zuo, Q., Su, S., Fang, B. (2020) A Survey on Access Control in the Age of Internet of Things. In IEEE Internet of Things Journal, 7(6):4682-4696, June 2020. doi: 10.1109/JIOT.2020.2969326.

[60] Feeney, L.M., Ahlgren, B., Westerlund, A. (2001) Spontaneous networking: an application oriented approach to ad hoc networking. IEEE Communications Magazine, vol. 39, no. 6, pp. 176–181, 2001.

[61] Mani, M., Nguyen, A.M., Crespi, N. (2010) SCOPE: a prototype for spontaneous P2P social networking. In 2010 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), pp. 220–225, Mannheim, Germany, 2010.

[62] Legendre, F., de Amorim, M.D., Fdida, S. (2004) Implicit merging of overlapping spontaneous networks. In IEEE 60th Vehicular Technology Conference, 2004. VTC2004-Fall, pp. 3050–3054, Los Angeles, CA, USA, 2004.

[63] IBM, "A Smarter Planet," 2012, http://www.ibm.com/smarterplanet.

[64] Alcaraz, C., Najera, P., Lopez, J., Roman, R. (2010) Wireless sensor networks and the Internet of Things: do we need a complete integration?. In 1st International workshop on the security of The internet of Things (SecIoT'10), Tokyo, Japan, 2010.

[65] Jadhav, D.S., Rokade, D.A. (2014) A survey on security based spontaneous wireless ad hoc networks for communication based elliptical curve

cryptography. International Journal on Recent and Innovation Trends in Computing and Communication, vol. 2, no. 11, pp. 3552–3555, 2014.

[66] Lacuesta, R., Palacios-Navarro, G., Cetina, C., Peñalver, L., Lloret, J. (2012) Internet of things: where to be is to trust. EURASIP Journal on Wireless Communications and Networking, vol. 2012, no. 1, 16 pages, 2012.

[67] Lacuesta, R., Lloret, J., Sendra, S., Peñalver, L. (2014) Spontaneous ad hoc mobile cloud computing network. The Scientific World Journal, vol. 2014, 19 pages, 2014.

[68] Lacuesta, R., Lloret, J., Garcia, M., Peñalver, L. (2013) A secure protocol for spontaneous wireless ad hoc networks creation. IEEE Transactions on Parallel and Distributed Systems, vol. 24, no. 4, pp. 629–641. April 2013. doi: 10.1109/TPDS.2012.168.

[69] Jadhav, S., Naik, P., Kahade, K. (2016) A survey on security based on user trust in spontaneous wireless ad hoc network creation. International Journal for Innovative Research in Science and Technology, vol. 2, no. 10, pp. 84–88,, 2016.

[70] Nandagawli, P.M., Tayal, A.R., Jaiswal, A. (2014) A survey on symmetric key protocol for spontaneous wireless ad hoc network creation. International Journal of Scientific and Technology Research, vol. 3, pp. 89–91, 2014.

[71] Reddy, N.S., Ponsam, J.G. (2014) Security based on user trust in spontaneous wireless ad hoc network creation. International Journal of Engineering Development and Research, vol. 2, no. 2, pp. 1473–1480, 2014.

[72] Satyannarayana, N.M.V., Veerababu, M.R. (2016) A secure protocol for spontaneous ad-hoc networks. International Journal of Computer Science and Information Technologies (IJCSIT), vol. 7, no. 6, pp. 2502–2506, 2016.

[73] Nimisha, P., Sindhu, M.P. (2016) An enhanced secure protocol for spontaneous wireless ad-hoc networks. In IOSR Journal of Computer Engineering (IOSR-JCE). International Conference on Emerging Trends in Engineering Management (ICETEM-2016), pp. 5–11, 2016.

[74] Kallada, J.B. (2015) A protocol for creating spontaneous ad hoc wireless network for secure communication. International Journal of Mechanical Engineering and Information Technology, vol. 3, no. 3, pp. 1061–1066, 2015.

[75] Shinde, K.V., Kaur, H., Patil, P. (2015) Enhance security for spontaneous wireless ad hoc network creation. In 2015 International Conference on Computing Communication Control and Automation, pp. 247–250, Pune, India, 2015.

