



PROYECTO FIN DE MÁSTER EN  
SISTEMAS INTELIGENTES

CURSO 2008-2009

---

**PROTOCOLO DE ENCAMINAMIENTO ACO  
HÍBRIDO PARA REDES MÓVILES AD HOC**

**Delfín Rupérez Cañas**

Director:

**Alfredo Fernández-Valmayor Crespo**

Departamento de Ingeniería del Software e Inteligencia Artificial

Colaborador externo de dirección:

**Luis Javier García Villalba**

Departamento de Ingeniería del Software e Inteligencia Artificial

---

MÁSTER EN INVESTIGACIÓN EN INFORMÁTICA

FACULTAD DE INFORMÁTICA

UNIVERSIDAD COMPLUTENSE DE MADRID



Él abajo firmante, matriculado en el Máster en Investigación en Informática de la Facultad de Informática, autoriza a la Universidad Complutense de Madrid (UCM) a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a su autor, el presente Trabajo Fin de Máster: *“Protocolo de Encaminamiento ACO Híbrido para Redes Móviles Ad hoc”*, realizado durante el curso académico 2008-2009 bajo la dirección de Alfredo Fernández-Valmayor Crespo y con la colaboración externa de dirección de Luis Javier García Villalba en el Departamento de Ingeniería del Software e Inteligencia Artificial, y a la Biblioteca de la UCM a depositarlo en el Archivo Institucional E-Prints Complutense con el objeto de incrementar la difusión, uso e impacto del trabajo en Internet y garantizar su preservación y acceso a largo plazo.

---

Delfín Rupérez Cañas



## *Abstract*

Mobile ad hoc networks (MANETs) are multihop wireless networks of mobile nodes without any fixed or preexisting infrastructure. The topology of these networks can change randomly due to unpredictable mobility of nodes and propagation characteristics. The objective of a mobile ad hoc network routing protocol is to allow sending a message from a node to other without a direct link. This work presents an AntHocNet-based hybrid ACO routing protocol for mobile ad hoc networks. The proposed protocol uses disjoint link/node routes, separation of the diffusion of the pheromones and exploration of routes using the number of hops of the best found routes. Simulation results show that the protocol incurs low overhead and high delivery packet ratio.

## *Keywords*

Mobile Ad Hoc Network, Routing Protocol, Hybrid, Ant Colony Optimization, ACO, Link-Disjoint Route, Node-Disjoint Route, Pheromone, AntHocNet.



## *Resumen*

Las redes móviles ad hoc (MANETs) son redes inalámbricas multisalto de nodos móviles sin infraestructura previa alguna. La topología de estas redes puede cambiar aleatoriamente debido a la movilidad imprevisible de los nodos y a las características de propagación. El objetivo de un protocolo de encaminamiento para redes móviles es conseguir el envío de un mensaje de un nodo a otro sin existir un enlace directo. Este trabajo propone un protocolo de encaminamiento ACO híbrido para este tipo de redes basado en el protocolo AntHocNet. El protocolo se caracteriza por la utilización de rutas de enlace/nodo disjuntos, por la separación del proceso de difusión de las feromonas y por el proceso de exploración de rutas, que tiene en consideración el número de saltos de las mejores rutas encontradas anteriormente. Los resultados de la simulación demuestran que el protocolo tiene una baja sobrecarga y un elevado ratio de entrega de paquetes.

## *Palabras clave*

Redes Móviles Ad Hoc, Protocolo de Encaminamiento, Híbrido, Optimización de la Colonia de Hormigas, ACO, Ruta de Enlace Disjunto, Ruta de Nodo disjunto, Feromona, AntHocNet.



## *Agradecimientos*

Agradezco a Javier García el que me haya dirigido y acompañado durante todo el proyecto.

También a Alfredo Fernández-Valmayor Crespo por ofrecerse a ser mi Director.

Agradezco a mi familia el que me haya animado y ayudado en todo momento.

A mis compañeros del Grupo de Análisis, Seguridad y Sistemas (GASS) que me ha dado ofrecido su apoyo y al Departamento de Ingeniería del Software e Inteligencia Artificial (DISIA).

En general doy las gracias a todas las personas que de alguna manera me han ayudado.



## *Lista de acrónimos*

ACO	Ant Colony Optimization
ADRA	Ant-based Distributed Route Algorithm
AEADMRA	Ant-based Energy Aware Disjoint Multipath Routing Algorithm
AODV	Ad hoc On-demand Distance Vector
AI	Artificial Intelligence
AMQR	Ant Colony based Multi-path QoS-aware Routing
ANTHOCNET	Adaptive Routing Algorithm for MANETs
ANSI	Ad hoc Networking with Swarm Intelligence
ANTOR	Ant Optimized Routing
AODV	Ad hoc On-demand Distance Vector
ARA	Ant-Colony-Based Routing Algorithm
ARAMA	Ant Routing Algorithm for Mobile Ad hoc networks
ARP	Address Resolution Protocol
CSMA	Carrier Sense Multiple Access
DARPA	Defense Advanced Research Projects Agency
DNR	Disjoint Node Route
DLR	Disjoint Link Route
DSDV	Destination-Sequenced Distance-Vector Routing
DSR	Dynamic Source Routing
DYMO	DYnamic Manet On-demand
FSR	Fisheye State Routing Protocol

GNU	General Public License
HOPNET	Hybrid ant colony OPTimization routing algorithm for mobile ad hoc NETwork
IEEE	Institute of Electrical and Electronic Engineers
IETF	Internet Engineering Task Force
IP	Internet Protocol
LFN	Link Failure Notification
MANET	Mobile Ad hoc NETwork
MPR	Multipoint Distribution Relay
NS-3	Network Simulator 3
OLSR	Optimized Link State Routing
PACONET	imProved Ant Colony Optimization algorithm for mobile ad hoc NETworks
PAN	Personal Area Networks
PBA	Proactive Backward Ant
PERA	Probabilistic Emergent Routing Algorithm
PFA	Proactive Forward Ant
POSANT	Position based Ant Colony Routing
QoS	Quality of Service
RF	Radio Frequency
RBA	Reactive Backward Ant
RFA	Reactive Forward Ant
RFC	Request For Comments

RIP	Routing Information Protocol
RRBA	Route Repair Backward Ant
RRFA	Route Repair Forward Ant
RREP	Route REPLY
RREQ	Route REQuest
TBRPF	Topology dissemination Based on Reverse-Path Forwarding
TC	Topology Control
TCP	Transmission Control Protocol
TORA	Temporally Ordered Routing Algorithm
TTL	Time To Live
UDP	User Datagram Protocol
VoIP	Voice over Internet Protocol
WG	Work Group
WiFi	Wireless Fidelity
ZRP	Zone-based hierarchical link state Routing Protocol



# ÍNDICE

<b>1. INTRODUCCIÓN.....</b>	<b>1</b>
1.1. OBJETO DE LA INVESTIGACIÓN .....	2
1.2. TRABAJOS RELACIONADOS .....	3
1.3. ESTRUCTURA DEL TRABAJO .....	4
<b>2. REDES MÓVILES AD HOC .....</b>	<b>7</b>
2.1. EVOLUCIÓN HISTÓRICA .....	7
2.2. CARACTERÍSTICAS.....	9
2.3. ESTÁNDAR IEEE 802.11 .....	12
2.4. CLASIFICACIÓN .....	13
2.5. APLICACIONES .....	15
<b>3. ENCAMINAMIENTO EN REDES MÓVILES AD HOC .....</b>	<b>17</b>
3.1. PROTOCOLOS DE ENCAMINAMIENTO EN REDES MÓVILES AD HOC.....	17
3.2. CLASIFICACIÓN .....	20
3.3. PROTOCOLOS REACTIVOS.....	23
3.3.1. AODV.....	23
3.3.2. DYMO .....	26
3.3.3. DSR .....	27
3.3.4. TORA .....	28
3.4. PROTOCOLOS PROACTIVOS .....	29
3.4.1. DSDV .....	29
3.4.2. OLSR.....	30
3.4.3. TBRPF.....	31
3.4.4. FSR.....	32
3.5. PROTOCOLOS HÍBRIDOS .....	33
3.5.1. ZRP.....	33
<b>4. EL ALGORITMO ANT COLONY OPTIMIZATION (ACO).....</b>	<b>35</b>
4.1. INTRODUCCIÓN .....	35
4.2. EXPERIMENTO DEL DOBLE PUENTE .....	36
4.3. HORMIGAS ARTIFICIALES .....	39
4.4. SIMPLE ANT COLONY OPTIMIZATION (S-ACO) .....	42
4.4.1. Introducción .....	42
4.4.2. Descripción.....	44
4.4.2.1. Modos de funcionamiento.....	44
4.4.2.2. Búsqueda de caminos.....	44
4.4.2.3. Trazado de ruta y actualización de feromona .....	45
4.4.2.4. Evaporación de las marcas de feromona.....	47
4.5. META-HEURÍSTICA ACO.....	48
<b>5. ENCAMINAMIENTO ACO EN REDES MÓVILES AD HOC.....</b>	<b>49</b>
5.1. ENCAMINAMIENTO ACO .....	49
5.2. DISEÑO DE ALGORITMOS ACO DE ENCAMINAMIENTO.....	51
5.3. APLICACIÓN DIRECTA DE ACO EN MANETS .....	52
5.4. MODIFICACIÓN DE ACO PARA SU APLICACIÓN EN MANETS .....	53
5.5. ANT-NET: EL PRIMER ALGORITMO ACO .....	54
5.6. PROTOCOLOS DE ENCAMINAMIENTO ACO PARA MANETS.....	55
5.6.1. Protocolos de Encaminamiento ACO Proactivos.....	55
5.6.1.1. Probabilistic Emergent Routing Algorithm.....	55
5.6.1.2. Ant Routing Algorithm for Mobile Ad hoc networks .....	56
5.6.2. Protocolos de Encaminamiento ACO Reactivos .....	57

