ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA
DE TELECOMUNICACIÓN

INGENIERÍA DE TELECOMUNICACIÓN - ITIS

PROYECTO FIN DE CARRERA

Estudio y modelado del retardo extremo a extremo en redes ad hoc inalámbricas de gran escala

Autora: MIHAELA IOANA CHIDEAN

Tutora: LORENA FERNÁNDEZ MARTÍNEZ
Co-tutor: EDUARDO MORGADO REYES

Curso académico: 2010/2011

Título:	*Estudio y modelado del retardo extremo a extremo en redes ad hoc inalámbricas de gran escala*
Autora:	*Mihaela Ioana Chidean*
Tutora:	*Lorena Fernández Martínez*
Co-tutor:	*Eduardo Morgado Reyes*

La defensa del presente Proyecto Fin de Carrera se realizó el día

siendo calificada por el siguiente tribunal:

Presidente:

Secretario:

Vocal:

Habiendo obtenido la siguiente calificación:

Calificación:

Presidente	**Secretario**	**Vocal**

«Drumul e lung, rădăcinile sunt
amare, dar fructele sunt dulci»
(«El camino es largo, las raíces
son amargas, pero los frutos son
dulces»)
Proverbio rumano.

Resumen

Las redes ad hoc inalámbricas, por definición, proporcionan portabilidad y movilidad a todos los nodos de la misma, siendo estas las características más requeridas por parte de los usuarios. Como valor añadido, todos los nodos de una red ad hoc inalámbrica tienen el mismo nivel en la jerarquía, siendo independientes de cualquier infraestructura. Este tipo de redes constituyen un reciente y muy diverso campo de investigación. Por ejemplo, se atiende a temas como el acceso al medio, el encaminamiento, las interferencias entre diferentes nodos, el retardo que se produce en una transmisión o la máxima movilidad que pueden tener los nodos, entre otros.

El objetivo principal de este Proyecto Fin de Carrera es estudiar y modelar el retardo extremo a extremo en las redes ad hoc inalámbricas en función de la distancia existente entre los nodos emisor y receptor. La consecución de esta meta requiere un paso previo, que es modelar el retardo extremo a extremo en una red ad hoc inalámbrica en función del número de saltos de la ruta que siguen los datos. En este estudio se tienen en cuenta tanto los protocolos de nivel MAC (*Medium Access Control*) como de encaminamiento utilizados. El modelo analítico es verificado mediante simulaciones por ordenador para redes ad hoc inalámbricas de gran escala, utilizando la herramienta de simulación `ns-2` (*Network Simulator 2*) y los protocolos de encaminamiento AODV (*Ad hoc On-demand Distance Vector*) y DSR (*Dynamic Source Routing*). El fin de utilizar dos protocolos de encaminamiento es, además de verificar las expresiones teóricas en distintos casos, realizar una comparativa entre ellos y concluir cual de ellos ofrece mejores prestaciones en cuanto al retardo extremo a extremo. Los resultados obtenidos demuestran que el protocolo de encaminamiento AODV es el que ofrece menor retardo extremo a extremo en las redes ad hoc inalámbricas de gran escala.

Agradecimientos

Este trabajo pone punto y final a una bonita etapa de mi vida y abre las puertas a otra fase, esperemos que más bonita e interesante. Me gustaría expresar mi más sincera gratitud a todas las personas que han intervenido en el desarrollo de mi proyecto de vida, que tiene como hito este proyecto.

A Lorena por la confianza que has depositado en mi, no solo para este trabajo sino desde la primera vez que nuestros caminos se cruzaron en la Universidad. Por tu paciencia y perseverancia, gracias a ti ya se poner la mayoría de las tildes.

A Eduardo por proporcionar un punto de partida y ser la guía a lo largo del trabajo. Por tu sentido crítico con los resultados; me ha enseñado que siempre hay que analizar desde varios puntos de vista.

A Antonio por ver siempre el siguiente paso en el camino.

A todos los compañeros de Departamento, por considerarme una más, desde el primer momento.

A mis compañeros de carrera, mis compañeros de sufrimiento. Por las reuniones de pasillo entre clase y clase. Por esas comidas interminables, nuestro pequeño momento de relax. Un especial gracias a Estefanía y Abraham.

A mis padres Aurel y Veronica, por dejarme elegir mi camino y apoyarme con todo vuestro entusiasmo en llegar a la meta. Mamá gracias por mostrarme que el estudio es una carrera de fondo y no un *sprint* de última hora. Por enseñarme que la escuela es mi trabajo. Papá gracias por la tranquilidad que transmites.

A mi hermano Mihai, por despertar mi interés por este mundo, por enseñarme que *Teleco* no es solo teoría y por ayudarme desde la distancia. *Merci frăţior!*

A Dora, que me ha mostrado que siempre hay que mirar hacia adelante; que por muy grandes que son los obstáculos, siempre se pueden superar.

A mis abuelas y abuelos por enseñarme que el mundo no es solo ciudad y asfalto, por cultivar mi interés por la naturaleza, el campo, la libertad. Queridas abuelas espero volver a veros muy pronto. Queridos abuelos os hecho muchos de menos.

A los familiares y amigos de Rumanía, gracias por vuestro apoyo, por estar siempre allí. Mara, la pequeña *Duracell*, gracias por la alegría de vivir que nos transmites a todos. Achi, quién iba a decir que nos graduamos el mismo año? *Nenea Ghiţă*, gracias por recibirme siempre con los brazos abiertos y con *mici* preparados.

A José, *mi boby*, por enseñarme que el amor realmente existe. Por el apoyo incondicional y tus inacabables palabras de ánimo.

Índice general

Índice de Figuras

Índice de Tablas

Lista de acrónimos

4G	Cuarta **G**eneración
ABR	**A**ssociativity-**B**ased **R**outing
ACK	**ACK**nowledgment
AODV	**A**d hoc **O**n-demand **D**istance **V**ector
bit	**B**inary dig**it**
CBR	**C**onstant **B**it **R**ate
CSMA/CA	**C**arrier **S**ense **M**ultiple **A**ccess with **C**ollision **A**voidance
DCF	**D**istributed **C**oordination **F**unction
DSDV	**D**estination-**S**equenced **D**istance **V**ector
DSR	**D**ynamic **S**ource **R**outing
IEEE	**I**nstitute of **E**lectrical and **E**lectronic **E**ngineers
IP	**I**nternet **P**rotocol
LLC	**L**ogical **L**ink **C**ontrol
MAC	**M**edium **A**ccess **C**ontrol
NASA	**N**acional **A**eronautics and **S**pace **A**dministration
`ns-2`	**N**etwork **S**imulator **2**
NSA	**N**ational **S**ecurity **A**gency
OSI	**O**pen **S**ystem **I**nterconnection
OSLR	**O**ptimized **L**ink **S**tate **R**outing
OSPF-OR	**OSPF-O**verlaping **R**elay

OSPF	**O**pen **S**hortest **P**ath **F**irst
OTcl	**O**bject oriented **T**ool **C**ommand **L**anguage
PDA	**P**ersonal **D**igital **A**ssistant
pgf	**P**robability **G**enerating **F**unction
PRNET	**P**acket **R**adio **NET**work
RAM	**R**andom **A**ccess **M**emory
regexp	**Reg**ular **Exp**ression
RERR	**R**oute **ERR**or
RFC	**R**equest **F**or **C**omments
RIP	**R**outing **I**nformation **P**rotocol
RREP-ACK	**RREP-ACK**nowledgment
RREP	**R**oute **REP**ly
RREQ	**R**oute **REQ**uest
TCP	**T**ransmission **C**ontrol **P**rotocol
TTL	**T**ime **T**o **L**ive
UDP	**U**ser **D**atagram **P**rotocol
ZRP	**Z**one **R**outing **P**rotocol

Capítulo 1

Introducción y objetivos

El objetivo de este primer capítulo es presentar brevemente el trabajo realizado en este Proyecto Fin de Carrera. Para empezar, en la Sección 1.1, se expone la motivación para la realización del mismo. A continuación, en la Sección 1.2, se enumeran los objetivos perseguidos con el proyecto y en la Sección 1.3 se muestra la organización del trabajo. Por último, en la Sección 1.4, se exponen los contenidos de los capítulos de esta memoria.

1.1. Motivación

La realización de este trabajo tiene tanto una motivación personal como una motivación profesional, por lo que se hace necesario mencionar ambas.

En primer lugar, la motivación personal se debe al interés por la particular naturaleza de las redes ad hoc inalámbricas y sus posibles escenarios de utilización. Lo más característico de las redes ad hoc es que son autónomas y sin infraestructura, haciendo posible su utilización en zonas que no disponen de infraestructura previa de comunicaciones y supone un coste económico alto o no es viable desplegarlas [1]. Uno de los escenarios típicos de uso de las redes ad hoc inalámbricas son las zonas devastadas por desastres naturales (terremotos, inundaciones) o por diversos accidentes provocados por el hombre (accidentes nucleares, ataques terroristas). En estos casos, es muy posible que las redes de comunicaciones existentes se vean afectadas y la rapidez del despliegue de las redes ad hoc es fundamental, ya que permite restablecer las comunicaciones de forma casi instantánea [2]. Otro escenario en el que la utilización de las redes ad hoc inalámbricas es fundamental es en el campo de batalla, ya que permite la comunicación entre los distintos nodos (soldados, vehículos o centros de mando) sin depender de ningún tipo de elemento fijo, aprovechando además la robustez frente al punto único de fallo [2]. Además, las redes ad hoc inalámbricas tienen cabida en otro tipo de escenarios más optimistas, que no involucran ni situaciones de emergencia ni guerras, como son las reuniones profesionales

(congresos o convenciones), las aplicaciones para el hogar, las comunicaciones intervehiculares o las redes de sensores inalámbricos. En primer lugar, en las reuniones profesionales se puede aprovechar el uso de nodos que actúan como pasarelas a redes externas (como Internet) para dar conectividad a todos los participantes. En segundo lugar, las redes ad hoc son adecuadas para las diferentes aplicaciones para el hogar, ya que permiten realizar conexiones entre todos los equipos electrónicos. En tercer lugar, las comunicaciones intervehiculares permiten dar cierta inteligencia a la red de carreteras para así conocer el estado actualizado de las mismas. En cuarto y último lugar, las redes de sensores inalámbricos permiten medir y controlar diversos parámetros en un área geográfica, como por ejemplo la humedad presente en un campo de cultivo, para realizar el riego en el momento preciso, o la temperatura en una zona forestal, para detectar incendios. En todos los escenarios presentados interviene, en mayor o menor medida, la movilidad de los nodos, siendo esta otra de las principales características de las redes ad hoc inalámbricas [1].

Por otra parte, la motivación profesional reside en realizar un avance en una de las líneas de investigación del Departamento de Teoría de la Señal y Comunicaciones de la Universidad Rey Juan Carlos llamada «Teoría del encaminamiento en redes ad hoc inalámbricas». El trabajo dentro del Departamento se centra en el estudio de las redes ad hoc en general y, mediante este trabajo se pretende dar una solución al modelado del retardo extremo a extremo en las ya mencionadas redes. De esta forma, con los resultados obtenidos, antes de desplegar un sistema real se pueden conocer o aproximar distintos parámetros de rendimiento del sistema.

1.2. Objetivos

Como acabamos de comentar, este proyecto se enmarca dentro de una de las principales líneas de investigación del Departamento de Teoría de la Señal y Comunicaciones de la Universidad Rey Juan Carlos.

El principal objetivo de este trabajo es estudiar y modelar el retardo extremo a extremo en redes ad hoc inalámbricas de gran escala desde una perspectiva multicapa (*cross-layer*) que nos permita obtener la variación del mismo dependiendo de los protocolos de nivel de enlace o de red utilizados, comprobando como varía en función de la distancia que separa los nodos. Este propósito fundamental se puede dividir a su vez en varios objetivos parciales enumerados a continuación:

- Estudio teórico de las redes ad hoc inalámbricas, poniendo especial interés en los protocolos de control de acceso al medio y en los protocolos de encaminamiento.
- Revisión del estado del arte del estudio del retardo en redes ad hoc inalámbricas.

- Modelado analítico del retardo en redes ad hoc inalámbricos.
- Familiarización con el *software* utilizado para el desarrollo del proyecto, como por ejemplo, la herramienta de simulación o los lenguajes de programación utilizados.
- Validación del modelo teórico propuesto.
- Extracción de conclusiones.
- Redacción de la memoria.

1.3. Metodología y planificación

En cuanto a la organización del trabajo realizado, este se divide en varias fases y, para cada una de ellas, se ha realizado una planificación temporal. En la Figura 1.1 se representan las distintas fases necesarias, realizando una ampliación en la tercera de ellas para indicar las subtareas que la componen. Durante la fase de documentación se lleva a cabo un extenso estudio bibliográfico de las redes ad hoc inalámbricas, centrándonos en el análisis del retardo y en los protocolos de control de acceso al medio y de encaminamiento de las mencionadas redes. Seguidamente, se propone el modelado analítico del retardo teniendo en cuenta las diferentes formas de obtener la exclusividad del canal para transmitir los datos. Posteriormente, durante la fase de validación del modelo analítico se adquiere experiencia con la herramienta de simulación y se repasan varios lenguajes de programación eligiendo el más adecuado para llevar a cabo las sucesivas tareas. A continuación, se preparan, automatizan y lanzan suficientes simulaciones para poder validar el modelo teórico propuesto. Una vez obtenidos los resultados necesarios, estos se analizan y, finalmente, se concluye si la formulación teórica del retardo se adecúa a la realidad simulada. La última fase del proyecto es la redacción de la presente memoria.

La planificación temporal seguida para llevar a cabo las diferentes fases del proyecto se muestra en la Figura 1.2 mediante un diagrama de Gantt donde la duración de las tareas se representa en el eje de abscisas. Varias de las tareas se solapan en el tiempo, ya que para llevarlas a cabo no fue necesario seguir un flujo secuencial.

1.4. Organización de la memoria

La presente memoria se estructura en seis capítulos, aparte del actual, cuyos contenidos se enumeran a continuación:

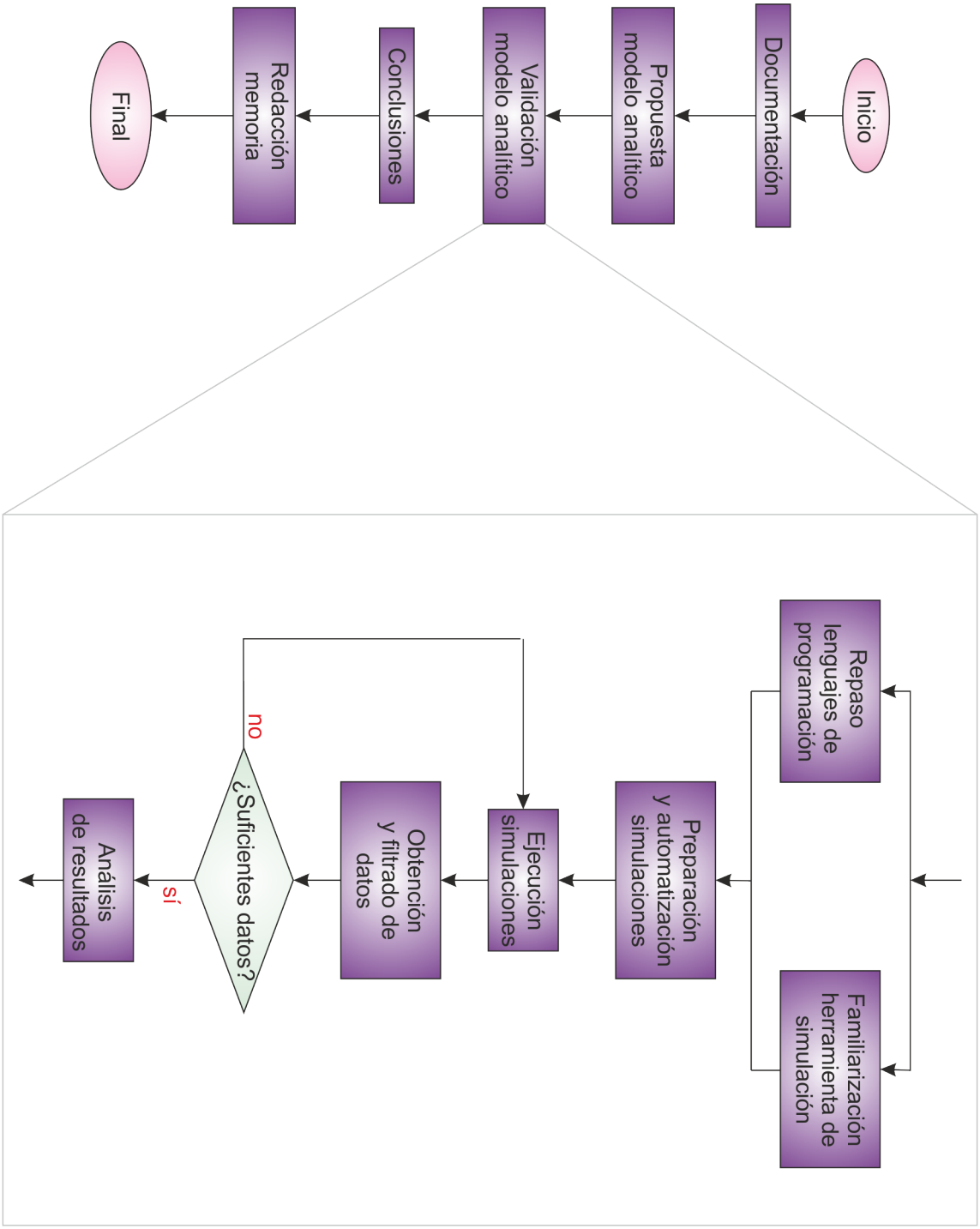

Figura 1.1: Diagrama de flujo seguido para analizar y modelar el retardo en redes ad hoc inalámbricas.

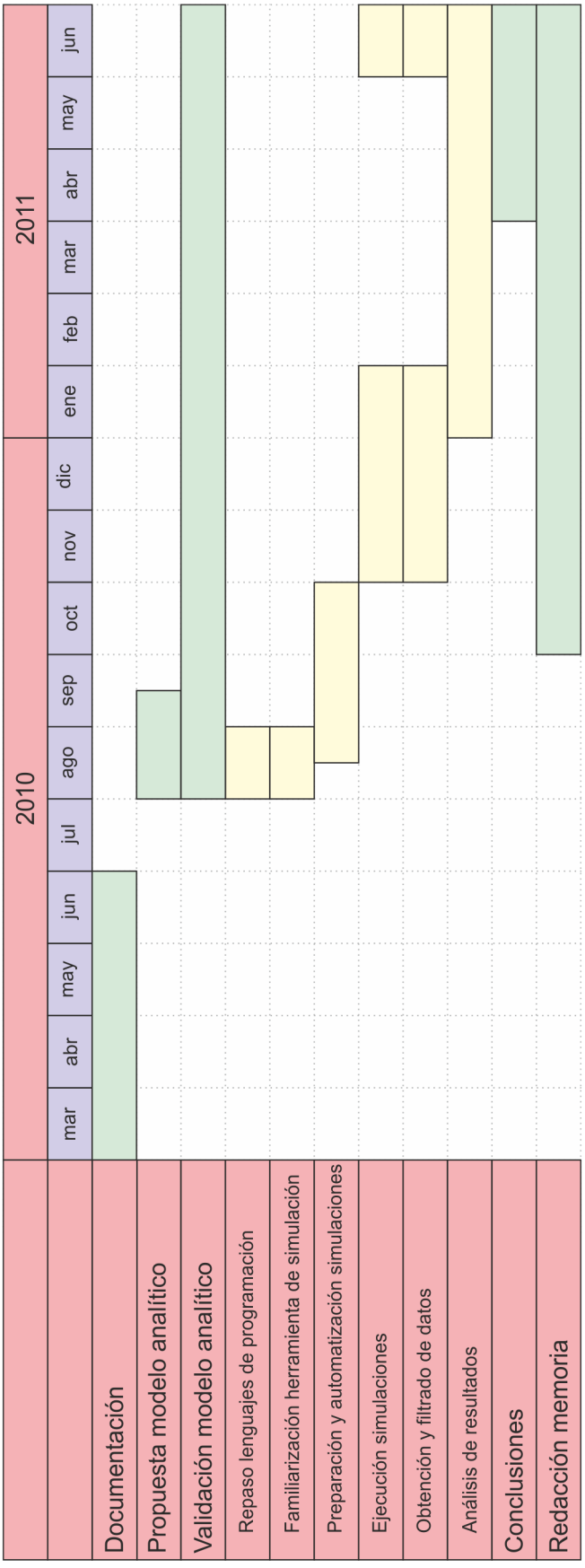

Figura 1.2: Planificación.

- **Capítulo 2: Redes ad hoc inalámbricas**. Se muestra una visión general sobre las redes ad hoc inalámbricas, poniendo más énfasis en las capas de los sistemas de comunicaciones que tienen más peso en el análisis propuesto.

- **Capítulo 3: Protocolos de encaminamiento AODV y DSR**. Se ofrece un análisis detallado sobre los protocolos de encaminamiento elegidos para realizar este estudio.

- **Capítulo 4: Materiales**. Se describen los recursos empleados para la realización de este Proyecto Fin de Carrera, tanto *hardware* como *software*. Adicionalmente, se señala un error en la herramienta de simulación de red utilizada, que se ha detectado durante el desarrollo de este proyecto.

- **Capítulo 5: Estudio teórico del retardo extremo a extremo y descripción de los experimentos**. Se presenta el estado del arte del estudio del retardo en redes ad hoc y, seguidamente, se detalla el modelo de red utilizado en el estudio. A continuación, se desarrolla la formulación matemática necesaria para evaluar el retardo en redes ad hoc y se describen las simulaciones realizadas para la obtención de los resultados.

- **Capítulo 6: Validación del modelo teórico**. Se analizan los datos obtenidos a partir de las simulaciones y se comparan con las expresiones teóricas propuestas.

- **Capítulo 7: Conclusiones y líneas de trabajo futuras**. Se exponen las conclusiones derivadas de los resultados obtenidos y se proponen varias líneas que dan continuidad a este estudio.

Capítulo 2

Redes ad hoc inalámbricas

El propósito de este capítulo es mostrar una visión general sobre las redes ad hoc inalámbricas, centrándonos en las capas de los sistemas de comunicaciones que intervienen posteriormente en el análisis del retardo extremo a extremo propuesto en este Proyecto Fin de Carrera. En primer lugar, se analizará el nivel de enlace de datos (Sección 2.2.1), que se encarga de controlar el acceso ordenado al medio y que en redes inalámbricas tiene que tener en cuenta las particularidades del medio de transmisión. En segundo y último lugar, se analiza el nivel de red (Sección 2.2.2), que se encarga de establecer comunicación entre nodos distantes mediante el encaminamiento.

2.1. Introducción

A principios de los años 90 la comisión IEEE 802.11 (*Institute of Electrical and Electronic Engineers*) aceptó por primera vez el término «ad hoc» para describir un tipo especial de redes: aquellas que se forman en cualquier lugar, en cualquier momento, para prácticamente cualquier aplicación y no utilizan ningún tipo de infraestructura previa [2–4]. Sin embargo, el primer proyecto de red ad hoc está documentado en 1972, cuando el Departamento de Defensa de los Estados Unidos de América encarga el proyecto PRNET (*Packet Radio NETwork*) con el objetivo de proporcionar redes de intercambio de paquetes para elementos móviles en entornos hostiles y sin infraestructura, como son los campos de batalla. Gracias a este proyecto y a otros que le siguieron, se consiguió demostrar la viabilidad y la eficiencia de este tipo de redes tanto en aplicaciones militares como civiles.

Las particulares características de las redes ad hoc hacen de ellas un campo importante en el mundo de la investigación en los últimos años [5–9]. En un futuro próximo se espera que incluso redes de uso cotidiano sean de tipo ad hoc. Este es el caso de la cuarta

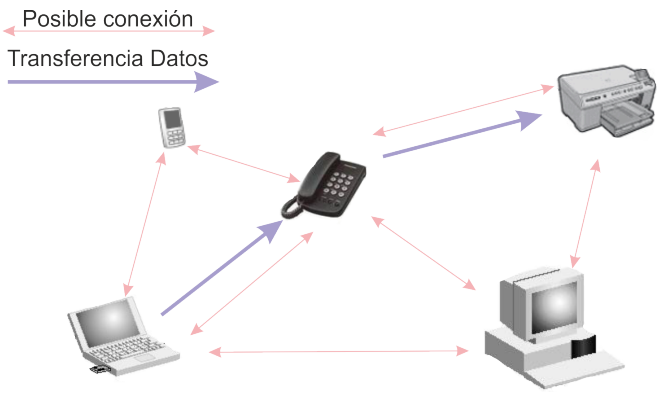

Figura 2.1: Ejemplo de red ad hoc.

generación de telefonía móvil (4G) que tendrá una estructura híbrida entre red ad hoc y celular [1, 10, 11].

Una red ad hoc inalámbrica está formada por nodos que se conectan entre sí mediante enlaces inalámbricos, no existe la figura de estación central o dominante y en cualquier momento un determinado nodo puede actuar de emisor de información, encaminador y receptor de dicha información. Además, se auto-organizan de forma dinámica, ya que la topología de la red puede cambiar de forma arbitraria a lo largo del tiempo [1, 3, 12]. En contraposición están las redes con infraestructura. Un ejemplo de redes con infraestructura son las redes celulares, donde existe una estación base que se encarga de coordinar las comunicaciones entre los nodos de menor nivel en la jerarquía de la red [13, 14].

En la Figura 2.1 se observa una red ad hoc en la que los nodos son aparatos de la vida cotidiana: un ordenador portátil, una PDA (*Personal Digital Assistant*), una impresora, un teléfono fijo y un ordenador de sobremesa. En dicha figura está ilustrada una hipotética situación donde el ordenador portátil necesita enviar un fichero a la impresora, pero al no tener comunicación directa con la misma utiliza un enlace a través del teléfono fijo.

2.2. Niveles OSI

El modelo de Interconexión de Sistemas Abiertos (OSI, *Open System Interconnection*) [15] es el marco de referencia seguido para la arquitectura de los sistemas de telecomunicación. El modelo define la «torre OSI» como un conjunto jerárquico de 7 capas o niveles, de modo que una determinada capa recoge datos de la capa antecesora, procesa dichos datos y se los entrega a la capa sucesora sin tener ningún conocimiento sobre las funciones de los demás niveles. El principal objetivo del modelo OSI es la definición de interfaces entre los distintos niveles, de forma que para cada nivel se puedan definir protocolos independientes del resto de la torre. Los siete niveles en orden ascendente de la torre OSI son:

1. Capa física: se encarga de las conexiones físicas entre el nodo y la red. Por tratarse de la salida del nodo hacia el canal, existen protocolos específicos para cada tipo de medio (por ejemplo cable coaxial, fibra óptica, aire, etc.) ofreciendo una interfaz unificada hacia la siguiente capa.

2. Capa de enlace de datos: se ocupa de la conexión y transmisión de datos fiable entre dos nodos conectados directamente (conectados al mismo cable en una red cableada o compartiendo área de cobertura en una red inalámbrica).

3. Capa de red: realiza la transmisión de datos entre nodos que no están conectados directamente al mismo medio físico.

4. Capa de transporte: conexión extremo a extremo, es decir, entre los nodos emisor y receptor sin intervenir ningún nodo intermedio. Hace de frontera entre las capas inferiores (ligadas a la red utilizada) y las capas superiores (ligadas a la aplicación). Prepara los datos para ser transmitidos por la red y recibe datos de la red para ser entregados a cada aplicación en particular.

5. Capa de sesión: proporciona la estructura de control para las comunicaciones entre las diferentes aplicaciones; establece, mantiene y termina comunicaciones (sesiones) entre aplicaciones.

6. Capa de presentación: representación homogénea de los datos de forma que, aunque distintos nodos tengan diferente representación interna, la siguiente capa siempre recibe la misma representación.

7. Capa de aplicación: proporciona a las aplicaciones una interfaz común para poder acceder a la red de transmisión.

En este Proyecto Fin de Carrera se estudia el retardo extremo a extremo desde punto de vista multicapa o a través de capas. Este concepto, también conocido como *cross-layer*, se utiliza para indicar que se incumplen las estrictas interconexiones entre los distintos niveles de la torre OSI. Al ser este trabajo una primera aproximación al estudio del retardo en el Departamento de Teoría de la Señal y Comunicaciones de la Universidad Rey Juan Carlos, únicamente se consideran los niveles de enlace de datos y de red, el resto de ellos considerándose ideales. El nivel de enlace de datos es necesario ya que influye en el orden de transmisión de los distintos paquetes de datos y el nivel de red se encarga de dar conectividad a dos nodos alejados.

2.2.1. Nivel de enlace de datos

Según el modelo OSI, el nivel de enlace de datos proporciona funciones y procedimientos para el establecimiento, el mantenimiento y el cierre de conexiones de datos entre

distintos nodos de una red. Además detecta y, opcionalmente, corrige errores producidos en el nivel inferior (el nivel físico). El nivel de enlace se divide en dos subcapas: Control de Enlace Lógico (LLC, *Logical Link Control*) y Control del Acceso al Medio (MAC, *Medium Access Control*). La subcapa LLC asume las funciones de detectar y corregir errores de transmisión mientras que la subcapa MAC determina si un nodo puede transmitir por el medio físico, es decir, proporciona un mecanismo de acceso múltiple al canal. En este trabajo, la subcapa MAC es la que presenta mayor interés, ya que influye directamente tanto en el retardo de las transmisiones como en las interferencias que produce y que recibe una transmisión de datos en una red ad hoc inalámbrica [16].

La regla básica de los protocolos MAC sobre canales inalámbricos es que si un nodo detecta una transmisión en curso no intenta transmitir a su vez datos para así evitar la colisión de los paquetes. Como consecuencia, en las redes inalámbricas se dan los problemas de nodo oculto y nodo expuesto [1, 16–18]. Como apoyo a la explicación se utiliza la Figura 2.2 que representa de forma esquemática cuatro nodos: S (emisor), R (receptor), H (nodo oculto) y E (nodo expuesto). Se representa igualmente el área de cobertura del nodo S como la circunferencia con frontera de color negro y la zona de cobertura del nodo R con la circunferencia de color naranja. Por lo tanto, los dos problemas de las redes inalámbricas ya mencionados son:

- Nodo oculto: este problema aparece cuando dos nodos no pueden detectar sus sendas transmisiones. Supongamos que S está transmitiendo un paquete hacia R. Si en el intervalo temporal que dura la transmisión H quiere enviar un paquete hacia R o hacia cualquier otro nodo de su propia área de cobertura, lo primero que hace es sensar el canal. Al no detectar la comunicación en curso, comienza el envío del paquete. En este caso los dos paquetes transmitidos colisionan y se pierden porque coinciden en un mismo intervalo de tiempo en la misma zona espacial. Este problema reduce la efectividad del sistema completo ya que es necesario realizar retransmisiones de los paquetes que han sufrido colisiones. El nodo oculto puede estar situado en cualquier punto del área de color verde de la Figura 2.2, ya que se cumple que está dentro del área de cobertura del nodo R y no está en el área de cobertura de S.

- Nodo expuesto: este problema se produce cuando un nodo detecta una transmisión y retrasa su propio envío sin necesidad. Supongamos nuevamente que S está transmitiendo un paquete hacia R. Si en el mismo intervalo de tiempo E quiere enviar un paquete a cualquier nodo fuera del área de cobertura de S, sensa el canal y, al detectar la transmisión en curso, pospone su propio envío. Sin embargo, esto no sería necesario ya que la interferencia entre ambos paquetes sería suficientemente pequeña como para no perder ninguno de ellos. Este problema, al igual que en el caso anterior, reduce la efectividad del sistema y además incrementa el retardo medio

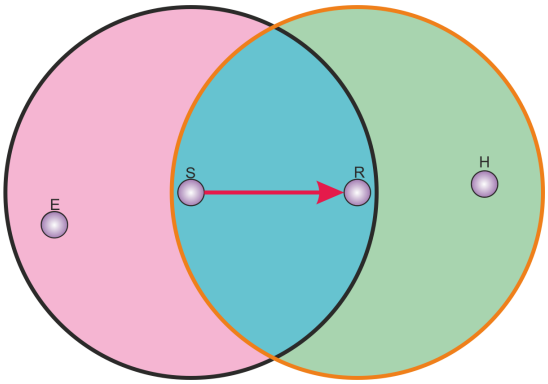

Figura 2.2: Representación esquemática de los problemas del nodo oculto y nodo expuesto incluyendo los nodos emisor (S), receptor (R), oculto (H) y expuesto (E).

de los paquetes ya que se producen esperas innecesarias. Y, además, se considera nodo expuesto a cualquier nodo situado en el área de color rosa de la Figura 2.2.

En función de la solución propuesta para estos problemas, los protocolos MAC se clasifican de la siguiente manera [16]:

- Clase 1: se prohíben las transmisiones simultáneas dentro del radio de cobertura del emisor (circunferencia con frontera de color negro en la Figura 2.2). Los protocolos pertenecientes a esta clase no resuelven ni el problema del nodo oculto ni el del nodo expuesto. El ejemplo típico de protocolo MAC perteneciente a esta clase es CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*) sin reserva [19].
- Clase 2: se prohíben las transmisiones simultáneas dentro del radio de cobertura del emisor (circunferencia con frontera de color negro en la Figura 2.2) y del receptor (circunferencia con frontera naranja en la Figura 2.2). Esta clase resuelve el problema del nodo oculto pero no el del nodo expuesto. A esta clase pertenece por ejemplo CSMA/CA con reserva [18, 19].
- Clase 3: se prohíben las transmisiones simultáneas dentro del radio de cobertura del receptor (circunferencia con frontera naranja en la Figura 2.2) y hacia nodos que se encuentran dentro del radio de cobertura del emisor (circunferencia con frontera de color negro en la misma Figura 2.2). Esta clase resuelva ambos problemas pero para ello necesita señalización adicional.

Con el fin de mantener una nomenclatura homogénea, se define la ausencia de protocolo MAC como Clase 0 [5] que es el peor escenario posible en cuanto acceso múltiple a la red, ya que no existe ningún control sobre las transmisiones simultáneas. A modo de conclusión, en la Tabla 2.1, se exponen qué acciones son permitidas para cada una de las distintas clases MAC para nodos situados en diferentes áreas afectadas por una determinada comunicación (nodos S y R en la Figura 2.2).

	Nodos dentro del radio de S pero fuera del radio de R (nodo situado en el área de color rosa en la Figura 2.2)		Nodos dentro del radio de R pero fuera del radio de S (nodo situado en el área de color verde en la Figura 2.2)		Nodos dentro de la zona común de cobertura de R y S (nodo situado en el área de color azul en la Figura 2.2)	
	Transmitir	Recibir	Transmitir	Recibir	Transmitir	Recibir
Clase 0	Sí	Sí	Sí	Sí	Sí	Sí
Clase 1	No (1)	Sí (2)	Sí (3)	Sí (2)	No	Sí (2)
Clase 2	No (1)	No (4)	No	No	No	No
Clase 3	Sí (5)	No	No	Sí (6)	No	No

Tabla 2.1: Transmisiones/recepciones prohibidas o admitidas para las diferentes clases de protocolos MAC.
1: Esto se podría permitir mientras que el nodo destino fuera externo al área de cobertura de R.
2: Desde nodos externos al área de cobertura de S.
3: Hacia cualquier nodo.
4: Si E tendría permiso para transmitir, el paquete colisionaría con el enviado por S.
5: Hacia nodos externos al área de cobertura de R.
6: Desde nodos externos al área de cobertura de R.

2.2.2. Nivel de red

El nivel de red es el encargado de establecer una ruta para la transmisión de los datos entre el nodo que desea enviar datos y el nodo que tiene que recibir dichos datos mediante un protocolo de encaminamiento. Además, un protocolo de encaminamiento tiene que ser capaz de mantener las rutas y de enviar los paquetes de datos a lo largo de las mismas [1, 3]. Para comenzar, se ofrece una visión general sobre las funciones de un protocolo de encaminamiento y los problemas que surgen en las redes sin infraestructura, justificando la necesidad de utilizar protocolos específicos para las redes ad hoc que tengan en cuenta sus particularidades. Finalmente se presentan varios criterios de clasificación para los distintos protocolos de encaminamiento.

2.2.2.1. Encaminamiento en redes ad hoc inalámbricas

El uso de un protocolo de encaminamiento es obligatorio [3, 6, 20], ya que el consumo de energía en redes ad hoc móviles normalmente tiene que ser limitado. Si a este hecho le añadimos que en las comunicaciones inalámbricas el canal de frecuencia es compartido [19], no se hace posible la transferencia de datos directamente entre emisor y receptor, salvo que estos estén a una distancia corta, ya que interferiría en todos los demás nodos de la red y la energía consumida sería muy grande.