[76] Rewadkar, D.N., Karve, S.B. (2014) Energy efficient self configured secure protocol (EESCSP) for wireless spontaneous adhoc network. In 2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), pp. 792–799, Kanyakumari, India, 2014.

[77] Firoj, R., Antil, A. (2016) Jain Enhanced security protocol for spontaneous wireless ad-hoc network. International Journal Of Engineering And Computer Science, vol. 5, no. 12, pp. 19419–19428, 2016.

[78] Varghane, N., Kurade, B. (2014) Secure protocol and signature based intrusion detection for spontaneous wireless ad hoc network. International Journal of Computer Science and Mobile Computing (IJCSMC), vol. 3, no. 5, pp. 758–768, 2014.

[79] Varghane, N., Kurade, B., Pote, C. (2014) Intrusion detection, secure protocol and network creation for spontaneous wireless ad hoc network. International Journal of Computer Science and Mobile Computing, vol. 3, no. 2, pp. 389–394, 2014.

[80] Srinivas, K., NarasimhaRao, G.B., Reddy, S.S. (2014) A self-configured secure protocol for the management of wireless ad hoc networks. International Journal of Computer Science and Information Technologies, vol. 5, no. 4, pp. 5529–5532, 2014.

[81] Pradeep, G., Shobak, A.P. (2014) Improving QoS in spontaneous ad hoc neworks. International Journal of Computer Science and Information Technologies, vol. 5, no. 4, pp. 5705–5707, 2014.

[82] Garcia, M., Bri, D., Boronat, F., Lloret, J. (2008). A new neighbour selection strategy for group-based wireless sensor networks. Fourth international conference on networking and services (ICNS 2008), Guadeloupe: Gosier, pp. 109–114, 2008. doi: 10.1109/ICNS.2008.18.

[83] AlFarraj, O., AlZubi, A., Tolba, A. (2018) Trust-based neighbor selection using activation function for secure routing in wireless sensor networks. J Ambient Intell Human Comput (2018). doi: 10.1007/s12652-018-0885-1.

[84] Mishra, S., Kaur, P. (2014) Energy efficient neighbor selection for flat wireless sensor networks. In WiMON 2014 conference proceedings, 1-7, June 2014. doi: 10.5121/csit.2014.4511.

[85] Azad, P., Sharma, V. (2013) Cluster head selection in wireless sensor networks under fuzzy environment. ISRN Sensor Networks, Volume 2013, Article ID 909086, 1-8. : 10.1155/2013/909086.

[86] Thein, M.C.M., Thein, T. (2010) An Energy Efficient ClusterHead Selection for Wireless Sensor Networks. 2010 International Conference on Intelligent Systems, Modelling and Simulation, 2010, pp. 287-291. doi: 10.1109/ISMS.2010.60.

[87] Crosby, G.V., Pissinou, N., Gadze, J. (2006) A framework for trustbased cluster head election in wireless sensor networks. Second IEEE Workshop on Dependability and Security in Sensor Networks and Systems, 2006, pp. 10-22. doi: 10.1109/DSSNS.2006.1.

[88] Boulis, A. (2011). Castalia - A simulator for wireless sensor networks and body area networks. Version 3.2 . User's Manual. https://es.scribd.com/document/78901825/castalia-user-manual. unpublished.

[89] Garcia, M., Martinez, C., Tomas, J., Lloret, J. (2007). Wireless Sensors self-location in an Indoor WLAN environment. In International Conference on Sensor Technologies and Applications SENSORCOMM 2007 (pp. 14–20). Spain: Valencia.

[90] Lloret, J., López, J. J., Turró, C., Flores, S. (2004). A fast design model for indoor radio coverage in the 2.4 GHz wireless LAN. In 1st International Symposium on Wireless Communication Systems (pp. 408-412).

[91] Garcia, M., Tomas, J., Boronat, F., Lloret, J. (2009) The Development of Two Systems for Indoor Wireless Sensors Self-location. Ad Hoc Sens. Wirel. Networks, 8 (3-4), 235-258, 2009.

[92] Sorribes JV, Peñalver L, Tavares Calafate C, Lloret J (2020) Collision-aware deterministic neighbor discovery in static ad hoc wireless networks. In Global conference on wireless and optical technologies (GCWOT'20), Malaga, Spain, Oct 2020 (1-8). Online.