5.6.2.1. Ad hoc Networking with Swarm Intelligence .....	57
5.6.2.2. Ant-Colony-Based Routing Algorithm.....	58
5.6.2.3. Position based Ant Colony Routing.....	59
5.6.2.4. Ant colony based Multi-path Qos-aware Routing .....	60
5.6.2.5. Distributed Ant Routing.....	60
5.6.2.6. Ant-based Distributed Route Algorithm.....	61
5.6.2.7. Ant-based Energy Aware Disjoint Multipath Routing Algorithm .....	62
5.6.2.8. ImProved Ant Colony Optimization Algorithm for Mobile Ad Hoc Networks.....	62
5.6.3. Protocolos de Encaminamiento ACO Híbridos.....	63
5.6.3.1. Ant-AODV.....	63
5.6.3.2. A hybrid ant colony optimization routing algorithm for mobile ad hoc network.....	64
<b>6. ANTHOCNET.....</b>	<b>66</b>
6.1. ESTRUCTURAS DE DATOS .....	67
6.2. FUNCIONAMIENTO.....	68
6.2.1. Establecimiento del camino .....	69
6.2.2. Encaminamiento de datos.....	73
6.2.3. Mantenimiento de las rutas establecidas y exploración de nuevas.....	74
6.2.4. Gestión de los fallos de enlace.....	75
6.3. VARIANTES DEL PROTOCOLO .....	77
6.3.1. Ant Colony Based Node Disjoint Hybrid Multi-path Routing .....	77
6.3.2. An Efficient Ant-based Routing Algorithm for MANETs .....	78
6.3.3. Integrated Routing Algorithm.....	78
6.4. EVALUACIÓN .....	79
<b>7. ANTOR: PROTOCOLO DE ENCAMINAMIENTO ACO HÍBRIDO .....</b>	<b>81</b>
7.1. PROTOCOLO DE ENLACE / NODO DISJUNTO.....	81
7.1.1. Rutas de nodo disjunto .....	82
7.1.2. Rutas de enlace disjunto.....	83
7.2. SEPARACIÓN DEL PROCESO DE DIFUSIÓN DE FEROMONA .....	85
7.3. UTILIZACIÓN DE LA MÉTRICA DISTANCIA EN LA EXPLORACIÓN DE CAMINOS.....	86
7.4. FUNCIONAMIENTO.....	86
<b>8. SIMULACIONES Y RESULTADOS.....</b>	<b>91</b>
8.1. RESULTADOS.....	92
8.1.1. Volumen de trabajo.....	92
8.1.2. Ratio de entrega de paquetes .....	93
8.1.3. Sobrecarga en el número de paquetes.....	93
8.1.4. Sobrecarga en el número de bytes .....	94
8.1.5. Retardo medio extremo a extremo .....	95
<b>9. CONCLUSIONES Y TRABAJO FUTURO .....</b>	<b>97</b>
9.1. TRABAJO FUTURO .....	98
<b>REFERENCIAS .....</b>	<b>99</b>
<b>ANEXO A: SIMULADOR DE REDES NS-3.....</b>	<b>107</b>
<b>ANEXO B: CARACTERÍSTICAS DEL PROTOCOLO .....</b>	<b>116</b>
<b>ANEXO C: DIAGRAMA DE CLASES.....</b>	<b>132</b>

## ÍNDICE DE TABLAS

7.1. Tabla de rutas para el nodo A. ....	87
7.2. Tabla de rutas para el nodo D. ....	88
7.3. Tabla de rutas para el nodo B. ....	88
7.4. Tabla de rutas para el nodo C. ....	88
7.5. Tabla de rutas para A en el proceso de difusión. ....	89
B.1 Vecinos .....	117
B.2 Encaminamiento.....	117



## ÍNDICE DE FIGURAS

4.1 Esquema del doble puente.....	39
8.1. Volumen de trabajo ( <i>ThroughPut</i> ). .....	92
8.2. Ratio de entrega de paquetes.....	93
8.3. Sobrecarga en el número de paquetes. ....	94
8.4. Sobrecarga en el número de bytes. ....	94
8.5. Retardo medio extremo-a-extremo. ....	95
B.1 Formato general de los paquetes AntOR.....	118
B.2 Formato de la cabecera de los mensajes.....	119
B.3 Formato del mensaje HELLO.....	120
B.4 Formato del mensaje Reactive Forward Ant (RFA).....	121
B.5 Formato del mensaje Reactive Backward Ant (RBA) .....	122
B.6 Formato del mensaje Proactive Forward Ant (PFA).....	123
B.7 Formato del mensaje Proactive Backward Ant (PBA).....	123
B.8 Formato mensaje Link Failure Notification (LFN).....	124
B.9 Formato del mensaje Route Repair Forward Ant (RRFA).....	125
B.10 Formato del mensaje Route Repair Backward Ant (RRBA) .....	125
C.1 Diagrama clases de AntOR (Disjunto de Enlace).....	131



## ÍNDICE DE ECUACIONES

4.1. Heurística de búsqueda de camino en S-ACO.....	45
4.2. Actualización de la pista de feromona en S-ACO. ....	46
4.3. Evaporación de la feromona en S-ACO.....	47
6.1. Estimación de tiempo utilizado en el establecimiento del camino. ....	70
6.2. Valor de estimación local en AntHocNet. ....	71
6.3. Tiempo medio necesitado para enviar un paquete en AntHocNet. ....	71
6.4. Valor de la feromona en AntHocNet.....	71
6.5. Actualización de un registro de la tabla de encaminamiento en AntHocNet.....	72
6.6. Proceso de envío en el descubrimiento de ruta en AntHocNet. ....	72
6.7. Encaminamiento de los datos en AntHocNet.....	73



# 1. INTRODUCCIÓN

---

Una red móvil ad hoc o *Mobile Ad hoc NETWORK* (MANET) es un conjunto de nodos móviles que se comunican entre sí a través de enlaces inalámbricos (*wireless*). Al contrario de las redes convencionales, una red móvil ad hoc no necesita la existencia de una infraestructura previa ya que cada nodo se apoya en los demás para conseguir comunicarse con otro creando la llamada comunicación multisalto.

Este tipo de redes tiene varios inconvenientes que una red convencional no presenta. La topología de este tipo de redes puede cambiar rápidamente y de una forma impredecible. Además, pueden surgir variaciones en las capacidades de los nodos y enlaces, errores frecuentes en la transmisión y falta de seguridad. Por último, se deben tener en cuenta los recursos limitados de los nodos ya que normalmente una red ad hoc estará formada por dispositivos alimentados por baterías.

Desde la perspectiva de la capa IP, una red ad hoc se presenta como una red multisalto de nivel 3 constituida por una colección de enlaces.

El resto de este capítulo está organizado como sigue: El apartado 1.1 presenta el objeto de investigación de este trabajo. En el apartado 1.2 se analizan algunos trabajos relacionados. En último lugar, el apartado 1.3 resume la estructura del resto del trabajo.

## 1.1. Objeto de la investigación

En redes móviles ad hoc los protocolos de encaminamiento convencionales (para redes cableadas) o bien tendrán un rendimiento muy pobre, o bien serán simplemente inaplicables. Como alternativa se desarrollan protocolos específicos de encaminamiento.

El concepto de encaminamiento básicamente comprende dos actividades. En primer lugar, determinar los caminos óptimos y, en segundo lugar, transferir los grupos de paquetes de información a través de la red. Los algoritmos utilizan varias métricas para calcular el mejor camino para que los paquetes lleguen a su destino. Estas métricas son medidas estándar como podría ser el número de saltos que son usados por el algoritmo para determinar el camino óptimo. El proceso para determinar el camino inicializa y mantiene tablas de encaminamiento que contienen la información total de cada ruta. La información que se almacena para cada ruta varía de un algoritmo a otro.

El objetivo de un protocolo de encaminamiento para redes móviles es conseguir el envío de un mensaje de un nodo a otro sin existir un enlace directo.

Este trabajo propone un protocolo *bioinspirado* de encaminamiento para redes móviles ad hoc basado en el algoritmo de la colonia de hormigas (ACO) y, más concretamente, en AntHocNet, que es una aplicación directa del algoritmo ACO en este tipo de redes.

## 1.2. Trabajos relacionados

Las redes móviles ad hoc presentan características especiales que deben tenerse en cuenta a la hora de implementar un protocolo de encaminamiento. Existen muchas soluciones (RFC 3626 [6], RFC 3561 [44], 4728 [31], 3684 [40], ...). Todas estos protocolos son soluciones válidas pero suelen tener como premisas de diseño unas determinadas características de topología y unos escenarios particulares, no siendo especialmente adecuados si hay cambios drásticos en la topología dinámica de la red ad hoc.

Existe un grupo de algoritmos o protocolos de encaminamiento denominados *bioinspirados* que tienen como característica esencial el hecho de ser adaptativos, algo especialmente reseñable en este tipo de ambientes. Dentro de estos algoritmos han sido especialmente referenciados en la literatura los basados en el concepto de inteligencia *swarm*, esto es, aquellos que aplican el comportamiento social de los insectos y de otros animales para resolver problemas. El algoritmo *Ant Colony Optimization* (ACO) o algoritmo de optimización de la colonia de hormigas constituye el punto de partida de estos algoritmos. Los algoritmos ACO se basan en el comportamiento colectivo de las hormigas en su búsqueda del alimento y en llevarlo de vuelta al hormiguero. Existen diversos trabajos que realizan propuestas de encaminamiento ACO encontrándose protocolos proactivos, reactivos e híbridos.

Como protocolos de encaminamiento ACO proactivos cabe señalar: *Probabilistic Emergent Routing Algorithm* (PERA) [2] y *Ant Routing Algorithm for Mobile Ad hoc networks* (ARAMA) [27].

Como protocolos de encaminamiento ACO reactivos cabe señalar: *Ad hoc Networking with Swarm Intelligence* (ANSI) [47], *Ant-Colony-Based Routing Algorithm* (ARA) [24], *Position based Ant Colony Routing* (POSANT) [33], *Ant Colony based Multi-path QoS-aware Routing* (AMQR) [35], *Distributed Ant Routing* (DAR) [49], *Ant-based Distributed Route Algorithm* (ADRA) [55], *Ant-based Energy Aware Disjoint Multipath Routing Algorithm* (AEADMRA) [54] e *imProved Ant Colony Optimization algorithm for mobile ad hoc NETworks* (PACONET) [42].

Como protocolos de encaminamiento ACO híbridos cabe señalar: *Ant-AODV* [37], *Hybrid ant colony OPTimization routing algorithm for mobile ad hoc NETwork* (HOPNET) [52] y *AntHocNet* [8] [11] [12], protocolo sobre el que se apoya la propuesta realizada en este trabajo.

### **1.3. Estructura del trabajo**

El resto del trabajo está organizado en 9 capítulos con la estructura que se comenta a continuación.

El Capítulo 2 realiza un estado del arte de las redes ad hoc incluyendo un repaso cronológico de la evolución de las redes ad hoc, un análisis de las características básicas de este tipo de redes, una presentación del protocolo de comunicación que actualmente se utiliza en ellas, una clasificación de las redes ad hoc y un resumen de las principales aplicaciones de las redes móviles ad hoc.

El Capítulo 3 analiza la problemática del encaminamiento en redes móviles ad hoc presentándose una clasificación de los protocolos de encaminamiento así como las características de los más representativos.

El Capítulo 4 presenta el algoritmo de optimización de la colonia de hormigas ó *Ant Colony Optimization* (ACO). Este algoritmo constituye una referencia indispensable para entender la propuesta de protocolo de encaminamiento que se presenta en este trabajo.

El Capítulo 5 aborda en detalle el denominado encaminamiento ACO en redes móviles ad hoc que consiste en la aplicación directa del algoritmo ACO al problema del encaminamiento en este tipo de redes.

El Capítulo 6 describe el protocolo AntHocNet, punto de partida del protocolo propuesto, analizando sus características más relevantes y presentando las diversas versiones del mismo.

El Capítulo 7 presenta el protocolo de encaminamiento ACO híbrido basado en AntHocNet denominado AntOR (*Ant Optimized Routing*) que constituye la propuesta de este trabajo.