Igualmente importante en los protocolos de encaminamiento para redes ad hoc es que

tienen que soportar grados de movilidad mayores que en las redes cableadas, ya que los nodos se mueven de forma aleatoria a distintas velocidades mientras que en las redes tradicionales los cambios se deben principalmente a fallos en los nodos. Por tanto, los protocolos utilizados en redes cableadas, como RIP (*Routing Information Protocol*) [21] o OSPF (*Open Shortest Path First)* [22], no son útiles en las redes móviles ya que tendrían que enviar continuamente mensajes de control [3]. Para este último protocolo se ha definido la extensión OSPF-OR (*OSPF-Overlaping Relay*) [23], que permite su implementación en redes ad hoc móviles. Esta mejora se encarga principalmente de reducir el tamaño de los paquetes de control y de optimizar el método de inundación para la actualización de rutas. La inundación es la técnica utilizada por los nodos para enviar mensajes a todos sus vecinos. El inconveniente de adaptar un protocolo ya existente es precisamente la misma adaptación, ya que en un principio OSPF no fue diseñado para cubrir las necesidades de las redes ad hoc móviles. En consecuencia, la tendencia es definir protocolos específicos para estas redes que atiendan desde el primer momento sus particularidades.

En esencia, en las redes ad hoc inalámbricas son necesarios protocolos de encaminamiento capaces de soportar el continuo cambio de la topología para poder cumplir con sus funciones. Es necesario mencionar que, aunque soportan distintos grados de movilidad, ningún protocolo puede hacer su cometido si para cada paquete es necesario encontrar una ruta nueva. En este caso la única técnica posible de encaminamiento es la inundación [3]. Por lo tanto, también los métodos para encaminar en redes móviles tienen limitaciones en cuanto al grado de movilidad pero menos restrictivas que en el caso de los protocolos tradicionales para redes fijas.

Un protocolo de encaminamiento tiene dos principales funciones [3]:

- Encontrar una ruta factible entre los nodos emisor y receptor. Esta misión se realiza antes de enviar los paquetes de datos y compone la «fase de descubrimiento».
- Transmitir los paquetes de datos entre los distintos nodos de la ruta. Esta función forma la «fase de conmutación».

Atendiendo a la fase de descubrimiento, existen varias maneras de clasificar los protocolos de encaminamiento. En las siguientes líneas solamente se comentan tres de ellas, ya que se consideran más relevantes [3, 12, 24]:

1. Encaminamiento en «origen» frente a encaminamiento «salto-a-salto». En el primer caso, la ruta se establece al iniciar una transmisión y es el nodo origen el que se encarga de guardar esta información para sucesivos envíos. El camino que tienen que seguir los datos se adjunta en el paquete, por lo que la tasa de transmisión efectiva disminuye. Por otro lado, en el encaminamiento salto-a-salto son los nodos

intermedios los que deciden el siguiente salto en la comunicación y guardan la información necesaria para enviar los paquetes hacia el nodo destino. Como ejemplo, el protocolo DSR [25] utiliza el encaminamiento en origen, mientras que el protocolo AODV [26] utiliza encaminamiento salto-a-salto.

2. Planificación de la fase de descubrimiento. En este caso se tiene en cuenta el momento en el que se buscan las rutas. Las tres opciones disponibles son:

   a) Tener siempre las rutas disponibles. Las rutas se buscan de forma periódica y todos los nodos que componen la red tienen un camino en el momento en el que quieran utilizarlo. Este es el enfoque del los protocolos llamados «proactivos» e implica tener una gran capacidad de almacenamiento en los nodos, ya que necesitan guardar todas las rutas posibles de la red en todo momento. Sin embargo, todos los paquetes de la misma comunicación de datos tardarán un tiempo parecido en ser transmitidos [1, 24].

   b) Buscar un camino solamente cuando es necesario. Al contrario que en el caso anterior, ahora solo se busca una ruta nueva cuando se necesita. Este planteamiento, llamado «reactivo», no necesita tanta capacidad de almacenamiento como el anterior pero el primer paquete de la transmisión tendrá un tiempo de transmisión sustancialmente mayor que el resto de paquetes debido a que tiene que esperar a que se encuentre una ruta entre los nodos emisor y receptor [1,24].

   c) Utilizar un planteamiento híbrido entre proactivo y reactivo. En este caso se aprovecha el conocimiento de la localización de los nodos para el descubrimiento de las rutas. Cuando el nodo destino se encuentra a gran distancia del nodo emisor se utiliza el enfoque reactivo, mientras que para distancias cortas y para los mensajes de mantenimiento se utiliza el enfoque proactivo [24].

   A modo de ejemplo, los protocolos DSDV (*Destination-Sequenced Distance Vector*) [27] y OSLR (*Optimized Link State Routing*) [28] utilizan el enfoque proactivo, los protocolos AODV (*Ad hoc On-demand Distance Vector*) [26] y DSR (*Dynamic Source Routing*) [25] utilizan el enfoque reactivo y el protocolo ZRP (*Zone Routing Protocol*) [29] utiliza al enfoque híbrido.

3. Elección de la ruta óptima. Uno de los criterios más utilizados es el de utilizar la ruta con menor número de nodos intermedios por los que pasan los datos, como es el caso de AODV. En el caso del protocolo ABR (*Associativity-Based Routing*) el criterio es el de seguir el camino que contiene nodos situados en la zona más densa de la red, asegurando de este modo un mayor tiempo de vida a la ruta [3, 12, 30].

Capítulo 3

Protocolos de encaminamiento AODV y DSR

Este capítulo ofrece un análisis más detallado sobre los protocolos de encaminamiento utilizados para el estudio que se realiza en este Proyecto Fin de Carrera: AODV (Sección 3.1) y DSR (Sección 3.2). La elección de los protocolos se ha hecho teniendo en cuenta que este estudio forma parte de uno mayor que se lleva a cabo en el Departamento de Teoría de la Señal y Comunicaciones de la Universidad Rey Juan Carlos, donde se utilizan los mismos protocolos de encaminamiento. Estos dos protocolos son los dos más característicos en cuanto a encaminamiento en origen y salto-a-salto. Como ya se ha mencionado, las redes ad hoc son un tipo especial de redes y, por lo tanto, necesitan protocolos de encaminamiento específicos, capaces de descubrir rutas entre el nodo que transmite los datos y el nodo que los recibe en entornos sin infraestructura y con movilidad de los nodos y los protocolos elegidos cumplen estas exigencias. La decisión de utilizar AODV y DSR se ha tomado atendiendo a que son protocolos ampliamente utilizados tanto en escenarios en los que las redes se encuentra desplegadas físicamente como en investigación [8,31–34]. A continuación detallamos su funcionamiento centrándonos en sus diferencias.

3.1. AODV

La primera vez que se menciona el protocolo AODV es en 1997 en formato *Internet-draft* [35] por Charles E. Perkins como documento de trabajo del grupo de trabajo MANET *(Mobile Ad Hoc NETworking)* de IETF *(Internet Engineering Task Force)*. La versión actual se rige por la RFC3561 (*Request For Comments*) donde se detallan las características del mismo [26].

El protocolo AODV fue diseñado para redes ad hoc móviles con gran población de nodos, típicamente de decenas a miles, con distintos grados de movilidad, desde nula o

baja a relativamente alta. Uno de los principales objetivos del AODV es la reducción del número de mensajes de control [36]. De esta forma se reduce la latencia de los mensajes de datos. Otra característica a tener en cuenta es que no incluye seguridad en la red [37], ya que fue proyectado para redes donde los nodos pueden confiar los unos en los otros.

Tal y como se mencionó en la Sección 2.2.2.1, el protocolo AODV utiliza encaminamiento salto-a-salto (la decisión sobre el siguiente salto la toman los nodos intermedios) y es de tipo reactivo. Utiliza el concepto de «número de secuencia» de un nodo, que es un contador que se incrementa inmediatamente antes de generar un mensaje de petición de ruta nueva o inmediatamente antes de responder a una petición de ruta. Con este mecanismo, todos los nodos utilizan la información relativa al número de secuencia mayor, es decir, la más nueva, y pueden descartar información respecto a rutas o mensajes que contienen un número de secuencia menor, con información obsoleta. Igualmente se evita la formación de bucles en los procedimientos de inundación de red, ya que los nodos únicamente procesan un mensaje con un determinado número de secuencia una vez y, si vuelven a recibir un paquete con un número de secuencia igual, lo descartan. Este número de secuencia también se utiliza para resolver el problema de «cuenta al infinito» de Bellman-Ford [26,38], por lo tanto la convergencia de la red mejora.

En la Figura 3.1 se muestra como se genera la cuenta al infinito en una red de cuatro nodos que no almacenan información sobre el siguiente nodo en la ruta. Conviene resaltar que los mensajes de tipo «Soy X» los envían todos los nodos a todos sus vecinos, pero en la figura solo se incluyen los que envía el nodo A por claridad de la misma. Con ayuda de estos mensajes todos los nodos conocen la distancia hacia sus vecinos más cercanos. En este ejemplo, cuando el enlace A-B se rompe en el instante temporal 7, B deja de recibir actualizaciones del nodo A pero sigue recibiendo información sobre una ruta hacia A desde el nodo C y actualiza su tabla de rutas con la nueva información recibida. De igual modo, en los siguientes instantes temporales, los demás nodos van actualizando sus tablas de rutas con la información que reciben de sus vecinos y en dichas tablas la distancia hasta el nodo A va incrementado. La cuenta al infinito se origina, tal y como se observa, debido a que los nodos simplemente almacenan la distancia al nodo y no incluyen en sus tablas información sobre el siguiente salto en la ruta. En el ejemplo de la Figura 3.1, si los mensajes de tipo «Soy X» incluyeran un número de secuencia y la información derivada de estos mensajes incluyeran el mismo número el problema de la cuenta al infinito no aparecería. Después la perder del enlace A-B los nodos no actualizarían la distancia hasta A ya que recibirían mensajes con el mismo número de secuencia y estos se descartarían.

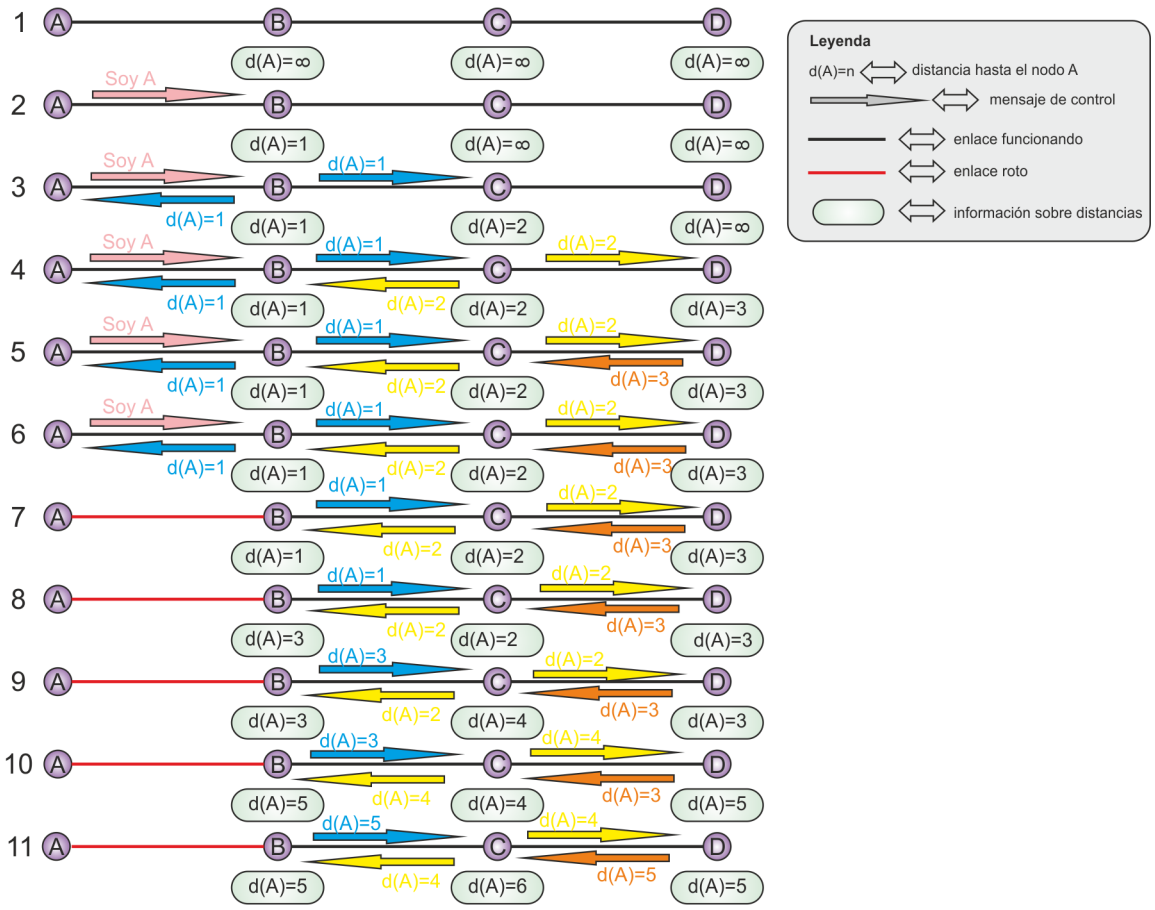

Figura 3.1: Ejemplo del problema de «cuenta al infinito» de Bellman-Ford para una red de 4 nodos.

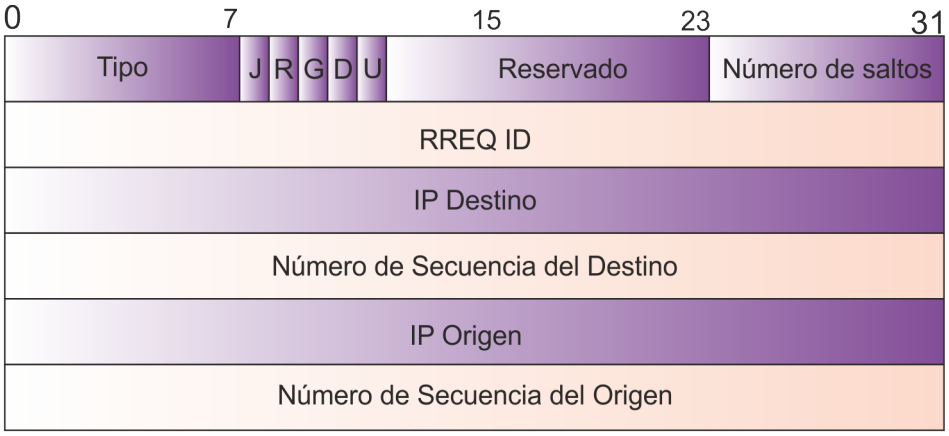

Figura 3.2: Formato de los mensajes de petición (RREQ) para AODV

3.1.1. Tipos de mensajes

El protocolo AODV dispone de cinco tipos de mensaje [26]: de petición (RREQ, *Route REQuest*), de respuesta (RREP, *Route REPly*), de asentimiento a mensajes de respuesta (RREP-ACK, *Route REPly ACKnowledgment*), de error (RERR, *Route ERRor*) y *Hello*. A continuación se detallan únicamente los dos primeros tipos ya que son los mensajes utilizados en la misión principal del protocolo: el descubrimiento de las rutas. Los mensajes RREP-ACK se utilizan especialmente en redes con enlaces unidireccionales acompañando a los de tipo RREP. Los de tipo RERR se utilizan cuando se detecta que uno de los enlaces que forma una ruta se ha roto y los de tipo *Hello* se emplean para el mantenimiento de los enlaces entre nodos vecinos.

3.1.1.1. Mensajes de petición

Los mensajes de petición del protocolo AODV son los de tipo RREQ. Un nodo decide enviar un mensaje del tipo RREQ cuando quiere enviar datos a otro nodo y no tiene disponible ninguna ruta activa hacia dicho destino. En la Figura 3.2 se presenta el formato de los mensajes RREQ y seguidamente se explican los campos relevantes:

- Tipo: se utiliza para distinguir entre los distintos tipos de mensaje. En este caso es 0.
- *Flags*:
  - J, R: reservados para un escenario *multicast*.
  - G: indica que se tiene que enviar un mensaje RREP «gratuito» hacia el destino en caso de que algún nodo intermedio conteste a este mensaje. Se utiliza para asegurar que el destino también dispone de una ruta hacia el origen en caso de

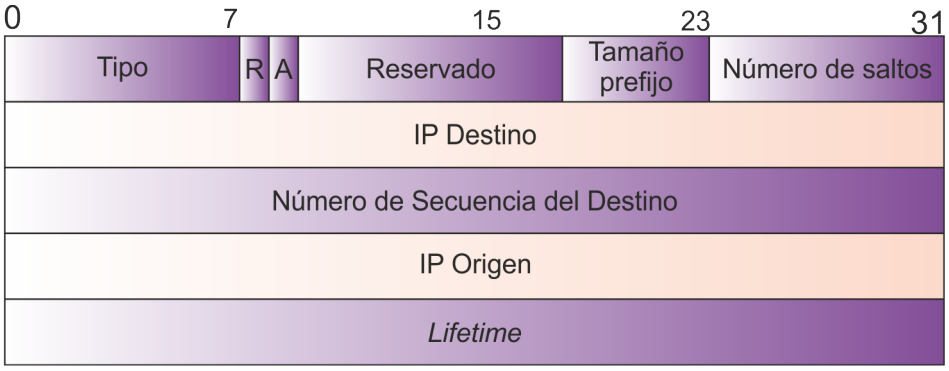

Figura 3.3: Formato de los mensajes de respuesta (RREP) para AODV

no intervenir en el descubrimiento de ruta. Esta situación se explicará con más detalle en la Subsección 3.1.4.2.

- D: señala que únicamente el destino puede responder a esta petición.
- U: el emisor no conoce el número de secuencia del destinatario.

- Número de saltos: almacena el número de saltos intermedios que ha realizado el paquete hasta llegar a este punto.
- RREQ ID: número de secuencia único que identifica una petición de ruta cuando se toma en conjunto con la dirección IP (*Internet Protocol*) del nodo origen.
- IP Destino: dirección IP del nodo hacia el que se desea enviar la información.
- Número de Secuencia del Destino: último número de secuencia del que se tiene constancia de una ruta que inició este mismo emisor hacia el mismo receptor. Si se desconoce, obligatoriamente se tiene que activar el *flag* U.
- IP Origen: dirección IP del nodo que inicia la búsqueda de una nueva ruta.
- Número de Secuencia del Origen: es el número de secuencia que utiliza el nodo inicial para identificar esta ruta en su tabla de encaminamiento.

3.1.1.2. Mensajes de respuesta

Los mensajes RREP se generan en respuesta a una petición de camino, es decir, un nodo forma y envía un mensaje de este tipo únicamente cuando recibe un mensaje RREQ y cumple las condiciones necesarias para contestar (por ejemplo, cuando es el nodo destino). Los campos existentes se muestran en la Figura 3.3 y, debido a la coincidencia de estos con los mencionados en la subsección anterior, se explican seguidamente únicamente las novedades:

- Tipo: 2

- *Flags*:
  - R: utilizado en escenario *multicast*.
  - A: requerido asentimiento.
- Tamaño Prefijo: se utiliza para configurar subredes dentro de la misma red ad hoc.
- *Lifetime*: tiempo (en milisegundos) indicando el periodo durante el que los nodos involucrados en la ruta la consideran como válida (activa).

3.1.2. Información almacenada

Todos los nodos que componen la red disponen de una tabla de encaminamiento donde se almacena información sobre los destinos conocidos por el propio nodo, tanto si se tiene una ruta válida como si la ruta ha caducado. Los campos de cada entrada de la tabla son los siguientes [26]:

- Dirección IP del destino
- Número de secuencia del destino: número de secuencia que el nodo indica en los mensajes RREQ y RREP para referenciarse unívocamente.
- *Flags*: almacenan información de distinta naturaleza, como el estado de la ruta o la validez del número de secuencia del destino.
- Interfaz de red: interfaz de red por la que se accede al nodo al que se refiere la entrada de la tabla. Se almacena porque AODV puede ser utilizado también para redes cableadas donde un nodo puede disponer de más de una interfaz.
- Número de saltos: número de saltos necesarios para llegar al nodo destino partiendo de este nodo.
- Siguiente salto: nodo vecino al que se le envían por *unicast* los datos dirigidos al destino.
- Lista de precursores: lista de los nodos que forman parte de una ruta. Los nodos que forman parte de esta lista reciben mensajes del tipo RERR si se detecta un enlace roto.
- *Lifetime*: tiempo durante el cual la ruta se mantiene válida (activa).

3.1.3. Parámetros de configuración del protocolo

El protocolo AODV es flexible a la hora de adaptar sus parámetros para la implementación en redes, es decir, se pueden modificar todos los valores numéricos presentados en la RFC3561 [26]. Sin embargo, la modificación de estos valores puede afectar al rendimiento del protocolo. A continuación se presenta una muestra de dichos parámetros:

PATH_DISCOVERY_TIME: tiempo durante el que se almacena la información relativa a un procedimiento de descubrimiento de ruta. Si durante este tiempo el nodo recibe de nuevo un mismo paquete de petición de ruta lo descarta.

TTL_START: valor utilizado para el campo TTL en el inicio de un descubrimiento de ruta. Por defecto toma el valor 1. En el caso de disponer de información relativa al nodo destino (porque se dispone de una entrada no válida en su tabla de encaminamiento), en lugar de utilizar el valor 1 se utiliza un TTL dos unidades mayor que el número de saltos hasta el destino.

TTL_INCREMENT: indica la cantidad de unidades que se incrementa el TTL utilizado en la búsqueda actual (que no ha dado resultado) para ser utilizado en la siguiente búsqueda. El valor por defecto es 2.

TTL_THRESHOLD: umbral por debajo del cual se utiliza el incremento en TTL explicado con anterioridad. Después de alcanzar el umbral se utiliza el valor máximo permitido. Esta variable toma por defecto el valor 7.

NET_DIAMETER: indica el numero máximo de saltos en una comunicación. Por tanto el valor más alto que se puede asignar al campo TTL. Su valor por defecto es 35.

RREQ_RETRIES: número máximo de reintentos de descubrimiento de ruta para un tiempo de vida dado. Por defecto, el número máximo de reintentos es 2.

MY_ROUTE_TIMEOUT: tiempo durante el cual el nodo afirma que la ruta es válida. El valor por defecto es de 6 segundos.

3.1.4. Descubrimiento de ruta

El procedimiento de descubrimiento de una nueva ruta comienza cuando un nodo, que en adelante denominaremos «emisor», quiere enviar datos a otro nodo, llamado «receptor», y observa que no tiene ninguna ruta activa en su tabla de encaminamiento. El método seguido se divide en dos pasos: la formación del camino de vuelta y la formación del camino de ida, explicados con más detalle a continuación.

3.1.4.1. Formación del camino de vuelta

Como acabamos de comentar, el inicio del procedimiento en cuestión se debe a que el emisor no tiene una ruta disponible para enviar datos al receptor. Esto puede ocurrir en dos casos:

- Los nodos que pretenden comunicarse no lo han hecho con anterioridad ni han formado parte ambos de una ruta entre otros nodos, por lo que no disponen una ruta entre ambos.
- Los nodos sí conocen la existencia el uno del otro, pero la ruta utilizada ha dejado de estar activa.

En cualquiera de estas situaciones, el emisor construye un paquete de tipo RREQ y lo envía por *broadcast* (dirección IP 255.255.255.255) [39] hacia sus vecinos. Para formar este mensaje, el nodo incrementa el valor de su número de secuencia para distinguirlo de peticiones de ruta anteriores. En sucesivos reenvíos del mismo mensaje se incrementa también el valor de RREQ ID. Antes de enviarlo, almacena en un *buffer* el identificador de la petición y su propia dirección IP durante un tiempo determinado. De esta forma evita procesar este mismo mensaje cuando sus vecinos lo envíen por *broadcast*.

Para evitar inundar la red con mensajes innecesarios de descubrimiento de ruta, el AODV realiza una búsqueda expansiva en anillo (*expanding ring search*) [26, 36] que consiste en enviar los paquetes RREP dentro de un radio que va incrementando. El control del radio utilizado se lleva a cabo a través del campo TTL (*Time To Live*), que indica el número de reenvíos máximos que puede sufrir un paquete. La primera búsqueda se realiza en una pequeña área de la red centrada en el nodo emisor con un radio determinado por la constante TTL_START o por el número de saltos necesarios para llegar al receptor si se tiene información sobre este dato; se puede disponer de dicho dato si el emisor tiene en su tabla de encaminamiento una ruta no válida (al caducar una ruta esta no se elimina de la tabla). En la Figura 3.4 se ilustran las dos primeras fases de la búsqueda expansiva en anillo, cuando el TTL es 1 y seguidamente se incrementa hasta 3. En este ejemplo se parte de la suposición que ninguno de los vecinos del emisor es el receptor y además no lo conoce indirectamente. En el primer caso, Figura 3.4(a), el emisor prepara el mensaje y lo envía. Sin embargo, como el receptor no es uno de sus vecinos más cercanos estos descartarán dicho mensaje. En el segundo caso, Figura 3.4(b), se muestra el escenario donde el receptor no fue encontrado en el radio mínimo de búsqueda y el valor del TTL se incrementó una vez. Se observa el decremento de una unidad en el valor del TTL en cada reenvío y la inundación (envío por *broadcast*) que se lleva a cabo en la red y, por lo tanto, la posibilidad de que varios mensajes RREQ lleguen al receptor.

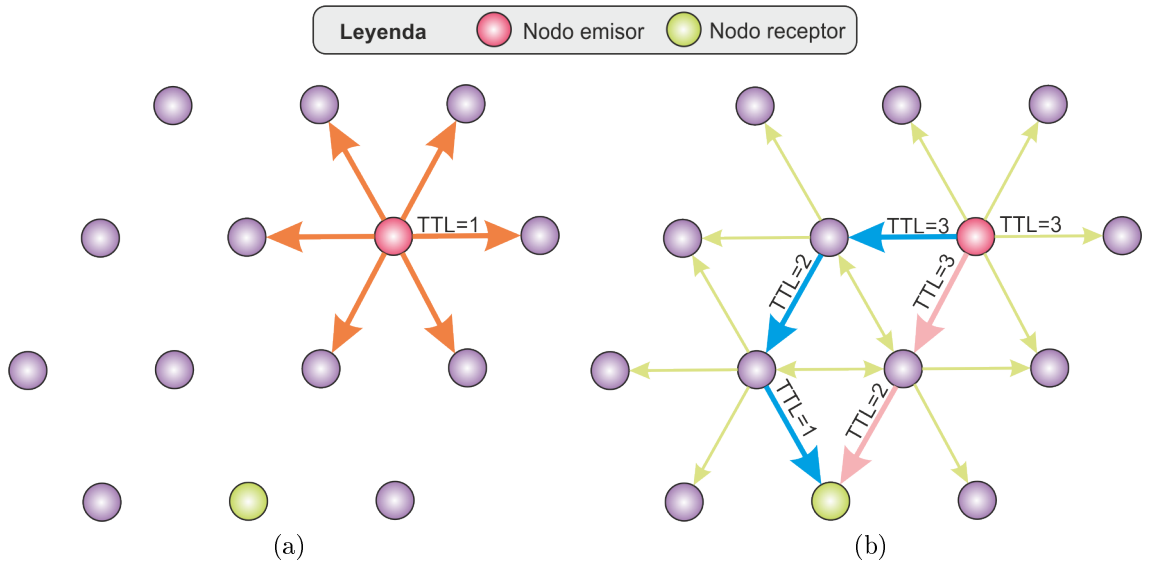

Figura 3.4: Envío de mensajes de tipo RREQ con TTL igual a 1 (a) y 3 (b) en el protocolo AODV.

Si la búsqueda no fue exitosa, el radio de la búsqueda se amplia sucesivamente hasta llegar a un umbral determinado (TTL_THRESHOLD). Si aún así no se ha encontrado al destinatario, las siguientes búsquedas se realizan dentro del máximo radio permitido (NET_DIAMETER). Si después de varios reintentos (tantos como indica la constante RREQ_RETRIES) de búsqueda todavía no se ha encontrado al nodo receptor, el nodo emisor desiste en su búsqueda y descarta toda la información que tenía que ser enviada.

Cuando un paquete RREQ llega a un nodo intermedio, la primera acción que realiza es actualizar su tabla de encaminamiento con la nueva información recibida sobre el nodo emisor. Seguidamente, con el fin de evitar la aglomeración de mensajes de descubrimiento de ruta en la red, si la misma petición fue recibida con anterioridad, este paquete se descarta [26]. En el caso de ser la primera vez que se recibe la solicitud, el nodo tiene dos posibilidades:

- Si conoce un camino hacia el destino y el *flag* D no está activado, se genera un paquete RREP en respuesta al paquete RREQ y se descarta este último. En este caso, si el *flag* G del paquete está activado es necesario enviar un RREP llamado «gratuito» hacia el nodo destino para asegurar que se forma una ruta completa.
- Actualizar el valor del Número de saltos (incrementándolo una unidad) y del TTL (decrementándolo una unidad). Si este último sigue siendo mayor que 0 se reenvía por *broadcast* a sus vecinos para seguir con el proceso de descubrimiento de ruta.

El proceso continua hasta que el emisor recibe al menos un mensaje RREP o mientras que, según se ha comentado con anterioridad, la búsqueda no produce ningún resultado dentro del máximo radio permitido. Esta fase del descubrimiento de ruta recibe el nombre

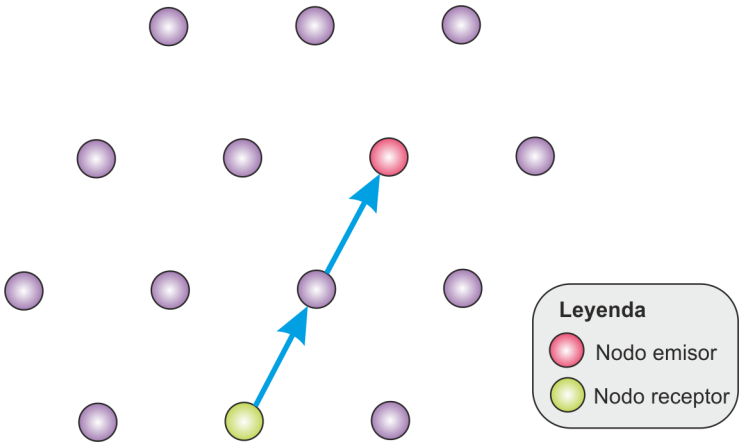

Figura 3.5: Envío de mensaje de tipo RREP en el protocolo AODV.

de «formación del camino de vuelta», ya que la información que almacenan los nodos intermedios sobre la ruta se utiliza para la transmisión receptor-emisor, es decir, se utiliza para la «vuelta» de los paquetes con información (los bits de datos que el emisor tiene que enviar al receptor).

3.1.4.2. Formación del camino de ida

La segunda fase del descubrimiento de ruta comienza con la generación de un paquete RREP, bien en un nodo intermedio bien en el receptor. El primer caso solamente se puede encontrar si se cumplen ciertas condiciones: el nodo dispone de una ruta activa hacia el destino y el *flag* D (que indica que únicamente el destinatario puede contestar) del mensaje RREQ no está activado. En adición a esta condición, si el mensaje tiene el *flag* G activado, además de enviar el RREP hacia el emisor se tiene que enviar otro RREP gratuito hacia el receptor. Una vez creado el mensaje RREP, este se envía hacia su destinatario mediante *unicast*.

En el caso de que el que genera el mensaje sea el destinatario final, este incrementa su número de secuencia de destino si el actual coincide con el número de secuencia del origen [26]. El incremento se realiza para que los números de secuencia de los extremos no coincidan. Sin embargo, cuando el mensaje lo genera un nodo intermedio, el valor que se pone en el campo NÚMERO DE SECUENCIA del destino es el que tiene almacenado dicho nodo en su tabla de encaminamiento. En la Figura 3.5 se puede observar el camino que seguirán los mensajes RREP desde el receptor hasta el emisor. Al llegar el mensaje RREP al emisor, la ruta queda establecida en ambas direcciones y los nodos pueden empezar a enviar datos. Al igual que en la Figura 3.4, el camino recorrido por los mensajes se muestra mediante flechas.

Cuando un nodo recibe un mensaje de tipo RREP busca en su tabla de encaminamiento una ruta hacia el nodo que tiene que recibir dicho mensaje. Seguidamente incrementa el

valor del número de saltos para poder almacenar o actualizar información hacia el nodo que generó el RREP. La actualización de la información se realiza solamente si se cumple una de las siguientes condiciones:

- El número de secuencia que lleva el mensaje está marcado como inválido en su tabla.
- El número de secuencia del destino que lleva el mensaje es mayor que el que tiene almacenado el nodo en su tabla.
- El número de secuencia del destino coincide pero la ruta está marcada como inactiva.
- El número de secuencia del destino coincide pero el número de saltos de este mensaje (contando el incremento hecho en el nodo que procesa el mensaje) es menor.

En cuanto a la validez de las rutas dentro de las tablas, los nodos intermedios se encargan de calcular el tiempo de validez en función del valor indicado en el mensaje y del tiempo que tardó en llegar hasta el mismo nodo. De esta forma, las entradas de todas las tablas para una misma ruta caducan en el mismo instante.

3.1.5. Mantenimiento de la ruta

El mantenimiento de rutas se realiza mediante la utilización de mensajes tipo *Hello* y RERR. Los mensajes *Hello* se utilizan para monitorizar la conectividad que tiene una estación con sus vecinos y se envían de forma periódica con un valor de TTL igual a 1. En el caso que un nodo deje de enviar los mensajes *Hello*, sus nodos vecinos dan el enlace por inexistente y en este momento intervienen los mensajes de tipo RERR, que se encargan de informar de esta situación a todos los nodos que forman parte de las rutas conocidas por la estación que detecta el fallo. Opcionalmente, el nodo que detecta el fallo de una conexión puede buscar una ruta alternativa utilizando una petición de ruta [26].

3.2. DSR

La primera descripción detallada del protocolo DSR se hace en 1996 [40], aunque en 1994 las bases del mismo son presentadas [41]. En la RFC4728 [25], publicada en 2007, se define como protocolo experimental de Internet.

DSR es un protocolo que se clasifica como «completamente reactivo» y con encaminamiento en origen. Completamente reactivo porque todos sus acciones se realizan únicamente bajo demanda y no tiene ningún tipo de mensaje periódico propio del protocolo que tenga la misión de conocer el estado de los enlaces de la red. Como consecuencia de lo

dicho anteriormente, se reduce el consumo de ancho de banda utilizado por mensajes de control. Al ser clasificado como encaminamiento en origen, el protocolo es «sin estado», es decir, un nodo intermedio no tiene la obligación de almacenar información sobre las rutas ya que cada paquete de datos tiene incluido el camino a seguir. Gracias a este hecho, si uno de los nodos perteneciente a una ruta tiene algún problema (por ejemplo, se reinicia), puede volver a formar parte de la misma de forma casi inmediata, ya que la ruta a seguir está almacenada en la cabecera de los paquetes. Así se evita utilizar una ruta alternativa o descubrir una nueva, en caso de no tener ninguna alternativa [25].

Al ser un protocolo de encaminamiento diseñado para redes ad hoc, DSR no necesita ningún tipo de infraestructura para funcionar. Fue diseñado principalmente para redes con menos de 200 nodos, aunque se puede utilizar para redes mayores a costa de un incremento considerable en la memoria necesaria para almacenar la información de las rutas [42]. Es capaz de funcionar con altos grados de movilidad y permite a los nodos tener múltiples rutas hacia un destino y elegir la que consideren más conveniente en cada momento. Por ejemplo, si se detecta algún fallo en la ruta utilizada, se puede utilizar inmediatamente otra sin tener que buscar una alternativa [25, 43].

3.2.1. Tipos de mensajes

En el protocolo DSR no se define un tipo específico de mensajes sino, que se definen opciones que se añaden en paquetes IP inmediatamente después de su cabecera. De esta forma, la información relativa a direcciones IP o TTL están almacenadas en el paquete IP y no en un mensaje DSR propiamente dicho.

Los paquetes que forman el protocolo DSR están compuestos por una parte fija de 4 octetos seguidos de una parte de longitud variable que depende del tipo de mensaje enviado [25]. En la Figura 3.6 se observa la estructura de todos los mensajes y, a continuación, se explican los campos que la componen.

- Siguiente cabecera: identifica el tipo de la siguiente cabecera, en caso de haber mas. Este campo está presente en todas las opciones de IP y forma una lista de opciones a analizar en los sucesivos nodos intermedios o en el nodo final, dependiendo del tipo de opción. Por ejemplo, las opciones referentes al encaminamiento se encuentran al inicio de la lista y son examinadas por todos los nodos, mientras que opciones sobre seguridad solamente son analizadas por los nodos extremos [44].