El Capítulo 8 contiene los resultados obtenidos de las simulaciones realizadas en el software NS-3 [51].

Por último, el Capítulo 9 muestra las principales conclusiones extraídas de este trabajo así como algunas líneas futuras de trabajo.



## 2. REDES MÓVILES AD HOC

---

Una red móvil ad hoc es un conjunto de nodos móviles que se comunican entre sí a través de enlaces inalámbricos (*wireless*). Al contrario de las redes convencionales, una red móvil ad hoc no necesita la existencia de una infraestructura previa ya que cada nodo se apoya en los demás para conseguir comunicarse con otro creando la llamada comunicación multisalto.

El capítulo se estructura de la siguiente forma: En el apartado 2.1 se hace un repaso cronológico de la evolución de las redes ad hoc. En el apartado 2.2 se analizan las características básicas de este tipo de redes. En el apartado 2.3 se presta atención al protocolo de comunicación que actualmente se utiliza en este tipo de redes, el IEEE 802.11 [29]. El apartado 2.4 realiza una clasificación de las redes ad hoc. Por último, el apartado 2.5 resume sus principales aplicaciones.

### 2.1. Evolución Histórica

En muy pocos años, el campo de las redes ad hoc ha tenido una rápida expansión visible en la proliferación de dispositivos inalámbricos de bajo coste como ordenadores portátiles, asistentes personales digitales (PDAs), teléfonos móviles, etc.

A comienzos de los años 70 un trabajo pionero en radio de la Universidad de Hawai introduce el primer sistema que usa el medio de la radio para la transmisión de información. Conocido ampliamente como ALOHA [1], fue desarrollado por Abramson y Kuo.

El trabajo realizado en Hawai llevó en 1972 al desarrollo de una arquitectura distribuida consistente en una red de difusión de radio con mínimo control

central llamada PARNET [16] bajo el patrocinio de DARPA (*Defense Advanced Research Projects Agency*). El proyecto ayudó a establecer el concepto de redes móviles ad hoc. PARNET permitía la comunicación directa entre usuarios móviles sobre grandes áreas geográficas, ancho de banda compartido y protección contra los efectos de múltiples caminos.

Los rápidos avances de la tecnología de la radio en los años 70 provocó la aparición de múltiples sistemas de comunicación móvil como teléfonos celulares e inalámbricos, sistemas de radio búsqueda, satélites móviles, etc.

Posteriormente, DARPA desarrolló el proyecto SURAN (*Survivable Radio Networks*) en 1983 que trata las tareas de escalabilidad de la red, seguridad, capacidad de proceso y gestión de energía. Se dedicaron esfuerzos para desarrollar dispositivos de bajo coste y con poco gasto de energía que pudieran soportar los avanzados protocolos de encaminamiento, escalar a miles de nodos las redes y dar soporte para ataques a la seguridad. El resultado fue la aparición de la tecnología conocida como LPR (*Low-cost Packet Radio*) en 1987.

A mitad de los 90 se produce un nuevo avance con la llegada de tarjetas de radio 802.11 para ordenadores personales y portátiles. En los artículos [19] [29] se propone por primera vez la idea de una colección de *hosts* móviles con una infraestructura mínima, y el IEEE (*Institute of Electrical and Electronic Engineers*) acuña el término *redes ad hoc*.

Durante el mismo tiempo, el Departamento de Defensa de Estados Unidos continuaba trabajando con proyectos como el GloMo (*Global Mobile Information Systems*) o el NTDR (*Near-term Digital Radio*). El objetivo del GloMo era permitir la conectividad multimedia de tipo *Ethernet*, en cualquier momento y en cualquier lugar, entre los dispositivos inalámbricos. NTDR son protocolos que se basan en dos componentes: agrupamiento y encaminamiento. Los algoritmos

de agrupamiento organizan dinámicamente una red en líderes de grupo y miembros de grupo. Los líderes forman la columna vertebral de la red y los miembros se comunican entre sí a través de dicha columna. NTDR inicialmente fue un prototipo para la Armada de los Estados Unidos y en la actualidad algunos países lo utilizan como base para otros protocolos.

La definición de estándares como IEEE 802.11 [19] provocó el rápido crecimiento de las redes móviles en campos no sólo militares, sino también en el mundo comercial.

## 2.2. Características

Como su propio nombre indica la característica principal de una red móvil ad hoc es la movilidad de los nodos, que pueden cambiar de posición rápidamente. La necesidad de crear redes de forma rápida en lugares sin infraestructura suele implicar que los nodos exploren el área y, en algunos casos, se deban unir para conseguir un objetivo. El tipo de movilidad que desarrollen los nodos puede tener una influencia a la hora de elegir el protocolo de encaminamiento que aumente el rendimiento de la red.

Otro de los aspectos importantes en las redes ad hoc es la llamada auto-organización que se estudia en profundidad en [20]. La idea principal se basa en la coordinación y colaboración de todos los nodos de la red para conseguir un mismo objetivo. Se han propuesto varios métodos de auto-organización para redes en general y para redes ad hoc en particular. La auto-organización puede desglosarse en las siguientes capacidades:

- *Auto-reparación*: mecanismos que permitan detectar, localizar y reparar automáticamente los fallos siendo capaces de distinguir la causa del error. Por ejemplo, sobrecarga o mal funcionamiento.

- Auto-configuración: métodos de generación de configuraciones adecuadas en función de la situación actual dependiendo de las circunstancias ambientales. Por ejemplo, conectividad o parámetros de calidad de servicio.
- Auto-gestión: capacidad de mantener dispositivos o redes dependiendo de los parámetros actuales del sistema.
- Adaptación: adecuación a los cambios de las condiciones ambientales. Por ejemplo, cambio en el número de nodos vecinos.

A continuación se presentan el resto de características de las redes móviles ad hoc:

- Ausencia de infraestructura: Al contrario que las redes convencionales que cuentan con la existencia de elementos físicos, las redes móviles se forman autónomamente.
- Topología dinámica: Los nodos se pueden mover arbitrariamente haciendo que algunos enlaces se destruyan y otros se creen cuando un nodo se acerque a otros que antes tenía fuera de su alcance.
- Ancho de banda limitado: En la mayoría de las ocasiones será menor que el de una conexión cableada, afectado además por las interferencias de las señales electromagnéticas.
- Variación en la capacidad de los enlaces y los nodos: Los nodos pueden disponer de varias interfaces de radio que difieren entre sí en capacidad de transmisión/recepción y en la banda de frecuencia en la que trabajan. Esta característica complica el desarrollo de los protocolos de encaminamiento en gran medida.

- Conservación de energía: Algunos o todos los nodos de una red móvil ad hoc son alimentados por baterías y no tienen posibilidad de recargarlas. Para estos nodos, el criterio más importante a la hora de diseñar sistemas y protocolos será la optimización de la conservación de energía.
- Escalabilidad: En muchas aplicaciones las redes ad hoc pueden llegar a tener miles de nodos lo que conlleva dificultad en tareas como direccionamiento, encaminamiento, gestión de localización, gestión de configuración, interoperabilidad, seguridad, etc.
- Falta de seguridad: La seguridad juega un papel importante en las redes ad hoc dado el carácter vulnerable de los enlaces inalámbricos que se forman. Los protocolos de encaminamiento deben proporcionar una comunicación segura. Existen áreas de investigación en este sentido que sugieren incluir datos de sensores externos e información geográfica y topográfica en el propio algoritmo de encaminamiento.
- Encaminamiento multisalto: Los nodos actúan como *routers* para retransmitir los paquetes intercambiados entre nodos cuyo alcance no permite una comunicación directa.
- Entorno imprevisible: Las redes ad hoc pueden darse en terrenos en los que las situaciones no son las más óptimas debido a condiciones peligrosas o desconocidas. Pueden darse casos donde los nodos se destruyan, se estropeen o comiencen a producir fallos.
- Comportamiento de los terminales: Una de las principales claves para que una red móvil ad hoc tenga un funcionamiento adecuado es la confianza que cada nodo debe tener sobre los demás. Sin esta confianza sería imposible crear un protocolo de encaminamiento ya que la información

debe transmitirse por varios nodos intermedios. Normalmente, los protocolos de encaminamiento que descubren los terminales intermedios se basan en las respuestas que dan los nodos sobre el coste de la comunicación. Existen nodos maliciosos que podrían intencionadamente informar de forma incorrecta sobre los costes con la finalidad de recibir todos los paquetes, poder manipularlos, alterarlos o incluso eliminarlos. Algunas soluciones al respecto se encuentran en [46].

### 2.3. Estándar IEEE 802.11

El IEEE 802.11 es un estándar de protocolo de comunicaciones que define el uso de los dos niveles más bajos de la arquitectura OSI (capa física y capa de enlace de datos), especificando sus normas de funcionamiento en una red inalámbrica. La primera propuesta de este estándar mantenía tasas de transmisión de 1 y 2 Mbps en la banda de frecuencias ISM (*Industrial Scientific and Medical*), situada en 2.4 GHz. Además, se especificaban como tecnologías en la capa física los infrarrojos y el canal radio. Con los años se ha llegado a distintas versiones del estándar. Se citan los más importantes a continuación:

- IEEE 802.11a: hasta 54 Mbps a 5 GHz. Utiliza la tecnología OFDM (*Orthogonal Frequency-Division Multiplexing*) en la capa física.
- IEEE 802.11b: hasta 11 Mbps a 2.4 GHz. Actualmente es el más utilizado. Utiliza la tecnología DSSS (*Direct Sequence Spread Spectrum*) en la capa física.
- IEEE 802.11e: pretende proporcionar calidad de servicio QoS (*Quality of Service*) para su uso en servicios como VoIP (Voz sobre IP) y *Streaming*. Una aproximación para otorgar calidad de servicio es la de diferenciar los paquetes clasificándolos en un número pequeño de tipos de servicios y

utilizar mecanismos de prioridad para proporcionar una calidad de servicio adecuada a cada tráfico.

- IEEE 802.11f: desarrolla especificaciones para la implementación de puntos de acceso y sistemas de distribución para evitar problemas de interoperabilidad entre distintos fabricantes y distribuidores de equipos.
- IEEE 802.11g: hasta 54 Mbps a 2.4 GHz. Soporta tanto OFDM como DSSS en la capa física

## 2.4. Clasificación

La terminología de redes ad hoc aún no está muy asentada y no existe una clasificación clara. A continuación se exponen varias clasificaciones situando el lugar en el que se encuentran las redes móviles ad hoc.

Existen redes ad hoc *con infraestructura* donde los nodos se mueven mientras se comunican con una estación base fija. Cuando un nodo se mueve fuera del rango de una estación fija entra en el alcance de otra estación. Por otro lado se encuentran las redes ad hoc *sin infraestructura* donde no existen estaciones base fijas y todos los nodos de la red necesitan actuar como *routers*. Las redes móviles ad hoc son redes ad hoc sin infraestructura.