- *Flag* F: tiene que estar fijado a 0 ya que solamente se utiliza en una extensión del protocolo llamada «*flow state*». Esta extensión reduce la sobrecarga del protocolo dando la posibilidad al nodo emisor de habilitar el encaminamiento salto-a-salto en los nodos intermedios.

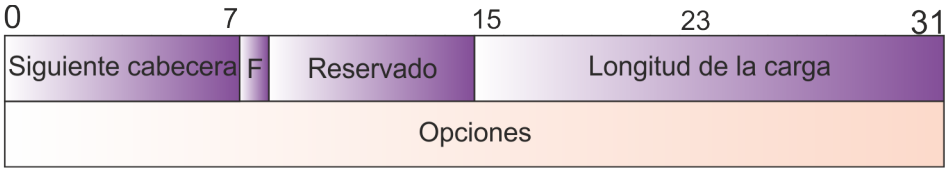

Figura 3.6: Formato de la cabecera de opciones en el protocolo DSR.

- Bits reservados: tienen que estar fijados a 0 y se ignoran.
- Longitud de la carga: se indica el tamaño de las opciones que vienen a continuación, excluyendo los 4 octetos de la cabecera fija.
- Opciones: este campo ya no forma parte de los 4 octetos fijos y es de longitud variable, dependiendo del tipo de mensaje DSR que se envíe. Aunque existen multitud de opciones, cada una identificada por un número único entendible por todos los nodos que implementan el protocolo, a continuación se explican solamente los mensajes de petición (opción *route request*) y los de respuesta (opción *route reply*) por ser los dos más importantes que utiliza el DSR.

3.2.1.1. Mensajes de petición

Los mensajes de petición en el protocolo DSR son los de tipo *route request* y se utilizan cuando un nodo emisor necesita conocer una nueva ruta hacia el destino. En la Figura 3.7 se revela el formato que tiene esta opción y, seguidamente, se explican los campos que la componen.

- Tipo de opción: número que identifica la opción *route request* de forma inequívoca.
- Longitud de la opción: indica el tamaño de la opción, excluyendo este campo y el anterior.
- Identificación: número único que genera el emisor de este mensaje para poder diferenciar entre peticiones sucesivas.
- IP destino: es la dirección del destinatario de esta petición de ruta.
- Dirección IP ((1),(2),...,(n)): cada uno de estos campos indica la dirección IP de los nodos de la ruta. En esta lista no está incluida la dirección IP del nodo que inicia el descubrimiento de ruta ya que está guardada en el campo correspondiente de la cabecera del paquete IP.

Con el envío de estos paquetes se inicia el procedimiento de descubrimiento de ruta que será presentado en la Sección 3.2.4.

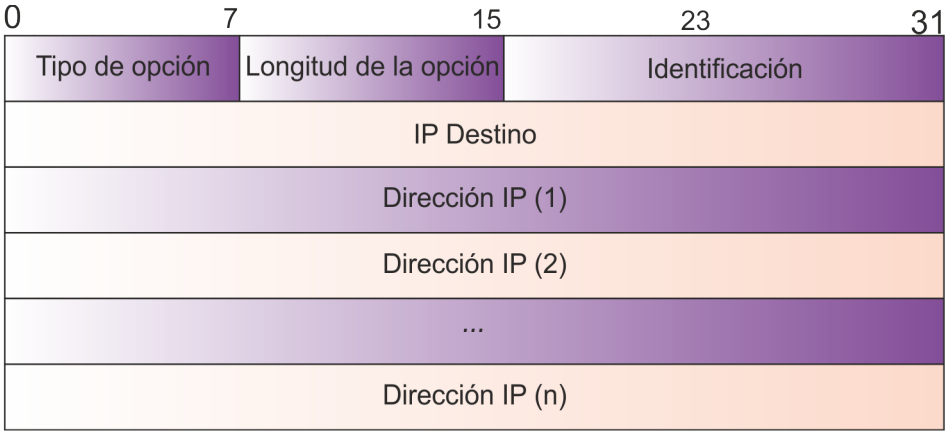

Figura 3.7: Formato de los mensajes de petición (opción *route request*) en el protocolo DSR.

3.2.1.2. Mensajes de respuesta

Los mensajes *route reply* se generan como respuesta a los paquetes de tipo *route request* en diferentes situaciones. Se puede observar la composición de la opción en cuestión en la Figura 3.8 y, a continuación, se ofrecen detalles sobre cada campo que forma parte de este mensaje:

- Tipo de opción: número que identifica la opción *route reply* de forma única.
- Longitud de la opción: indica el tamaño de la opción, excluyendo este campo y el anterior.
- *Flag* L: se utiliza cuando la red tiene una conexión con otra red externa.
- Bits reservados: tienen que estar fijados a 0 y se ignoran.
- Dirección IP ((1),(2),...,(n)): almacena la ruta que tienen que seguir los paquetes enviados por el emisor del *route request*, terminando en la dirección IP (n).

Es importante destacar que dentro de un mismo paquete IP pueden viajar varias opciones *route reply*, cada una indicando una ruta disponible entre el emisor y el receptor. Este tipo de mensajes se envía de manera *unicast* hacia el nodo que inició el descubrimiento de ruta. Otro hecho a destacar es que algunas opciones pueden no estar alineadas (no empezar en el byte 0), tal y como se observa en el caso del *route reply* en la Figura 3.8. Esta forma de colocar los bytes asegura que la opción termina con 32 bits de información y no de relleno.

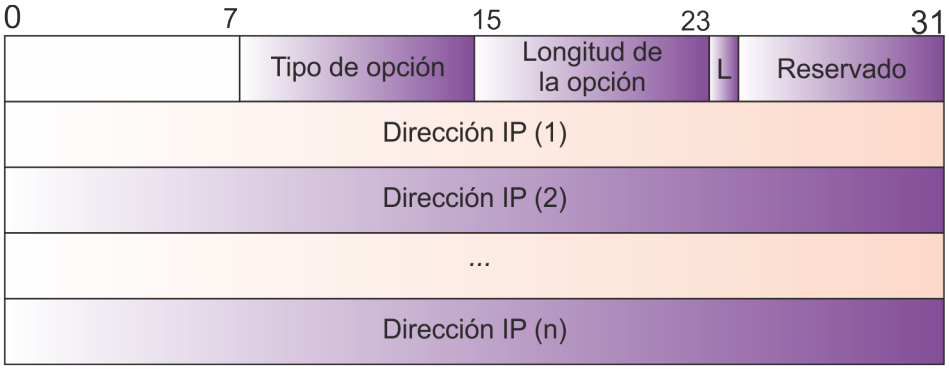

Figura 3.8: Formato de los mensajes de respuesta (opción *route reply*) en el protocolo DSR.

3.2.2. Información almacenada

Aunque DSR es un protocolo de encaminamiento en origen y los nodos no tienen que almacenar información sobre el siguiente salto, sí que se tiene que guardar información referente a las transmisiones que están en curso [25]. A continuación se muestran las estructuras de datos que un nodo tiene que utilizar para el funcionamiento básico de DSR:

- Se necesita una memoria para almacenar las rutas conocidas llamada *route cache*. Se añaden datos cuando se reciben mensajes *route request*, *route reply* o paquetes de datos que contiene una ruta completa. Esta información se elimina cuando se sabe que un enlace está roto o pasa un tiempo sin que una ruta sea utilizada.
- Los nodos tienen que almacenar en un *buffer* durante un tiempo determinado todos los datos que tienen que ir a un destino del que no se conoce el camino. Mientras el paquete de datos está en este «*send buffer*» el nodo tiene que enviar *route request* periódicos para encontrar una ruta hacia el destino. El paquete de datos se borra si no ha sido enviado antes de un tiempo determinado.
- Se mantiene una tabla para los *route request* que fueron originados o reenviados por este nodo. En esta tabla se guarda la información necesaria para realizar el descubrimiento de ruta, como el tiempo desde que se envió el último *route request* o el número de mensajes enviados sin recibir un *route reply*.

3.2.3. Parámetros de configuración del protocolo

El protocolo DSR dispone de varias constantes que se deben modificar para adaptarse a cada red en particular [25]. A continuación se presentan una muestra de las mismas:

RouteCacheTimeout: tiempo durante el cual se mantiene una ruta en la memoria *cache* de rutas. El valor por defecto es de 300 seg.

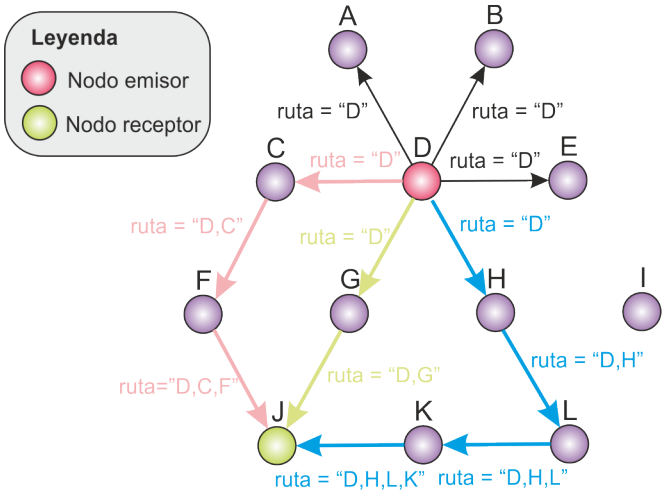

Figura 3.9: Envío de mensajes de descubrimiento de ruta en el protocolo DSR.

SendBufferTimeout: tiempo máximo durante el que se guardan datos a enviar en el *buffer* a la espera de disponer de un camino. El valor por defecto es de 30 seg.

MaxRequestPeriod: tiempo máximo que puede durar el procedimiento de descubrimiento de ruta. Una vez cumplido, se detiene la búsqueda de un nuevo camino y se descartan los datos. El valor por defecto es de 10 seg.

Max_salvage_count: indica el número máximo de veces que se puede salvar un paquete. Su valor por defecto es de 15.

3.2.4. Descubrimiento de ruta

Al igual que el protocolo AODV, el procedimiento de descubrimiento de ruta se utiliza cuando un nodo emisor quiere enviar datos a otro nodo, receptor, y no dispone de ninguna ruta en su memoria *cache* de caminos. Este procedimiento empieza con la construcción de un mensaje de tipo *route request* por parte del nodo que necesita la ruta y el envío de dicho mensaje por *broadcast* [25].

Cuando un nodo intermedio recibe el *route request*, si no es una petición que recibió con anterioridad, añade su dirección IP a la lista de direcciones y lo reenvía a todos sus vecinos. En la Figura 3.9 se muestran varias rutas que puede seguir un paquete *route request* iniciada por el nodo D y se observa como la lista de direcciones IP va aumentando a medida que se va pasando por los nodos intermedios. Como acabamos de comentar, el nodo emisor no está incluido en la lista de direcciones IP del mensaje pero en el ejemplo se ha incluido para un mejor entendimiento del mismo.

Al recibir el *route request* el nodo destino, este genera un *route reply* que se intenta enviar hacia el emisor. En el ejemplo ilustrado en la Figura 3.9, el destino generaría un mensaje *route reply* en respuesta al primer *route request* que recibe. En este momento el

destino tiene dos opciones: utilizar la ruta inversa a la que se generó por el mensaje de petición o no utilizarla. Si se tiene que utilizar una ruta distinta (por ejemplo, en una red donde los enlaces son unidireccionales) se puede dar el caso de no conocer ninguna. Entonces el destino iniciará un nuevo procedimiento de descubrimiento de ruta. Para evitar un bucle infinito, el mensaje *route reply* (la respuesta al primer descubrimiento de ruta) se añade como segunda opción en el paquete que lleva el *route request* recién generado (el nuevo descubrimiento de ruta).

En la RFC4728 [25] se definen una serie de procedimientos adicionales que los nodos pueden implementar en el descubrimiento de ruta. En las siguientes líneas se presentan algunas de las posibles mejoras:

- Cuando un nodo recibe un *route request* puede almacenar información relativa a la ruta generada hasta ese punto para poder utilizarla en caso de necesidad. Por ejemplo, en la Figura 3.9, el nodo L almacenaría una ruta hacia D a través de H.
- Si un nodo que recibe un *route request* conoce la ruta hacia el destino puede contestar él mismo a esta petición generando un *route reply*, añadiendo a la ruta existente los nodos que forman la ruta hasta el destino, asegurándose de que no hay nodos repetidos. Si hay nodos repetidos en la ruta que ve el nodo intermedio, este no puede generar el *route reply*.
- También se puede implementar el mecanismo de búsqueda expansiva en anillo (ver Sección 3.1.4.1) para evitar que haya demasiados *route request* circulando por la red.

Finalmente, para una mejor compresión del procedimiento de descubrimiento de ruta, en la Figura 3.10 se presenta el diagrama de flujo que sigue un nodo al recibir un paquete *route request*, en el que los procedimientos básicos están escritos con letra de color negro y los procedimientos adicionales con color rosa.

3.2.5. Mantenimiento de ruta

Al igual que el protocolo AODV, DSR también dispone de un procedimiento de mantenimiento de rutas. Sin embargo, mientras que AODV hace uso de mensajes periódicos entre nodos vecinos, DSR utiliza un mecanismo ya existente en las redes inalámbricas [40]. Dicho mecanismo consta de envíos de asentimientos (ACK *ACKnowledgment*) a nivel de enlace para asegurar la fiabilidad de las transmisiones [19]. Cuando un nodo no recibe estos mensajes de uno de los vecinos, el nivel de enlace comunica que no se puede transmitir un paquete y el protocolo DSR decide que el enlace hacia ese nodo está roto. Sin embargo, no es la única forma para detectar un enlace roto, ya que puede ser que el

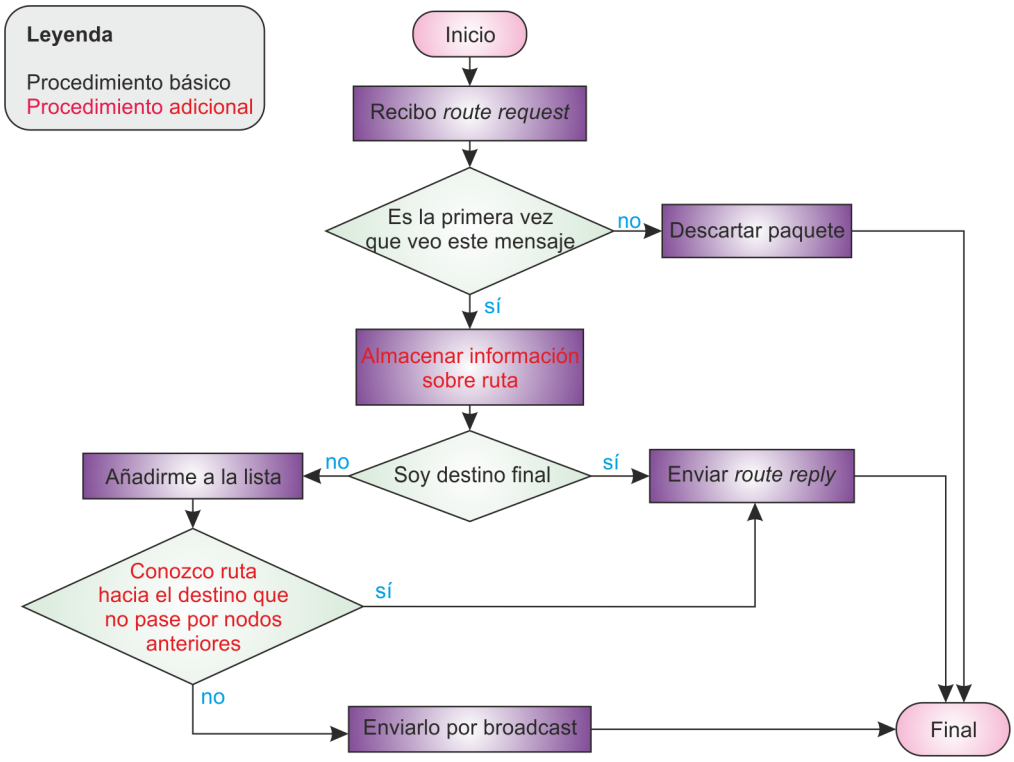

Figura 3.10: Diagrama de flujo que sigue un nodo al recibir un mensaje de tipo *route request* en el protocolo DSR.

protocolo de nivel de enlace no tenga implementados los ACK. Por lo tanto, DSR puede utilizar los llamados «ACK pasivos»: si un nodo detecta que otro nodo está transmitiendo es que la comunicación entre los dos se puede establecer. En el protocolo DSR también se ofrece la posibilidad de enviar mensajes específicos de asentimiento (la opción llamada *Acknowledgement Request*) [25].

Este procedimiento de mantenimiento de ruta solamente se utiliza cuando el nodo emisor está enviando datos haca el destino. Durante este periodo, cada nodo es responsable del estado del siguiente enlace en la ruta. Si por alguno de los métodos mencionados un nodo detecta que el siguiente enlace de la ruta está roto, tiene que enviar un mensaje de tipo *route error* hacia el emisor. Por ejemplo, en la Figura 3.9, si el nodo G detecta que el enlace G-J está roto tiene que enviar un *route error* hacia D. Todos los nodos que reciben el mensaje que informa de un enlace roto deben eliminar esta ruta de la *cache* y el emisor tiene que utilizar una ruta alternativa. Si no existe ninguna conocida, realiza un nuevo descubrimiento de ruta.

Al igual que en el descubrimiento de ruta, el procedimiento de mantenimiento de ruta dispone de varias optimizaciones [25]:

- Cuando un nodo intermedio detecta que el siguiente enlace está roto pero conoce otra ruta hacia el destino, en lugar de descartar el paquete lo puede «salvar» y usar su ruta hacia el destino. Para realizar esta acción, el nodo tiene que reemplazar la

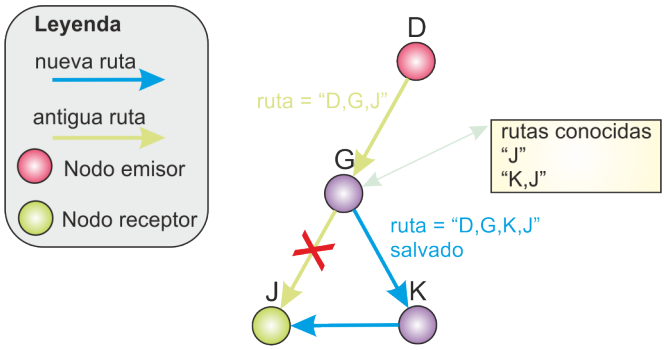

Figura 3.11: Ejemplo ilustrativo del procedimiento llamado «salvar la ruta» en el protocolo DSR.

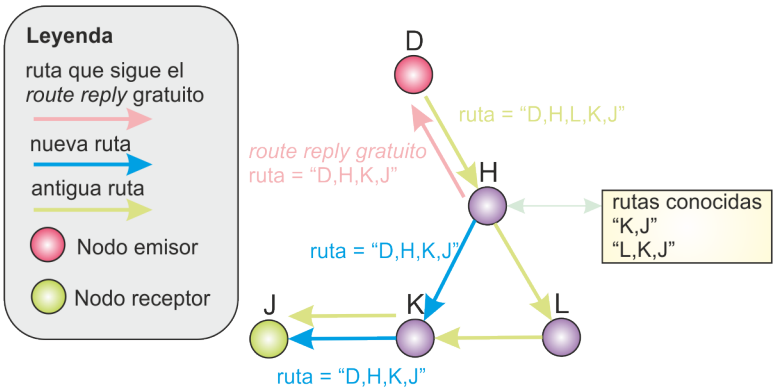

Figura 3.12: Ejemplo ilustrativo del procedimiento de acortamiento de rutas en el protocolo DSR.

ruta a seguir por el paquete por la que conoce. En la Figura 3.11, si G detecta que el enlace G-J está roto y conoce una ruta alternativa hacía J la utiliza. El nodo G puede tener conocimiento de la ruta alternativa de forma independiente a la transmisión D-J.

- En el caso de que uno de los nodos conozca una ruta más corta hacia el destino puede utilizar aquella en lugar de la que aparece en el paquete. Al realizar esta acción el nodo debe enviar un *«route reply* gratuito» hacia el emisor para que los sucesivos paquetes de datos también utilicen la ruta en la que intervienen un menor número de nodos. En la Figura 3.12, si el nodo H tiene conocimiento de una ruta más corta hacia el nodo J la utiliza y envía un mensaje *«route reply* gratuito» al emisor.

Capítulo 4

Materiales

En este capítulo se presentan los recursos empleados durante el desarrollo de este trabajo. Para ello, en primer lugar, se enumeran los equipos de trabajo utilizados, describiendo sus especificaciones *hardware* y *software* (Apartado 4.1). En segundo lugar, se presenta el simulador de red utilizado para la validación de los modelos teóricos mostrando sus principales características y, además, se detalla el proceso de resolución de uno de los problemas encontrados en dicho simulador (Apartado 4.2). En tercer y último lugar, se mencionan otros programas utilizados durante la realización de este proyecto (Apartado 4.3), como el lenguaje de programación empleado para automatizar las simulaciones, la herramienta *software* utilizada para analizar y representar los datos obtenidos a partir de las simulaciones y los dos programas utilizados conjuntamente para la redacción de esta memoria.

4.1. Equipos de trabajo

Durante la elaboración de este proyecto se ha contado con varios equipos de trabajo para la realización de las distintas tareas. Los equipos se diferencian en dos grupos: los de propósito general y los encargados exclusivamente de ejecutar las simulaciones.

Los equipos de propósito general se han utilizado principalmente para codificar los ficheros utilizados en la simulaciones y para el análisis y representación de los datos resultantes de las mismas. Las especificaciones y las principales tareas realizadas por estos equipos son:

- Equipo 1: utilizado para la codificación de los ficheros y para el análisis posterior de los datos. Con este equipo también se han realizado simulaciones de forma puntual. Se trata de un *netbook* EeePC de la marca ASUS con una CPU *Intel Atom N270* con frecuencia de reloj de 1,60GHz. Dispone de una memoria RAM (*Random Access*

Memory) de $2GB$ y un disco duro ATA de $160GB$. En cuanto al sistema operativo, este equipo tiene instalado Windows XP y *Ubuntu 10.4 Lucid* con la versión del *kernel 2.6.32-27-generic*.

- Equipo 2: utilizado para el análisis de los datos obtenidos de las simulaciones. Se trata de un ordenador marca *Sun Microsystems* que consta de una CPU *Dual-Core AMD Opteron Processor 1218* con frecuencia de reloj de $2{,}60GHz$. Dispone de $2GB$ de memoria RAM y de $500GB$ de disco duro (repartidos en dos discos iguales de la marca Hitachi). Al igual que el Equipo 1, tiene instalado tanto Windows XP, como *Ubuntu 10.10 Maverik* con la versión del *kernel 2.6.35-22-generic*.

Los equipos encargados exclusivamente para realizar las simulaciones por ordenador son los siguientes:

- Equipo 3: ordenador marca *Sun Microsystems* de iguales características *hardware* que el Equipo 2. El sistema operativo instalado es *Ubuntu 8.04 Hardy* con la versión del *kernel 2.6.14-19-generic*.
- Equipos 4 y 5: ordenadores marca Apple que constan de una CPU *PowerPC G5 (3.0)* con frecuencia de reloj de $1{,}60GHz$, $512MB$ de memoria RAM y $75GB$ de disco duro. El sistema operativo instalado es *Mac OS X 10.4.11* con la versión del *kernel Darwin 8.11.0*.

4.2. Simulador de red

Para validar el modelo teórico del retardo extremo a extremo en redes ad hoc inalámbricas presentado en este Proyecto de Fin de Carrera se recurre a un simulador de red de código libre llamado `ns-2` *(Network Simulator 2)* [45, 46]. La primera versión de `ns` fue desarrollada en 1989 como una variante del simulador de red REAL. La versión utilizada en este estudio, `ns-2`, fue desarrollada en la Universidad de California, Berkeley, como sucesora de la primera versión [47]. Actualmente, esta herramienta es mantenida por voluntarios y periódicamente se añaden nuevas actualizaciones y correcciones de errores.

Se considera `ns-2` es el simulador adecuado ya que está siendo utilizado de forma intensiva en investigación [6, 42, 48–53] y es muy flexible a la hora de modelar diferentes tipos de redes, entre los cuáles se encuentran las redes ad hoc inalámbricas. Además, al ser de código libre permite modificar todas las características de todos los protocolos implementados y añadir protocolos propios, en caso de necesitarlo. En contrapartida a esta libertad, `ns-2` presenta diversos errores que pueden entorpecer el desarrollo de un estudio.

El núcleo de `ns-2` utiliza `C++` debido a su rapidez de procesamiento y, de cara a un mejor manejo por parte del usuario, se utiliza `OTcl` (*Object oriented Tool Command Language*) como interfaz con el núcleo del simulador. Ambos lenguajes de programación utilizan una jerarquía de clases similares, de forma que se pueden traducir los objetos definidos por el usuario en `OTcl` a objetos compilados en `C++`. Sin embargo, la facilidad de uso por parte del usuario no es el principal motivo por el que se utilizan dos lenguajes de programación. Esta elección se debe a que el simulador tiene dos grandes tareas que realizar y ningún lenguaje es capaz de realizar ambas [46]:

- La simulación detallada de los protocolos de la torre OSI requiere un lenguaje de programación que sea eficaz trabajando con bytes, paquetes, cabeceras e implementación de algoritmos que utilicen conjuntos grandes de datos. En estas tareas la velocidad de ejecución adquiere mucha importancia y se utiliza `C++` para ello.

- La simulación de redes con distintas características requiere cambiar de forma rápida y fácil los parámetros o los escenarios de simulación. En este caso, lo que se necesita es una velocidad de iteración mayor (realizar cambios y volver a lanzar la simulación) y el lenguaje utilizado es `OTcl`.

El simulador `ns-2` está basado en eventos discretos y utiliza un planificador (también conocido como *scheduler*) centralizado que se encarga de ejecutar todos los eventos programados durante la simulación. En la versión actual del `ns-2`, el *scheduler* está implementado como un hilo de ejecución único. Por lo tanto, dos eventos programados en el mismo instante temporal no se pueden ejecutar a la vez y se ejecutan de forma secuencial [46].

Para realizar una simulación, el usuario tiene que codificar un *script* en `OTcl`, en adelante denominado *«script* de usuario», donde se proporcionan todos los parámetros necesarios para la simulación, como por ejemplo los protocolos MAC y de encaminamiento utilizados, la potencia de transmisión de los nodos, la posición de los mismos, el tipo de canal, el tipo de comunicación que se transmitirá y las parejas de nodos que se transmiten datos, entre otros.

Los resultados de las simulaciones son unos ficheros de trazas que ofrecen información sobre lo ocurrido en la red. En la Figura 4.1 se puede observar un extracto de un fichero de trazas generado por `ns-2` utilizando el protocolo DSR, explicándose por la presencia de los *flags* `-It DSR` y `-P dsr` resaltados en color rojo en la primera línea de la traza, y en la Tabla 4.1 se explican los *flags* más relevantes. Cada una de las líneas de la traza se genera a partir de un nuevo evento y comienza con un descriptor de evento, seguido por el instante de tiempo del mencionado evento (en segundos), el nodo emisor y el nodo receptor de dicho evento. El resto de la línea son *flags*. Conviene resaltar que los

```
# s -t 26.397655737 -Hs 408 -Hd -1 -Ni 408 -Nx -2850.00 -Ny -1299.04 -Nz 0.00
-Ne -1.000000 -Nl MAC -Nw -- -Ma 0 -Md ffffffff -Ms 198 -Mt 800 -Is 530.255 -Id
539.255 -It DSR -Il 2050 -If 0 -Ii 3 -Iv 32 -P dsr -Ph 29 -Pq 1 -Ps 2 -Pp 0 -Pn
2 -Pl 0 -Pe 0->29 -Pw 0 -Pm 0 -Pc 0 -Pb 0->0
# f -t 26.398163721 -Hs 420 -Hd -1 -Ni 420 -Nx -3000.00 -Ny -1039.23 -Nz 0.00
-Ne -1.000000 -Nl RTR -Nw -- -Ma 0 -Md ffffffff -Ms 15c -Mt 800 -Is 530.255 -Id
539.255 -It DSR -Il 1992 -If 0 -Ii 3 -Iv 32 -P dsr -Ph 29 -Pq 1 -Ps 2 -Pp 0 -Pn
2 -Pl 0 -Pe 0->29 -Pw 0 -Pm 0 -Pc 0 -Pb 0->0
# r -t 26.414056737 -Hs 474 -Hd -1 -Ni 474 -Nx -3000.00 -Ny -1558.85 -Nz 0.00
-Ne -1.000000 -Nl MAC -Nw -- -Ma 0 -Md ffffffff -Ms 198 -Mt 800 -Is 530.255 -Id
539.255 -It DSR -Il 1992 -If 0 -Ii 3 -Iv 32 -P dsr -Ph 29 -Pq 1 -Ps 2 -Pp 0 -Pn
2 -Pl 0 -Pe 0->29 -Pw 0 -Pm 0 -Pc 0 -Pb 0->0
```

Figura 4.1: Extracto de un fichero de trazas generado por `ns-2`.

Flag	Significado
`-t`	Instante de tiempo del evento expresado en segundos. Comienza en 0.0 y el último valor indica la duración de la simulación.
`-N(x, y, z)`	Coordenada en abscisas, ordenadas y azimut, respectivamente, del nodo emisor del paquete.
`-Nl`	Nivel del evento. Puede ser MAC (nivel de enlace), RTR (nivel de red), AGT (nivel superior al de red).
`-Nw`	Motivo por cual se descartó el paquete. Solamente tiene un valor distinto a `'--'` en eventos de tipo `d`.
`-Hs`	Identificador del nodo emisor de este paquete.
`-Hd`	Identificador del nodo destino de este paquete. Además puede tomar los valores -1 (corresponde a un paquete de *broadcast*) o -2 (el nodo destino no está asignado).
`-P`	Tipo del paquete. Posibles valores: dsr, aodv, tora etc.
`-Pn`	Tipo del paquete. Posibles valores: cbr, tcp.
`-Iv`	Valor del TTL.
`-Il`	Tamaño en bytes del paquete.
`-Pi`/`-Ps`	Número de secuencia del paquete dentro de la comunicación para cbr o tcp, respectivamente.
`-Pf`	Número de veces que el paquete ha sido reenviado.

Tabla 4.1: Explicación de los *flags* que aparecen en los ficheros de trazas generados por `ns-2`.

```
# s -t 25.000000000 -Hs 0 -Hd -2 -Ni 0 -Nx 300.00 -Ny 0.00 -Nz 0.00 -Ne
-1.000000 -Nl RTR -Nw -- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 0.255 -Id -1.255 -It AODV
-Il 48 -If 0 -Ii 0 -Iv 30 -P aodv -Pt 0x2 -Ph 1 -Pb 1 -Pd 1 -Pds 0 -Ps 0 -Pss 4
-Pc REQUEST
```

Figura 4.2: Extracto del fichero de trazas resultado de una simulación utilizando el protocolo AODV sin modificación.

descriptores de eventos disponibles son: `s` (envío de paquete), `r` (recepción de paquete), `d` (paquete descartado), `f` (reenvío de paquete). Para poder distinguir en la Figura 4.1 entre las distintas líneas del extracto, se marca con # el comienzo de cada una de ellas, aunque en el fichero de trazas este símbolo no aparece. Al analizar una línea de la traza se tiene que tener en cuenta que, dependiendo del tipo de mensaje que se envía, se utilizan unos *flags* u otros. Por ejemplo, el *flag* `-P` se utiliza para los protocolos de encaminamiento y también para los mensajes de tipo ARP del protocolo MAC [19]. Sin embargo no se utiliza para los paquetes que contienen datos. Otro ejemplo significativo son los *flags* `-Pi` y `-Ps` utilizados en las comunicaciones tipo cbr (*Constant Bit Rate*) y tcp (*Transmission Control Protocol*), respectivamente, que tienen la misma función. Por lo tanto, al mencionar la Figura 4.1 es necesario nombrar a que protocolo hacen referencia los eventos representados.

Aunque no es objetivo de este Proyecto Fin de Carrera la programación de un nuevo algoritmo ni la modificación de uno presente, a continuación se presentan fragmentos del código del protocolo AODV. Se considera necesario utilizar estos fragmentos, puesto que, al iniciar el estudio, se han detectado errores en la implementación. Como valor añadido se presentan las correcciones realizadas sobre los ficheros fuente.

4.2.1. Errores detectados en el código fuente para el protocolo AODV y soluciones propuestas

Anteriormente se ha mencionado que el simulador de red `ns-2` es de código abierto. El código implementado puede contener ciertos errores o puede que no se cumplan totalmente las especificaciones de algunos protocolos. La existencia de esta sección dentro de este Proyecto Fin de Carrera se debe precisamente a la inexactitud a la hora de seguir la RFC3651 [26] del protocolo AODV.

El primer problema detectado es la inexistencia de la búsqueda expansiva en anillo que, según se explica en la Sección 3.1.4.1, tiene que seguir el protocolo AODV en la formación del camino de vuelta. En la Figura 4.2 se muestra una línea de una de las simulaciones realizadas. A continuación procedemos a explicar la información que aporta y la razón de llegar a la conclusión de que el protocolo tiene errores de implementación.

La línea presentada en la Figura 4.2 aporta información sobre el primer paquete RREQ

```
/* Various constants used for the expanding ring search
*/
#define TTL_START 5
#define TTL_THRESHOLD 7
#define TTL_INCREMENT 2
```

Figura 4.3: Extracto del fichero fuente `aodv.h` original, correspondiente a las constantes utilizadas en la búsqueda expansiva en anillo.

```
// Determine the TTL to be used this time.
// Dynamic TTL evaluation - SRD
rt->rt_req_last_ttl = max(rt->rt_req_last_ttl,rt->rt_last_hop_count);
if (0 == rt->rt_req_last_ttl) {// first time query broadcast
     ih->ttl_ = TTL_START; }
else { // Expanding ring search.
    if (rt->rt_req_last_ttl < TTL_THRESHOLD)
        ih->ttl_ = rt->rt_req_last_ttl + TTL_INCREMENT;
    else { // network-wide broadcast
        ih->ttl_ = NETWORK_DIAMETER; rt->rt_req_cnt += 1;
} }
```

Figura 4.4: Extracto del fichero fuente `aodv.cc` original, correspondiente a las instrucciones que implementan la búsqueda expansiva en anillo.

que genera el nodo emisor al iniciar el descubrimiento de ruta. Tal y como se comentó con anterioridad, las líneas de las trazas tienen distintos campos, como son el nodo emisor, el nodo receptor, el protocolo de encaminamiento o el tiempo de vida del paquete, entre otros. En este caso, al ser el primero paquete RREQ enviado utilizando la búsqueda expansiva en anillo (ver Apartado 3.1.4.1), se esperaba encontrar el campo que indica el TTL (`-Iv`) de la siguiente manera: `-Iv 1`, indicando un tiempo de vida de un salto. Sin embargo, tal y como se observa en la Figura 4.2 marcado en color rojo, el valor asignado es `30`.

Una vez detectado el error se procede a su reparación y, para ello, se analizan los ficheros fuente del simulador presentes en la carpeta `ns-2.34/aodv` dentro de la carpeta de instalación del programa [46]. En primer lugar se revisa el fichero de cabeceras `aodv.h` que almacena, entre otras cosas, el valor de las constantes del protocolo. Una de estas constantes, TTL_START, tiene asignado el valor 5, tal y como se puede observar en la Figura 4.3 marcado en color rojo. Aunque esta situación es viable, ya que los valores definidos en la RFC3561 [26] se pueden modificar libremente, con el fin de seguir la especificación del protocolo AODV se le asigna a la constante su valor por defecto, que es 1.

Debido a que el análisis del fichero `aodv.h` fue infructífero en cuanto a la resolución del problema encontrado en la búsqueda expansiva en anillo, seguidamente se examina

```
# s -t 25.000000000 -Hs 0 -Hd -2 -Ni 0 -Nx 300.00 -Ny 0.00 -Nz 0.00 -Ne
-1.000000 -Nl RTR -Nw -- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 0.255 -Id -1.255 -It AODV
-Il 48 -If 0 -Ii 0 -Iv 1 -P aodv -Pt 0x2 -Ph 1 -Pb 1 -Pd 1 -Pds 0 -Ps 0 -Pss 4
-Pc REQUEST
# ...
# s -t 25.500000000 -Hs 0 -Hd -2 -Ni 0 -Nx 300.00 -Ny 0.00 -Nz 0.00 -Ne
-1.000000 -Nl RTR -Nw -- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 0.255 -Id -1.255 -It AODV
-Il 48 -If 0 -Ii 0 -Iv 3 -P aodv -Pt 0x2 -Ph 1 -Pb 1 -Pd 1 -Pds 0 -Ps 0 -Pss 6
-Pc REQUEST
```

Figura 4.5: Extracto del fichero de trazas resultado de una simulación utilizando el protocolo AODV con las modificaciones realizadas.

el fichero de implementación de las funciones `aodv.cc`. Con el objetivo de reducir la explicación de las pruebas realizadas modificando el mencionado fichero, en la Figura 4.4 se presenta en color rojo la línea que, al estar presente, impide que se realice la búsqueda expansiva en anillo. Por lo tanto, al eliminar la línea se consigue que el descubrimiento de ruta se realice de forma correcta y la Figura 4.5 muestra, en color rojo, el valor del TTL utilizado, que es precisamente el que se esperaba encontrar al iniciar el análisis de los ficheros de trazas. Igualmente se expone la línea correspondiente al segundo paquete generado ya que el nodo receptor no es vecino del emisor, y se observa que, efectivamente, se aplica la búsqueda en anillo con un incremento del TTL igual a 2.