Otra clasificación de las redes ad hoc incluye las *redes de un solo salto* y las *redes multisalto*. Los nodos de las redes de un solo salto se comunican únicamente con los nodos que tiene a su alcance. En las redes ad hoc multisalto, los nodos que no pueden comunicarse directamente utilizan nodos intermedios para retransmitir la información. Las redes móviles ad hoc son redes ad hoc multisalto.

Por último hay una clasificación que incluye las redes móviles ad hoc como un tipo independiente. Se incluyen tres tipos de redes ad hoc:

- Redes móviles ad hoc.
- Redes de sensores: También denominadas WSN (*Wireless Sensor Networks*). Formadas de dispositivos sensoriales, generalmente compuestos por un sensor tradicional y un conversor analógico-digital. La unidad de proceso está compuesta de un microprocesador y una pequeña memoria. Pueden incluir sistemas de localización y sistemas de movilidad. En estas redes el número de nodos suele ser mucho mayor que en una red móvil ad hoc pero la movilidad se considera escasa o nula (solamente cambia la topología con la pérdida o desconexión de nodos). Es habitual el flujo de información desde muchos orígenes hasta un nodo llamado sumidero (*sink*) que se encarga de procesar la información y enviársela al destino.
- Redes híbridas: También denominadas mixtas, son redes ad hoc que usan infraestructuras IP si están disponibles.

A su vez las redes móviles ad hoc se pueden dividir en dos tipos en función de si están conectadas o no a otras redes:

- Redes móviles ad hoc autónomas: Son redes que no están conectadas a ninguna otra red. Los nodos de la red se pueden identificar unívocamente a través de una dirección IP con la única premisa de que sea distinta a la de cualquier otro nodo de la red.
- Redes móviles ad hoc subordinadas: Son redes conectadas a una o más redes externas. Se obliga a usar un direccionamiento IP topológico correcto y encaminable globalmente. Un ejemplo típico de red móvil ad hoc subordinada es una red móvil ad hoc que es parte de Internet.

## 2.5. Aplicaciones

Es fácil encontrar situaciones donde se ve la utilidad de las redes móviles ad hoc. Uno de los ejemplos más clásicos (aunque también discutido) es una reunión de trabajo: un grupo de personas con ordenadores portátiles o PDAs. Son de distintas empresas y por tanto sus direcciones son distintas. Tal vez en la sala haya acceso a Internet y puedan usar por ejemplo IP móvil, pero ¿para qué pasear sus datagramas por toda la ciudad o todo el país cuando están en la misma habitación? Sus equipos probablemente estén dotados de puertos de infrarrojos o Bluetooth que les permitan formar una red para la ocasión. En algunos casos, simplemente no habrá infraestructuras de apoyo. Pensemos en poblaciones aisladas o de orografía difícil, situaciones de emergencia, desastres naturales donde las infraestructuras hayan desaparecido, etc.

Otro ejemplo son las denominadas PAN (*Personal Area Networks*): redes formadas por los dispositivos de una persona, como su reloj, su agenda y su teléfono móvil. Una red así puede querer entrar en contacto con la red de otra persona que en ese momento esté próxima.

La capacidad de desplegarse inmediatamente y la no dependencia de un único punto de fallo hace a estas redes muy interesantes para el uso militar. El campo militar es posiblemente el más desarrollado actualmente. Así, el ejército estadounidense ya dispone de un sistema basado en este tipo de redes, el FBCB2 (*Force XXI Battle Command, Brigade-and-Below*). Uno de sus objetivos es distinguir las fuerzas propias de las fuerzas del enemigo, ofreciendo a los soldados una visión del campo de batalla similar a la de un videojuego. Los equipos de la generación inmediatamente anterior estaban basados en comunicaciones por satélite, con latencias de cinco minutos. En abril de 2003 el FBCB2 se utilizó en la Segunda Guerra del Golfo, lo que supuso probablemente el primer uso bajo fuego real de una red móvil ad hoc.

Otro motivo por el que una red móvil ad hoc puede ser ventajosa es el coste. Aunque exista una infraestructura de red, si pertenece a una entidad ajena es muy posible que nos cobre por su uso, mientras que si tenemos nuestros equipos desplegados dispondremos ya de una red sin coste adicional. Por ejemplo, los coches que pasan por una autopista podrían formar fácilmente una red móvil ad hoc, independiente de su capacidad de conectarse a otras redes como GSM o similar. Por último, supongamos que tenemos estaciones capaces de comunicarse empleando un satélite. Estos equipos de comunicaciones son caros, pero bastaría con que algunos tengan capacidad de conectarse al satélite para que todos dispusieran de conectividad. Y no todos los capaces de conectarse al satélite necesitarían estar conectados simultáneamente.

Resulta evidente que el potencial de este tipo de redes es muy grande.

## 3. ENCAMINAMIENTO EN REDES MÓVILES AD HOC

---

### 3.1. Protocolos de Encaminamiento en Redes Móviles Ad Hoc

En redes móviles ad hoc los protocolos convencionales o bien tendrán un rendimiento muy pobre, o bien serán simplemente inaplicables. Como alternativa se desarrollan protocolos específicos de encaminamiento. Con frecuencia se les denomina de nivel 2.5, ya que es habitual encontrarlos por encima de protocolos de enlace como IEEE 802.11 y por debajo del protocolo de red IP.

El concepto de encaminamiento básicamente comprende dos actividades. En primer lugar, determinar los caminos óptimos y, en segundo lugar, transferir los grupos de paquetes de información a través de la red. Los algoritmos utilizan varias métricas para calcular el mejor camino para que los paquetes lleguen a su destino. Estas métricas son medidas estándar como podría ser el número de saltos que son usados por el algoritmo para determinar el camino óptimo. El proceso para determinar el camino inicializa y mantiene tablas de encaminamiento que contienen la información total de cada ruta. La información que se almacena para cada ruta varía de un algoritmo a otro.

Las redes móviles ad hoc se construyen de forma dinámica cuando un conjunto de nodos crean rutas entre sí para conseguir la conectividad entre ellos. Los nodos de la red móvil ad hoc pueden actuar como origen o destino de una comunicación, pero también como *routers* cuando una relación entre nodos no se puede realizar directamente por motivos de alcance. De esta forma se crean comunicaciones multisalto. Un protocolo de encaminamiento de una red móvil ad hoc necesita proveer un mecanismo que mantenga las rutas hacia los

destinos frente al movimiento de los nodos que puede provocar que las rutas se destruyan, y sea necesario encontrar una ruta alternativa para mantener la comunicación entre los nodos.

El objetivo de un protocolo de encaminamiento para redes móviles es conseguir el envío de un mensaje de un nodo a otro sin existir un enlace directo. La mayoría de protocolos de encaminamiento para redes móviles ad hoc provienen de adaptaciones realizadas sobre protocolos de redes fijas, siendo su principal problema la cantidad de fallos que se producen en la comunicación debido a la movilidad de los nodos.

Los protocolos de encaminamiento para redes móviles ad hoc deben satisfacer básicamente los siguientes criterios [4]:

- *Señalización mínima*: La reducción de los mensajes de control ayuda a conservar la capacidad de las baterías y la comunicación de los nodos.
- *Mantenimiento dinámico de topología*: El algoritmo deberá ser capaz de localizar una nueva ruta rápidamente cuando se rompe un enlace.
- *Libre de bucles*: Se pretende evitar el problema de tener paquetes circulando perdidos por la red.
- *Capacidad multisalto*: Debe asegurarse el reenvío de paquetes a través de los nodos de la red dado que habitualmente el destino no se encuentra dentro del alcance de la fuente.
- *Tiempo de procesamiento mínimo*: Se requieren algoritmos con cálculos computacionales que no sean excesivamente complejos para disminuir el tiempo de procesamiento y alargar de esta forma el tiempo de vida de la batería.

Además, debe admitir diversos modos de operación [36]:

- Distribuido: Propiedad esencial de las redes MANET.
- Inactivo: Los protocolos de encaminamiento deberán estar preparados para afrontar aquellos periodos de tiempo en los cuales los nodos frenan su actividad y permanecen inactivos para ahorrar energía.
- Bajo demanda: La adaptación del encaminamiento a los patrones de tráfico particulares de cada situación hace posible reducir el gasto de ancho de banda y energía, aunque se amplía el tiempo de obtención de la ruta.
- Soporte de enlaces unidireccionales: Los protocolos de encaminamiento en muchas ocasiones han sido diseñados y funcionan correctamente sólo con enlaces bidireccionales y esto no debería ser así, porque en la práctica podemos encontrarnos con la existencia de enlaces unidireccionales que sean clave para el intercambio de información en redes móviles ad hoc.

Se han diseñado numerosos protocolos de encaminamiento para redes MANET atendiendo a estos criterios.

La finalidad del MANET WG [38] del IETF es estandarizar la funcionalidad de un protocolo de encaminamiento IP para aplicaciones de encaminamiento inalámbrico dentro de topologías tanto estáticas como dinámicas como consecuencia de la movilidad de los nodos u otros factores. Los enfoques están destinados a ser relativamente generales pues deben ser adecuados en múltiples entornos inalámbricos y hardware y dirigidos a escenarios donde las redes móviles ad hoc estén desplegadas en la frontera de una infraestructura IP. Las infraestructuras *mesh* híbridas (por ejemplo, una mezcla de *routers* móviles y fijos) deberían ser soportados por las especificaciones de las redes móviles ad hoc.

## 3.2. Clasificación

Desde que se empezaron a estudiar las redes móviles ad hoc se han propuesto diversas clasificaciones de los protocolos de encaminamiento que se resumen en [30].

En función de la información de estado que almacenan los nodos de la red los protocolos pueden ser clasificados en protocolos basados en la topología y protocolos basados en el destino. En los primeros cada nodo toma decisiones basándose en una completa información de la topología de la red. Los segundos son protocolos que manejan vectores de distancias, en los que cada nodo intercambia con sus vecinos las distancias que conoce a otros nodos.

Otra clasificación propone dividir los protocolos en función de la estructura, diferenciándose varios niveles.

El primer nivel se refiere a la homogeneidad o heterogeneidad de las funciones de los nodos en el encaminamiento, distinguiéndose dos tipos:

- Protocolos uniformes o de estructura plana: Ningún nodo de la red realiza un papel distinto al de los demás, todos ellos envían y responden a los mensajes de control del mismo modo.
- Protocolos no uniformes: Típicos de estructuras jerárquicas en las que algunos nodos desarrollan papeles especiales e incluso pueden dotarse de capacidades particulares en términos de cómputo, energía o almacenamiento entre otros. Esto les permite soportar algoritmos más complejos, reducir la sobrecarga debida a la comunicación y ofrecer la posibilidad de balanceo de carga mientras mantienen sus características, incluso ante incrementos del número de nodos en la red. Por el contrario,

generan cierto coste de mantenimiento de la estructura y necesitan en muchos casos la disponibilidad de nodos heterogéneos.

Dentro de estas dos categorías anteriores, los protocolos presentan una nueva peculiaridad relativa al procedimiento adoptado para el descubrimiento del camino a establecer y su mantenimiento. Ésta clasificación, sin duda, es la más difundida surgiendo los siguientes tipos de protocolos:

- Protocolos proactivos: En este tipo de encaminamiento cada nodo mantiene información de cómo llegar a cualquier otro nodo de la red e intercambia esta información con todos sus vecinos. La información de encaminamiento es normalmente almacenada en un número diferente de tablas. Periódicamente se actualizan las tablas si la topología de red cambia. La diferencia entre los protocolos de este tipo se encuentra en la forma de actualizar y detectar la información de encaminamiento y el tipo de información que se guarda en cada tabla. La ventaja que aportan estos protocolos es la baja latencia ya que las rutas están siempre disponibles. Sin embargo, esto conlleva un consumo de energía muy alto en los nodos y se puede producir una sobrecarga de mensajes en la red debido a la inundación periódica de mensajes. DSDV (*Destination-Sequenced Distance-Vector*) [45], WRP (*Wireless Routing Protocol*) [39], GSR (*Global State Routing*) [5], FSR (*Fisheye State Routing Protocol*) [21], OLSR (*Optimized Link State Routing*) [6] o TBRPF (*Topology Broadcast Reverse Path Forwarding*) [40] son ejemplos de protocolos de encaminamiento proactivos. En general, estos protocolos tratan de evitar bucles en las rutas, consumo excesivo de memoria y reducción del tamaño de los paquetes que contienen la información de las tablas de encaminamiento. Dentro de los protocolos proactivos se pueden distinguir dos subtipos de protocolos según su comportamiento: los conducidos por eventos (*event-driven*), que envían paquetes con información sobre las rutas sólo cuando

éstas sufren algún cambio, y los que refrescan la información periódicamente (*regular updated*).

- Protocolos reactivos: Estos protocolos tratan de reducir la sobrecarga que producen los protocolos proactivos. Para ello proponen que los nodos de la red móvil ad hoc, cuando no tienen una ruta a un destino, la calculen sólo cuando es necesaria, es decir, cuando el nodo tenga que comenzar un intercambio de paquetes con el destino. El descubrimiento de una ruta normalmente se realiza por inundación de mensajes de solicitud por toda la red. Estos protocolos conllevan una alta latencia, provocada por el descubrimiento de rutas. Sin embargo, la sobrecarga de mensajes por la red se reduce. AODV (*Ad hoc On-demand Distance Vector*) [44], DYMO (*DYnamic Manet On-demand*) [4], DSR (*Dynamic Source Routing*) [31], ROAM (*Routing On-demand Acyclic Multi-path*) [48], LMR (*Lightweight Mobile Routing*) [7], LAR (*Location Arder Routing*) [34] o TORA (*Temporally-Ordered Routing Algorithm*) [43] son algunos de los protocolos de encaminamiento reactivos. La mayoría de ellos tienen el mismo coste de encaminamiento en el peor escenario posible ya que casi todos siguen la misma filosofía para el descubrimiento de rutas.
- Protocolos híbridos: Combinando los protocolos proactivos y reactivos nacen los protocolos híbridos que pretenden minimizar los inconvenientes de ambos. La idea de estos protocolos es que los nodos de la red trabajen de forma proactiva con los nodos más cercanos y de forma reactiva con el resto de nodos. La parte reactiva controla la sobrecarga y el consumo de memoria al calcular las rutas sólo cuando son necesarias. En contraste, la parte proactiva necesita actualizar periódicamente la información almacenada y mantiene rutas que quizás nunca serán utilizadas, añadiendo una innecesaria sobrecarga. El caso más conocido de protocolo híbrido es ZRP (*Zone Routing Protocol*) [25].

El Grupo de Trabajo MANET tiene previsto desarrollar dos especificaciones de protocolo de encaminamiento estándar, denominadas *Reactive MANET Protocol* (RMP) y *Proactive MANET Protocol* (PMP), si bien también puede decidir un enfoque mixto. Soportará IPv4 e IPv6, requisitos de seguridad y otros aspectos, y prestará atención especial al protocolo OSPF-MANET [39] que viene desarrollando el OSPF WG [41]. El OSFP WG desarrolla extensiones del protocolo OSPF para diferentes escenarios, siendo OSPF-MANET la extensión de OSPF a redes móviles ad hoc.

### 3.3. Protocolos reactivos

#### 3.3.1. AODV

Creado por Charles E. Perkins (*Ad Hoc On-Demand Distance Vector* [44]) como evolución de su anterior protocolo DSDV (*Destination-Sequenced Distance-Vector Routing* [45]). Mantuvo la idea de mantener números de secuencia y tablas de encaminamiento pero añadió el concepto de encaminamiento bajo demanda (protocolo reactivo), es decir, sólo se almacena información de los nodos que intervengan en la transmisión de datos. La optimización primordial que se consiguió en relación a su anterior diseño fue el decremento del tiempo de proceso, disminución del gasto de memoria y reducción del tráfico de control por la red. Además AODV es muy cuidadoso con las rutas, manteniéndolas en caché mientras son necesarias e inhabilitándolas cuando su información no es útil.

El encaminamiento se produce salto a salto, mediante un vector de distancias, este protocolo se caracteriza de forma global en que:

- Nadie tiene un grafo completo del estado de la red.

- Para cada posible destino sólo se conocen el primer salto por donde debe encaminarse y la distancia a la que se encuentra.

Para distinguir la información moderna de la antigua, se emplean horas lógicas (identificador de nodo, número de secuencia). Esto se usa de forma local, es decir, no hay un reloj absoluto. Toda la información que se incluye en la tabla de estado de enlaces o tabla de encaminamiento lleva la hora lógica de quien la generó, ya que es posible que cierta información recién recibida sea más antigua que otra recibida anteriormente.

Este mecanismo se realiza debido a que toda información acaba caducando a los pocos segundos.

Seguidamente se describe el funcionamiento general de este protocolo:

Cuando se necesita una ruta desde un origen hasta un destino, se inunda la red con peticiones RREQ (*Route REQuest*). Cuando un RREQ llega al destino buscado, o a alguien que conoce una ruta para el destino, se genera una respuesta RREP (*Route REsPonse*). Este RREP sabe volver al origen porque la inundación del mensaje RREQ fue creando el camino de vuelta. Cuando el RREP va volviendo al origen, va creando el camino de ida. Una vez que el origen ha recibido el RREP, ya puede enviar datagramas, que seguirán el camino de ida generado por el envío de mensajes anteriores.

Para la formación del camino de vuelta, cuando las peticiones de ruta inundan la red, las tablas apuntan hacia el origen. Si un nodo no sabe responder la petición de ruta, la reenvía. Las tablas almacenan, en cada entrada, la hora lógica en el origen. La petición de ruta viaja en sentido inverso.

En cuanto a la formación del camino de ida, el RREQ tiene éxito si llega al destino o a alguien que sabe llegar al destino. Cuando el RREP vuelve, las tablas

de los nodos apuntan hacia el destino. También se conserva la hora lógica del destino. Si transcurrido cierto tiempo no llega un RREP, se borra la información.

Cada petición RREQ lleva encapsulado:

- Origen de la petición y destino buscado.
- Identificador de petición (para controlar inundación).
- Hora lógica en que el destino envió la última ruta conocida (cero si no se conocía ninguna). Esto garantiza que nadie responda una ruta más vieja de la ya conocida.
- Hora lógica en que el origen envió la petición (Si el origen se mueve y envía nuevas RREQ, la información sobre el nuevo camino de vuelta debe reemplazar a la información antigua).
- TTL (*Time To Live*).

Cada respuesta RREP (viajando desde el destino hacia origen) incluye:

- Origen de la petición y destino buscado.
- Hora lógica del destino.
- Distancia del origen al destino.

Para el mantenimiento de rutas se produce el siguiente mecanismo. La detección de un movimiento se percibe como un cambio en el vecindario, además cada nodo mantiene información sobre sus vecinos con mensajes HELLO periódicos, de TTL = 1.

Cada nodo mantiene información sobre rutas vivas (activas), por este motivo si un cambio en el vecindario afecta a una ruta viva, se notifica un error para generar nuevas búsquedas. Por otro lado, la desaparición de un vecino que no participa en ninguna ruta viva no provoca ninguna acción.

### 3.3.2. DYMO

DYMO (*Dynamic MANET On-demand* [4]) es un protocolo en el que el descubrimiento de nuevas rutas se realiza mediante paquetes de control especiales.

Cuando es necesario descubrir nuevas rutas, el nodo origen manda mediante difusión un paquete especial llamado *Route Request* (RREQ), que es una petición de ruta hacia el nodo destino. Este mensaje se retransmite de nodo a nodo, y cada uno de estos nodos intermedios que reciben y reenvían el mensaje lo modifican anotándose a ellos como parte de la ruta hacia el nodo destino.

Cuando este mensaje RREQ finalmente llega hasta el nodo destino, crea un paquete llamado *Route Reply* (RREP), que se comporta igual que el RREQ: es reenviado por la red hasta llegar al nodo origen.

De esta forma, al acabar el proceso, se habrán establecido dos rutas para llegar desde origen a destino, y de destino a origen.

El mantenimiento de las rutas también se tiene en cuenta en DYMO. Cuando un nodo recibe un paquete de datos que debe retransmitir hacia un destino al cual no tiene ruta conocida, notifica al origen mediante un paquete *Route Error* (RERR). Al recibir este RERR, el nodo origen sabrá que la ruta ya no es válida y la borrará.

### 3.3.3. DSR

DSR (*Dynamic Source Routing* [31]), es un protocolo para direcciones IPv4 que funciona totalmente bajo demanda. No utiliza en ningún momento el envío de paquetes de forma periódica, por lo que este protocolo tiene una sobrecarga mínima.

Se basa en dos mecanismos que trabajan juntos: el de descubrimiento de rutas (*Route Discovery*) y el del mantenimiento de las rutas (*Route Maintenance*).

El primero de los mecanismos se encarga de descubrir la ruta del nodo origen al nodo destino solamente cuando se envíe un mensaje. Para ello envía en difusión un paquete *Route Request* y los nodos intermedios lo retransmiten de nuevo también en difusión, de esta forma habrá momentos en los que un nodo reciba más de una vez el mismo paquete, para que esto no sea un problema, estos mensajes llevan un número de secuencia que va aumentando en cada salto e indica cual es el que tiene la mejor ruta. Cuando el paquete llega a su destino, el receptor envía un *Route Reply* al origen por el mismo camino que el paquete ha llegado. Gracias a esto las rutas del emisor al receptor quedan descubiertas.

Cada nodo, además de guardar esta ruta, guarda en su caché otras rutas alternativas, para que en caso de pérdida de la ruta principal, se pueda invocar de nuevo al proceso de *Route Discovery* o poder utilizar una de las rutas alternativas. Estas rutas, también pueden servir para mejorar propiedades de la red, como el balanceo de carga o la robustez.

En el caso de que un nodo intermedio ya tenga la ruta en su caché al nodo destino, en vez de retransmitirla, envía un *Route Reply* al emisor con la información de la ruta.