Como resultado adicional a la correcta implementación del protocolo AODV en el simulador de red `ns-2` se consigue una disminución drástica del tamaño de los ficheros de trazas, debido especialmente a la reducción del número de paquetes de control enviados. Esta reducción es notable en casos donde los nodos emisor y receptor se encuentran a menos de 8 saltos ya que para encontrar un nodo situado a 8 saltos de distancia se utiliza como TTL el valor almacenado en la constante NET_DIAMETER. A modo de ejemplo, de un fichero de 1.8 MB para simulaciones de 1 salto antes de las modificaciones, se pasa a ficheros de menos de 15KB y menos de 700KB para simulaciones de 1 salto y 7 saltos, respectivamente, después de las modificaciones realizadas. Los datos son aproximados y se han obtenido realizado varias simulaciones de prueba para comprobar el tamaño de los ficheros de traza. Este efecto puede ser significativo si se desean almacenar todos los ficheros obtenidos ya que se reduce considerablemente el espacio necesario.

4.3. *Software* adicional

Además del simulador de red, durante el desarrollo de este proyecto se han utilizado programas para realizar otras tareas de igual importancia que la simulación. Entre las mismas se encuentra automatizado de las simulaciones, el análisis y la representación de

los datos y la redacción de la memoria. A continuación se menciona brevemente el *software* utilizado para las tareas comentadas con anterioridad.

4.3.1. *Software* utilizado para automatizar las simulaciones y realizar el postprocesado

Python es un lenguaje de programación de alto nivel creado por Guido van Rossum en los años ochenta. El nombre recibido se debe a las series de comedia *Monty Python's Flying Circus* como homenaje del creador a Monty Python [54].

La elección de Python como lenguaje para crear los *script*s encargados de las automatizaciones se debe a la calidad del *software* generado, la alta productividad del desarrollador, la alta portabilidad de los *scripts* y el soporte de librerías del que se dispone. Se considera que es de alta calidad debido a que el lenguaje Python fue diseñado para ser fácilmente leído y, por lo tanto, reutilizado y mantenido. Al utilizar Python se incrementa la productividad del desarrollador ya que, debido a su uniformidad, un *script* en este lenguaje ocupa típicamente entre un tercio y un quinto menos que un programa equivalente en otros lenguajes de programación. Los *scripts* editados en Python se pueden utilizar sin realizar ningún cambio en múltiples plataformas tales como: Windows, Linux o Mac OS. Finalmente, las librerías que ofrece Python fueron determinantes para su elección ya que es necesario interactuar con programas externos (el simulador de red `ns-2`) y la librería `os` ofrece precisamente esta opción [54–56].

El lenguaje de programación elegido se utiliza en multitud de productos desarrollados por importantes compañías, tales como [54]:

- *Google*: utiliza Python en el sistema de búsqueda en las páginas web.
- *Youtube*: su servicio de compartición de vídeos es mayoritariamente codificado en Python.
- NSA (*National Security Agency*): utiliza Python para criptografía y análisis.
- NASA (*Nacional Aeronautics and Space Administratio*n): utiliza Python en los programas científicos.

Por último es necesario aclarar que Python es un lenguaje de propósito general que se aplica con frecuencia en tareas de *scripting*. Es decir, no es únicamente un lenguaje de *scripting,* sino que también incluye programación orientada a objetos, conexión a bases de datos o utilización en plataformas externas, como por ejemplo Django [54,57]. Por lo tanto, para nombrar los ficheros codificados en Python se puede utilizar tanto la denominación «programa» como «*script*».

4.3.2. *Software* de análisis y representación de los datos

El *software* utilizado para analizar y representar los datos recogidos en las simulaciones es Matlab [58]. Matlab es un lenguaje de programación de alto nivel utilizado para computación técnica y ofrece un entorno interactivo para visualización y análisis de datos, entre muchas otras tareas. La adquisición de datos se puede realizar de múltiples maneras, pero lo que interesa en el desarrollo de este proyecto es el hecho de que se pueden leer ficheros escritos en texto (extensión .txt) en un formato conocido por el programador. Los datos obtenidos así se pueden representar de múltiples maneras para dar la posibilidad a una mejor interpretación de los mismos.

4.3.3. *Software* utilizado para la redacción de la memoria

Al redactar la memoria se acude al procesador de texto visual llamado LyX [59]. El programa elegido combina la potencia y la flexibilidad de TeX/LaTeX [60] con la facilidad de uso de una interfaz gráfica. LaTeX es un sistema de composición de textos y es el estándar *de facto* para redactar documentos científicos y técnicos. LyX fue diseñado para crear documentos utilizando LaTeX sin necesidad de conocer los comandos de este por lo que se facilita la tarea de redactar.

En cuanto a las citas bibliográficas, se utiliza el editor KBibTeX [61] que se encarga de mantener una base de datos bibliográfica que puede ser accedida directamente desde el procesador de texto LyX. KBibTeX utiliza internamente la herramienta BibTeX [62] que se utiliza para dar formato a las listas de referencia de los documentos escritos con LaTeX.

Capítulo 5

Estudio teórico del retardo extremo a extremo y descripción de los experimentos

Como ya se indica en el Capítulo 1, el propósito principal de este Proyecto Fin de Carrera es el estudio del retardo que experimenta un paquete de datos desde que es enviado por el nodo fuente hasta que llega al nodo destino, dentro de una red ad hoc, en función de la distancia que une estos nodos. Para alcanzar este objetivo, en primer lugar, se expresa el retardo en función de la longitud de la ruta en número de saltos, H, y en segundo y último lugar, se expresa el mismo retardo en función de la distancia Euclídea entre los extremos de dicha ruta, R. En este capítulo se explica detalladamente la formalización teórica de dicho retardo y las simulaciones que se llevan a cabo para la validación de la teoría. Para comenzar, en el Apartado 5.1 se hace un estudio del estado del arte del estudio del retardo en redes ad hoc. Seguidamente, en el Apartado 5.2, se define el modelo de red utilizado en este trabajo con el fin de conocer sus características. A continuación, en el Apartado 5.3, se deducen las expresiones teóricas por las que se rige el retardo y después, en el Apartado 5.4, se describen las simulaciones realizadas. Para concluir con el capítulo, en el Apartado 5.5, se exponen los procesos de automatización de las simulaciones y del filtrado de los datos resultantes.

5.1. Estado del arte del análisis del retardo en redes ad hoc inalámbricas

Este apartado pretende reflejar el estado del arte del retardo en redes ad hoc. En las fuentes consultadas [6,9,20,63–78] se analiza este problema desde distintos puntos de vista

y bajo diversas suposiciones. En las siguientes líneas se presentan brevemente los trabajos mencionados en orden cronológico:

- En [63], publicado en 2003, se propone un modelo analítico para calcular el tiempo medio de servicio para un paquete transmitido en una red ad hoc saturada con el protocolo IEEE 802.11 [19] en el nivel de enlace. Se le llama red saturada a aquella donde todos los nodos tienen en todo momento un paquete para transmitir. En este caso, el tiempo medio de servicio equivale al retardo sufrido por un paquete desde que el emisor lo genera hasta que es transmitido completamente por el canal físico hacia el siguiente nodo. En el análisis hecho en este artículo no se menciona el protocolo de encaminamiento utilizado, centrándose en el retardo medido a través de los parámetros del protocolo IEEE 802.11. El modelo propuesto se evalúa mediante simulaciones por ordenador. Se simulan redes de entre 8 y 56 nodos estáticos situados en un área de 20 m x 20 m y se llega a la siguiente conclusión: a menor tamaño de paquete, menor es el tiempo medio de servicio. Además, para aumentar la eficiencia de la red, el tamaño de los paquetes tiene que ser elegido acorde con el tamaño de la misma.

- En [64], publicado en 2003, se propone un nuevo algoritmo de encaminamiento para redes ad hoc que aprovecha la movilidad de los nodos para garantizar que el retardo de los paquetes sea menor que un determinado umbral. La red utilizada está compuesta por n nodos estáticos y m nodos móviles situados en un disco de área unidad. Los nodos móviles siguen un modelo de movilidad uniforme: el nodo elige una dirección de forma aleatoria y se mueve una distancia aleatoria d a una velocidad v elegida uniformemente entre el intervalo $(0, v_{máx}]$; cuando el nodo recorre la distancia d se repite el proceso anterior. Tanto el nodo emisor con el receptor se eligen entre los n posibles nodos estáticos, acotando así el retardo. De esta forma, para el protocolo de encaminamiento que proponen consiguen un retardo máximo de $\frac{2D}{v}$, siendo D el diámetro de la red.

- En [65], publicado en 2004, se modela analíticamente el tiempo de servicio en un nodo atendiendo nuevamente a las características del protocolo de nivel de enlace IEEE 802.11. En este caso, el tiempo de servicio se define como el tiempo transcurrido desde que un paquete llega a la cola de transmisión de un nodo hasta que abandona dicha cola para ser transmitido al siguiente nodo. El análisis proporciona la *pgf* (*Probability Generating Function*; representación en serie de potencias de la función de masa de probabilidad de una variable aleatoria discreta) del tiempo de servicio en función de la *pgf* del número de veces que el nodo intenta transmitir, la *pgf* del tiempo adicional inducido por otras estaciones transmitiendo simultáneamente y la *pgf* de la longitud del paquete a transmitir. Para verificar las expresiones

propuestas se realizan simulaciones por ordenador utilizando distintos modelos de red. Estos se componen de 10 y de 20 nodos distribuidos de forma aleatoria un una región rectangular de 1500 m x 1500 m. El protocolo de encaminamiento utilizado es DSR y la velocidad de transmisión de $2Mbps$. Los resultados de las simulaciones demuestran la veracidad del modelo propuesto.

- En [66], publicado en 2004, se mide mediante simulaciones el retardo medio extremo a extremo y el retardo medio de descubrimiento de ruta para el protocolo de encaminamiento AODV. Las simulaciones se realizan con 3, 16 y 40 nodos distribuidos en una zona de 1000 m x 1000 m con distintos grados de movilidad. Los resultados obtenidos muestran un incremento del retardo a medida que se aumenta el número de nodos y la movilidad.

- En [67], publicado en 2004, se modela el retardo mediante modelos epidemiológicos. Se equipara la transmisión de los paquetes de datos a la propagación de una enfermedad entre personas. En cuanto al modelo de red se distingue entre redes que solamente utilizan como máximo un nodo intermedio en las rutas (para que el paquete llegue al destino, este tiene que estar dentro del área de cobertura del nodo intermedio o, gracias a la movilidad de los nodos, acercarse en un tiempo indeterminado al nodo intermedio) y redes que utilizan tantos nodos intermedios como sean necesarios. Las expresiones propuestas en este artículo se verifican mediante simulaciones por ordenador, demostrando que las redes que permiten la utilización de más de un nodo intermedio obtienen mejores resultados en cuanto al retardo.

- En [68], publicado en 2004, se demuestra que el retardo extremo a extremo en una red ad hoc con nodos móviles escala según $\Theta(\log k)$, donde k es el número de nodos que compone la red. La notación $\Theta(\cdot)$ se utiliza para indicar que un parámetro está acotado tanto superior como inferiormente [79]. Para la validación de las expresiones propuestas se simulan redes compuestas por 64, 128, 256, 512 y 1024 nodos, sin especificar más detalles sobre el modelo de red. La principal aportación de esta publicación es que verifican que las redes ad hoc inalámbricas no escalan en cuanto a las aplicaciones de tiempo real.

- En [69], publicado en 2005, se demuestra que el tiempo necesario para el acceso al canal (tiempo que transcurre desde que la capa de aplicación genera el paquete hasta que este es enviado por el medio físico) se debe a cuatro parámetros de la red: la probabilidad de acceso al canal, la potencia de transmisión del nodo emisor, la carga de la red y la densidad de nodos de la misma. Por lo tanto, para tener un retardo acotado es necesario establecer los parámetros de la red de forma óptima. Además, se deriva una expresión para el retardo extremo a extremo y se analiza el efecto de los parámetros de la red en las fórmulas propuestas. En este artículo no se

realizan simulaciones por ordenador. Sin embargo, mediante las fórmulas propuestas se establece que existe un determinado radio de cobertura y una determinada probabilidad de acceso al canal que ofrecen un retardo extremo a extremo óptimo.

- En [70], publicado en 2005, se deriva el retardo extremo a extremo en una red ad hoc inalámbrica multisalto bajo la suposición de tráfico no saturado, es decir, que no todos los nodos tienen en todo momento disponible un paquete para transmitir. El protocolo de nivel de enlace que se analiza es, nuevamente, el IEEE 802.11 sin tener en cuenta el protocolo de encaminamiento que se utiliza en la red. Los resultados obtenidos analíticamente se corroboran mediante simulaciones por ordenador. La red simulada está formada por 120 nodos estáticos situados en un área de 1500 m x 1500 m. Cada nodo tiene un radio de cobertura de 250 m y se considera que las rutas entre los nodos son conocidas.

- En [71], publicado en 2005, se amplían los resultados obtenidos en [68], artículo publicado por los mismos autores. En este caso se demuestra que el retardo extremo a extremo escala según $\Theta\left(k \cdot \log k\right)$. Se obtiene un resultado distinto ya que utilizan un modelo de movilidad más cercano a la realidad. En cuanto al retardo, los resultados obtenidos son peores que en el caso anterior porque indican que el retardo se incrementa a un ritmo mayor.

- En [20], publicado en 2006, se caracteriza el retardo extremo a extremo medio para redes ad hoc inalámbricas multisalto. Se divide el retardo en dos partes: el retardo debido a la transmisión propiamente dicha y el retardo debido a la espera del paquete de datos en las colas de transmisión. Para obtener expresiones analíticas se modela el protocolo de nivel de enlace y las colas de transmisión que tienen los nodos. Las expresiones obtenidas se verifican mediante simulaciones. Se consideran redes de 500, 600 y 800 nodos con diferentes tasas de transmisión de paquetes. El retardo medio extremo a extremo se obtiene promediando el retardo obtenido para cada topología utilizada. Los resultados muestran que a mayor número de nodos, mayor es el retardo extremo a extremo medio medido. Este retardo también se incrementa a medida que aumenta el número de paquetes transmitidos.

- En [72], publicado en 2006, se proporcionan expresiones para el retardo medio en una red con N nodos móviles. El retardo medio, en este caso, es definido como el tiempo medio necesario para establecer una ruta entre cualquier pareja emisor-receptor. El modelo analítico muestra que a mayor número de nodos el retardo medio aumenta. No se realizan simulaciones para comprobar las expresiones obtenidas. Además demuestran que la movilidad de los nodos provoca un incremento del retardo en zonas menos pobladas y una disminución del mismo en zonas con más densidad de nodos.

- En [73], publicado en 2006, se estudia el retardo crítico y el retardo para rutas de dos saltos en redes ad hoc inalámbricas con diferentes tipos de movilidad. El retardo crítico se define como el mínimo retardo medio tolerado. Se utiliza este parámetro para decidir que modelo de movilidad proporciona un retardo menor. Se demuestra, de forma teórica, que el retardo crítico es inversamente proporcional a la distancia que recorre un nodo sin cambiar de dirección (en línea recta).

- En [74], publicado en 2006, se analiza la dependencia entre el retardo y la tasa de transmisión efectiva. Dicha dependencia se estudia tanto para redes estáticas como para redes con nodos móviles. Las redes analizadas están formadas por n nodos distribuidos en una zona del espacio de área unidad de forma toroidal, para evitar efectos de borde. En caso de tener nodos móviles, estos se desplazan de forma uniforme dentro de una pequeña zona cercana al punto inicial. El tamaño de los paquetes de datos se considera lo suficientemente pequeño como para que en un mismo intervalo temporal se pueda enviar más de un paquete. Se trata de una publicación extensa que tiene en cuenta múltiples escenarios, demostrando para cada uno de ellos la dependencia entre el retardo y la tasa de transmisión efectiva. Por ejemplo, considerando una tasa de transmisión efectiva constante (escala con $\Theta(1)$), demuestran que el retardo escala según $\Theta(n \cdot \log n)$ para redes con nodos dotados de movilidad.

- En [75], publicado en 2006, se extiende el trabajo anterior, [74], considerando un tamaño de paquetes de datos constante. Se demuestra que la dependencia entre el retardo y la tasa de transmisión efectiva es la misma, independientemente del tamaño de los paquetes de datos.

- En [76], publicado en 2006, se derivan expresiones para calcular el tiempo medio que tarda un paquete en llegar a un nodo destino fijo, siendo el resto de nodos que componen la red móviles. Además, proporcionan un algoritmo para calcular el mismo tiempo en el caso de que el destino también sea un nodo móvil. Para calcular las expresiones se tiene en cuenta la pérdida de paquetes y las retransmisiones, demostrando que elegir correctamente el intervalo de tiempo entre retransmisiones conlleva a minimizar el tiempo medio de transmisión.

- En [77], publicado en 2006, se propone un nuevo protocolo MAC, basado en IEEE 802.11, que garantiza el retardo medio de forma que las aplicaciones de tiempo real en redes ad hoc inalámbricas puedan funcionar correctamente. Se evalúa el funcionamiento del protocolo realizando simulaciones por ordenador, llegando a la conclusión de que este protocolo realmente proporciona garantías en cuanto al retardo. Además, es un protocolo simple que no introduce una gran carga computacional a los nodos y tampoco una sobrecarga en la red.

- En [78], publicado en 2008, al igual que en [74] y [75], se analiza la relación entre el retardo y la tasa de transmisión efectiva en una red ad hoc, pero utilizando otros tipos de movilidad en los nodos. Para cada escenario analizado, supuesto conocido el retardo D, se proporciona una cota superior para la tasa de transmisión efectiva.

- En [6], publicado en 2009, se propone un modelo analítico para el retardo en redes ad hoc multisalto. Para definir el modelo se atiende únicamente el protocolo de nivel de enlace y se tiene en cuenta la influencia del problema del nodo oculto (ver Sección 2.2.1). El modelo propuesto se valida mediante simulaciones por ordenador. La red simulada se extiende por una superficie de 900 m x 900 m y se utilizan distintas densidades de nodos. Se demuestra que a mayor radio de cobertura mayor es el retardo, debido al incremento de las retransmisiones provocadas por las colisiones de los paquetes.

- En [9], publicado en 2010, se propone un nuevo protocolo para el control de la potencia de transmisión de los nodos en aplicaciones sensibles al retardo. En el diseño se atiende al compromiso entre la potencia de transmisión y el retardo: a mayor potencia de transmisión menor retardo en la comunicación. Sin embargo, se produce una mayor interferencia hacia otros nodos cercanos que no pueden transmitir, reduciendo de esta forma la tasa de transmisión efectiva. Para evaluar el correcto funcionamiento del protocolo se realizan simulaciones por ordenador. La red utilizada se compone de 15 nodos estáticos distribuidos de forma uniforme en un área de 200 m x 200 m. El protocolo propuesto únicamente ofrece resultados aceptables para redes de tamaño pequeño o mediano.

En la Figura 5.1 se muestra, de forma esquemática, una clasificación de los distintos tipos de artículos resumidos en las líneas anteriores. Cabe destacar que esta clasificación solamente tiene en cuenta las fuentes consultadas, pudiendo existir distintas asunciones, parámetros y resultados que no hayan sido considerados.

Por tanto, una vez realizada un exhaustivo estudio bibliográfico, podemos concluir que este Proyecto Fin de Carrera supone una novedosa aportación al estudio del retardo extremo a extremo en redes ad hoc inalámbricas. En este trabajo proponemos un modelo analítico para el retardo extremo a extremo en redes ad hoc inalámbricas de gran escala, validando los resultados teóricos obtenidos con dos protocolos de encaminamiento con características diferentes. Este retardo extremo a extremo se estudia en función de la distancia que une los nodos emisor y receptor, siendo esta la principal aportación de este proyecto, ya que en ninguna de las publicaciones analizadas se estudia el retardo extremo a extremo teniendo en cuenta esta métrica. Además, los modelos analíticos existentes no tienen en cuenta de forma conjunta ni el protocolo de encaminamiento, ni las rutas que proporciona dicho protocolo ni las distintas clases de protocolo MAC en el estudio del

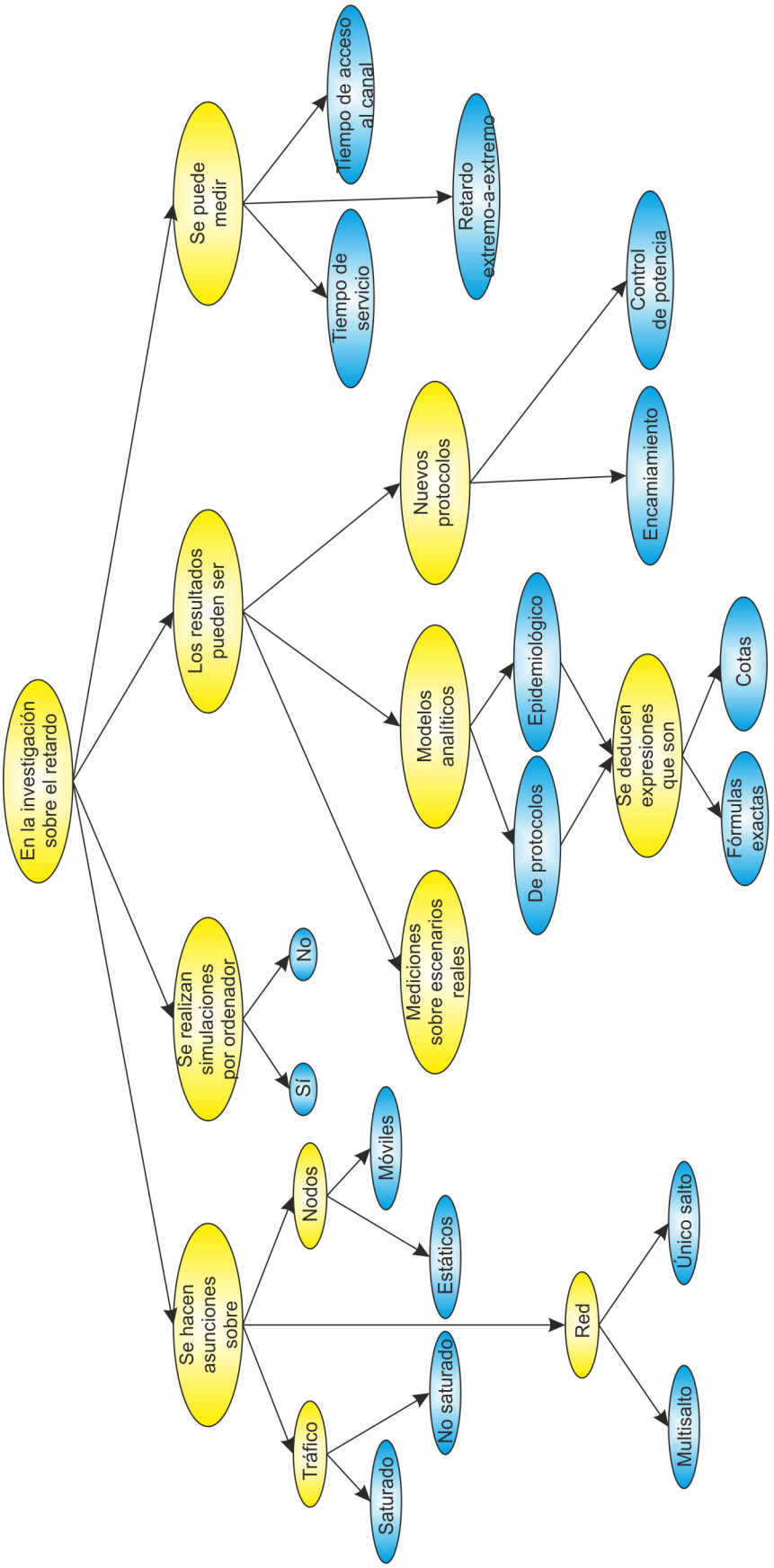

Figura 5.1: Representación esquemática del análisis del retardo en las redes ad hoc inalámbricas.

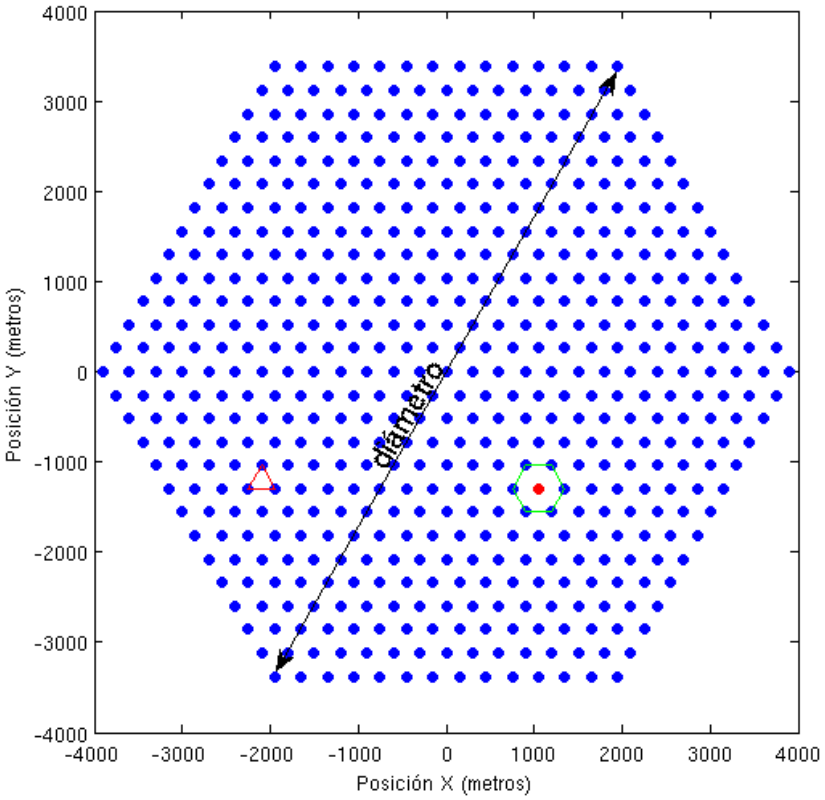

Figura 5.2: Representación del escenario de red utilizado.

retardo, es decir, siguiendo un enfoque multicapa (*cross-layer*), tal y como se realiza en este estudio.

5.2. Modelo de red

En esta sección describimos de forma detallada el modelo de red que se utiliza en este estudio. Esta misma estructura de red ya ha sido utilizada en estudios previos [5, 80, 81] del Departamento de Teoría de la Señal y Comunicaciones de la Universidad Rey Juan Carlos, donde se realiza este proyecto. La red utilizada, representada en la Figura 5.2, es una red regular, es decir, la distancia entre un nodo cualquiera y sus vecinos más cercanos es constante. Los nodos que forman la red no están dotados de movilidad, por lo que la distancia que los separa no se ve modificada en ningún momento. En esta misma figura se puede observar que el mallado de la red es triangular (los nodos se emplazan sobre un espacio bidimensional formando triángulos equiláteros, como el triángulo rojo de la Figura 5.2). Mediante línea con doble fecha se indica el diámetro de la misma, es decir, la mayor separación posible en línea recta entre dos extremos de la red. Para cumplir la condición de red de gran escala, la red tendrá un elevado número de nodos.

El área de cobertura de un nodo, determinada por la potencia de transmisión junto con la distancia entre nodos, engloba únicamente a los seis vecinos más cercanos a él (por ejemplo, los nodos representados por el hexágono verde en la Figura 5.2 con respecto al nodo de color rojo), es decir, se utiliza alcance a primeros vecinos [5, 81, 82].

Se desea asegurar la ausencia de interferencias externas, por lo que en la red hay una única transferencia de datos activa en todo momento. En dicha transferencia de datos se permite el envío de uno o más paquetes de información, pero estos serán lo suficientemente pequeños para que no sea necesaria su fragmentación.

La capa física se considera ideal, por lo que no se producen pérdidas de paquetes asociadas a errores o desvanecimientos. En cuanto a la capa de control de acceso al medio, se utilizan protocolos pertenecientes a las cuatro clases MAC (ver Sección 2.2.1). A nivel de red, se considera conocida la ruta entre el emisor y el receptor en el momento de iniciar la transferencia de datos ya que se asegura que los protocolos reactivos utilizados (AODV y DSR) han terminado el procedimiento de descubrimiento de ruta. El resto de capas que forman el modelo OSI se consideran irrelevantes en este estudio, por lo que no se atiende a las posibles configuraciones o estados que presentan.

5.3. Caracterización analítica del retardo en redes ad hoc inalámbricas

Como ya se ha comentado, el objetivo de este Proyecto Fin de Carrera es analizar el retardo extremo a extremo en redes ad hoc inalámbricas en función de la longitud de la ruta en número de saltos, H, y de la distancia Euclídea entre los extremos de la ruta, R. En la Figura 5.3 se puede observar R, distancia entre el nodo emisor C y el nodo receptor K, que establece una ruta de $H = 5 saltos$ (C-B-A-D-G-K). La longitud de esa ruta, que denotaremos como L, se puede expresar como $L = H \cdot d$, siendo d la distancia entre dos nodos adyacentes.

El resto de variables consideradas a la hora de estudiar el retardo extremo a extremo han sido los protocolos MAC y de encaminamiento utilizados y el número de paquetes que se transmiten en una comunicación. El protocolo de nivel MAC es relevante ya que, dependiendo de la clase a la que pertenece, el esquema de acceso a la red es diferente, pudiendo ocasionar más o menos retardo adicional. El protocolo de encaminamiento es fundamental en este trabajo, ya que es el encargado de proporcionar la ruta por la que viajan los paquetes, y se caracteriza por la «directividad» de las mismas, es decir, la semejanza que existe entre la ruta y la recta que une emisor y receptor. Si el protocolo de encaminamiento encuentra la ruta óptima en términos de distancia, el retardo que sufren los datos es más pequeño. Por último, el número de paquetes de la misma comunicación

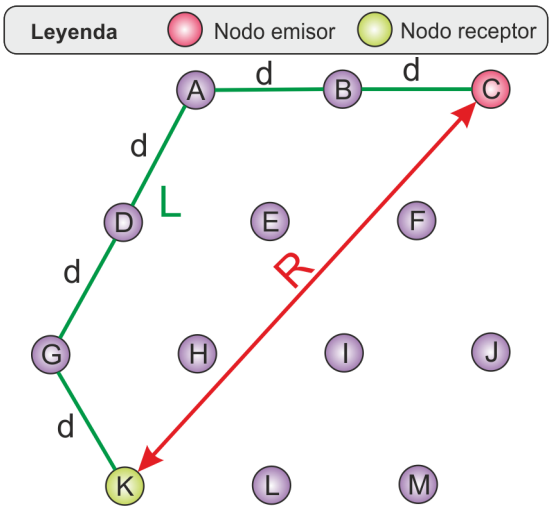

Figura 5.3: Representación gráfica de los valores escalares R, d y L sobre una red regular.

tiene una relación directa con el retardo: el tiempo de transmisión aumenta cuanto mayor es la cantidad de paquetes a transmitir.

Analizar el retardo extremo a extremo en función de H y en función de L es equivalente ya que, como acabamos de ver, la relación es directa a través de d. Sin embargo, no arroja el mismo resultado realizar el análisis en función de R o en función de L. Suponiendo que el protocolo de encaminamiento proporciona la ruta más corta (mínimo L), puede existir diferencia entre este valor mínimo de L y la distancia entre extremos, tal y como se muestra en la Figura 5.4(a). Además, por diversas circunstancias de la red, el protocolo de encaminamiento puede no proporcionar la mejor ruta en términos de distancia. En la Figura 5.4(b) se representan posibles rutas que unen emisor y receptor. Tres de ellas son rutas óptimas, ya que todas tienen el menor número de saltos necesarios para establecer la comunicación entre extremos cuando tenemos un alcance a primeros vecinos.

Expresar el retardo extremo a extremo con respecto a la distancia que une dichos extremos tiene una dificultad añadida ya que el protocolo de encaminamiento puede proporcionar rutas de diferente longitud para la misma distancia R, dependiendo del estado que presenta la red completa durante la fase de descubrimiento. Por lo tanto, la obtención de este modelo requiere un paso previo que relacione el retardo con el número de saltos que une a los nodos emisor y destino. El desarrollo del modelo teórico en función del número de saltos de la ruta se realiza en el Apartado 5.3.1 y en función de la distancia Euclídea entre extremos se realiza en el Apartado 5.3.2.

5.3.1. Retardo en función del número de saltos de la ruta

A la hora de evaluar el retardo extremo a extremo nos encontramos con dos casos bien diferenciados:

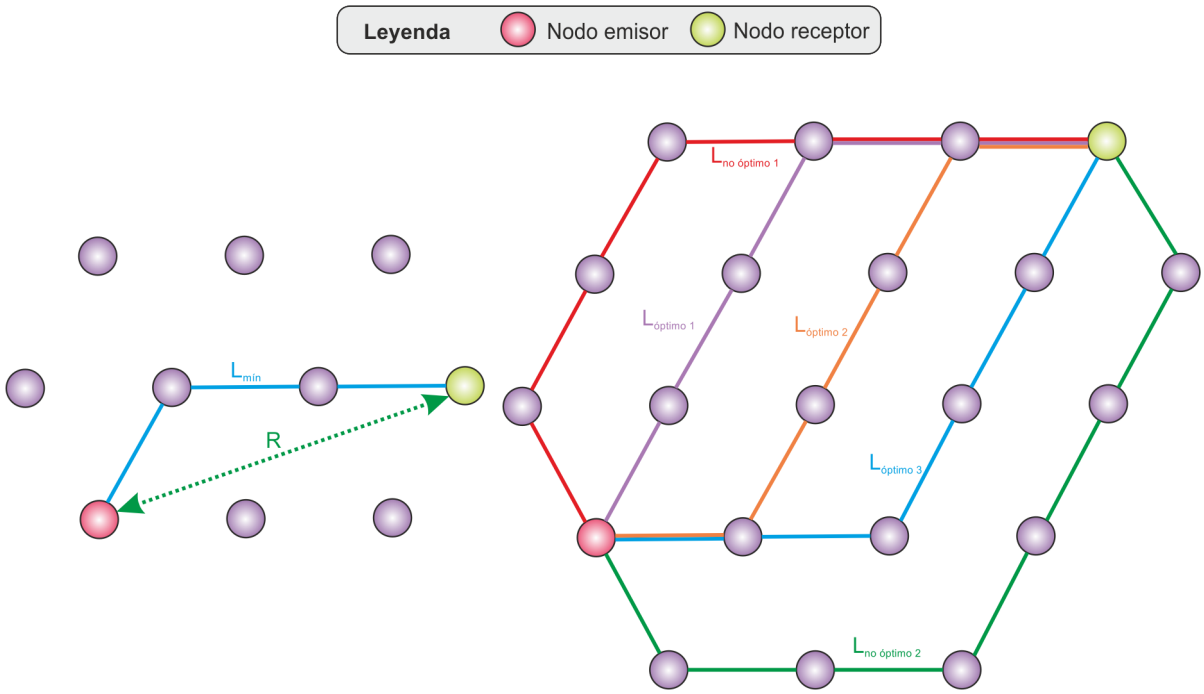

Figura 5.4: (a) Diferencia entre mínima longitud de la ruta y distancia ente dos nodos. (b) Representación de diferentes rutas entre dos nodos.

1. Se transmite un único paquete por comunicación.

2. Se transmiten dos o más paquetes por comunicación.

Estas diferencias se deben a las distintas clases MAC, ya que las características particulares de cada una de ellas aparecen cuando hay más de un paquete transmitiéndose en la red. En la Figura 5.5 se puede observar que dependiendo de la clase MAC utilizada el tiempo total necesario para la transmisión del mismo número de paquetes es distinto. En el primer caso, en la Figura 5.5(a) se representa la transmisión de tres paquetes de datos bajo la suposición de tener un protocolo MAC de clase 0, es decir, no disponer de ningún control sobre las transmisiones simultáneas. En el segundo caso, en la Figura 5.5(b) se representa la transmisión de tres paquetes de datos en el caso de tener un protocolo MAC de clase 1, es decir, cuando se prohíben transmisiones simultáneas dentro del radio del emisor. Por último, en la Figura 5.5(c) se indica la misma situación que en los casos anteriores, pero teniendo protocolos MAC de clases 2 y 3. La diferencia en el esquema de transmisión entre ambas clases aparece cuando se tiene más de una pareja intercambiando datos. Tal y como se ha comentado en el Apartado 5.2, en este trabajo se tiene una única transmisión de datos en toda la red, por lo que el mismo esquema sirve para ambas clases.

Comenzamos analizando el caso más sencillo, un único paquete transmitiéndose entre los nodos emisor y receptor, obteniendo de esta forma un resultado independiente de la clase del protocolo MAC utilizado. Por lo tanto, se define el retardo en función del número

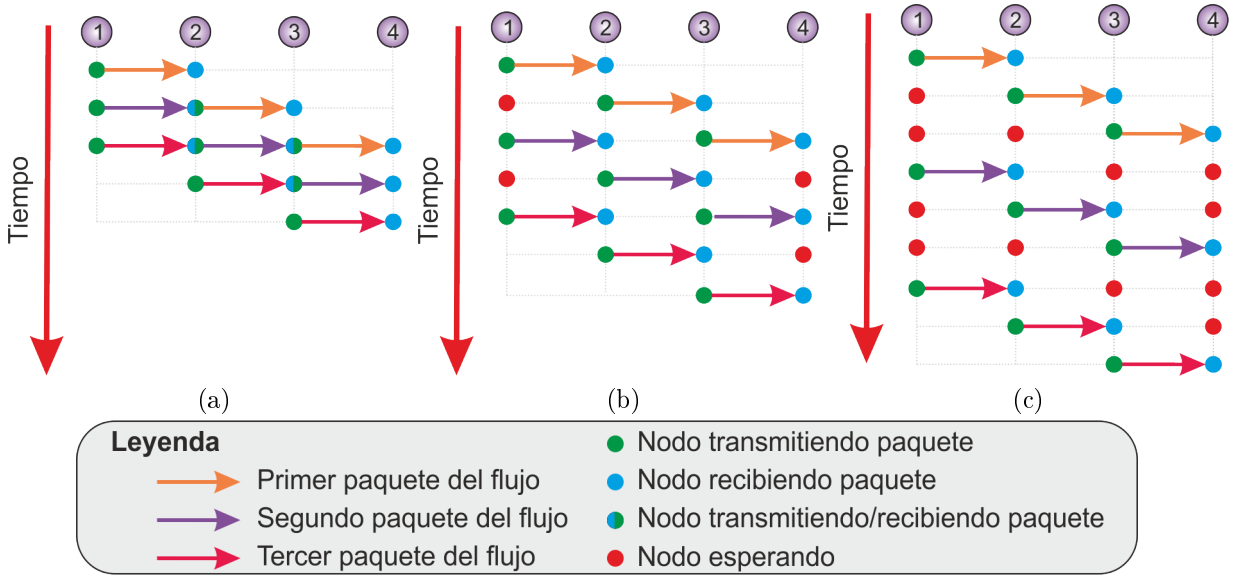

Figura 5.5: Esquema seguido por una comunicación entre dos nodos con 3 paquetes por comunicación en ausencia de protocolo MAC (a), utilizando un protocolo MAC de clase 1 (b) y un protocolo MAC de clase 2 o 3 (c).

de saltos, $\Delta t\,(H)$, como el número de saltos de una ruta por el tiempo transcurrido en cada salto (tiempo de salto, t_h). Como ya se ha comentado con anterioridad, la red utilizada es regular, por lo que un paquete de datos tarda el mismo tiempo en ser transmitido entre cualesquiera dos nodos consecutivos de la ruta. Por tanto, el retardo en función del número de saltos sigue la expresión:

$$\Delta t\,(H) = H \cdot t_h \tag{5.1}$$

siendo:

$$t_h = t_{tx} + t_{prop} \tag{5.2}$$

donde t_{tx} es el tiempo necesario para transmitir los datos de un paquete y t_{prop} es el tiempo de propagación de dichos datos por el aire. Así, la expresión (5.2) quedaría como:

$$t_h = t_{tx} + t_{prop} = \frac{N}{v_{tx}} + \frac{d}{c} \tag{5.3}$$

siendo N el tamaño del paquete expresado en *bits*, v_{tx} la velocidad de transmisión expresada en *bps* (bits por segundo) y c la velocidad de la luz. Tomando como ejemplo $N = 128bytes$, $v_{tx} = 256Kbps$ y $d = 300m$ (valores razonables para este entorno) se obtiene un valor de $t_{tx} = 4ms$ y un valor de $t_{prop} = 1\mu s$. Se puede observar que el tiempo de propagación es considerablemente más pequeño que el tiempo de transmisión. Para que ambos valores fueran comparables sería necesaria una velocidad de transmisión del orden

Paquetes / comunicación	Clase 0	Clase 1	Clases 2 y 3
1	$H \cdot t_{tx}$	$H \cdot t_{tx}$	$H \cdot t_{tx}$
2	$(H+1) \cdot t_{tx}$	$(H+2) \cdot t_{tx}$	$(H+3) \cdot t_{tx}$
3	$(H+2) \cdot t_{tx}$	$(H+4) \cdot t_{tx}$	$(H+6) \cdot t_{tx}$
...	...	...	...
n	$(H+n-1) \cdot t_{tx}$	$(H+2 \cdot (n-1)) \cdot t_{tx}$	$(H+3 \cdot (n-1)) \cdot t_{tx}$
η	$(n-1) \cdot t_{tx}$	$2 \cdot (n-1) \cdot t_{tx}$	$3 \cdot (n-1) \cdot t_{tx}$

Tabla 5.1: Cálculo del parámetro η en función del número de paquetes por comunicación para las distintas clases de protocolos MAC, considerando H saltos entre emisor y receptor.

de los *Gbps* para un N semejante al utilizado en el ejemplo, hecho que no se presenta en este tipo de redes ya que, con la tecnología actual no es usual obtener velocidades de transmisión tan elevadas en el medio inalámbrico. En conclusión, la fórmula para calcular t_h se puede aproximar por:

$$t_h \approx t_{tx} = \frac{N}{v_{tx}} \tag{5.4}$$

Conviene recortar en este punto que el modelo de red utilizado considera la capa física ideal, por lo que no hay retransmisiones de paquetes debido a errores en los datos. Por tanto no se modela el tiempo invertido en las posibles retransmisiones.

Con el objetivo de considerar la posibilidad de transmitir más de un paquete por comunicación, se añade a la expresión el parámetro η para modelar este hecho. El retardo en función del número de saltos de la ruta teniendo en cuenta la posibilidad de tener más de un paquete por comunicación se puede expresar en la forma:

$$\Delta t\,(H) = H \cdot t_{tx} + \eta \tag{5.5}$$

siendo η el parámetro que recoge la influencia de la clase del protocolo MAC utilizado. En la Tabla 5.1 se puede observar el proceso de deducción del valor de η en función del número de paquetes por comunicación, n, y la clase del protocolo MAC. Las expresiones obtenidas son:

$$\eta = \begin{cases} (n-1) \cdot t_{tx}, & \text{para clase 0} \\ 2\,(n-1) \cdot t_{tx}, & \text{para clase 1} \\ 3\,(n-1) \cdot t_{tx}, & \text{para clases 2 y 3} \end{cases} \tag{5.6}$$

5.3.2. Retardo en función de la distancia entre emisor y receptor

El siguiente paso en este proyecto es proporcionar las expresiones que relacionan el retardo extremo a extremo con la distancia Euclídea entre los nodos emisor y receptor. Para ello se hace uso de las expresiones obtenidas con anterioridad y del trabajo realizado en [5], donde se obtiene una expresión analítica que indica la probabilidad de tener H saltos en una ruta que une dos nodos situados a una distancia R, es decir, $P(H|R)$. Este modelo, conocido como «Hipótesis de Escala», tiene la siguiente expresión analítica:

$$P\left(H|R\right) = k\frac{1}{r^{\psi}} \cdot \left(\frac{H}{r^{\psi}}\right)^{-g_l} \cdot f_1\left(\frac{H}{r^{\psi}}\right) \cdot f_2\left(\frac{H}{H_{máx}^{\psi}}\right) \tag{5.7}$$

siendo k una constante de proporcionalidad, $r = {}^{R}/_{d}$ y ψ y g_l dos parámetros a ajustar para cada protocolo de encaminamiento utilizado. Por su parte, $f_1\left(x\right)$ y $f_2\left(x\right)$ se definen en la forma:

$$f_1\left(x\right) = \exp\left(-a \cdot x^{-\phi_1}\right) \tag{5.8}$$

$$f_2\left(x\right) = \exp\left(-b \cdot x^{-\phi_2}\right) \tag{5.9}$$

siendo a, ϕ_1, b y ϕ_2 nuevos parámetros a ajustar.

En este caso, se define el retardo extremo a extremo en función de la distancia que separa los nodos emisor y receptor, $\Delta t\left(R\right)$, como:

$$\Delta t\left(R\right) = \sum_{h=1}^{H_{máx}} P\left(h|R\right) \cdot \Delta t\left(h\right) \tag{5.10}$$

siendo $H_{máx}$ el TTL máximo permitido.

De esta forma, para obtener el valor de $\Delta t\left(R\right)$ se tiene en cuenta la influencia del protocolo de encaminamiento utilizado a través de $P(H|R)$, ya que el modelo de Hipótesis de Escala tiene en cuenta las características del protocolo utilizado a la hora de proporcionar valores, y del protocolo MAC a través de $\Delta t\left(h\right)$, ya que, tal y como se ha visto con anterioridad, en su desarrollo se ha tenido en cuenta la clase de protocolo MAC utilizado.

Finalmente, utilizando las expresiones (5.5) y (5.6) se obtiene:

$$\Delta t\left(R\right) = t_{tx} \cdot \sum_{h=1}^{H_{máx}} P\left(h|R\right) \cdot \left(h + \left(n-1\right)\right), \text{ para la clase 0} \tag{5.11}$$

$$\Delta t\left(R\right) = t_{tx} \cdot \sum_{h=1}^{H_{máx}} P\left(h|R\right) \cdot \left(h + 2\left(n-1\right)\right), \text{ para la clase 1} \tag{5.12}$$

$$\Delta t\left(R\right) = t_{tx} \cdot \sum_{h=1}^{H_{máx}} P\left(h|R\right) \cdot \left(h + 3\left(n-1\right)\right), \text{ para las clases 2 y 3} \tag{5.13}$$

Para validar estas expresiones de forma experimental hay que tener en mente que dependen de los siguientes parámetros:

- El protocolo MAC.
- El protocolo de encaminamiento.
- La distancia entre los nodos emisor y receptor, R.
- El número de paquetes por comunicación, n.

por lo que, a la hora de realizar las simulaciones, es necesario considerar diferentes protocolos MAC y de encaminamiento y realizar un barrido tanto en la distancia entre emisor y receptor como en el número de paquetes por comunicación.

5.4. Descripción de las simulaciones

Al realizar simulaciones con la herramienta `ns-2` se tiene que definir el escenario utilizado y configurar el simulador para atender al modelo de red definido en el Sección 5.2. Después, ha de elaborarse el *script* de usuario (fichero ejecutable donde se proporcionan todos los parámetros necesarios para una simulación) definido en la Sección 4.2 adaptado al escenario generado con anterioridad para después poder realizar las simulaciones necesarias. En las siguiente subsecciones se exponen de forma detallada los pasos del proceso mencionado.

5.4.1. Generación del escenario

En la Sección 4.2 se menciona que en el *script* de usuario se indica tanto la posición de los nodos en el espacio como las parejas de nodos que establecen una comunicación. Estos datos se pueden incluir directamente en el *script* de usuario pero, debido a la gran cantidad de nodos presentes, se utilizan ficheros de texto externos. El primer fichero, al que denominaremos «fichero de posiciones», almacena las coordenadas cartesianas de los nodos y el segundo, llamado «fichero de parejas», indica los nodos que forman las parejas fuente-destino.

El fichero de posiciones utilizado es el mismo que en [80] y la estructura del mismo se puede observar en la Figura 5.6(a). Cada nodo necesita dos líneas para tener identificada su

```
Coordenada X del nodo 1
Coordenada Y del nodo 1
Coordenada X del nodo 2
Coordenada Y del nodo 2
Coordenada X del nodo 3
Coordenada Y del nodo 3
...
Coordenada X del nodo 547
Coordenada Y del nodo 547
```

(a)

```
300.0
0.0
150.0
259.80
-150.0
259.80
...
0.0
0.0
```

(b)

Figura 5.6: (a)Estructura del fichero de posiciones. (b) Primeras líneas del fichero de posiciones utilizado.

```
Identificador del nodo emisor de la pareja 1
Identificador del nodo receptor de la pareja 1
Identificador del nodo emisor de la pareja 2
Identificador del nodo receptor de la pareja 2
Identificador del nodo emisor de la pareja 3
Identificador del nodo receptor de la pareja 3
...
Identificador del nodo emisor de la pareja N
Identificador del nodo receptor de la pareja N
```

(a)

```
57
9
376
35
59
211
...
414
103
```

(b)

Figura 5.7: (a) Estructura del fichero de parejas. (b) Ejemplo de un posible fichero de parejas.

posición: la primera línea indica la coordenada en el eje X y la segunda línea la coordenada en el eje Y. En la Figura 5.6(b) se muestra un extracto del fichero de posiciones utilizado en este proyecto. Las posiciones indicadas en el fichero de posiciones sitúan los nodos de tal forma que la red que forman es regular, igual a la definida en el Apartado 5.2.

El formato que tiene el fichero de parejas se representa en la Figura 5.7(a). Para identificar cada una de las parejas se utilizan dos líneas: la primera de ellas identifica el emisor y la segunda indica el receptor. A modo de ejemplo, en la Figura 5.7(b) se muestra un extracto de un posible fichero de parejas.

5.4.2. Configuración de la herramienta de simulación de red `ns-2`

La versión del simulador de red `ns-2` utilizada para la realización de las simulaciones es la versión 2.34, última versión estable a la fecha de inicio de este proyecto [83]. Durante la fase de familiarización con la herramienta de simulación de red se detecta la necesidad de modificar los parámetros que presenta el simulador de red en cuanto a la configuración de

los protocolos de red utilizados. En la RFC3561 [26], para AODV, y en la RFC4728 [25], para DSR, se indica que las constantes asociadas a los respectivos protocolos se pueden modificar para adaptarlas a las necesidades de cada red en particular.

Aunque las características concretas del modelo de red utilizado se detallarán en el Apartado 5.4.3, llegados a este punto necesitamos aclarar que el diámetro de la red, representado en la Figura 5.2, es 26. Por lo tanto, a la hora de realizar las simulaciones tenemos que asegurar que los nodos situados en los extremos pueden alcanzarse. El simulador `ns-2` permite modificar los parámetros que fijan el número máximo de saltos que puede recorrer un paquete mediante la constante MAX_SR_LEN para DSR y NET_DIAMETER para AODV.

En primer lugar, la constante MAX_SR_LEN se define en el fichero `ns-2.34/dsr/hdr_sr.h` dentro de la carpeta de instalación del programa [46], y presenta inicialmente el valor 16. En consecuencia, se asigna el valor 29 a la constante y seguidamente se compilan de nuevo los ficheros fuente para poder realizar las simulaciones con la configuración deseada.

En segundo lugar, la configuración que presenta el protocolo AODV en cuanto a la longitud máxima de las rutas es válida para la red utilizada en este proyecto, ya que la constante NET_DIAMETER está fijada al valor 30. Aunque no presenta el valor por defecto, mencionado en la Sección 3.1.3, se decide no modificarlo ya que la RFC3561 [26] permite configurar las constantes y en ningún caso será necesario un TTL tan alto. La constante NET_DIAMETER está definida en el fichero `ns-2.34/aodv/aodv.h` dentro de la carpeta de instalación del programa [46].

En el caso del protocolo AODV se han mencionado con anterioridad, en la Sección 4.2.1, varias modificaciones realizadas en su código original. Dichos cambios no se han considerado como configuración y no están incluidos en esta sección ya que con ellos se ha conseguido un correcto funcionamiento del protocolo, no una modificación de una variable.

5.4.3. Edición del *script* de usuario

El *script* de usuario, definido en la Sección 4.2, se utiliza para la configuración del escenario de red que se utilizará en todas las simulaciones. Este fichero se utiliza como argumento en la ejecución del simulador de red `ns-2`. Por lo tanto, este apartado tiene la misión de mostrar la forma de codificar en `OTcl` el escenario de red utilizado. La estructura secuencial del *script* se presenta a continuación:

1. **Definición de opciones**. Se indican diversas características de los nodos y del canal de transmisión. En la Tabla 5.2 se exponen los valores utilizados con un

Opción	Valor	Significado
adhocRouting	*AODV o DSR*	Protocolo de encaminamiento utilizado.
macType	*Mac/802_11*	Protocolo MAC utilizado.
antType	*Antenna/OmniAntenna*	Tipo de antena.
propType	*Propagation/FreeSpace*	Modelo de propagación radio. En este caso es el espacio libre.
phyType	*Phy/WirelessChannel*	Tipo de la interfaz de red.
channelType	*Channel/WirelessChannel*	Tipo de canal utilizado. En este caso inalámbrico.
stop	*200*	Duración máxima de la simulación en segundos.
tr	*fichero_trazas_ejemplo.tr*	Nombre del archivo de trazas.
*Pt_*	*0.1*	Potencia de transmisión utilizada en Watios.

Tabla 5.2: Valores utilizados en la configuración del escenario de simulación en `ns-2`.

breve comentario sobre su significado. En primer lugar, se indica el protocolo de encaminamiento utilizado, en nuestro caso AODV o DSR. En segundo lugar, se indica el protocolo MAC. Conviene resaltar que, aunque en el manual del `ns-2` [46] se indica que hay varios tipos de protocolos MAC, únicamente hemos conseguido realizar simulaciones con `Mac/802_11` que se corresponde al estándar IEEE 802.11 [19]. Por lo tanto, las expresiones propuestas se pueden verificar solamente para protocolos MAC de Clase 2 [18]. A continuación, se definen parámetros como el tipo de antena, el modelo de propagación, la interfaz de red de los nodos y el tipo de canal utilizado. Seguidamente se fija el tiempo máximo que dura la simulación para el que se elige un valor de 200 segundos ya que durante la fase de familiarización con el simulador se observa que las simulaciones tienen una duración menor. Conviene destacar que una vez finalizado el envío de paquetes el simulador se detiene aunque no se hayan alcanzado los 200 segundos fijados como tiempo máximo. Para terminar con la definición de opciones, se define el nombre del fichero de trazas resultante, la salida que devuelve `ns-2`, y se asigna el nivel de potencia de transmisión empleado por los nodos, en este caso 0,1W.

2. **Definición de la red**. Se indica el número de nodos que compone la red y el número de parejas que realizarán una transmisión de datos. En este caso el número de nodos de la red se fija a 547, con una distancia entre vecinos de 300 metros, y el número de parejas a 2. Mediante las simulaciones interesa recoger el retardo de la transmisión de datos sin tener en cuenta el tiempo necesario para descubrir la ruta entre extremos, asegurando que no han sido afectados por ningún tipo de interferencia.. Para ello, se utiliza una primera pareja para realizar el descubrimiento de la ruta y una segunda pareja para obtener el tiempo correspondiente a la transmisión de dichos datos. Simulando la misma pareja dos veces seguidas, con suficiente separación temporal como para que haya concluido completamente el descubrimiento de la ruta, se aprovecha que los protocolos de encaminamiento mantienen activa la

ruta encontrada durante un periodo de tiempo.

3. **Configuración del formato y contenido de la traza**. El `ns-2` cuenta con dos formatos para los ficheros de traza [46]. El formato inicial es el utilizado por defecto en la simulaciones. El formato `newtrace` es la versión mejorada del anterior y está diseñado especialmente para redes ad hoc, por lo que, es seleccionado para proporcionar los ficheros de trazas en este trabajo. El simulador de red `ns-2` considera tres tipos de trazas: a nivel MAC, a nivel de red y a nivel de agente, que engloba todos los niveles superiores de la torre OSI. Con el fin de obtener información a todos los niveles se habilitan los tres tipos de trazas mediante las opciones *macTrace*, *routerTrace* y *agentTrace*, fijadas a ON.

4. **Definición del patrón de tráfico**. En este caso se indica, para cada una de las parejas que realizan la simulación, el protocolo de transporte utilizado (UDP, *User Datagram Protocol*) y el tipo de tráfico (CBR). Además, se indica el tamaño de los paquetes (128*bytes*) y se asigna el número de paquetes por comunicación, que varía entre 1 y 6.

5. **Arranque de la simulación**. Se indica que la simulación debe concluir una vez hayan terminado las comunicaciones programadas. A continuación, se arranca la simulación.

5.5. Automatización de las simulaciones y del filtrado de datos

A la hora de demostrar la veracidad de las expresiones propuestas en este estudio, se necesita suficiente representatividad estadística. Para ello, es necesario disponer de un elevado número de valores medidos del retardo. Para obtener todos los valores de R posibles en el escenario de red utilizado, se realiza un amplio barrido en el conjunto de todas las parejas posibles. En cuanto a los paquetes de datos enviados, su cantidad se varía para cada protocolo de encaminamiento.

El modelo de red definido en el Apartado 5.2 señala que solamente se simula una pareja en cada simulación para así evitar interferencias externas. La estimación del tiempo necesario para simular es de 42 semanas, contando con un único ordenador y que, dependiendo de la potencia de la máquina, el tiempo necesario para realizar una simulación oscila entre 0,5 y 2 minutos. Se hace evidente la necesidad de un programa capaz de lanzar automáticamente el *script* de usuario y de realizar un filtrado posterior de las trazas obtenidas. Este filtrado se realiza para disponer de los datos útiles en un formato predefinido para facilitar el posterior análisis de los mismos. En consecuencia se decide utilizar el lenguaje

```
set opt(mac)__;# MAC type
set opt(adhocRouting)__;# routing protocol
set opt(tr)__;# archivo de trazas
...
$cbr_(0) set maxpkts_ __
```

(a)

```
set opt(mac) Mac/802_11;# MAC type
set opt(adhocRouting) DSR;# routing protocol
set opt(tr) fichero_trazas_ejemplo.tr;# archivo de trazas
...
$cbr_(0) set maxpkts_ 1
```

(b)

Figura 5.8: (a) Extracto de la plantilla de *script* de usuario. (b) Ejemplo de *script* de usuario resultante.

de programación Python [55], considerándose adecuado por permitir la interactuación con el simulador de red `ns-2` y el análisis de los ficheros de trazas resultantes.

5.5.1. Creación de la plantilla del *script* de usuario

Considerando que el programa que realice las simulaciones tiene que tener a su disposición un *script* de usuario como el explicado en la Sección 5.4.3, se decide crear una plantilla de dicho *script* que contenga marcas que se pueden rellenar con los datos utilizados en cada simulación concreta. Una marca es una secuencia de caracteres fácilmente reconocible dentro de un fichero de texto.

La plantilla tiene que tener marcas que identifiquen las posiciones del fichero donde se definen la clase MAC, el protocolo de encaminamiento y el número de paquetes por comunicación, ya que son los parámetros a variar dentro de las expresiones definidas para el retardo en la Sección 5.3. Se decide que la marca a utilizada sea `'__'` (dos símbolos *underscore* seguidos). Además, se incluye como parámetro variable, es decir, que también llevará una marca, el nombre del fichero resultado de trazas, dejando así la posibilidad de asignar un nombre único a cada fichero resultante. En la Figura 5.8(a) se puede observar un extracto de la plantilla utilizada y en la Figura 5.8(b) un posible *script* de usuario resultante, donde se muestra en rojo los parámetros que necesariamente tienen que poder ser modificados y en azul el parámetro opcional (el nombre del fichero de trazas). Con esta plantilla se realizan todas las simulaciones necesarias para poder verificar las fórmulas definidas en la Sección 5.3.

La utilización de las marcas que indican la posición dentro del fichero de los parámetros a modificar y el hecho de que el entorno de dichos parámetros es único en cada caso,

anima a que se empleen expresiones regulares [84, 85] para sustituir `'__'` por el valor correspondiente en cada caso. Una expresión regular (o *regexp*) se puede definir como un patrón de texto y se puede utilizar para multitud de tareas en aplicaciones y lenguajes de programación, como por ejemplo: comprobar que una entrada a un programa tiene un determinado formato, reemplazar texto que encaja en un patrón por otro distinto o dividir bloques de texto [84]. Las *regexp* son la herramienta por excelencia para realizar filtros y búsquedas dentro de ficheros de texto de cualquier tamaño y están presentes en cualquier lenguaje de programación que se precie (incluyendo Python) y esto las hace adecuadas para rellenar la plantilla del *script* de usuario.

5.5.2. Edición del *script* encargado de ejecutar la simulación y de realizar un filtrado de los datos

Finalmente, para completar la automatización de las simulaciones y realizar el filtrado de los datos, es necesario editar un *script* que se encargue de estas tareas. Tal y como se acaba de mencionar, el lenguaje de programación elegido para este cometido es Python. Antes de comenzar con la explicación propiamente dicha, es necesario aclarar los símbolos que se utilizarán en los sucesivos diagramas de flujo. Los rectángulos indican acciones que se llevan a cabo, los rombos indican puntos de decisión dentro del programa y los triángulos representan ficheros. En cuanto a estos últimos, un triángulo apoyado sobre su lado más largo representa un fichero temporal mientras que un fichero apoyado sobre el vértice opuesto al lado más largo representa un fichero permanente.

El *script* creado se llama `simula_y_analiza_trazas.py`. En la Figura 5.9 se puede observar el diagrama de flujo que sigue, recibiendo como entradas por la línea de comandos los protocolos MAC y de encaminamiento a utilizar y el máximo número de paquetes por comunicación que se permiten. Igualmente, es necesario disponer de dos ficheros adicionales en la misma carpeta llamados `fichero_parejas_python.txt` y `retardo_plantilla.tcl`. El primero de ellos contiene todas las parejas que se desean simular en cada ejecución y el segundo es la plantilla para el *script* de usuario. Como salida, este *script* proporciona un fichero (llamado «fich_resultados») que contiene únicamente la información relevante para este estudio, extraída de cada traza, cuyo nombre se crea para que sea único e inconfundible con otros ficheros de resultados producidos en otras ejecuciones.

El *script*, para cada pareja de nodos proporcionada, ejecuta `ns-2` tantas veces como número de paquetes por comunicación se indica. En todas las simulaciones realizadas con una ejecución de `simula_y_analiza_trazas.py` se utilizan los mismos protocolos MAC y de encaminamiento. Con los datos utilizados y con el fichero `retardo_plantilla.tcl` se crea el *script* de usuario (el fichero `retardo_python.tcl`) utilizando *regexp* para encontrar

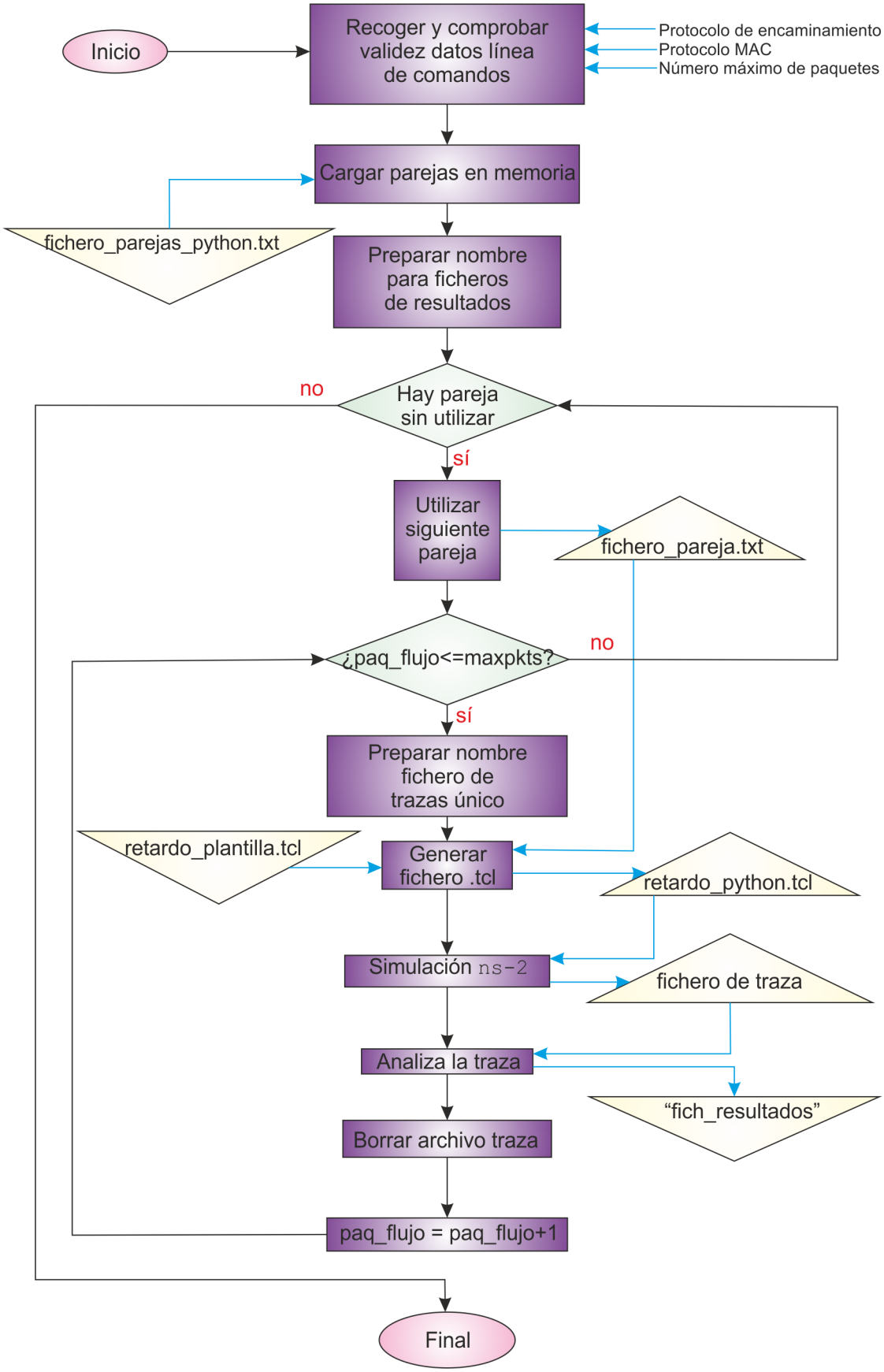

Figura 5.9: Diagrama de flujo del *script* `simula_y_analiza_trazas.py`.

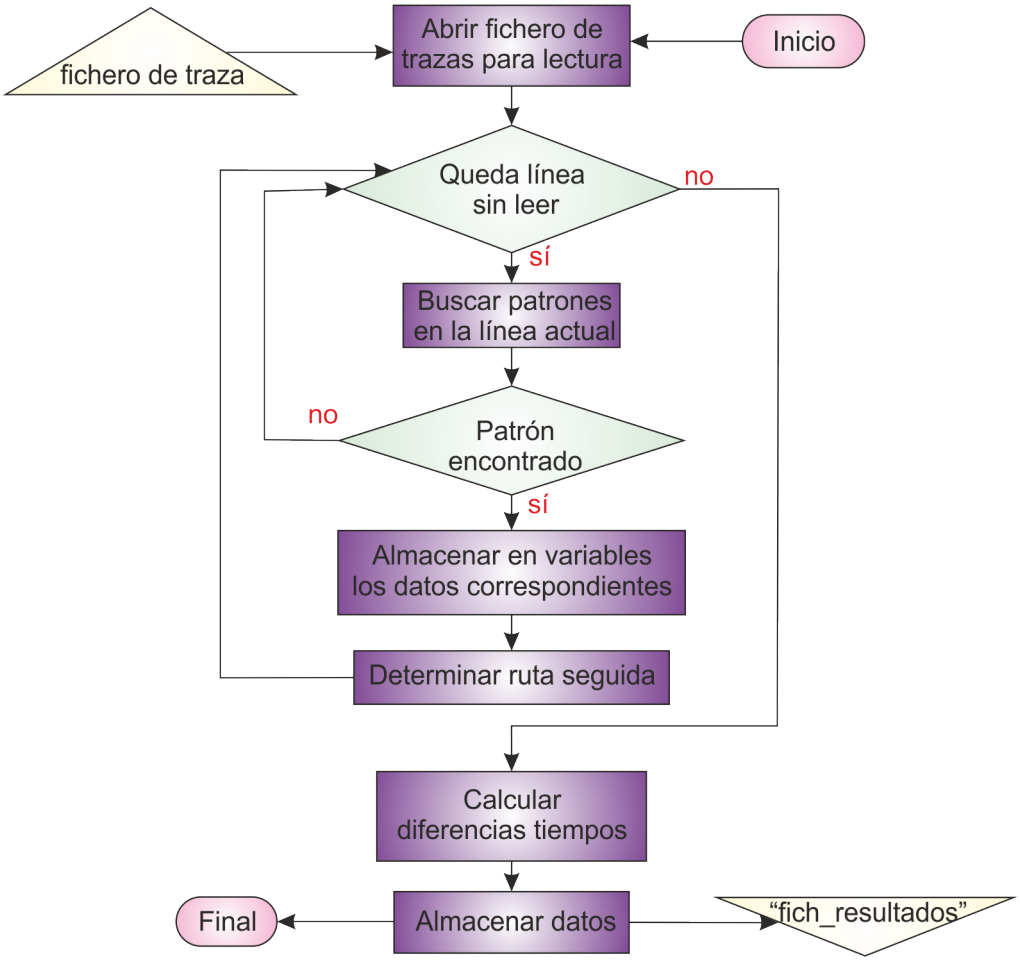

Figura 5.10: Diagrama de flujo del análisis de trazas.

```
# r -t 30.011582000 -Hs 139 -Hd 139 -Ni 139 -Nx 1350.00 -Ny 1299.04 -Nz 0.00 -Ne
-1.000000 -Nl AGT -Nw -- -Ma 13a -Md 8b -Ms bb -Mt 800 -Is 241.1 -Id 139.1 -It
cbr -Il 128 -If 0 -Ii 11 -Iv 31 -Pn cbr -Pi 0 -Pf 2 -Po 0
```

(a)

```
r -t (.*) -Hs (.*) -Hd.*AGT.* -Pn cbr -Pi (.*) -Pf (.*) -Po .*
```

(b)

```
instante_temporal = 30.011582000
emisor = 139
número_secuencia = 0
número_nodos_atravesados = 2
```

(c)

Figura 5.11: (a) Extracto de un fichero de trazas al que se le realiza el filtrado. (b) Ejemplo de expresión regular utilizada para analizar las trazas. (c) Resultado de filtrar (a) con (b).

las marcas (`'__'`) y sustituirlas por los respectivos valores. El fichero de parejas que se le proporciona al simulador no es `fichero_parejas_python.txt`, sino que se genera uno nuevo para cada simulación. El fichero generado se llama `fichero_pareja.txt` y contiene la pareja utilizada dos veces (tal y como espera el *script* de usuario explicado en la Sección 5.4.3). Además, en todas las simulaciones se genera un nombre único para el fichero de trazas para así dar la posibilidad al usuario de almacenar todas las trazas.

Después de la simulación propiamente dicha (el simulador de red se ha invocado una vez) se analiza la traza resultante. En la Figura 5.10 se puede observar el diagrama de flujo del análisis de trazas, teniendo en cuenta que es una «ampliación» de la acción llamada Analiza la traza de la Figura 5.9. En esta parte del *script* se analizan todas las líneas de la traza para extraer el instante temporal del inicio y del final de una comunicación (*flag* `-t`), el número de saltos de la ruta (*flag* `-Pf`), los índices del nodo emisor (*flag* `-Hs`) y receptor (*flag* `-Hr`), el número de paquetes transmitidos (*flag* `-Pi`) y los protocolos MAC y de encaminamiento utilizados. Todos estos valores, especialmente los instantes temporales, se obtienen a partir de la primera y la última línea a nivel AGT, que se corresponden al envío por parte del emisor del primer paquete y a la recepción por parte del receptor del último paquete, utilizado *regexp*.

En la Figura 5.11(a) se muestra, a modo de ejemplo, una línea de una de los ficheros de trazas obtenidas. En verde se indica que la línea corresponde a un evento de recepción a nivel AGT y con tipo de tráfico CBR. En azul se señalan los valores buscados y en rojo se muestra el entorno de los mismo, único para cada uno de ellos. La expresión regular utilizada para detectar las líneas de este tipo dentro de los ficheros de trazas se representa en la Figura 5.11(b). Se observa que en la *regexp* se incluyen las particularidades de la

línea (evento de recepción de nivel AGT y con CBR como tipo de tráfico) y el entorno de los valores buscados. Mediante los paréntesis, que son caracteres especiales, se indica la situación de los valores que se desean almacenar. Como a priori no se conoce el número de caracteres que tiene cada valor se utilizan, nuevamente, los caracteres especiales para las *regexp* . y * que, utilizados en conjunto, significan literalmente «cualquier número de caracteres». Para concluir con el ejemplo, en la Figura 5.11(c) se enumeran los valores obtenidos al aplicar la expresión regular de la Figura 5.11(b) a la línea de traza de la Figura 5.11(a). Los valores obtenidos son exactamente los de color azul de la línea de traza.