El *Route Maintenance* se encarga de que mientras el nodo origen esté enviando algo al destino, si hay un cambio en la red o una pérdida de la ruta, el paquete sea capaz de tomar otro camino, pudiendo elegir entre activar el mecanismo de *Route Discovery* o usar una de las rutas almacenadas en caché.

Este protocolo, minimiza la sobrecarga, ya que no usa ningún tipo de mensajes periódicos, y es totalmente bajo demanda. Si es el caso de que ya se encuentran las rutas aprendidas por los nodos, la sobrecarga añadida es nula.

Si se tiene una topología de mucho movimiento de nodos donde se pierden rutas frecuentemente, la única sobrecarga que se crea, es solamente la que se produce al aprender de nuevo esas rutas perdidas, todos estos cambios no afectan a las que están actualmente en uso.

### **3.3.4. TORA**

TORA (*Temporally-Ordered Routing Algorithm* [43]) es un protocolo especializado en grandes redes MANET que, a diferencia de la mayoría, no es un protocolo basado en vector distancia ni en estado del enlace. Pertenece a un grupo llamado inversión de enlace (*link-reversal*). El protocolo crea diferentes rutas sin bucles para retransmitir el tráfico y puede funcionar como reactivo o proactivo.

TORA es un protocolo distribuido, los nodos guardan información solamente de sus vecinos. Cuando realiza un proceso reactivo, el protocolo va descubriendo las rutas siendo este proceso mucho más conveniente cuando hay grandes cambios de topología de la red, al mismo tiempo puede estar realizando el proceso proactivo, usando las tablas internas de cada nodo, este proceso puede ser más conveniente para rutas que son requeridas frecuentemente.

No usa rutas del camino más corto, algo inusual en la mayoría de los protocolos, lo que hace es fijar unas alturas a cada nodo, de forma que solamente se pueda ir de uno a otro si es de una altura mayor a una menor, y al ser rutas acíclicas se asegura que no pueda subir en ningún momento. No busca la ruta más óptima, sino que intenta minimizar la sobrecarga de la red.

Una de las principales ventajas de este protocolo, es el descubrimiento y la reparación automática de pérdidas de la ruta en la red, si una ruta desaparece, se hace un *link-reversal*, que consiste en cambiar las alturas necesarias de los nodos para establecer un nuevo camino (siempre de alturas mayores a menores) hacia el destino.

### **3.4. Protocolos Proactivos**

#### **3.4.1. DSDV**

DSDV (*Destination-Sequenced Distance-Vector Routing* [44]) es un protocolo *unicast* proactivo adaptado del tradicional RIP (*Routing Information Protocol* [26]). Su principal objetivo es evitar los problemas de bucles en la actualización de las tablas de encaminamiento. Por lo cual añade un nuevo campo a las tablas RIP, el número de secuencia que permite distinguir entre una tabla antigua y una más reciente.

Como su nombre indica, DSDV implementa un algoritmo basado en vector de distancias. Eso significa que mantiene tablas con todos sus destinos accesibles junto con el siguiente salto, la métrica, y un número de secuencia de la entrada en la tabla generado por el nodo destino. Las tablas se mandan con mensajes de difusión de forma periódica o cuando ocurre un cambio significativo de la topología de red. Una ruta es considerada mejor que otra si

tiene un número de secuencia mayor o, en caso de empate, si la distancia al destino es menor.

Cuando un nodo B detecta que la ruta hacia cierto destino D se ha perdido, inunda la red con una actualización de esa entrada en la que se incrementa el número de secuencia en uno y la distancia se marca como infinita.

Cuando un nodo A recibe este mensaje, incorpora a su tabla la actualización de la entrada hacia D a través de B siempre que no tuviera una entrada mejor para alcanzar D.

Para conseguir una cierta consistencia en las tablas de encaminamiento de cada nodo al cambiar la topología de la red, las actualizaciones deben ser frecuentes y suficientemente rápidas para que cada nodo pueda tener una visión realista de la red en un momento dado.

El problema fundamental de DSDV es la elevada sobrecarga de control que genera. Al no haber una especificación estándar, no hay productos comerciales basados en este protocolo.

Sin embargo, es la base sobre cual se han desarrollado otros protocolos como por ejemplo AODV.

### **3.4.2. OLSR**

OLSR (*Optimized Link State Routing Protocol* [6]) es un protocolo de encaminamiento proactivo con un mecanismo de inundación controlada.

Al ser un protocolo proactivo, se deben tener las rutas hacia todos los nodos constantemente actualizadas. Para descubrir los vecinos inmediatos, con los que hay conexión a nivel de enlace, se emiten periódicamente mensajes HELLO,

que aparte de darse a conocer a los nodos con distancia de un salto, informan además de los vecinos inmediatos ya conocidos del nodo.

Una vez que se tiene conocimiento de los vecinos (y de los vecinos de éstos nodos, de distancia dos saltos), el nodo elige un conjunto de nodos para que actúen como sus *Multipoint Distribution Relay* (MPR). Estos nodos deben ser elegidos de forma que garanticen llegar a todos los nodos de dos saltos de distancia, de esta forma abrirán la ruta hacia cualquier nodo de la red.

Además, los MPR tienen la función de anunciar información sobre la topología de la red mediante inundación controlada, para que todos los participantes conozcan la ruta hacia el resto de la red. La inundación se realiza mediante mensajes *Topology Control* (TC), generados por nodos que han sido seleccionados como MPR, y que se reenvían de forma eficiente entre los MPR en lugar de hacerse mediante difusión masiva.

### **3.4.3. TBRPF**

TBRPF (*Topology Dissemination Based on Reverse-Path Forwarding* [40]) es un protocolo que provee un encaminamiento mediante el camino más corto posible, seleccionado mediante una modificación del *algoritmo de Dijkstra*.

Cada nodo tiene un árbol que almacena caminos a todos los nodos alcanzables basado en información parcial guardada en su tabla de rutas.

Para minimizar la sobrecarga, cada nodo sólo transmite parte de su árbol a sus vecinos. Este protocolo combina una transmisión periódica con una transmisión diferencial (transmite sólo si hay un cambio en la red)

El protocolo se diferencia en dos módulos independientes:

- Neighbor Discovery (TND): se basa en el intercambio de mensajes HELLO diferenciales, es decir, solamente contienen los cambios en el enlace, por lo tanto son más pequeños y se pueden enviar más frecuentemente y por lo tanto se detectarán los cambios en los vecinos mucho más rápidamente. Funciona a nivel de interfaces, es decir, cada interfaz enviará sus HELLO independientemente de las otras.
- Módulo de encaminamiento: cada nodo envía un subárbol del árbol de rutas mínimas que contiene, este subárbol se envía periódicamente cada un cierto tiempo y también se envía en actualizaciones diferenciales por cambios en la red, este método asegura la rápida actualización de la topología. Estos cambios del subárbol no son retransmitidos, si no que se espera a que llegue la siguiente transmisión periódica o diferencial para retransmitirlo.

#### 3.4.4. FSR

FSR (*Fisheye State Routing Protocol* [21]) es un algoritmo que introduce el concepto de ámbito multi nivel (*multi-level scope*) para reducir la sobrecarga de la actualización de rutas en redes grandes.

Los nodos almacenan el estado del enlace para cada destino de la red y cada cierto tiempo los estados de los enlaces se actualizan por difusión a sus vecinos. La frecuencia de esta acción viene determinada por la distancia en saltos al destino, es decir, el ámbito relativo a cada uno.

Actualizaciones a nodos lejanos, que son propensos a ser menos usados, se actualizan menor número de veces que los cercanos, evitando una sobrecarga innecesaria.

En el viaje de un paquete la ruta irá siendo más óptima según se acerque al destino.

## 3.5. Protocolos híbridos

### 3.5.1. ZRP

ZRP (*Zone Routing Protocol* [25]) es un protocolo de encaminamiento híbrido, ya que combina las mejores propiedades de los protocolos proactivos y reactivos.

ZRP se basa en separar la red en zonas. Se diferencian claramente una zona cercana o de vecindario, compuesta por los nodos que estén a un máximo de N saltos, y el resto de la red.

En ZRP se usan dos componentes para el encaminamiento. En la zona cercana se usa el componente *IntrA-zone Routing Protocol* (IARP) que actúa como un protocolo proactivo, manteniendo todas las rutas de los nodos que se encuentren a N saltos o menos, siendo N variable. El mecanismo usado para descubrir los nodos vecinos es el intercambio periódico de mensajes HELLO.

Para el encaminamiento global hacia los nodos fuera de la zona interior o cercana, ZRP cuenta con el componente *IntEr-zone Routing Protocol* (IERP), que se comporta como un protocolo reactivo.

Cuando se necesita la ruta hacia un nuevo nodo, usando el componente IERP se pide esta ruta con el mecanismo *Bordercast Resolution Protocol* (BRP). Este mecanismo funciona mandando mensajes de petición de ruta a los nodos que pertenecen a la zona interior pero que además están en el borde, es decir, están al número máximo de saltos para ser considerados de la zona interior.

Si alguno de estos nodos del borde conoce la ruta, mandará un mensaje indicándosela al nodo que originó la petición. Si no es así, reenviarán la petición por toda la red hasta que llegue a un nodo que conozca una ruta hacia el destino. Entonces se enviará la respuesta de vuelta hasta el origen, guardando los nodos intermedios por los que pasa el mensaje para usarlos como ruta hacia el destino buscado.

Como se ha dicho antes, el radio (en número de saltos) de la zona interior es ajustable según las necesidades de la red. Como casos extremos tenemos que si este radio es pequeño, como mínimo uno, ZRP se comportará como un protocolo puramente reactivo. Si por el contrario el radio es infinito, el comportamiento será proactivo.

## 4. EL ALGORITMO ANT COLONY OPTIMIZATION (ACO)

---

### 4.1. Introducción

El algoritmo de optimización de la colonia de hormigas, tradicionalmente conocido por sus siglas en inglés, ACO (*Ant Colony Optimization*), se compone de un conjunto de métodos y técnicas que se aplican en problemas genéricos de optimización.

El algoritmo ACO forma parte de los denominados algoritmos *bioinspirados*, y dentro de éstos, de aquellos basados en el concepto de inteligencia *swarm*, que aplica el comportamiento social de los insectos y de otros animales para resolver problemas.

Mención especial merece el colectivo de las hormigas. La hormiga como individuo único tiene una efectividad limitada, pero, como parte integrante de una colonia bien organizada, se convierte en un agente poderoso que trabaja para el desarrollo de la colonia. Las hormigas viven para la colonia y existen sólo como parte de ella.

Las hormigas tienen la posibilidad de comunicarse, de aprender, de cooperar, de organizarse, etc., y todas juntas pueden llevar a cabo una misión concreta.

Existe un número considerable de investigadores, principalmente biólogos, que han estudiado el comportamiento de las hormigas en detalle. Uno de los patrones de comportamiento que más sorprende de las hormigas es la habilidad de ciertas especies para encontrar la comida por el camino más corto. Los biólogos han demostrado experimentalmente, que se comunican por medio de la feromona.