Con expresiones regulares semejantes a la mostrada en la Figura 5.11(b) se busca el inicio de la transmisión de datos y eventos que indican reenvíos en nodos intermedios para así calcular el tiempo transcurrido entre el inicio y el fin de la transmisión y la ruta seguida.

Una vez analizada la traza completa se añade al fichero de resultados una nueva línea conteniendo los datos obtenidos: el nodo emisor y receptor, el valor del retardo, el número de saltos de la ruta, número de paquetes, los protocolos MAC y de encaminamiento y la ruta seguida. La acción BORRAR ARCHIVO TRAZA de la Figura 5.9 se puede eliminar si el usuario desea almacenar todas las trazas generadas. Para almacenar los ficheros de resultado de trazas es necesario disponer de suficiente memoria, ya que cada uno de ellos tiene un tamaño entre $10KB$ y $8MB$ dependiendo del número de saltos de la ruta y del número de paquetes transmitidos. La ejecución del *script* finaliza cuando se han simulado todas las parejas proporcionadas para los distintos valores de paquetes en la misma comunicación.

Capítulo 6

Validación del modelo teórico

El cometido de este capítulo es comprobar la veracidad de las expresiones matemáticas definidas en el capítulo anterior. Para ello, utilizando los resultados experimentales de los dos protocolos de encaminamiento utilizados, AODV y DSR, se analiza en primer lugar la relación del retardo con el número de saltos de la ruta (Apartado 6.1). Para seguir con la estructura del apartado, en primer lugar, se realiza este análisis teniendo en cuenta dos situaciones: cuando se transmite un único paquete en cada comunicación y en el caso en el que se transmiten más de un paquete por comunicación. Teniendo en cuenta que los distintos niveles introducen cabeceras, para la realización de este análisis es necesario calcular el tamaño de los paquetes a nivel físico y modelar el tiempo adicional introducido por la torre de protocolos.

En segundo lugar, en el Apartado 6.2, se amplía el modelo teórico propuesto en el caso de considerar el retardo en función de la distancia Euclídea entre el nodo transmisor y el nodo receptor. El desarrollo de este modelo implica realizar un ajuste de los parámetros libres del modelo de Hipótesis de Escala para cada protocolo de encaminamiento utilizado.

En tercer y último lugar, en el Apartado 6.3, se realiza una comparativa entre los dos protocolos utilizados en este estudio basada en los resultados obtenidos. Para ello, se tiene en cuenta el retardo que experimenta un paquete de datos en una red ad hoc inalámbrica con las características del modelo de red utilizad en este Proyecto Fin de Carrera, diferenciando entre AODV y DSR como protocolos de encaminamiento.

6.1. Retardo en función del número de saltos de la ruta

El objetivo de este apartado es comprobar la bondad de la expresión (5.5), es decir, se analiza la relación entre el retardo extremo a extremo y el número de saltos de la ruta seguida. Para ello, se representa la media del retardo de todos los valores medidos experimentalmente para rutas de un mismo número de saltos, junto con el intervalo de

confianza al 95 %. El intervalo de confianza asegura que el 95 % de los valores medidos están dentro de sus límites [86–88]. El análisis se realiza teniendo en cuenta dos situaciones:

- Conexiones con un único paquete de datos.
- Conexiones con dos o más paquetes de datos.

En la primera de ellas los resultados son independientes de la clase del protocolo MAC utilizado, tal y como se ha comentado en la Sección 5.3.1, mientras que en la segunda la transmisión de los datos está condicionada por la política de acceso al medio de la clase MAC a la que pertenece el protocolo.

En el segundo caso, el término η utilizado es el correspondiente a la clase 2, ya que durante las simulaciones, únicamente se ha conseguido ejecutar en `ns-2` el protocolo MAC `Mac/802_11`. Este se corresponde al protocolo IEEE 802.11 utilizando CSMA/CA con reserva, es decir, es un protocolo MAC perteneciente a la clase 2. Prueba de ello son los mensajes a nivel MAC que se intercambian los nodos antes de enviar un paquete de datos. Si `Mac/802_11` utilizara CSMA/CA sin reserva, estos mensajes no aparecerían ya que, en este caso, el nodo únicamente sensa el canal y no envía mensajes para la reserva del mismo [19]. A modo de ejemplo, en la Figura 6.1 se muestra un extracto de un fichero de trazas correspondiente a los mensajes de control mencionados anteriormente. El primer evento (línea verde) corresponde al envío de un paquete generado por el nodo origen (con índice 69) desde la capa de red (nivel RTR en `ns-2`) hacia la capa MAC, para ser enviado hacia el nodo destino (con índice 51). Después se observa el envío de dos mensajes de control (segundo y cuarto evento), resaltando en azul el hecho de que son mensajes *broadcast* y en rojo que los mensajes se corresponden al protocolo MAC. El sexto evento se corresponde al envío del paquete (cbr) de datos a nivel MAC por el nodo origen hacia el nodo destino.

Al representar (5.5) es necesario definir el valor de t_{tx}. El simulador `ns-2` utiliza la constante `bandwith_` para indicar la velocidad de transmisión en el nivel más bajo de la torre de protocolos (en el nivel físico). En las simulaciones se ha utilizado el valor por defecto fijado a $1Mbps$, definido en el fichero `ns-2.34/tcl/lib/ns-default.tcl` situado en la carpeta de instalación del programa. Conociendo la velocidad de transmisión y el tamaño del paquete de datos, se calcula el tiempo de transmisión utilizando la expresión (5.4). Debido a que disponemos del valor de la velocidad de transmisión a nivel físico, la cantidad de bits transmitidos es distinta al valor fijado en el *script* de usuario (comentado en el Apartado 5.4.3) ya que los distintos protocolos añaden sus correspondientes cabeceras a medida que avanzamos en la pila de protocolos. Por lo tanto, en el Apartado 6.1.1 se realiza el cálculo del tamaño de los paquetes a nivel físico en `ns-2`. Por esta razón es necesario añadir el retardo introducido por la torre de protocolos de los nodos, modelado en la Apartado 6.1.2.

```
# ...
# s -t 32.000000000 -Hs 69 -Hd 51 -Ni 69 -Nx 150.00 -Ny -1299.04 -Nz
0.00 -Ne -1.000000 -Nl RTR -Nw -- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 69.1
-Id 199.1 -It cbr -Il 148 -If 0 -Ii 1 -Iv 30 -Pn cbr -Pi 0 -Pf 0 -Po
0
# s -t 32.000575000 -Hs 69 -Hd -2 -Ni 69 -Nx 150.00 -Ny -1299.04 -Nz
0.00 -Ne -1.000000 -Nl MAC -Nw -- -Ma 8ee -Md 33 -Ms 45 -Mt 0
# r -t 32.000928000 -Hs 51 -Hd -2 -Ni 51 -Nx 300.00 -Ny -1039.23 -Nz
0.00 -Ne -1.000000 -Nl MAC -Nw -- -Ma 8ee -Md 33 -Ms 45 -Mt 0
# s -t 32.000938000 -Hs 51 -Hd -2 -Ni 51 -Nx 300.00 -Ny -1039.23 -Nz
0.00 -Ne -1.000000 -Nl MAC -Nw -- -Ma 7b4 -Md 45 -Ms 0 -Mt 0
# r -t 32.001243000 -Hs 69 -Hd -2 -Ni 69 -Nx 150.00 -Ny -1299.04 -Nz
0.00 -Ne -1.000000 -Nl MAC -Nw -- -Ma 7b4 -Md 45 -Ms 0 -Mt 0
# s -t 32.001253000 -Hs 69 -Hd 51 -Ni 69 -Nx 150.00 -Ny -1299.04 -Nz
0.00 -Ne -1.000000 -Nl MAC -Nw -- -Ma 13a -Md 33 -Ms 45 -Mt 800 -Is
69.1 -Id 199.1 -It cbr -Il 206 -If 0 -Ii 1 -Iv 30 -Pn cbr -Pi 0 -Pf
0 -Po 0
# ...
```

Figura 6.1: Extracto de un fichero de trazas generado por `ns-2` correspondiente a los mensajes de control utilizados para la reserva del canal inalámbrico.

6.1.1. Cálculo del tamaño de los paquetes de datos a nivel físico en `ns-2`

El tamaño de los paquetes en un nivel alto de la torre de protocolos como, por ejemplo, a nivel de aplicación (AGT en la simulaciones con `ns-2`) es distinto al tamaño a un nivel más bajo (por ejemplo, a nivel MAC). Eso se debe a que todos los protocolos involucrados introducen su propia cabecera y, opcionalmente, un código de corrección de errores. A modo de ejemplo, la cabecera del protocolo IP es de 20*bytes*, sin tener en cuenta las opciones [44], mientras que la cabecera del protocolo IEEE 802.11 es de 32*bytes*, incluyendo además un código de corrección de errores de 4*bytes* [19]. No obstante, el simulador de red utilizado no cumple fielmente los estándares en cuanto a las cabeceras utilizadas. Prueba de ello es que en la capa Mac (`MAC/802_11`) correspondiente al estándar IEEE 802.11, el tamaño total de la cabecera en el `ns-2` es de 24*bytes* y no de 32 como se ha comentado con anterioridad. Este hecho se puede observar en el fichero `ns-2.34/mac/mac-802_11.h`, dentro de la carpeta de instalación del programa, donde se define, entre otras cosas, el formato de la cabecera.

Por lo tanto, para obtener el tamaño de los paquetes a nivel bajo se analizan las trazas obtenidas tras las simulaciones, ya que en el campo `-Il` se indica el tamaño del paquete expresado en bytes. En este punto hay que diferenciar entre los dos protocolos de encaminamiento que se han utilizado debido a que el método de enrutado es distinto: DSR utiliza encaminamiento en origen mientras que AODV es un protocolo salto-a-salto. El

impacto del encaminamiento en origen en cuanto al tamaño del paquete es significativo: a mayor número de saltos, más grande es el paquete a transmitir, ya que dentro de la cabecera IP, colocada en el campo de opciones, se almacena la ruta completa a seguir. Por otro lado, al utilizar el protocolo AODV, el tamaño de las cabeceras se mantiene fijo, independientemente de la longitud de la ruta, ya que la información sobre el camino a seguir se almacena en cada nodo intermedio (siguiente salto).

En conclusión, el tamaño del paquete en bytes, N, a nivel MAC en el simulador de red se rige por las siguientes reglas:

$$\text{AODV}: \ N = N_{datos} + N_{cabeceras} \tag{6.1}$$

$$\text{DSR}: \ N = N_{datos} + N_{cabeceras} + 4 \cdot (H - 1) \tag{6.2}$$

donde $N_{cabeceras} = 78 bytes$ es el tamaño de todas las cabeceras introducidas por los protocolos en el simulador `ns-2` y H representa el número de saltos de la ruta. Conviene recordar que, en el caso del protocolo de encaminamiento DSR, la dirección IP del origen no se añade a las opciones ya que está ya almacenada en un campo específico.

6.1.2. Modelado del retardo adicional introducido por la torre de protocolos

Este apartado es considerado fundamental dentro de este trabajo y debe su existencia a que los datos no son transmitidos por la capa física hacia el medio en el mismo instante en el que la capa de aplicación los genera, en el caso de enviar un mensaje, y no son entregados a la capa de aplicación en el mismo instante en el que la capa física recoge el mensaje del medio, en el caso de recibir un mensaje. Los motivos de este retardo adicional son el procesamiento de los datos que realizan todos los niveles (por ejemplo, añadir o quitar cabeceras) y, sobre todo, el mecanismo de acceso al medio del protocolo IEEE 802.11 [19, 63, 65] utilizado en la simulaciones.

El estándar IEEE 802.11 define el control de acceso al medio y varios tipos de capas físicas para ser utilizado en redes inalámbricas donde los nodos pueden ser estáticos o móviles. El mecanismo de acceso al medio que utiliza es CSMA/CA y se controla mediante la función de coordinación distribuida (DCF, *Distributed Coordination Function*). Proporciona otras funciones de coordinación para casos especiales. Sin embargo, la DCF es la que provoca un retardo variable en las comunicaciones [63, 65]. Este retardo se debe al mecanismo de *backoff* que se inicia cuando un nodo quiere enviar datos y detecta otra transmisión en curso. En ese instante se activa un temporizador que indica al nodo cuando

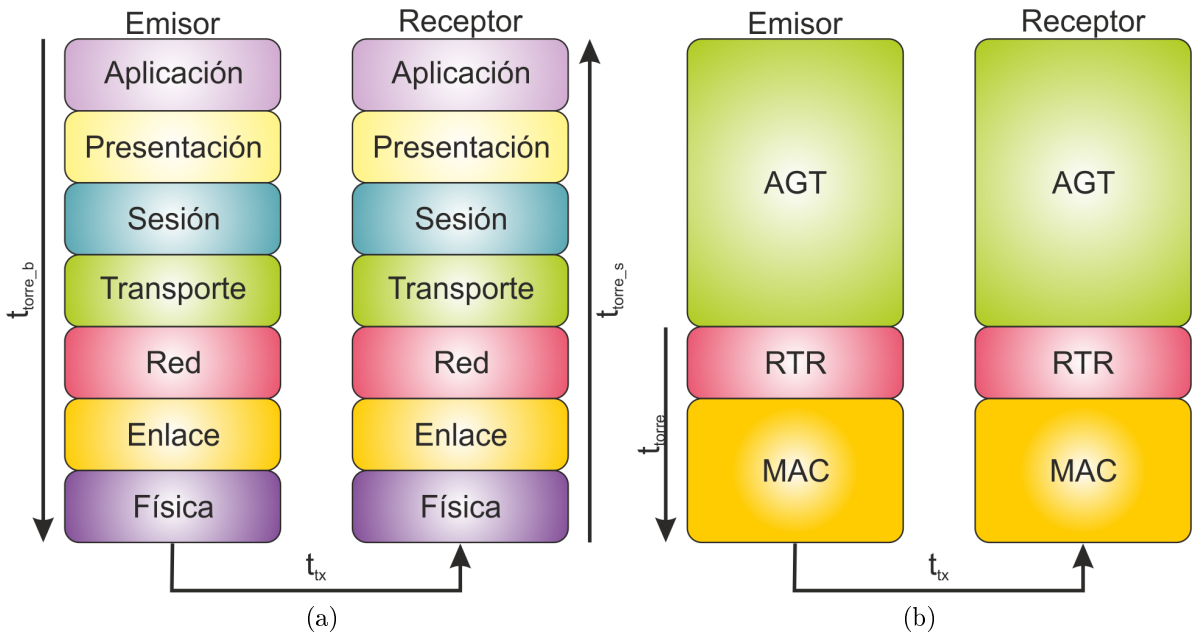

Figura 6.2: (a) Descomposición del retardo en una transmisión de un salto siguiendo la torre OSI. (b) Descomposición del retardo en una transmisión de un salto en `ns-2`.

puede volver a intentar transmitir el paquete. El valor del temporizador se elige de forma aleatoria dentro del rango $(0, W-1)$, donde W es la ventana de contienda. El valor de la ventana de contienda depende del número de veces que la transmisión de un mismo paquete no ha sido exitosa. El primer intento se asigna la constante $W_{mín}$ definida en el estándar y con cada fracaso se dobla el valor.

En primera instancia se intenta utilizar un modelado analítico de la DCF existente y por ello se recurre a las publicaciones científicas que abordan este tema. El problema encontrado es que los modelos analíticos son dependientes de la topología de la red. Por ejemplo, en [63] se considera que la red está saturada (todos los nodos tienen en todo momento un paquete para transmitir) y en nuestro caso esto no es cierto, ya solo los nodos que forman parte de la ruta que unen emisor y receptor transmiten algún paquete. Por otro lado, en [7,65,77,89,90] se proponen modelos analíticos sin especificar el modelo de red; pero en este caso no se dispone de todos los parámetros necesarios, como por ejemplo, de la probabilidad de que un paquete no sea descartado, de la probabilidad de que un paquete sea transmitido después de n reintentos o del número medio de nodos que intentan acceder al canal simultáneamente.

Al no poder utilizar un modelo analítico ya definido, se mide de forma experimental el retardo sufrido por los paquetes debido a su transición por la torre de protocolos. En la Figura 6.2(a) se puede observar la descomposición del retardo en una transmisión de un paquete en un escenario donde los nodos tienen implementada la torre OSI completa. El retardo adicional al tiempo de transmisión es la suma entre t_{torre_b} (tiempo de bajada) y t_{torre_s} (tiempo de subida). Teniendo en cuenta que se mide el retardo adicional presente

en las simulaciones realizadas por ordenador, hay que analizar la misma situación teniendo en cuenta la implementación de los protocolos en `ns-2`. En la Figura 6.2(b) se muestran los tipos de eventos presentes en los ficheros de traza, representando la equivalencia entre estos eventos y los distintos niveles de la torre OSI. Igualmente, se representa el tiempo adicional introducido por las diferentes capas de `ns-2`. En este caso, el único retardo presente se da entre la capa de red y la capa MAC en la bajada (denotado como t_{torre}). En el resto de casos, incluyendo la recepción de un paquete, el tiempo de cómputo por parte de los nodos se considera instantáneo. Para medir este retardo, se realiza una serie de simulaciones distinguiendo entre los protocolos de encaminamiento AODV y DSR. La media del retardo experimental medido es de:

AODV	DSR
$t_{torre} = 1,3 milisegundos$	$t_{torre} = 1,5 milisegundos$

En todas las simulaciones se mide el tiempo entre el envío del paquete de datos por la capa de red hasta que el mismo paquete es enviado por la capa MAC. Este escenario se puede observar en la Figura 6.1 y el tiempo medido es el transcurrido entre los dos eventos representados por las líneas coloreadas en verde.

6.1.3. Conexiones de un paquete por comunicación

Una vez concluido el anterior desarrollo, la expresión definitiva del retardo extremo a extremo en una red ad hoc inalámbrica en función del número de saltos de la ruta seguida es:

$$\Delta t\,(H) = H \cdot (t_{tx} + t_{torre}) + \eta \tag{6.3}$$

siendo η:

$$\eta = \begin{cases} (n-1) \cdot t_{tx}, & \text{para clase 0} \\ 2\,(n-1) \cdot t_{tx}, & \text{para clase 1} \\ 3\,(n-1) \cdot t_{tx}, & \text{para clases 2 y 3} \end{cases} \tag{6.4}$$

Tal y como se ha comentado en la introducción de este capítulo, la validación de (6.3) se realizará atendiendo dos situaciones diferenciadas: el caso de tener un único paquete por comunicación y el caso de tener dos o más paquetes por comunicación. El cometido de este apartado es realizar el análisis en el primer caso, es decir, teniendo un único paquete por comunicación. En este caso, el término η se anula independientemente de

H	AODV	DSR	H	AODV	DSR
1	483	316	16	2120	1383
2	736	483	17	2048	1475
3	1188	698	18	1863	1336
4	1175	891	19	1778	1299
5	1639	975	20	1562	1281
6	1492	1146	21	1524	1138
7	2310	1146	22	1303	1156
8	1648	1319	23	1252	1060
9	1940	1509	24	1071	954
10	2031	1416	25	928	985
11	2163	1442	26	791	942
12	2253	1531	27	661	921
13	2201	1533	28	520	1325
14	2236	1526	29	398	-
15	2226	1449	30	303	-
			Total	43843	32635
			Únicos	42260	30968

Tabla 6.1: Número de valores medidos de retardo extremo a extremo en función del número de saltos.

la clase de protocolo MAC que se utiliza, ya que en todos los casos está presente el término multiplicativo $(n-1)$, donde n indica el número de paquetes. La fórmula teórica resultante para el retardo en función del número de saltos en caso de tener un único paquete transmitiéndose es:

$$\Delta t\,(H) = H \cdot (t_{tx} + t_{torre}) \tag{6.5}$$

La Tabla 6.1 muestra el número de simulaciones llevadas a cabo para cada H, y, por tanto, de valores de retardo extremo a extremo. En las últimas dos filas se indica el total de las simulaciones realizadas para cada protocolo y el número de simulaciones no repetidas (distintas parejas de nodos), respectivamente. No se han excluido las parejas repetidas, por lo que pueden existir valores medidos iguales. Este hecho no representa un problema a la hora de analizar los resultados, ya que se dispone de una elevada cantidad de datos que contrarresta su influencia. Para una mejor visualización, en la Figura 6.3 se representan estos valores en forma de histograma.

A continuación se presentan los resultados obtenidos para los dos protocolos de encaminamiento analizados, AODV y DSR, a partir de las simulaciones realizadas por ordenador enfrentados a las expresiones teóricas.

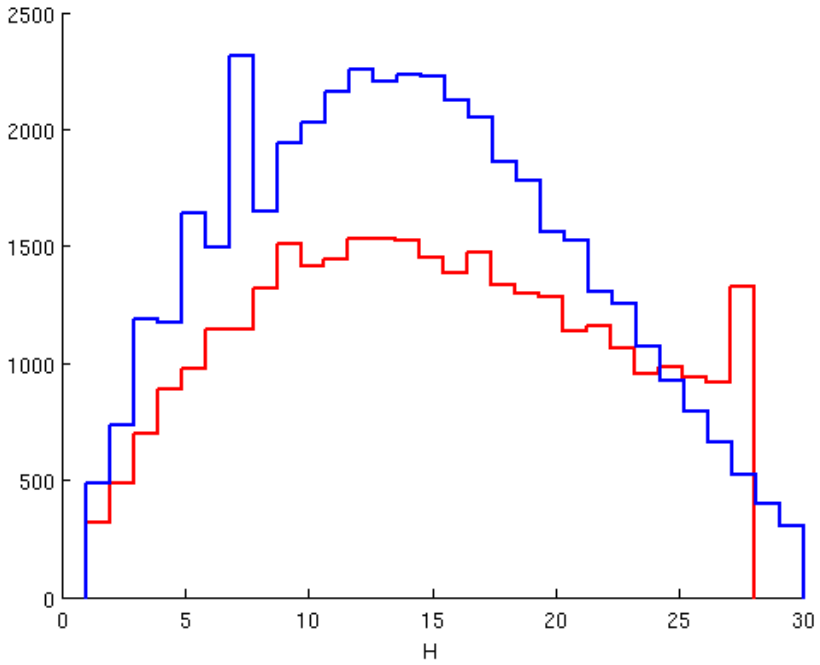

Figura 6.3: Histograma del número de simulaciones realizadas para los protocolos AODV (azul) y DSR (rojo) en función del número de saltos de la ruta.

6.1.3.1. Análisis de resultados para el protocolo AODV

Con el fin de evaluar el retardo extremo a extremo para el protocolo AODV en la Figura 6.4 se muestra mediante los puntos verdes la función teórica para valores de H comprendidos entre 1 y el TTL máximo establecido (ver Sección 5.4.2) y la Figura 6.5 representa el detalle de la anterior para observar mejor el ajuste para los valores de H pequeños. Mediante los puntos rojos se representa la media de los valores del retardo medidos en las simulaciones y los segmentos horizontales azules marcan los límites inferior y superior del intervalo de confianza al 95 %. Se considera la media experimental como buen estadístico para representar los datos gracias a la ausencia de valores atípicos. En la Figura 6.6 se representa el histograma normalizado del retardo extremo a extremo medido para las rutas de $20 saltos$. Mediante la ampliación se observa una pequeña diferencia entre la media y la mediana experimental de estos resultados. Sin embargo, esta diferencia es ínfima, demostrando así la ausencia de valores atípicos, siendo esta la razón de la utilización únicamente de la media experimental en la representación de los resultados.

En todos los casos los resultados obtenidos en las simulaciones hechas por ordenador se ajustan a la expresión teórica, es decir, la representación de la función teórica se encuentra dentro del intervalo de confianza al 95 % y cercana a la media. Esto demuestra la veracidad de la expresión (6.3) para el protocolo AODV con un paquete por comunicación.

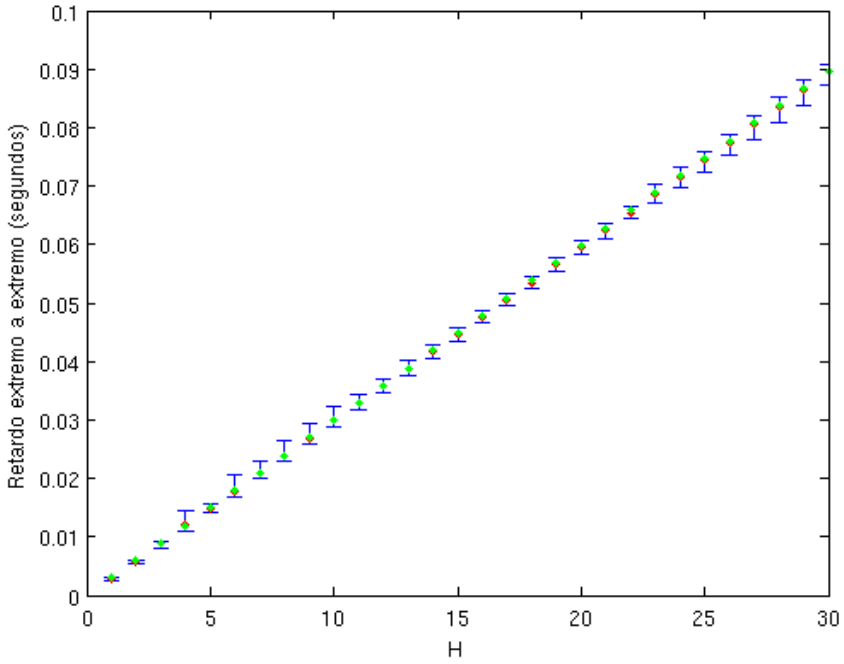

Figura 6.4: Comparación del modelo teórico (verde) frente al experimental (rojo) para el protocolo AODV con un paquete por comunicación, en función del número de saltos de la ruta.

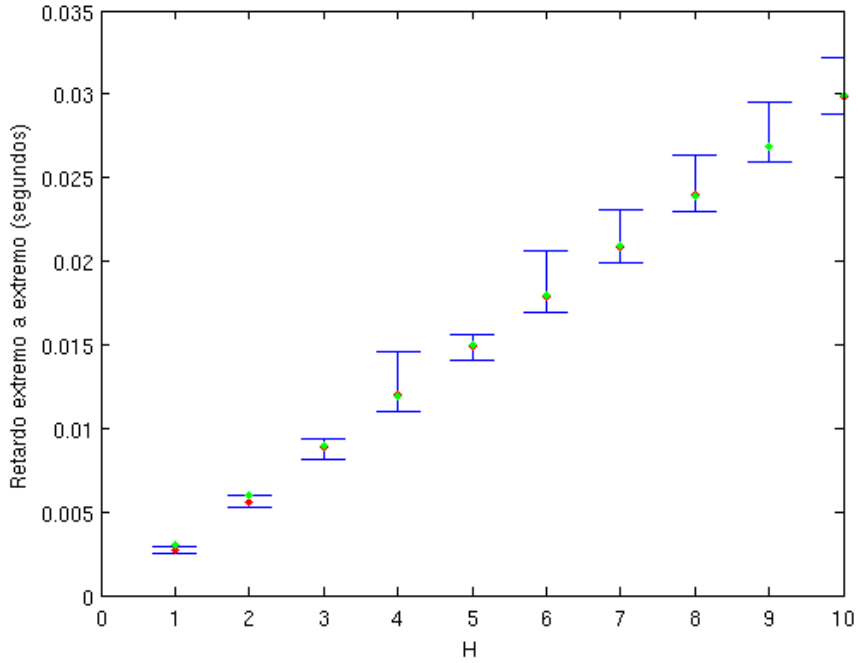

Figura 6.5: Detalle de la Figura 6.4.

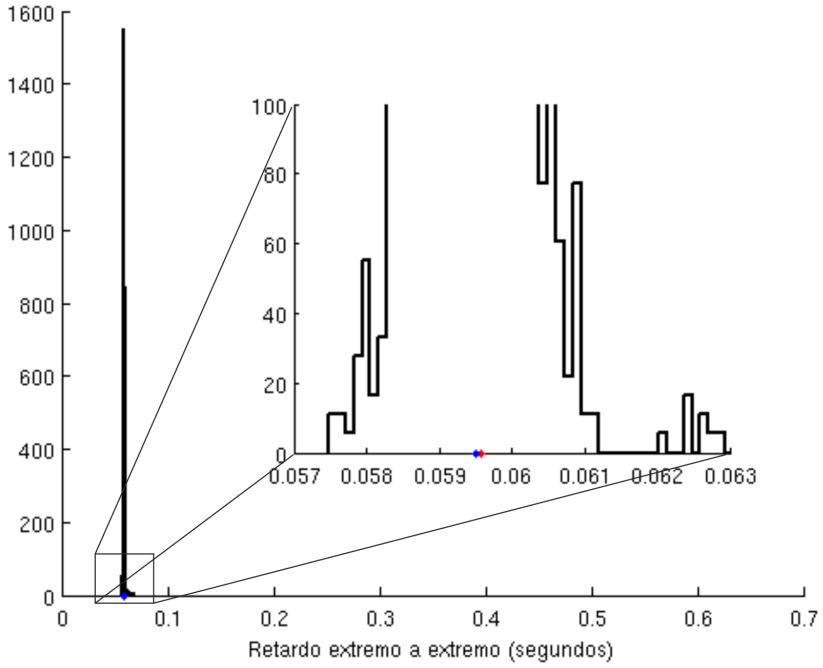

Figura 6.6: Histograma normalizado del retardo extremo a extremo medido para $H = 20$, con AODV como protocolo de encaminamiento. La media y la mediana se representan mediante un punto rojo y azul, respectivamente.

6.1.3.2. Análisis de resultados para el protocolo DSR

En la Figura 6.7 se presentan los resultados experimentales obtenidos frente al modelo teórico para el protocolo de encaminamiento DSR y la Figura 6.5 representa el detalle de la anterior para observar mejor el ajuste para los valores de H pequeños. Al igual que para AODV, los puntos rojos representan la media experimental, medida a partir de las simulaciones realizadas con `ns-2` y los puntos verdes el modelo teórico. Nuevamente, mediante los segmentos horizontales azules se indican los límites superior e inferior del intervalo de confianza al 95 % de los datos experimentales. En este caso se indica también la mediana mediante cuadrados de color rojo, ya que aporta más información que la media debido a la distribución de los datos recogidos de las simulaciones, que se representan en la Figura 6.9 $H = 20 saltos$. En este histograma normalizado se puede observar que la gran mayoría de valores se agrupan bajo el valor 0,075*segundos*, pero que existen valores atípicos a lo largo de todo el rango representado. En este tipo de distribuciones la media se ve muy afectada por la existencia de valores atípicos y la mediana aporta más información sobre la gran mayoría de los datos, tal y como se puede apreciar en la ampliación representada. Los valores atípicos también influyen en los límites del intervalo de confianza, haciendo que el límite superior se vea desplazado hacia arriba, como se puede observar en la Figura 6.7.

Los valores experimentales se ajustan a la función teórica en todos los casos. Se observa

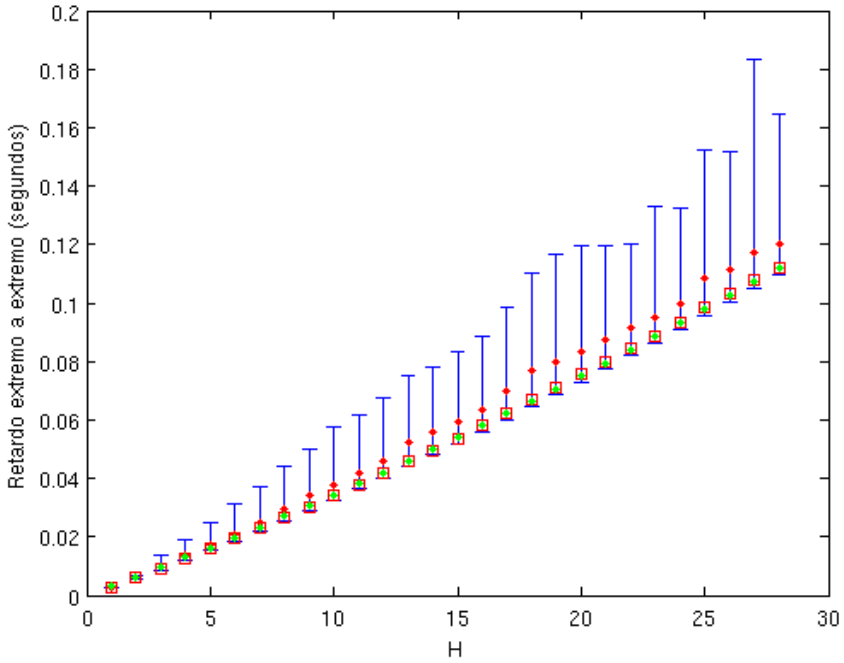

Figura 6.7: Comparación del modelo teórico (verde) frente al experimental (rojo) para el protocolo DSR con un paquete por comunicación, en función del número de saltos de la ruta. La media y la mediana experimental se representan mediante puntos y cuadrados, respectivamente.

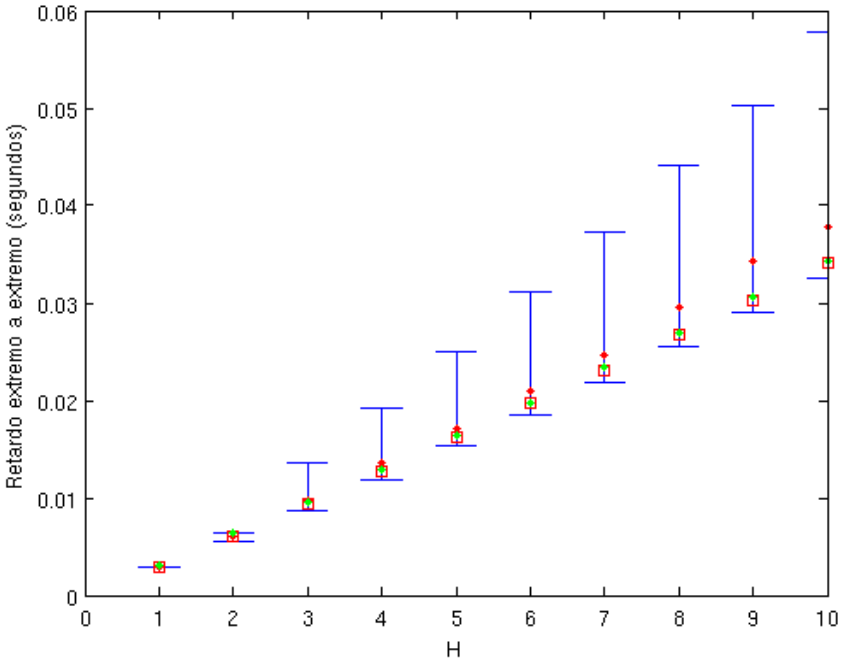

Figura 6.8: Detalle de la Figura 6.7.

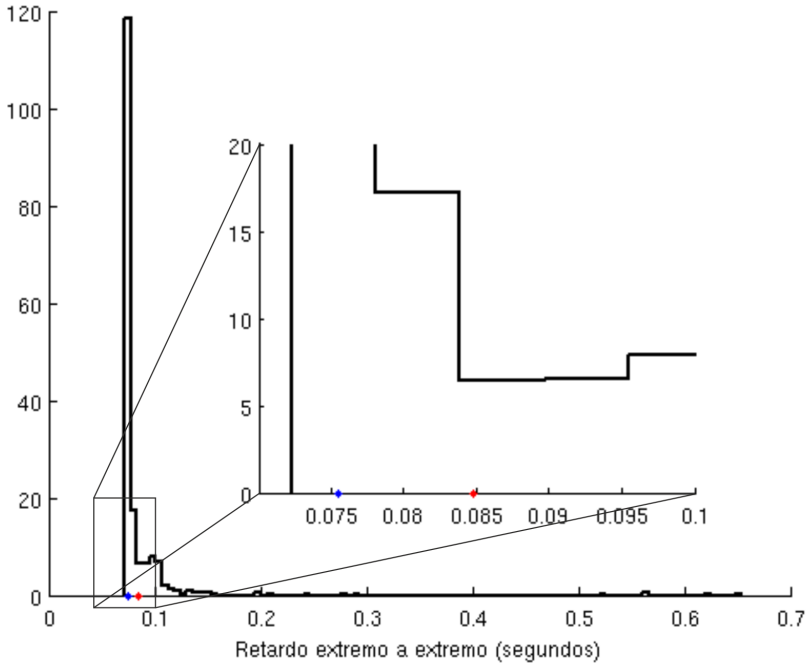

Figura 6.9: Histograma normalizado del retardo extremo a extremo medido para $H = 20$, con DSR como protocolo de encaminamiento. La media y la mediana se representan mediante un punto rojo y azul, respectivamente.

que la función teórica se aproxima mucho mejor a la mediana que a la media en todos los casos medidos por lo que hemos dicho antes. A la vista de los resultados obtenidos, la veracidad de la expresión (6.3) queda demostrada para el caso del protocolo DSR con un paquete de datos por comunicación.