Las hormigas, en su necesidad de encontrar la comida y traerla de vuelta al hormiguero, exploran una extensa área y se lo indican a otras al hacer el recorrido de vuelta a la colonia.

De esta forma, las hormigas conocen el camino desde su hormiguero hasta su destino, sin necesidad de tener una visión global del terreno. La mayoría de las veces encuentran el camino más corto y se adaptan a los cambios del terreno, demostrando su eficiencia en esta tarea.

Este patrón de comportamiento inspiró a los investigadores a desarrollar algoritmos de optimización que ayudasen a superar los diferentes problemas de encaminamiento existentes.

Los primeros intentos aparecieron cerca de los noventa, y han seguido evolucionando hasta nuestros días.

## **4.2. Experimento del Doble Puente**

El experimento del doble puente [23] consiste en observar cómo se comporta una colonia de hormigas de la especie *Argentina I. humilis* ante el problema de encontrar una fuente de comida.

Estas hormigas están completamente ciegas, realizando la comunicación entre ellas y su entorno por medio de una sustancia química llamada feromona. La señal que dejan en el suelo suele denominarse *pista de feromona* y, lógicamente, es fundamental para su vida social. Este tipo de feromona la usan algunas especies de hormigas para marcar la ruta sobre la tierra y de este modo conocer el camino hacia la comida y el de vuelta al hormiguero. Las hormigas huelen la feromona y eligen la ruta marcada por la mayor concentración de feromona.

Se ha comprobado que hay dos tipos de comportamientos: asentamiento de la pista de feromona y seguimiento de la misma. Estos comportamientos han sido estudiados en experimentos controlados. El más conocido es el denominado *experimento del doble puente* que consiste en analizar el comportamiento de las hormigas que buscan comida en su trayectoria desde un punto *a* (el hormiguero) a otro punto *b* (la comida). En el experimento se cambia 3 veces la relación  $r = l_l / l_s$  de la longitud de los ramales del doble puente, donde  $l_l$  será la longitud de rama más larga y  $l_s$  la longitud de la más corta.

- Primer Caso: El puente tiene dos ramales de igual longitud ( $r = 1$ ). Las hormigas salen del hormiguero y empiezan a moverse libremente buscando un camino que les lleve hasta la comida. Se encuentran en el camino con dos ramales y un porcentaje de ellas elige un ramal y otro porcentaje elige el otro ramal. Se las observó durante un tiempo. El resultado fue que, aunque en la fase inicial la elección fue aleatoria, siguieron eligiendo los ramales en un porcentaje parecido. En la Fig. 4.1 se muestra un esquema del doble puente.

- Segundo Caso: La relación de longitud entre los dos ramales es 2 ( $r = 2$ ). De esta forma el ramal largo es el doble que el corto. Se comprobó que, en un principio, la elección de camino fue aleatoria, como en el caso anterior, pero después de algún tiempo las hormigas eligieron el ramal más corto.

Esto tiene la siguiente explicación: cuando comienza una prueba no hay feromona en ninguno de los dos ramales. Las hormigas no tienen una preferencia y eligen los dos ramales con la misma probabilidad. Como las hormigas depositan feromona mientras caminan, el camino más corto será recorrido más veces, por lo que al cabo de un tiempo la cantidad de feromona depositada en ese recorrido será mayor, y como eligen el que más feromona tiene, que es el más corto, irán por él ya que la mayor cantidad de feromona estimula a más hormigas a seguir ese camino. Sin embargo, no todas las hormigas usan el ramal corto, un pequeño número sigue por el ramal más largo. Esto se puede interpretar como un *mantenimiento de ruta*.

- Tercer Caso: Se estudió el comportamiento de las hormigas cuando se les presenta un solo camino y una vez que se *acostumbran* a él, se les ofreció otro camino más corto. Es decir, inicialmente se les presentó sólo un camino, que fue recorrido por las hormigas durante 30 minutos, y después se les añadió un ramal que les hacía el recorrido más corto. La rama corta fue recorrida esporádicamente, pero la colonia siguió en la rama larga. Y es que la alta concentración de feromona en la rama más larga se mantenía por su lenta evaporación. Este comportamiento seguía reforzando el ramal más largo, a pesar de que ya tenían un ramal más corto.

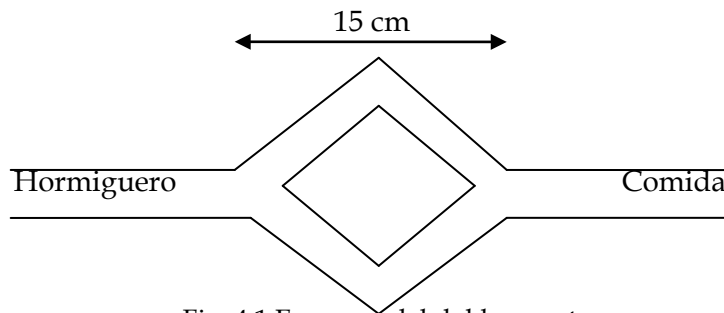


Fig. 4.1 Esquema del doble puente

### 4.3. Hormigas Artificiales

Las hormigas artificiales son agentes que colaboran para resolver problemas computacionales. En nuestro problema concreto la hormiga artificial será un agente computacional simple, que intentará dar soluciones al problema de cálculo del camino mínimo, explotando los rastros de feromona disponibles y la información heurística. En algunos casos ofrecerá soluciones que no son adecuadas y serán *penalizadas*, descartándose o no dependiendo del nivel de error de la solución.

En general, la hormiga artificial tiene las siguientes propiedades:

- Busca solucionar el problema del *coste mínimo*.
- Tiene una memoria interna que almacena información sobre el camino seguido hasta el momento. Esta memoria sirve para:
  - (i) Construir soluciones válidas.
  - (ii) Evaluar la solución generada.
  - (iii) Reconstruir el camino que ha seguido la hormiga.

- Tiene un estado inicial con una o más condiciones de parada (estados finales).
- Comienza en el estado inicial y se mueve construyendo incrementalmente la solución.
- Cuando un nodo está en un estado determinado y ha seguido la secuencia de nodos visitados puede moverse a cualquier vecino.
- El movimiento se lleva a cabo aplicando una regla de transición, que está en función de la feromona depositada, de los valores heurísticos de la memoria interna de la hormiga y de las restricciones del problema.
- Cuando una hormiga se mueve de un nodo a otro puede actualizar el rastro de feromona asociado al *arco* entre los dos nodos. Este proceso se llama *actualización en línea de los rastros de feromona paso a paso*.
- El proceso de construcción acaba cuando se satisface alguna condición de parada.
- Una vez que la hormiga tiene una solución puede reconstruir el camino recorrido y actualizar los rastros de feromona de los arcos/componentes visitados/as utilizando un proceso llamado *actualización en línea a posteriori*.

Las colonias de hormigas naturales y artificiales comparten una serie de características, puesto que interaccionan y colaboran para solucionar una tarea determinada. Seguidamente se resumen las más importantes:

- Tanto las hormigas naturales como las artificiales modifican su *entorno* a través de una comunicación *estigérgica*<sup>1</sup> basada en la feromona. En el caso de las hormigas artificiales, los rastros artificiales de feromona son valores numéricos que están disponibles únicamente de manera local.
- Las hormigas naturales y las artificiales comparten una tarea común: la búsqueda del camino más corto (construcción iterativa de una solución de coste mínimo) desde un origen (el hormiguero) hasta un estado final (la comida).
- Las hormigas artificiales construyen las soluciones de forma reiterada aplicando una estrategia de transición local estocástica o probabilística para moverse entre estados adyacentes, al igual que hacen las hormigas naturales.

Sin embargo, estas características por sí solas no permiten desarrollar algoritmos eficientes para problemas combinatorios difíciles. Las hormigas artificiales tienen algunas capacidades adicionales:

- Las hormigas artificiales pueden utilizar la información de forma heurística para encontrar el camino que les lleve al destino.
- Tienen una memoria que almacena el camino seguido.
- La cantidad de feromona depositada por la hormiga artificial está en función de la calidad de la solución encontrada. Otra gran diferencia está en el momento de depositar la feromona. Normalmente, las hormigas artificiales sólo depositan feromona después de generar una solución completa.

---

<sup>1</sup> Colaboración a través del medio físico.

- La evaporación de feromona en los algoritmos ACO es diferente a como se presenta en la naturaleza, ya que la inclusión del mecanismo de evaporación es una cuestión fundamental para evitar que el algoritmo se quede estancado en la primera solución. La evaporación de feromona permite a la colonia de hormigas artificiales olvidar lentamente su historia y buscar nuevos caminos. Esto evita quedarnos en una solución prematura.
- Para mejorar la eficiencia y eficacia del sistema, los algoritmos ACO se enriquecen con habilidades adicionales como la capacidad de mirar más allá de la siguiente transición (*lookhead*), la optimización local, el *backtracking* y la llamada lista de candidatos, que contiene un conjunto de vecinos que pueden mejorar la eficiencia del algoritmo.

## 4.4. Simple Ant Colony Optimization (S-ACO)

### 4.4.1. Introducción

A partir del experimento del doble puente [23] Dorigo recrea el comportamiento de las hormigas para hacer un algoritmo que encuentra el camino más corto en un grafo.

Comienza considerando un grafo estático  $G = (N, A)$ , donde  $N$  es el conjunto de vértices del grafo y  $A$  es el conjunto de aristas no dirigidas que los conecta. A los dos puntos entre los que se quiere establecer el camino más corto los llama fuente y destino, o como pasa con las hormigas reales, el hormiguero y el alimento.

El primer problema que aparece como consecuencia de la actualización de la feromona es la creación de ciclos. Este problema puede acontecer mientras las hormigas construyen la solución y, como tienden a ir donde más feromona hay, repiten el camino que les parece mejor. Incluso si las hormigas escapan de los ciclos ya no se vería favorecida la ruta más corta entre la fuente y el destino.

Por tanto, parece razonable eliminar la actualización de feromona a medida que la hormiga avanza. Sin embargo, si se suprime este mecanismo, el sistema deja de funcionar, incluso para el caso más simple del doble puente.

La razón es muy simple: Volviendo a las hormigas reales, la orientación de una hormiga se basa en el rastro que dejan en el suelo, de tal manera que cuando encuentra una fuente de comida debe volver al hormiguero. Si no marca su rastro con feromona no será capaz de volver, porque no recuerda el camino que ha seguido para alcanzar la comida.

En S-ACO Dorigo extiende las capacidades básicas de las hormigas artificiales dotándolas de una memoria interna capaz de almacenar la ruta que han seguido así como el coste de la misma. Gracias a esta memoria las hormigas artificiales son capaces de implementar una serie de comportamientos que les permiten construir eficientemente una solución:

- A medida que la hormiga avanza, construye probabilísticamente una solución basada en rutas de feromona, sin mecanismo de actualización de feromona.
- Hace un recorrido inverso determinista con eliminación de ciclos, a la vez que se actualizan las rutas de feromonas.
- Evalúa la calidad de la solución generada y, en base a dicha evaluación, se determina la cantidad de feromona que deposita.