6.1.4. Conexiones de dos o más paquetes por comunicación

En caso de tener más de un paquete por comunicación, estos se tienen que transmitir entre los nodos de la ruta siguiendo una planificación como la mostrada en la Figura 5.5(c). Un posible ejemplo se puede observar en la Figura 6.10, donde se representa una instantánea de una transmisión en curso suponiendo que el protocolo MAC utilizado pertenece a la clase 2. La red representada es la misma que se ha definido en el Apartado 5.2 (red regular de 547 nodos con topología triangular). Mediante las líneas de colores se representa la ruta que siguen los paquetes, que puede haber sido descubierta tanto por el protocolo de encaminamiento AODV como por DSR. Cada vez que un paquete es transmitido hacia el siguiente nodo de la ruta, en la figura se indica mediante una línea del correspondiente color que une los dos nodos que forman el salto. Dicha línea no se borra, obteniendo al final una línea quebrada para cada paquete de la comunicación. Se observa que para un determinado paquete, el paquete anterior de la comunicación está dos saltos por delante en la ruta mientras que el paquete posterior está dos saltos por detrás.

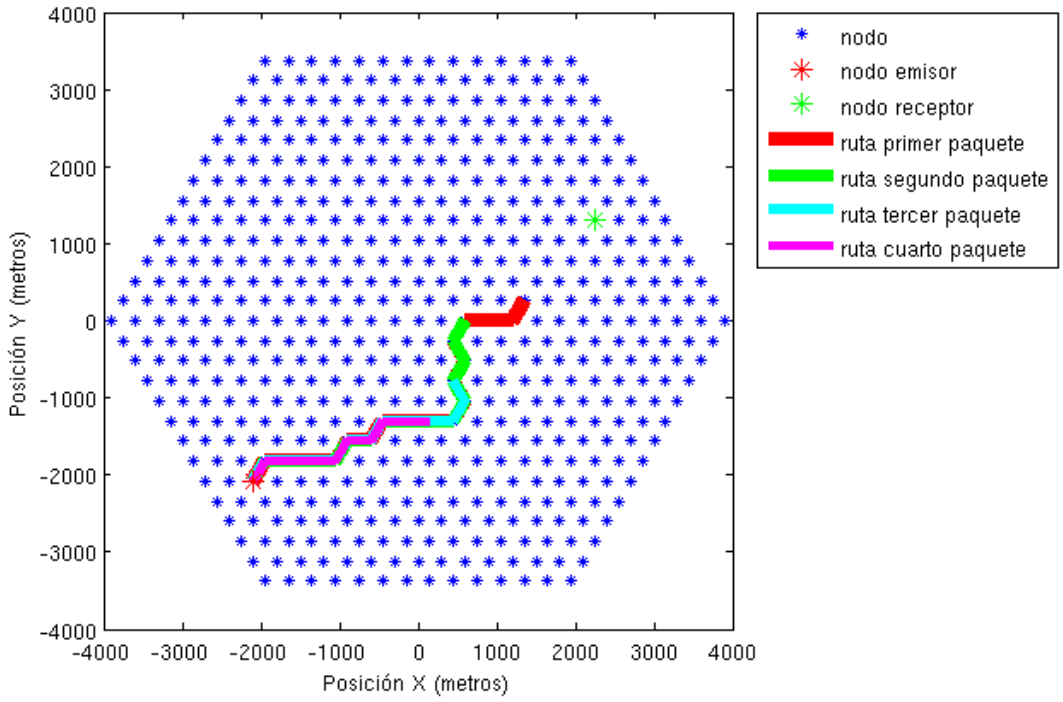

Figura 6.10: Instantánea del recorrido de los paquetes de datos en un escenario donde el protocolo MAC es de clase 2.

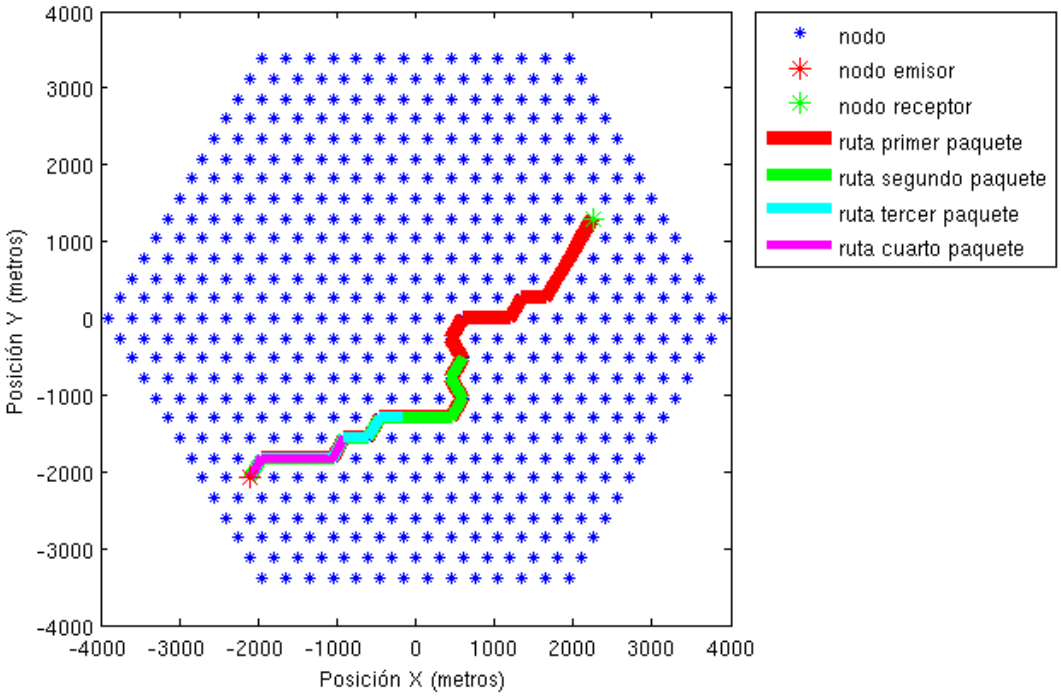

Figura 6.11: Instantánea del recorrido de los paquetes de datos en un escenario donde el protocolo MAC es de clase 2 pero no funciona correctamente (resultado obtenido en `ns-2`).

Como ya se ha mencionado con anterioridad, el simulador `ns-2` dispone de un planificador para ejecutar los eventos programados. El hecho de que el *scheduler* esté implementado mediante un único hilo de ejecución y que los protocolos no estén implementados según los estándares (en particular el protocolo IEEE 802.11) provoca que estos no funcionen según la planificación esperada. En especial, se ha estudiado con más detalle la forma que tiene el `ns-2` de organizar la cola de eventos utilizada durante la simulación. El *scheduler* utilizado en las simulaciones es el llamado *heap* [46], es decir, la cola se organiza con una estructura de tipo árbol donde la información está ordenada. La conclusión extraída después de este análisis es que la inserción o la extracción de los eventos de la cola presenta algún defecto, haciendo que la transmisión de los datos no siga la planificación esperada para un protocolo MAC de clase 2. Esta conclusión no se da por definitiva ya que este análisis no es un de los objetivos de este Proyecto Fin de Carrera y el análisis del código fuente del simulador de red `ns-2` es muy costoso en cuanto al tiempo dedicado al mismo.

A modo de ejemplo, en la Figura 6.11 se puede observar una instantánea de una transmisión en curso entre la misma pareja que la representada en la Figura 6.10. En este caso se observa que, mientras que el primer paquete ha llegado al destino, el segundo está diez saltos por detrás en lugar de solo dos, como quedaría en el caso de que el protocolo MAC funcionase conforme con la política de la clase a la que pertenece.

Según la definición hecha de las clases MAC, en este proyecto se calcula el retardo extremo a extremo mínimo, sin embargo, este no es el tiempo que proporciona el simulador de red, ya que el protocolo IEEE 802.11 implementado en `ns-2` no funciona como si perteneciera a la clase 2. Por lo tanto no se puede demostrar la exactitud de las expresiones propuestas en la Sección 5.3 en caso de tener más de un paquete por comunicación.

6.2. Retardo en función de la distancia entre emisor y receptor

El segundo paso para demostrar las expresiones de la Sección 5.3 es demostrar la veracidad de las fórmulas que relacionan el retardo con la distancia entre el emisor y el receptor. Para ello, nuevamente se realizan simulaciones con el ordenador para un protocolo MAC de clase 2 y para los protocolos de encaminamiento AODV y DSR.

Al igual que en el caso del retardo extremo a extremo en función del número de saltos de la ruta, la expresión teórica del retardo extremo a extremo en función de la distancia entre emisor y receptor también sufre una ligera modificación, ya que es necesario añadir el retardo adicional introducido por la torre de protocolos de los nodos, t_{torre}. Además, para el valor de t_{tx} se hace uso nuevamente del número de bits que contiene un paquete de datos, N, calculado en el Apartado 6.1.1. Por lo tanto, la expresión a validar es:

$$\Delta t\left(R\right)=\sum_{h=1}^{H_{max}}P\left(h|R\right)\cdot\left(t_{tx}+t_{torre}\right)\cdot\left(h+3\left(n-1\right)\right) \tag{6.6}$$

A lo largo de esta sección, debido al elevado número de distancias posibles, en todas las figuras dependientes de R, los valores experimentales se representa mediante una línea continua que une los valores obtenidos y el intervalo de confianza al 95 % se indica mediante una zona sombreada que engloba los límites superior e inferior del mismo.

6.2.1. Ajuste de los parámetros libres del modelo de Hipótesis de Escala

Como se ha comentado con anterioridad, para utilizar el modelo de Hipótesis de Escala es necesario realizar un ajuste de los parámetros libres. A modo de recordatorio, la expresión del modelo es:

$$P\left(H|R\right)=k\frac{1}{r^{\psi}}\cdot\left(\frac{H}{r^{\psi}}\right)^{-g_l}\cdot f_1\left(\frac{H}{r^{\psi}}\right)\cdot f_2\left(\frac{H}{H_{max}^{\psi}}\right) \tag{6.7}$$

con $f_1\left(x\right)=\exp\left(-a\cdot x^{-\phi_1}\right)$ y $f_2\left(x\right)=\exp\left(-b\cdot x^{-\phi_2}\right)$. Los parámetros a ajustar son ψ, g_l, a, ϕ_1, b y ϕ_2. Estas variables se tienen que determinar para cada escenario de red. En nuestro caso, al utilizar la misma red (red regular triangular de 547 nodos estáticos) para ambos protocolos de encaminamiento solo hay que realizar dos ajustes: uno para el protocolo AODV y otro para el protocolo DSR.

Los ajustes se realizan representado la $P\left(H|R\right)$ experimental y analítica, barriendo los parámetros hasta que la diferencia entre ambas es mínima. El ajuste a grosso modo se realiza mediante comparación visual de las gráficas obtenidas, para obtener una aproximación de los parámetros y observar la influencia que tienen estos en la expresión analítica. El ajuste fino se realiza minimizando el error cuadrático medio entre la $P\left(H|R\right)$ experimental y la analítica. Los valores obtenidos en ambos protocolos son:

AODV		DSR	
$\psi=1{,}010$	$g_l=8{,}5$	$\psi=1{,}005$	$g_l=7$
$a=1$	$\phi_1=100$	$a=1$	$\phi_1=100$
$b=1$	$\phi_2=20$	$b=1$	$\phi_2=20$

En las Figuras 6.12 y 6.13 se compara la $P\left(H|R\right)$ obtenida por simulación (azul) con la $P\left(H|R\right)$ analítica (rojo) para el protocolo AODV y DSR, respectivamente. No hay siempre

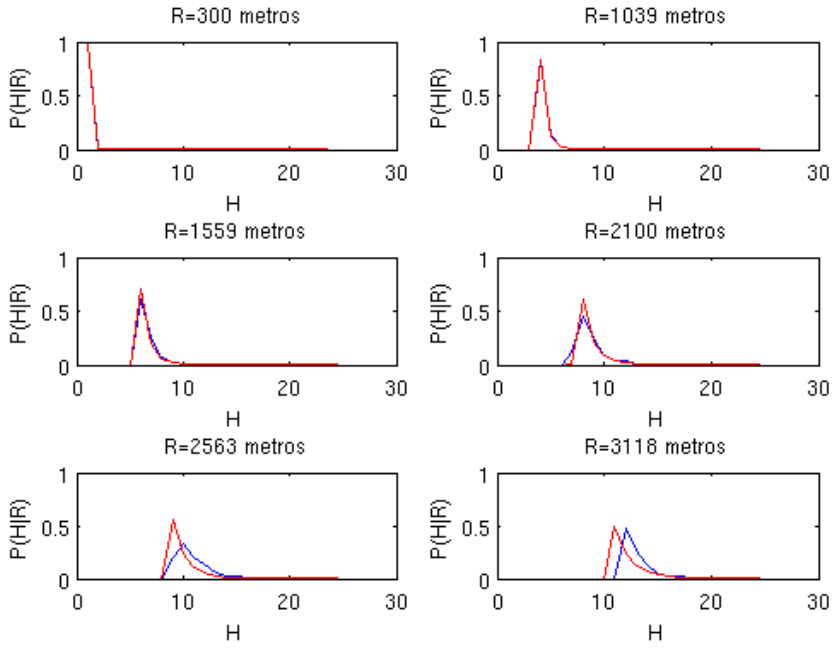

Figura 6.12: $P(H|R)$ obtenida de forma experimental (azul) y de forma analítica (rojo) para el protocolo AODV y distintos valores de R.

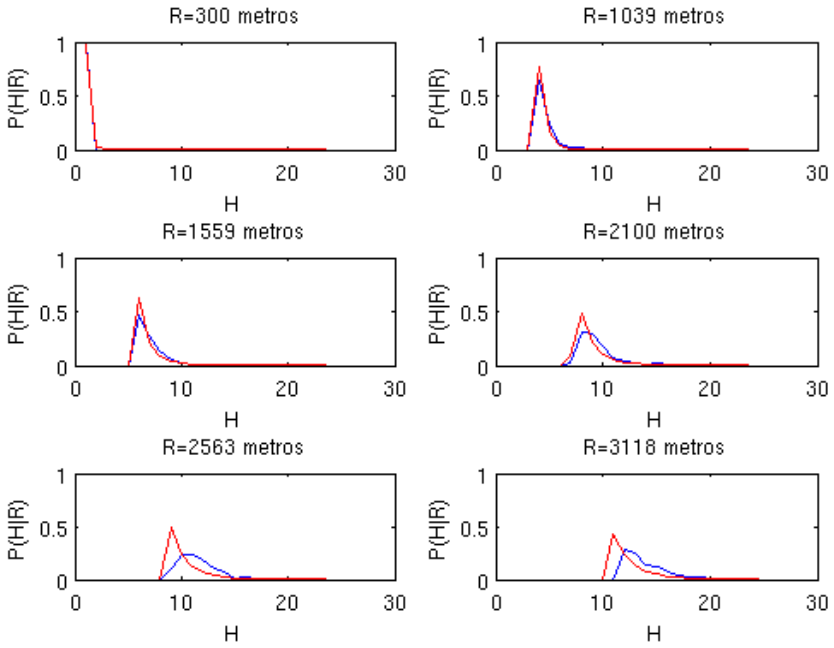

Figura 6.13: $P(H|R)$ obtenida de forma experimental (azul) y de forma analítica (rojo) para el protocolo DSR y distintos valores de R.

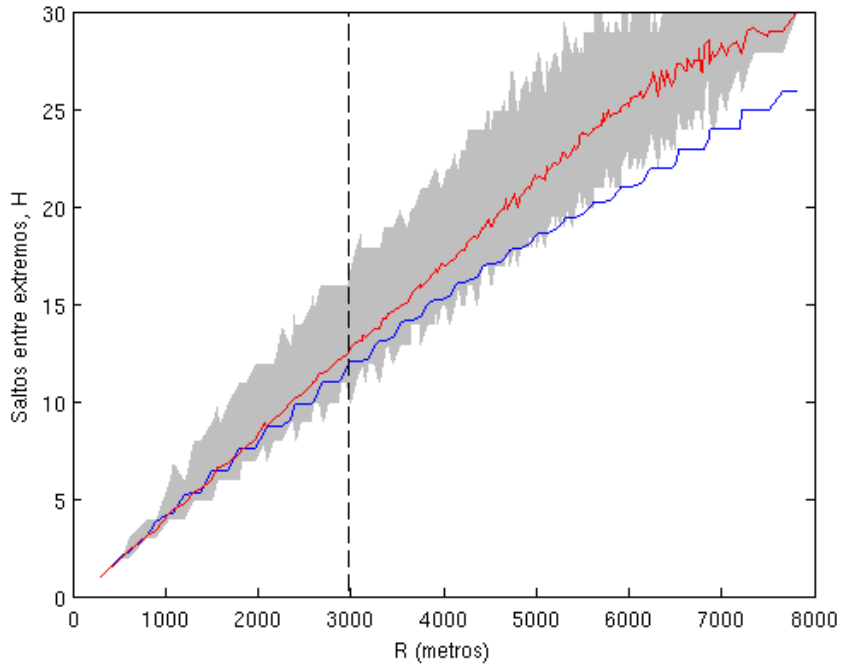

Figura 6.14: Número de saltos, H, de una ruta en función de la distancia Euclídea, R, que separa nodo emisor y receptor para el protocolo AODV, representando el modelo de Hipótesis de escala (azul) y las medidas experimentales (rojo).

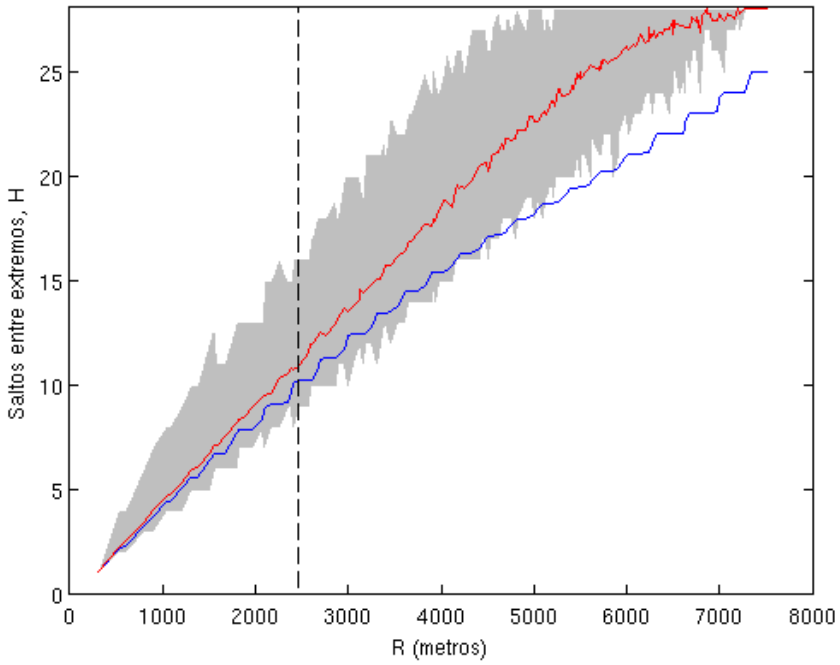

Figura 6.15: Número de saltos, H, de una ruta en función de la distancia Euclídea, R, que separa nodo emisor y receptor para el protocolo DSR, representando el modelo de Hipótesis de escala (azul) y las medidas experimentales (rojo).

la misma similitud entre los valores experimentales y analíticos ya que el modelo de Hipótesis de Escala tiene un límite superior de H (y por tanto de R) hasta el que es capaz de modelar el funcionamiento del protocolo de encaminamiento de forma adecuada. Este límite es el llamado «radio de persistencia» [5,80–82], denotado por ξ. Este parámetro es utilizado en trabajos paralelos del Departamento de Teoría de la Señal y Comunicaciones de la Universidad Rey Juan Carlos, bajo la línea de investigación de Encaminamiento en Redes Ad Hoc Inalámbricas. Aglutina los factores necesarios para evaluar la eficiencia de un protocolo de encaminamiento, como por ejemplo la movilidad de los nodos o la estructura de la red. En el caso de una red con mallado triangular con 547 nodos el valor del radio de persistencia es $\xi = 9,92 saltos$ (equivalente a $R \simeq 3000 metros$) para AODV, calculado a partir de los datos obtenidos en [82], y $\xi = 8{,}18 saltos$ (equivalente a $R \simeq 2400 metros$) para DSR, calculado en [5]. El ajuste se pierde para distancias mayores, cuando las rutas experimentales son menos directivas (se alejan más de la recta que une emisor y receptor) que las determinadas por el modelo analítico.

Se comprueba que (6.7), la expresión analítica que proporciona la probabilidad de tener H saltos en una ruta que une dos nodos separados R metros, se ajusta de forma adecuada dentro del intervalo $H < \xi$, que es la cota hasta la cual se debe garantizar la similitud entre el modelo y los datos experimentales. El ajuste realizado es adecuado incluso para distancias mayores que el radio de persistencia, concretamente hasta $R \simeq 4000 metros$ y $R \simeq 3500 metros$ para AODV y DSR, respectivamente. Esto se puede observar en las Figuras 6.14 y 6.15, donde se representa el valor medio de H con un intervalo de confianza al 95 % obtenido a partir de las simulaciones realizadas frente al modelo analítico para los protocolos AODV y DSR, respectivamente. Mediante la línea vertical discontinua se indica el radio de persistencia correspondiente a cada protocolo de encaminamiento. La tendencia seguida por el modelo de Hipótesis de Escala es que, a medida que la distancia Euclídea entre extremos aumenta, este proporciona menor número de saltos, es decir, es más optimista en cuanto a la ruta seguida.

6.2.2. Conexiones de un paquete por comunicación

Al igual que en la Sección 6.1, en primera instancia la validación de las expresiones teóricas se ha de realizar haciendo la separación entre dos casos: tener un único paquete por comunicación y tener dos o más paquetes por comunicación. Sin embargo, debido a que no se cumple la planificación necesaria para considerar que el protocolo MAC implementado en `ns-2` pertenece a la clase 2, tal y como se ha demostrado en el Apartado 6.1.4, no es posible realizar la validación en el caso de tener más de un paquete por comunicación.

Por tanto, al analizar los resultados en el caso de tener un único paquete por comunicación, el término η se anula nuevamente, ya que el valor de n es 1, y la expresión a

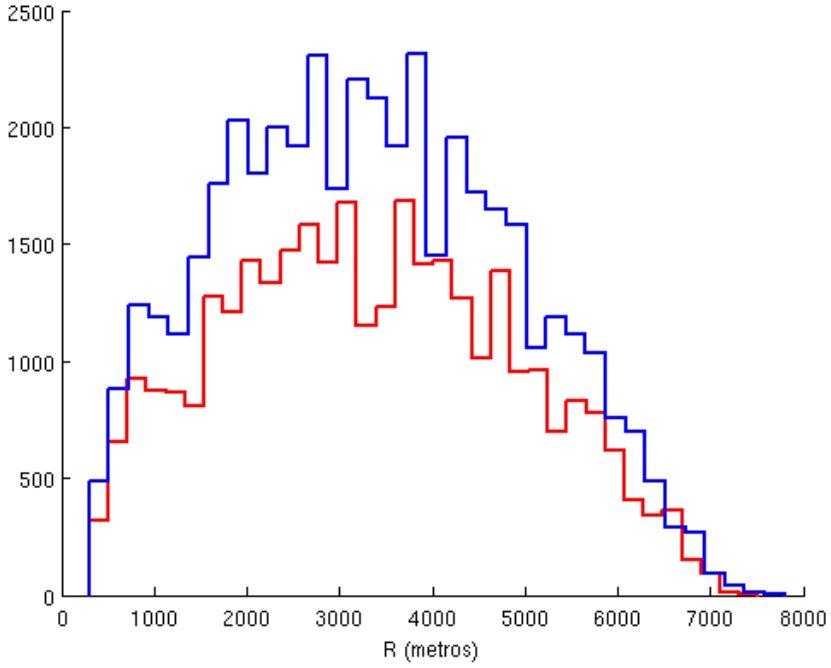

Figura 6.16: Histograma del número de simulaciones realizadas para los protocolos AODV (azul) y DSR (rojo) en función de la distancia Euclídea entre emisor y receptor.

validar es:

$$\Delta t\,(R) = \sum_{h=1}^{H_{max}} P\,(h|R) \cdot (t_{tx} + t_{torre}) \cdot h \tag{6.8}$$

Utilizando en cada caso (AODV o DSR) los valores de las constantes calculadas en el Apartado 6.2.1, a continuación se analizan los resultados experimentales obtenidos en comparación con el modelo teórico propuesto.

La Tabla 6.2 muestra la cantidad de datos para cada distancia entre emisor y receptor. Las dos últimas filas de la tabla indican el total de las simulaciones realizadas y aquellas que son únicas. Los valores proporcionados son iguales que en la Tabla 6.1, ya que en ambos análisis se utilizan los mismo datos. Nuevamente, para una mejor visualización de estos valores, en la Figura 6.16 se representa el histograma de las simulaciones realizadas para los protocolos de encaminamiento AODV y DSR en función de la distancia Euclídea entre los nodos emisor y receptor. En las siguientes subsecciones se presentan los resultados obtenidos para los dos protocolos de encaminamiento analizados, AODV y DSR, a partir de las simulaciones realizadas por ordenador enfrentados a las expresiones teóricas.

6.2.2.1. Análisis de resultados para el protocolo AODV

En la Figura 6.17 se muestran los resultados experimentales obtenidos mediante simulaciones por ordenador frente al modelo teórico. Mediante la línea roja se representa

R	AODV	DSR	R	AODV	DSR	R	AODV	DSR	R	AODV	DSR	R	AODV	DSR
300	483	316	3000	240	181	4439,59	288	184	5507,27	149	120	6510,76	52	30
519,62	425	336	3044,67	509	343	4479,96	270	212	5556,08	277	179	6538,35	44	39
600	545	317	3117,69	236	159	4500	138	96	5604,46	106	84	6579,51	78	70
793,73	815	596	3132,09	469	350	4529,9	217	179	5620,5	116	84	6600	27	20
900	425	329	3160,7	426	308	4539,82	250	174	5700	171	150	6620,42	43	36
1039,23	406	266	3174,9	418	340	4618,44	220	166	5715,77	77	49	6634	27	20
1081,67	779	610	3245	444	318	4657,25	246	162	5723,64	225	173	6681,32	38	29
1200	367	308	3300	211	154	4676,54	120	83	5747,17	123	93	6701,49	31	22
1307,67	745	559	3340,66	466	366	4686,15	261	190	5786,19	112	79	6755	47	34
1374,77	713	538	3380,83	410	315	4714,87	465	369	5793,96	104	81	6761,66	29	19
1500	373	270	3407,35	409	301	4762,35	232	193	5840,38	113	86	6781,59	51	20
1558,85	357	273	3459,77	839	627	4800	108	75	5855,77	111	71	6794,85	33	14
1587,45	721	526	3536,95	389	306	4828,04	442	316	5901,69	93	62	6814,69	12	12
1670,33	701	475	3600	193	138	4911,21	211	163	5909,31	96	77	6860,76	18	11
1800	336	261	3637,31	603	413	4938,62	180	188	5977,46	103	88	6873,86	23	15
1824,83	653	455	3649,66	395	280	4956,81	360	312	5992,5	168	154	6900	8	9
1873,5	717	492	3686,46	341	284	4993	202	144	6000	54	31	6919,54	44	26
1967,23	655	483	3747	416	281	5010,99	185	147	6022,46	159	137	6977,82	26	11
2078,46	309	237	3758,99	373	288	5046,78	200	138	6067,12	80	66	6990,71	11	8
2100	904	712	3830,14	351	269	5100	85	73	6089,33	75	58	7054,79	21	12
2163,33	587	431	3900	507	374	5117,62	164	126	6126,17	78	55	7073,9	17	12
2264,95	591	468	3923,01	350	261	5126,4	180	136	6155,49	69	43	7092,95	18	15
2343,07	567	432	3934,46	315	257	5196,15	85	80	6199,19	120	108	7168,68	9	8
2381,18	551	457	3968,63	336	252	5204,81	345	264	6235,38	43	18	7200	9	8
2400	287	220	4036,09	307	264	5230,68	190	145	6242,6	65	69	7218,73	10	8
2455,61	538	392	4058,32	327	262	5256,42	163	124	6264,18	86	55	7274,61	8	8
2563,2	557	402	4124,32	315	230	5273,52	159	126	6285,7	60	41	7354,59	8	8
2598,08	274	213	4156,92	167	103	5307,54	153	109	6300	100	77	7391,21	8	8
2615,34	549	411	4167,73	317	233	5332,92	152	110	6321,39	54	49	7500	8	8
2666,46	494	383	4200	460	339	5400	75	41	6349,8	56	41	7517,98	8	8
2700	272	169	4232,02	301	242	5408,33	156	97	6385,14	57	33	7654,41	8	8
2749,55	534	404	4253,23	314	211	5424,94	141	90	6413,27	49	48	7800	8	-
2861,82	1008	699	4326,66	296	200	5458,02	164	138	6455,23	55	49	Total	43843	32635
2893,1	520	386	4357,75	267	209	5474,49	156	132	6489,99	48	29	Únicos	42260	30968
2954,66	464	337	4419,28	562	409	5499,09	149	96	6496,92	115	89			

Tabla 6.2: Número de valores medidos de retardo extremo a extremo en función de la distancia entre emisor y receptor.

la media experimental para cada valor de R simulado y mediante la línea verde se indica el valor del retardo obtenido a partir de la teoría para todos los valores de R posibles dentro del modelo de red. La línea vertical representa el radio de persistencia para el protocolo AODV. En la Figura 6.18 se muestra el detalle para los valores de R pequeños, correspondientes a $H < \xi$.

La curva teórica no se ajusta en todo momento a los resultados experimentales, como ya se ha comentado, al contrario de lo obtenido al representar el retardo en función del número de saltos en el Apartado 6.1.3.1, que ajustaba la H. El desajuste se debe, principalmente, a que el modelo de Hipótesis de Escala no es adecuado para distancias mayores que el radio de persistencia (ver Apartado 6.2.1). Para $H > \xi$ el modelo de Hipótesis de Escala proporciona, por norma general, mejores resultados (el retardo es menor) en cuanto al número de saltos de una ruta que los que se obtienen mediante simulaciones. Por ejemplo, para un valor de R fijo, el modelo teórico proporciona, por norma general, un valor de H menor que el obtenido en las simulaciones. Como el retardo extremo a extremo tiene una relación directa con el número de saltos de la ruta, para un mismo valor de R, el modelo analítico estima un menor retardo que el medido en las simulaciones.

Al representar la Figura 6.18 se observa poca variación del retardo para $R \lesssim 520 metros$. Esto se debe a que no todos los valores de R son posibles en la red utilizada y, para el caso de R pequeñas, se nota especialmente este efecto. En la Figura 6.19 se representan las posibles R menores de $600 metros$ presentes en una red con mallado triangular y nodos estáticos.

Teniendo en cuenta los resultados anteriores, la expresión (6.6) se considera validada para el protocolo AODV con un paquete por comunicación para una distancia $R < 3000 metros$ ($H < \xi = 9,92 saltos$). Además, el modelo propuesto se puede considerar válido incluso para valores de R mayores que los indicados por el radio de persistencia, concretamente hasta $R \simeq 4000 metros$.

6.2.2.2. Análisis de resultados para el protocolo DSR

Al igual que en el análisis llevado a cabo para el protocolo AODV, en la Figura 6.20 se muestran los resultados obtenidos para el protocolo DSR. Nuevamente, para representar la media experimental se utiliza la línea roja y para el modelo teórico se utiliza la línea verde. En azul se representa la mediana experimental ya que en este caso identifica mejor los datos, tal y como se ha comentado en el Apartado 6.1.3.2, y la línea vertical representa el radio de persistencia. En la Figura 6.21 se representa la figura anterior aumentada para observar mejor los resultados para distancias pequeñas, abarcando $H < \xi$.

Nuevamente, la función teórica se ajusta a los datos experimentales dentro del intervalo de distancias que se corresponden a $H < \xi$. En el caso del protocolo DSR el radio de

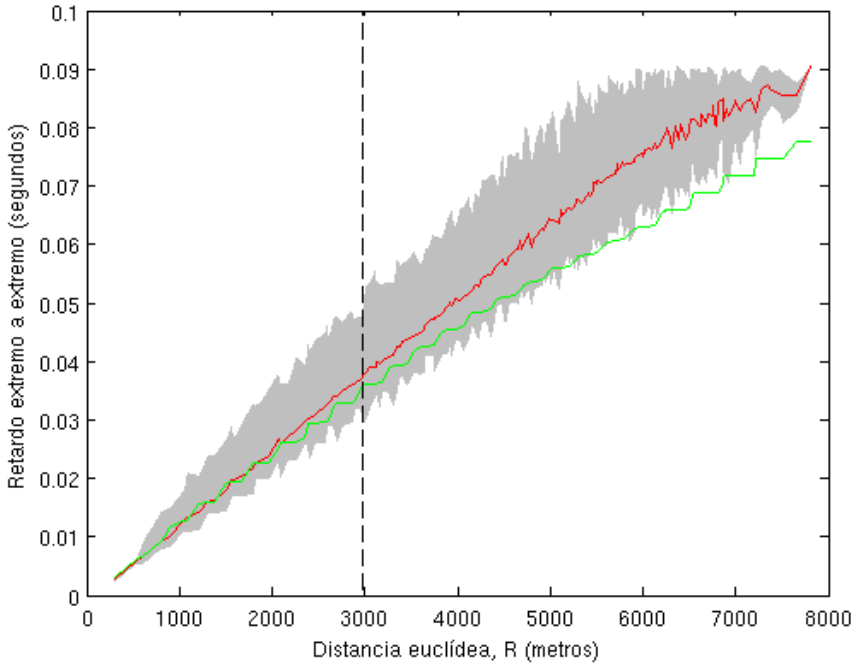

Figura 6.17: Comparación del modelo teórico (verde) frente al experimental (rojo) para el protocolo AODV con un paquete por comunicación, en función de la distancia Euclídea entre emisor y receptor.

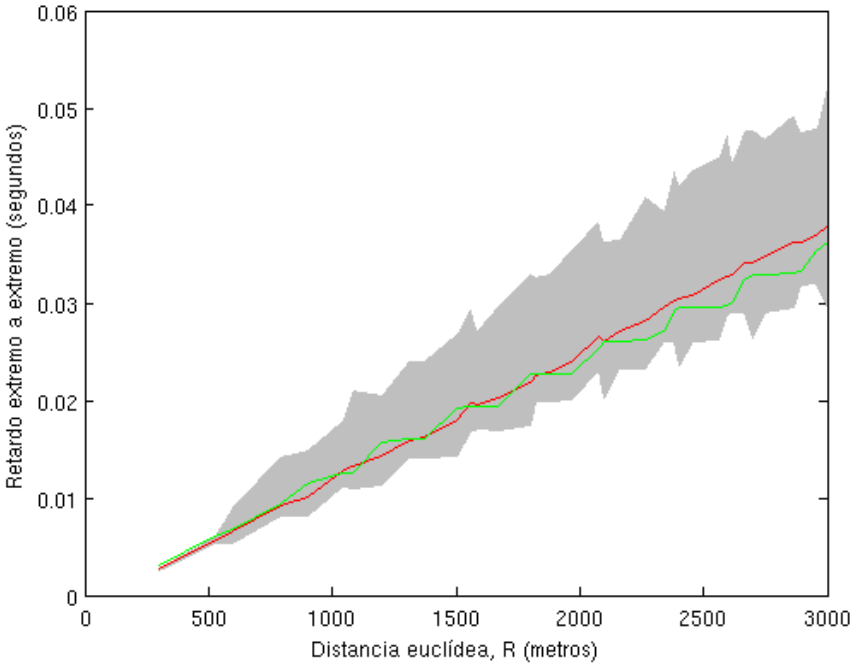

Figura 6.18: Detalle de la Figura 6.17, representando R correspondientes a $H < \xi$.

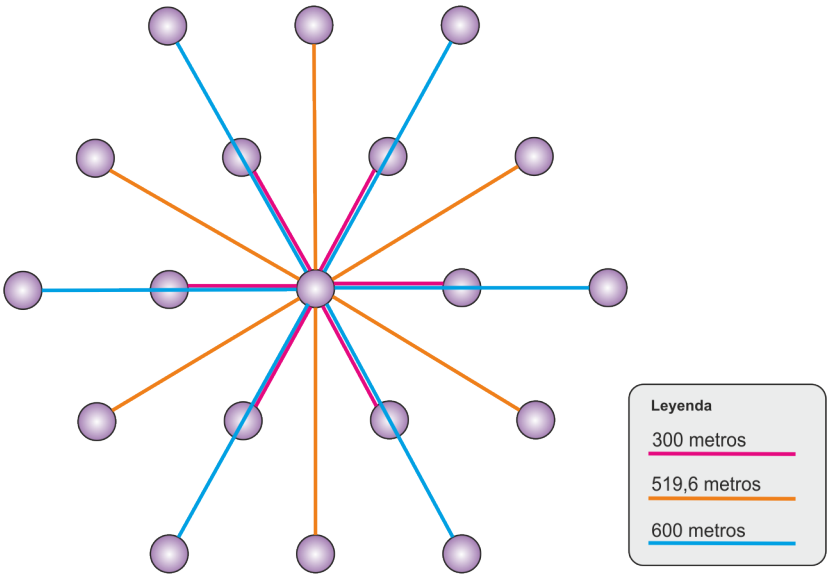

Figura 6.19: Representación de las distancias R menores de $600 metros$ posibles.

persistencia es $\xi = 8{,}18 saltos$, equivalente a $R \simeq 2400 metros$ [5]. Al igual que en el análisis realizado en función del número de saltos de la ruta, el modelo propuesto se ajusta mejor a la mediana de los datos experimentales que a la media.

Por lo tanto, para el protocolo DSR con un paquete por comunicación, la expresión (6.6) queda demostrada para distancias menores de $2400 metros$. La validación se puede realizar para valores de R incluso mayores que el radio de persistencia, concretamente hasta $R \simeq 3500 metros$.

6.3. Comparativa entre los protocolos AODV y DSR

El objetivo de este apartado es realizar una comparación entre los dos protocolos utilizados en este estudio. La comparativa se realiza teniendo en cuenta el retardo extremo a extremo experimentado por los datos bajo los dos protocolos de encaminamiento utilizados, AODV y DSR. Adicionalmente, se atiende también al rendimiento del simulador de red `ns-2` al utilizar AODV o DSR como protocolos de encaminamiento.

6.3.1. Comparativa en función del retardo extremo a extremo

En la Figura 6.22 se representa el modelo teórico propuesto en este proyecto del el retardo extremo a extremo en función del número de saltos de la ruta para los protocolos de encaminamiento AODV y DSR. Dado que en la Sección 6.1 se ha demostrado la veracidad de este modelo teórico, esta comparativa se puede considerar válida.

Se observa que, a medida que las rutas tiene un mayor número de saltos, la diferencia en el retardo entre ambos protocolos aumenta. Los paquetes enrutados con AODV tienen

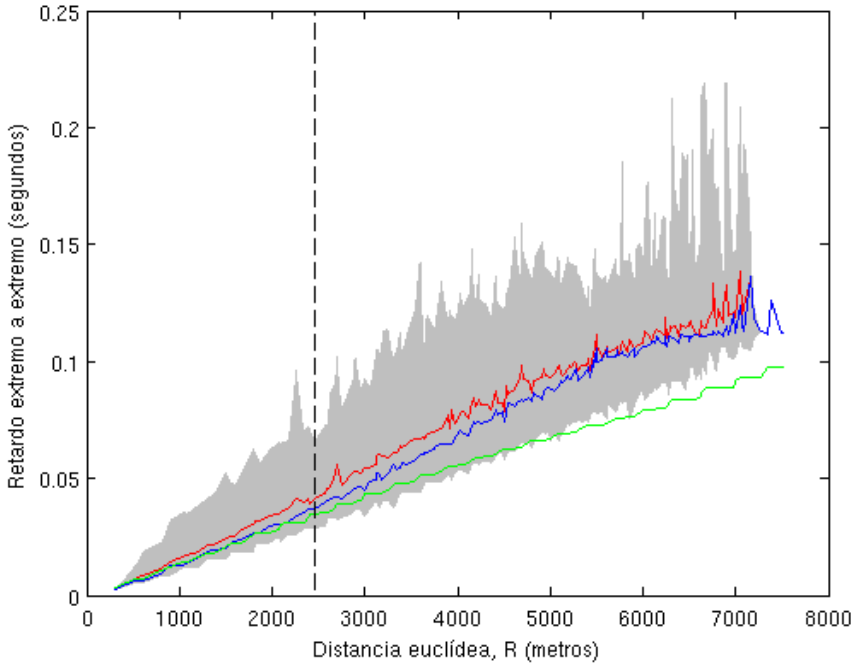

Figura 6.20: Comparación del modelo teórico (verde) frente la media experimental (rojo) y la mediana experimental (azul) para el protocolo DSR con un paquete por comunicación, en función de la distancia Euclídea entre emisor y receptor

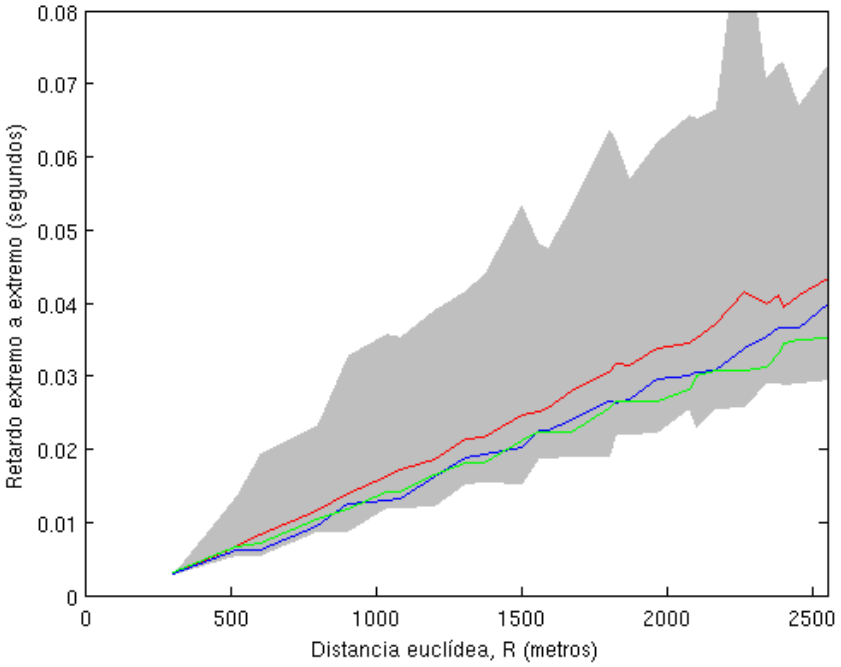

Figura 6.21: Detalle de la Figura 6.20, representando R correspondientes a $H < \xi$.

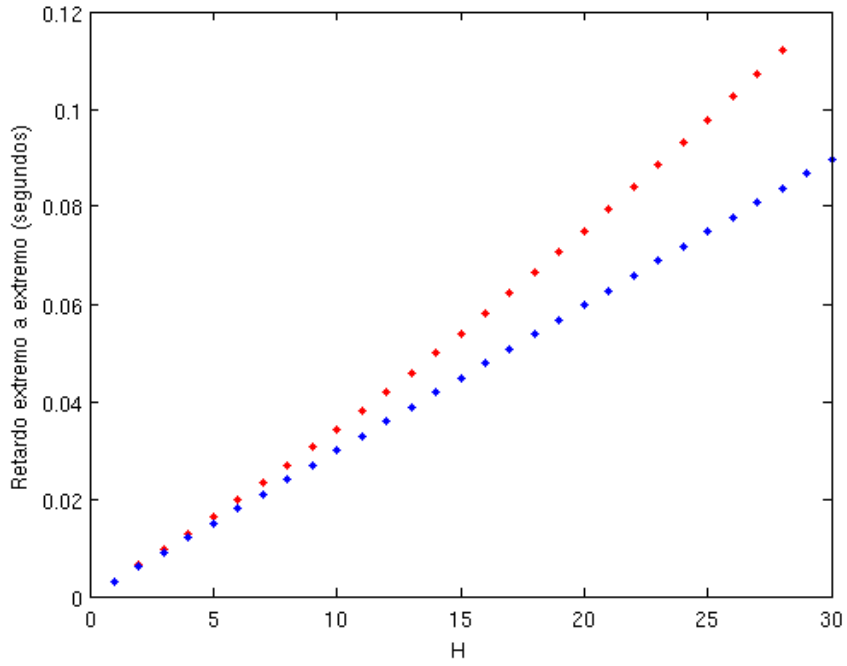

Figura 6.22: Representación del modelo teórico del retardo extremo a extremo en función del número de saltos para los protocolos AODV (azul) y DSR (rojo).

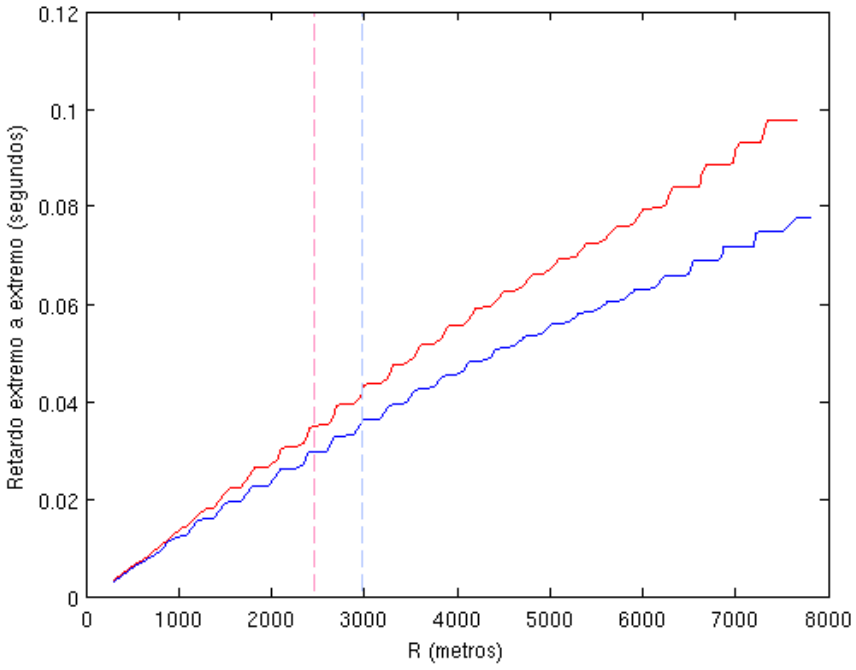

Figura 6.23: Representación del modelo teórico del retardo extremo a extremo en función de la distancia entre los nodos emisor y receptor para los protocolos AODV (azul) y DSR (rojo).

una relación cuasi-lineal con respecto al número de saltos de la ruta, mientras que en los paquetes enrutados con DSR el retardo aumenta a un ritmo mayor. AODV es un protocolo de encaminamiento salto-a-salto, por lo tanto, independientemente del número de saltos de la ruta, la cantidad de bytes a transmitir es la misma. Por otro lado, DSR es un protocolo con encaminamiento en origen y la ruta completa a seguir se almacena en el mismo paquete. Por este motivo, cuanto mayor sea el número de saltos de la ruta, mayor será la cantidad de bytes a transmitir y, por tanto, mayor el retardo extremo a extremo.

En la Figura 6.23 se muestra el modelo teórico en función de la distancia Euclídea entre los nodos emisor y receptor para los dos protocolos de encaminamiento utilizados. Mediante las líneas verticales azul y roja se indica el radio de persistencia para los protocolos de encaminamiento AODV y DSR, respectivamente. Esta comparación se puede realizar, ya que en la Sección 6.2 se ha demostrado la validez del modelo analítico en función a la distancia Euclídea entre el nodo emisor y el nodo receptor. Nuevamente, se observa que el retardo en caso del protocolo DSR es superior al que se experimenta en caso del protocolo AODV.

6.3.2. Comparativa en función del rendimiento en `ns-2`

En cuanto a la simulación en la herramienta `ns-2`, hay una mayor estabilidad del protocolo AODV en cuanto a los resultados que del protocolo DSR. Esto se deduce a partir de las Figuras 6.4, 6.7, 6.17 y 6.20, ya que los intervalos de confianza para AODV son menores que para DSR. Es decir, para el protocolo DSR se dan más valores atípicos y estos están más alejados de la media que en el caso de AODV.

Capítulo 7

Conclusiones y líneas de trabajo futuras

El objetivo de este último capítulo es realizar una reflexión sobre el trabajo realizado, los resultados obtenidos en este proyecto y las líneas de trabajo futuras. En primero lugar, en la Sección 7.1 se exponen las conclusiones derivadas de los resultados obtenidos. En segundo y último lugar, en la Sección 7.2 se proponen varias líneas de investigación con las que se podría extender este trabajo.

7.1. Conclusiones

A partir de los resultados experimentales obtenidos en el Capítulo 6 se determina la viabilidad del modelado analítico del retardo propuesto en este proyecto. Igualmente, se analizan las prestaciones de la herramienta de simulación de red `ns-2`, teniendo en cuenta las diferencias observadas en el funcionamiento de los protocolos AODV y DSR. Las conclusiones obtenidas se enumeran a continuación:

- **El modelo analítico propuesto refleja de forma adecuada la relación existente entre el retardo extremo a extremo y el número de saltos de la ruta seguida.** Mediante las simulaciones realizadas por ordenador con el programa `ns-2` se ha demostrado la veracidad de las expresiones analíticas que relacionan el retardo extremo a extremo con el número de saltos de la ruta seguida por los datos. La comprobación se ha realizado para los protocolos de encaminamiento AODV y DSR. Se ha utilizado un único paquete por comunicación, haciendo que los resultados sean independientes de la clase del protocolo MAC.

- **El modelo analítico propuesto refleja de forma adecuada la relación existente entre el retardo extremo a extremo y la distancia Euclídea entre los nodos emisor y receptor.** Las expresiones analíticas se ajustan a los resultados experimentales para distancias incluso mayores que el radio de persistencia, que es

el límite superior de la región donde el modelo de Hipótesis de Escala garantiza el ajuste. Los datos experimentales se han obtenido mediante simulaciones por ordenador utilizando el programa `ns-2`. Los protocolos de encaminamiento utilizados han sido AODV y DSR. La utilización de un único paquete por comunicación hace que, nuevamente, los resultados sean independientes de la clase del protocolo MAC.

- **El protocolo de encaminamiento AODV ofrece menor retardo extremo a extremo que DSR.** Mediante la comparativa realizada entre AODV y DSR se ha llegado a la conclusión de que el primero de ellos ofrece menor retardo extremo a extremo que el segundo. Esto se debe a que AODV sigue un encaminamiento salto a salto, por lo que la cantidad de bytes transmitidos es la misma, independientemente del número de saltos que tiene la ruta, mientras que DSR es encaminamiento en origen y la ruta a seguir se almacena en las cabeceras del paquete. Por tanto, a medida que aumenta la distancia entre los nodos emisor y receptor, por consiguiente el número de saltos de la ruta, un paquete enviado con DSR experimenta un retardo extremo a extremo mayor.

- **El protocolo de encaminamiento que presenta menor error en el ajuste realizado es AODV.** La comparación realizada entre la bondad del ajuste para AODV y para DSR revela que el primero de ellos tiene una diferencia porcentual entre los datos experimentales y el modelo analítico menor. Esto revela que las expresiones analíticas propuestas modelan de forma más precisa el retardo extremo a extremo cuando en la red gobierna el protocolo de encaminamiento AODV que cuando se utiliza DSR.

- **La herramienta de simulación `ns-2` ofrece menos variabilidad en las medidas experimentales para el protocolo de encaminamiento AODV.** El análisis de las medida experimentales obtenidas revela que la variabilidad de las mismas es menor en el caso de utilizar el protocolo de encaminamiento AODV que DSR, teniendo en cuenta que el resto de las condiciones de la red simulada son idénticas. Este hecho se puede deber a una implementación más óptima del protocolo AODV en `ns-2` que del protocolo DSR.

7.2. Líneas de trabajo futuras

En este Apartado se proponen varías líneas de trabajo que pueden complementar el actual estudio. Estas propuestas son:

- Utilizar distintas clases MAC. Finalmente, en este proyecto ha sido imposible simular otras clases MAC que no sea la 2, ya que en `ns-2` no hemos conseguido que funcionen.

En un escenario real, es posible que el problema del nodo oculto o nodo expuesto no presente realmente un inconveniente o que se desee dar una mayor protección contra los mismos y entonces se utilice un protocolo MAC perteneciente a otra clase. Por lo tanto, es interesante conocer como se comportaría el retardo en este caso.

- Analizar el comportamiento de otros protocolos de encaminamiento para determinar cuál de ellos es el que mejores resultados en cuanto al retardo extremo a extremo ofrece. De esta forma, para una red no tolerante a retardo se puede elegir un protocolo de encaminamiento u otro en función de los resultados obtenidos.

- Simular otros modelos de red incluyendo un mallado de los nodos distinto e incluso incluyendo movilidad de los mismos (a velocidad constante o variable). De esta forma, un nodo puede libremente entrar o salir de la red y desplazarse a lo largo de la misma.

- Considerar un modelo de capa física no ideal que introduzca determinada probabilidad de pérdida de paquetes, con la consiguiente retransmisión de los mismos y, por lo tanto, un retardo adicional en la transmisión.

- Realizar un estudio del retardo medio en una transmisión teniendo en cuenta que en un escenario real hay más de una comunicación simultánea y esta puede producir interferencias. Además se podrían añadir comunicaciones con un nivel de prioridad más elevado para poder simular un escenario de una zona devastada. De esta forma, los mensajes más urgentes tendrían preferencia en la transmisión a través de los nodos intermedios. En este caso, al considerar un número reducido de mensajes urgentes, estos tendrían el menor retardo posible, comparable incluso con el obtenido en este estudio, donde solamente hay una única comunicación en la red.

- Realizar las mismas simulaciones utilizando otra herramienta de simulación de red para poder comparar los resultados y eliminar así la influencia de `ns-2` en el estudio. Los resultados obtenidos a través de cualquier herramienta de simulación por ordenador no son los resultados que se obtendrán cuando la red se implemente físicamente. Por tanto, al realizar simulaciones con diferentes herramientas, se puede realizar una mejor aproximación en cuanto a los resultados que ofrecerá la red implementada.

Bibliografía

[1] Stefano Basagni, Marco Conti, Silvia Giordano, and Ivan Stojmenovic, editors. *Mobile Ad Hoc Networking*. John Wiley & Sons, 2004.

[2] Rajeev Shorey, A. Ananda, Mun Choon Chan, and Wei Tsang Ooi. *Mobile, Wireless and Sensor Networks. Technology, Applications, and Future Direcctions*. John Wiley & Sons, 2006.

[3] George Aggelou. *Mobile Ad Hoc Networks: From Wireless LANs to 4G Networks*. McGraw-Hill, 2005.

[4] L. Gavriloska and R Prasad. *Ad Hoc Networking Towards Seamless Communications*. Springer, 2006.

[5] Eduardo Morgado Reyes. *Prestaciones de las Redes Ad Hoc Inalámbricas: Teoría a través de Capas*. PhD thesis, Dpto. de Teoría de la Señal y Comunicaciones, Universidad Rey Juan Carlos, Spain, July 2009.

[6] A.A. Abdullah, F. Gebali, and Lin Cai. Modeling the Throughput and Delay in Wireless Multihop Ad Hoc Networks. In *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*, pages 1–6, December 2009.

[7] Xiaopeng Fan, Jiannong Cao, and Weigang Wu. Contention-aware data caching in wireless multihop ad hoc networks. In *Mobile Adhoc and Sensor Systems, 2009. MASS '09. IEEE 6th International Conference on*, pages 1–9, oct. 2009.

[8] Nitiket N. Mhala and N. K. Choudhari. An Implementation Possibilities For AODV Routing Protocol In Real World. In *International Journal of Distributed and Parallel Systems (IJDPS)*, volume 1, pages 118–127, November 2010.

[9] Qi Qu, L.B. Milstein, and D.R. Vaman. Cross-Layer Distributed Joint Power Control and Scheduling for Delay-Constrained Applications over CDMA-Based Wireless Ad-Hoc Networks. *Communications, IEEE Transactions on*, 58(2):669–680, February 2010.

[10] A. Lo and I. Niemegeers. Multi-hop relay architectures for 3GPP LTE-advanced. In *Communications (MICC), 2009 IEEE 9th Malaysia International Conference on*, pages 123–127, December 2009.

[11] Tao Chen, Gilles Charbit, and Sami Hakola. Time Hopping for Device-To-Device Communication in LTE Cellular System. In *Wireless Communications and Networking Conference (WCNC), 2010 IEEE*, pages 1–6, April 2010.

[12] Ozan K. Tonguz and Gianluigi Ferrari. *Ad Hoc Wireless Networks: A Communication-Theoretic Perspective*. John Wiley & Sons, September 2006.

[13] P. Bosch, L. Samuel, S. Mullender, P. Polakos, and G. Rittenhouse. Flat Cellular (UMTS) Networks. *Wireless Communications and Networking Conference, 2007.WCNC 2007. IEEE*, pages 3861–3866, March 2007.

[14] U.R. Patel and B.N. Gohil. Cell identity assignment techniques in cellular network: A review. volume 2, pages 594–596, July 2010.

[15] ISO/IEC 7498-1:1994. Information Technology - Open Systems Interconnection - Basic Reference Model: The Basic Model. Also published as ITU-T Recommendation X.200, November 1994.

[16] Ramin Hekmat. *Ad-hoc Networks: Fundamental Properties and Network Topologies*. Springer, 2006.

[17] F. Tobagi and L. Kleinrock. Packet Switching in Radio Channels: Part II–The Hidden Terminal Problem in Carrier Sense Multiple-Access and the Busy-Tone Solution. *Communications, IEEE Transactions on*, 23(12):1417–1433, December 1975.

[18] Matthew S. Gast. *802.11 Wirreless Networks. The Definitive Guide*. O'Really, 2 edition, April 2005.

[19] IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. Technical report, 2007.

[20] N. Bisnik and A.A. Abouzeid. Queuing Delay and Achievable Throughput in Random Access Wireless Ad Hoc Networks. In *Sensor and Ad Hoc Communications and Networks, 2006. SECON '06. 2006 3rd Annual IEEE Communications Society on*, volume 3, pages 874–880, September 2006.

[21] C. Hedrick. Routing Information Protocol. RFC 1058, Internet Engineering Task Force, June 1988.

[22] J. Moy. OSPF Version 2. Standards Track 2328, Internet Engineering Task Force, April 1998.

[23] A. Roy and M. Chandra. Extensions to OSPF to Support Mobile Ad Hoc Networking. RFC Experimental 5820, Internet Engineering Task Force, March 2010.

[24] Michel Barbeau and Evangelos Kranakis. *Principles of Ad Hoc Networking*. John Wiley & Sons, 2007.

[25] David B. Johnson, Y. Hu, and D. Maltz. The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4. RFC Experimental 4728, Internet Engineering Task Force, February 2007.

[26] Charles E. Perkins, Elizabeth M. Belding-Royer, and Samir R. Das. Ad hoc On-Demand Distance Vector (AODV) Routing. RFC Experimental 3561, Internet Engineering Task Force, July 2003.

[27] Charles E. Perkins and Pravin Bhagwat. Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. In *ACM Conference on Communications Architectures, Protocols and Applications, SIGCOMM '94, London, UK*, pages 234–244. ACM, ACM, August 1994.

[28] Thomas Clausen and Philippe Jacquet. Optimized Link State Routing Protocol (OLSR). RFC 3626, Internet Engineering Task Force, October 2003.

[29] Z.J. Haas. A new routing protocol for the reconfigurable wireless networks. In *Universal Personal Communications Record, 1997. Conference Record., 1997 IEEE 6th International Conference on*, volume 2, pages 562–566 vol.2, oct 1997.

[30] Chai-Keong Toh. Associativity-Based Routing For Ad-Hoc Mobile Networks. In *Wireless Personal Communications Journal, Special Issue on Mobile Networking and Computing Systems*, volume 4, pages 103–139. Kluwer Academic Publishers, March 1997.

[31] E.M. Royer and Charles E. Perkins. An implementation study of the AODV routing protocol. In *Wireless Communications and Networking Conference, 2000. WCNC. 2000 IEEE*, volume 3, pages 1003–1008 vol.3, 2000.

[32] Charles E. Perkins, Elizabeth M. Belding-Royer, Samir R. Das, and Ian D. Chakeres. AODV Public Iimplementations. `http://moment.cs.ucsb.edu/AODV/aodv.html`. Consultado el 8 de Marzo de 2011.

[33] L. Klein-Berndt. Kernel AODV from National Institute of Standards and Technology (NIST). `http://w3.antd.nist.gov/wctg/aodv_kernel/`. Consultado el 8 de Marzo de 2011.

[34] P. Sankar. Implementation of DSR algorithm using VHDL in wireless ad-hoc network. In *Solid-State and Integrated Circuits Technology. Proceedings. 7th International Conference on*, volume 2, pages 1364–1367, October 2004.

[35] Charles E. Perkins. Ad Hoc On-Demand Distance Vector Routing Protocol. Internet-draft, IETF MANET Working Group, November 1997. Expiration: May 20, 1998.

[36] Woonkang Heo and Minseok Oh. Performance of Expanding Ring Search Scheme in AODV Routing Algorithm. volume 2, pages 128–132, December 2008.

[37] Hongmei Deng, Wei Li, and Dharma P. Agrawal. Routing Security in Wireless Ad Hoc Networks. *IEEE Communications Magazine*, pages 70–75, October 2002.

[38] Geetha Jayakumar and Gopinath Ganapathy. Performance Comparison of Mobile Ad-hoc Network Routing Protocol. *IJCSNS International Journal of Computer Science and Network Security*, 7(11):77–84, November 2007.

[39] S. Bradner and V. Paxson. IANA Allocation Guidelines For Values In the Internet Protocol and Related Headers. RFC Best Current Practice 2780, Internet Engineering Task Force, March 2000.

[40] David B. Johnson and David A. Maltz. Dynamic Source Routing in Ad Hoc Wireless Networks. 1996.

[41] David B. Johnson. Routing in Ad Hoc Networks of Mobile Hosts. *Proceedings of The IEEE Workshop on Mobile Computing System and Aplications (WMCSA), IEEE Computer Society, Santa Cruz, CA*, pages 158–163, December 1994.

[42] I. Broustis, G. Jakllari, T. Repantis, and M. Molle. A Comprehensive Comparison of Routing Protocols for Large-Scale Wireless MANETs. In *Sensor and Ad Hoc Communications and Networks, 2006. SECON '06. 2006 3rd Annual IEEE Communications Society on*, volume 3, pages 951–956, September 2006.

[43] A. Rawat, P.D. Vyavahare, and A.K. Ramani. Enhanced DSR with secured multipath route discovery and concurrent data transmission. pages 612–613, December 2006.

[44] Jon Postel. Internet Protocol. DARPA Internet Program. Protocol Specification. RFC 791, Information Sciences Institute, University of Southern California, September 1981.

[45] The Network Simulator - `ns-2`. `http://www.isi.edu/nsnam/ns/`. Consultado el 16 de Agosto de 2010.

[46] Kevin Fall and Kannan Varadhan. The *ns* Manual (formerly *ns* Notes and Documentation). `http://www.isi.edu/nsnam/ns/ns-documentation.html`. Consultado el 16 de Agosto de 2010.

[47] A.U. Salleh, Z. Ishak, N.M. Din, and M.Z. Jamaludin. Trace Analyzer for `ns-2`. In *Research and Development, 2006. SCOReD 2006. 4th Student Conference on*, pages 29–32, June 2006.

[48] Li Xu, Hui Bo, Liu Haixia, Yang Mingqiang, Song Mei, and Guo Wei. Research and Analysis of Topology Control in `NS-2` for Ad-hoc Wireless Network. In *Complex, Intelligent and Software Intensive Systems, 2008. CISIS 2008. International Conference on*, pages 461–465, March 2008.

[49] A. Hegedus, G.M. Maggio, and L. Kocarev. A `ns-2` simulator utilizing chaotic maps for network-on-chip traffic analysis. In *Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on*, pages 3375–3378 Vol. 4, May 2005.

[50] S. Gowrishankar, T.G. Basavaraju, M. Singh, and Subir Kumar Sarkar. Scenario based Performance Analysis of AODV and OLSR in Mobile Ad hoc Networks. *24th South East Asia Regional Computer Conference*, November 2007.

[51] C. Hanle and M. Hofmann. Performance comparison of reliable multicast protocols using the network simulator `ns-2`. In *Local Computer Networks, 1998. LCN '98. Proceedings., 23rd Annual Conference on*, pages 222–237, October 1998.

[52] Qin long Qiu, Jian Chen, Ling di Ping, Qi fei Zhang, and Xue zeng Pan. LTE/SAE Model and its Implementation in `NS 2`. In *Mobile Ad-hoc and Sensor Networks, 2009. MSN '09. 5th International Conference on*, pages 299–303, December 2009.

[53] D. Mahrenholz and S. Ivanov. Real-Time Network Emulation with `ns-2`. In *Distributed Simulation and Real-Time Applications, 2004. DS-RT 2004. Eighth IEEE International Symposium on*, pages 29–36, October 2004.

[54] Mark Lutz. *Learning Python*. O'Really, third edition, October 2007.

[55] Python Programming Language – Official Website. `http://www.python.org/`. Consultado el 20 de Agosto de 2010.

[56] The Python Standard Library. `http://docs.python.org/release/2.6.5/library/index.html`. Consultado el 20 de Agosto de 2010.

[57] Django. The Web framework for perfectionists with deadlines. `http://www.djangoproject.com/`. Consultado el 17 de Noviembre de 2010.

[58] MATLAB - The Language Of Technical Computing. `http://www.mathworks.com/products/matlab/`. Consultado el 17 de Noviembre de 2010.

[59] LyX - The Document Processor. `http://www.lyx.org/`. Consultado el 10 de Octubre de 2010.

[60] LaTeX - A document preparation system. `http://www.latex-project.org/`. Consultado el 10 de Octubre de 2010.

[61] KBibTeX - KDE3. `http://www.unix-ag.uni-kl.de/~fischer/kbibtex/`. Consultado el 10 de Octubre de 2010.

[62] BibTeX. `http://www.bibtex.org/`. Consultado el 10 de Octubre de 2010.

[63] M.M. Carvalho and J.J. Garcia-Luna-Aceves. Delay analysis of IEEE 802.11 in single-hop networks. In *Network Protocols, 2003. Proceedings. 11th IEEE International Conference on*, pages 146–155, November 2003.

[64] N. Bansal and Z. Liu. Capacity, Delay and Mobility in Wireless Ad-Hoc Networks. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 2, pages 1553–1563 vol.2, March 2003.

[65] O. Tickoo and B. Sikdar. Queueing analysis and delay mitigation in IEEE 802.11 random access MAC based wireless networks. In *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, volume 2, pages 1404–1413 vol.2, March 2004.

[66] S.R. Chaudhry, H.S.A. Raweshidy, and S.S.A. Obayya. Delay analysis in ad hoc on demand distance vector based networks. In *Multitopic Conference, 2004. Proceedings of INMIC 2004. 8th International*, pages 250–255, 2004.

[67] P.L. Royds and J.M.H. Elmirghani. Delay characteristics of diverse ad hoc networks. *Electronics Letters*, 40(7):439–440, 2004.

[68] Heberto Del Rio and Dilip Sarkar. Logarithmic expected packet delivery delay in mobile ad hoc wireless networks. *Wireless Communications and Mobile Computing*, 4:281–287, 2004.

[69] S.S. Kunniyur and S. Narasimhan. Modelling the effect of network parameters on delay in wireless ad-hoc networks. In *Sensor and Ad Hoc Communications and Networks, 2005. IEEE SECON 2005. 2005 Second Annual IEEE Communications Society Conference on*, pages 340–349, September 2005.

[70] Dong Linfang, Shu Yantai, Chen Haiming, and M. Maode. Estimation and application of end-to-end delay under unsaturation traffic in wireless ad hoc networks. In *Mobile Technology, Applications and Systems, 2005 2nd International Conference on*, pages 6 pp.–6, 2005.

[71] H. del Rio, D. Sarkar, and L.D. Stelling. Packet delay estimation for ad hoc networks. In *Computer Communications and Networks, 2005. ICCCN 2005. Proceedings. 14th International Conference on*, pages 289–296, 2005.

[72] G. Farhadi and N.C. Beaulieu. On the Connectivity and Average Delay of Mobile Ad Hoc Networks. In *Communications, 2006. ICC '06. IEEE International Conference on*, volume 8, pages 3868–3872, 2006.

[73] G. Sharma, R. Mazumdar, and N. Shroff. Delay and Capacity Trade-Offs in Mobile Ad Hoc Networks: A Global Perspective. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pages 1–12, 2006.

[74] A. El Gamal, J. Mammen, B. Prabhakar, and D. Shah. Optimal Throughput-Delay Scaling in Wireless Networks - Part I: The Fluid Model. *Information Theory, IEEE Transactions on*, 52(6):2568–2592, June 2006.

[75] A. El Gamal, J. Mammen, B. Prabhakar, and D. Shah. Optimal Throughput-Delay Scaling in Wireless Networks - Part II: Constant-Size Packets. *Information Theory, IEEE Transactions on*, 52(11):5111–5116, 2006.

[76] E. Gelenbe. Travel delay in a large wireless ad hoc network. In *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, 2006 4th International Symposium on*, pages 1–6, 2006.

[77] Y. Yang and R. Kravets. Achieving Delay Guarantees in Ad Hoc Networks through Dynamic Contention Window Adaptation. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pages 1–12, 2006.

[78] Lei Ying, Sichao Yang, and R. Srikant. Optimal Delay-Throughput Tradeoffs in Mobile Ad Hoc Networks. *Information Theory, IEEE Transactions on*, 54(9):4119–4143, September 2008.

[79] G. Brassard and P. Bratley. *Fundamentos de Algoritmia.* Prentice Hall, 1997.

[80] Juan José Vinagre Díaz. *Teoría del encaminamiento en Redes Ad Hoc Inalámbricas.* PhD thesis, Dpto. de Teoría de la Señal y Comunicaciones, Universidad Carlos III de Madrid, Spain, July 2007.

[81] Maryna Kurmanava. Estudio de la eficiencia de encaminamiento del protocolo DSR en redes ad hoc inalámbricas de gran escala. Master's thesis, Dpto. de Teoría de la Señal y Comunicaciones, Universidad Rey Juan Carlos, Spain, July 2009.

[82] María Elena Gil Jiménez. Estudio de la eficiencia de encaminamiento del protocolo AODV en redes ad hoc inalámbricas de gran escala. Master's thesis, Dpto. de Teoría de la Señal y Comunicaciones, Universidad Rey Juan Carlos, Spain, July 2009.

[83] Mohammad Reza Sahraei. Intergration of `ns-BGP` with `ns-2.34` (`ns-2.34-BGP`). Technical report, ENSC 891: Directed Studies, 2009.

[84] Jan Goyvaerts and Steven Levithan. *Regular Expressions Cookbook*. O'Really, May 2009.

[85] Tony Stubblebine. *Regular Expressions. Pocket Reference*. O'Really, second edition, July 2007.

[86] Solomon Garfunkel, Jody L. Doran, and Eugenio Hernández. *Las matemáticas en la vida cotidiana versión en español*. Addison-Wesley Iberoamericana España, 3 edition, 2006.

[87] Daniel Peña. *Fundamentos de estadística*. Alianza, 2001.

[88] John E. Freund and Benjamin M. Perles. *Statistics: a first course*. Prentice Hall, 2004.

[89] P. Chatzimisios, A.C. Boucouvalas, and V. Vitsas. IEEE 802.11 packet delay-a finite retry limit analysis. In *Global Telecommunications Conference, 2003. GLOBECOM '03. IEEE*, volume 2, pages 950–954 Vol.2, December 2003.

[90] G. Bianchi and I. Tinnirello. Remarks on IEEE 802.11 DCF performance analysis. *Communications Letters, IEEE*, 9(8):765–767, August 2005.

Definiciones

	Significado
Directividad	Mantenimiento en cada salto de la dirección trazada previamente.
Extremo a extremo	Entre el nodo emisor y el nodo receptor de un mensaje.
Flag	Campo que únicamente almacenan información binaria (0/1).
Inundación	Técnica utilizada por los nodos para enviar mensajes a todos sus vecinos.
Nodo intermedio	Nodo que forma parte de una ruta establecida entre una pareja.
Pareja	Dos nodos (emisor y receptor) que intercambian datos.
Red saturada	Todos los nodos tienen en todo momento un paquete para transmitir.
Ruta activa	Se puede utilizar para enviar datos.
Ruta óptima	Tiene el menor número de saltos.
Salto	Comunicación entre dos nodos sin pasar por nodos intermedios.
Script de usuario	Fichero ejecutable donde se proporcionan todos los parámetros necesarios para una simulación en `ns-2`.