## 4.4.2. Descripción

### 4.4.2.1. Modos de funcionamiento

Las hormigas artificiales del S-ACO pueden funcionar de dos modos: *hacia delante* y *hacia atrás* (ó *forward* y *backward*, respectivamente). Se encuentran en modo *forward* cuando se mueven desde el hormiguero hacia la fuente de comida y en modo *backward* cuando regresan al hormiguero.

Cuando una hormiga en modo *forward* alcanza la fuente de comida cambia su modo de comportamiento al modo *backward* y comienza su viaje de regreso hacia la colonia. En S-ACO las hormigas *forward* construyen una solución eligiendo de manera probabilística el nodo vecino siguiente al que se va a mover.

La elección probabilística está influenciada por la feromona depositada previamente por otras hormigas en ese arco. Las hormigas artificiales en modo *forward* no depositan ninguna cantidad de feromona mientras se mueven. Este hecho, junto con el comportamiento determinista de las hormigas en modo *backward*, evita la aparición de ciclos.

### 4.4.2.2. Búsqueda de caminos

Cada hormiga colabora en la solución del problema, comenzando desde el nodo fuente y tomando decisiones en cada nodo. La información que cada nodo almacena sobre la zona de sus vecinos es percibida por la hormiga y utilizada de una manera probabilística para decidir a qué nodo siguiente debe ir.

Al comienzo de cada proceso de búsqueda se asigna una cantidad constante de feromona a todos los arcos. Cuando la hormiga  $k$  se encuentra en el nodo  $i$ , utiliza el sendero de feromona  $\tau_0$  para calcular la probabilidad de elegir el nodo  $j$  de la siguiente manera:

$$p_{ij}^k = \begin{cases} \frac{\tau_{ij}^\alpha}{\sum_{l \in N_i^k} \tau_{il}^\alpha} & \text{si } j \in N_i^k \\ 0 & \text{si } j \notin N_i^k \end{cases}$$

Ecuación 4.1. Heurística de búsqueda de camino en S-ACO.

donde  $N_i^k$  es la lista de los nodos disponibles a los que puede ir la hormiga  $k$  cuando se encuentra en el nodo  $i$ .

En S-ACO un nodo contiene la vecindad de todos los nodos que están conectados con él, exceptuando el nodo del que procede la hormiga. De esta manera se evita que las hormigas vuelvan a su nodo origen. Sólo en el caso de que la vecindad de un nodo sea el conjunto, se permite a la hormiga volver sobre sus pasos. Hay que señalar que este proceder puede inducir fácilmente a la generación de caminos cíclicos en el grafo.

#### 4.4.2.3. Trazado de ruta y actualización de feromona

El uso de una memoria explícita permite a una hormiga artificial volver por el camino que le ha llevado hasta la fuente de comida.

Cuando una hormiga alcanza su destino cambia de comportamiento de *forward* a *backward* y comienza a construir el camino de retorno. Antes de comenzar el regreso a la fuente la hormiga elimina los ciclos que pueda haber en el camino que ha construido mientras buscaba el nodo objetivo. El problema de los ciclos es que, mientras la hormiga realiza su viaje de retorno, puede recibir varias veces las aportaciones de feromona, generando un fenómeno de auto-reforzamiento de los ciclos.

Mientras la hormiga regresa, deposita una cantidad fija de feromona en los arcos que ha visitado. En particular, si una hormiga  $k$  en modo *backward* atraviesa el arco  $(i, j)$  cambia la cantidad de feromona del arco:

$$\tau_{ij} \leftarrow \tau_{ij} + \Delta\tau^k$$

Ecuación 4.2. Actualización de la pista de feromona en S-ACO.

Con esta regla una hormiga que utilice el arco que conecta  $i$  y  $j$  incrementa la probabilidad de que lo usen el resto de hormigas.

La cantidad de feromona que se deposita está en función de la cantidad de hormigas que lo transitan.

En S-ACO la hormiga memoriza el camino que le ha llevado hasta la solución junto con el coste de los arcos que ha recorrido. De tal modo, que puede evaluar el coste de la solución obtenida y utilizarlo para ajustar la cantidad de feromona que deposita en cada arco que recorre, por medio de una función variable en relación al coste del camino, para que la búsqueda de caminos más cortos lleve más rápidamente hacia las mejores soluciones.

#### 4.4.2.4. Evaporación de las marcas de feromona

La evaporación de feromona puede ser vista como un mecanismo que evita la rápida convergencia de las hormigas hacia una ruta que no es óptima. De hecho, la disminución de feromona que se encuentra en el camino favorece la exploración de nuevas rutas durante el proceso de búsqueda global. Dorigo [10] señala que en las hormigas reales este mecanismo también está presente, aunque no juega un papel fundamental.

Pensamos que esto no es así, porque si no hubiese evaporación de feromona, las hormigas seguirían siempre el mismo camino. Gracias al proceso de evaporación las hormigas cambian periódicamente las zonas de exploración. Este es justamente el mecanismo que les permite sobrevivir y explorar zonas nuevas y tener disponibles otras rutas.

Por lo que hemos visto, el mecanismo de evaporación de la feromona natural y el de la artificial juega un papel clave, porque sin este mecanismo el sistema no funcionaría bien, como se ha podido ver en el tercer caso del experimento del doble puente.

En las hormigas reales la intensidad de la feromona presente en el medio disminuye en función del tiempo. En S-ACO dicha evaporación es simulada aplicando una regla de reducción de feromona, que se muestra a continuación:

$$\tau_{ij} \leftarrow (1 - \rho)\tau_{ij} \quad \rho \in (0,1]$$

Ecuación 4.3. Evaporación de la feromona en S-ACO.

La evaporación de feromona hace que se construyan cada vez mejores soluciones, debido a que se evapora la feromona asociada a las primeras soluciones.

## 4.5. Meta-heurística ACO

ACO constituye una meta-heurística. En otras palabras, es un método heurístico para resolver un tipo de problema computacional general aplicándose a problemas que no tienen un algoritmo que dé una solución satisfactoria o cuando no es posible implementar la solución óptima.

ACO resuelve problemas de optimización combinatoria, como encontrar la ruta más corta.

El proceso distribuido de encontrar la ruta más corta es una fuente importante de investigación en inteligencia artificial (AI). Los algoritmos ACO trabajan de una manera interactiva. En cada iteración todas las hormigas artificiales contribuyen a dar una solución al problema utilizando la denominada matriz de feromona artificial. La matriz de feromona es actualizada con los valores asociados a las soluciones encontradas.

ACO se aplicó por primera vez al conocido problema del viajante ó *Traveling Salesman Problem* (TSP) [50]: un viajante partiendo de una ciudad tiene que visitar N ciudades sin repetir las y volver al origen en el menor tiempo posible. En este problema se aplicó *Ant System* (AS) [15]. En AS cada arista tiene asociado un valor de feromona artificial. El algoritmo trabaja de una manera interactiva. Al comienzo de cada iteración cada hormiga se coloca aleatoriamente en una ciudad inicial. Empezando desde ella y moviéndose de ciudad en ciudad dan una solución al TSP.

## 5. ENCAMINAMIENTO ACO EN REDES MÓVILES AD HOC

---

### 5.1. Encaminamiento ACO

Se denomina encaminamiento ACO al conjunto de protocolos de encaminamiento que hacen uso de las técnicas ACO.

La aproximación ACO para adquirir información de encaminamiento es bastante robusta porque la pérdida de una hormiga no es importante. Un aspecto esencial de todo el proceso es que las hormigas siempre muestran rutas completas entre el origen y el destino, si bien esto conduce al incremento de la sobrecarga, que influye negativamente en la eficacia del encaminamiento ACO. La aproximación ACO para el encaminamiento es, pues, diferente de la aproximación de vector distancia, donde la información de encaminamiento procede de la información proporcionada por los nodos vecinos y de la aproximación de estado de los enlaces, donde la información de encaminamiento se actualiza con los mensajes recibidos de los nodos de la red.

Otra característica es la manera en que las hormigas eligen una ruta. Ellas construyen el camino salto a salto de manera probabilística usando la información de la feromona. El uso de información de feromona permite construir sobre la experiencia adquirida previamente por las hormigas. Esta es la clave de un proceso altamente distribuido. El hecho de que las hormigas construyan sus caminos de una manera probabilística permite la exploración de rutas múltiples. Esto hace que el algoritmo se adapte a los cambios de la red, incrementando la robustez a través de la disponibilidad de rutas de reserva (*backup*) y el *throughput* de la red.

Una tercera característica es el reenvío aleatorio de los paquetes de datos basado en la información de feromona. Esto está relacionado con el uso de múltiples rutas. En este proceso el uso de feromona asegura que los datos se encaminen por las mejores rutas. Si la feromona se mantiene actualizada por el uso de suficientes hormigas, el balanceo de carga sigue automáticamente los cambios en la red.

Todo lo anterior hace que los algoritmos ACO presenten unas propiedades muy interesantes:

- Trabajan de una manera completamente distribuida, en la que la información no se encuentra en un nodo central, sino que está contenida en cada nodo.
- Tienen una alta adaptabilidad a los cambios de la red y del tráfico.
- Utilizan agentes móviles (llamados hormigas) para determinar los caminos para el envío de los datos. Estos agentes son paquetes de control que se envían por la red.
- Proporcionan encaminamiento multicamino.
- Presentan una excelente tolerancia a fallos, esto es, ofrecen un buen comportamiento ante el fallo de los agentes.
- Eligen automáticamente la ruta para el envío de datos.

## 5.2. Diseño de Algoritmos ACO de Encaminamiento

Pueden identificarse algunos pasos a la hora de abordar el encaminamiento utilizando esta meta-heurística. Estos pasos se pueden resumir en las seis tareas de diseño que se enumeran a continuación:

1. Representar el problema como un conjunto de componentes y transiciones o como un grafo ponderado que recorren las hormigas para construir soluciones.
2. Definir de manera apropiada el significado de las marcas de feromona para la toma de decisión. Este es un paso crucial en la implementación de un algoritmo ACO y no es una tarea trivial.
3. Elegir la preferencia heurística de cada decisión que debe tomar una hormiga mientras construye una solución. Remarcamos que la información heurística es crucial para un buen rendimiento cuando se aplica a algoritmos de búsqueda local.
4. Implementar una búsqueda local eficiente.
5. Elegir un algoritmo ACO específico y apropiado.
6. Ajustar los parámetros del algoritmo ACO. Un buen punto de partida es utilizar configuraciones que han demostrado ser buenas en problemas similares. Otra posible alternativa es utilizar procedimientos automáticos de ajuste de parámetros.

Queremos hacer notar que los pasos más importantes son los cuatro primeros, ya que una elección poco acertada en esos puntos hace que no sea fácil poderlo corregir con ajustes de parámetros.

### 5.3. Aplicación directa de ACO en MANETs

La aplicación directa del algoritmo tal y como está descrito por Dorigo no es aconsejable en redes móviles ad hoc por la lenta convergencia que ofrece (cuando es el caso).

La propuesta que da, la desarrolla en una topología estática de la red, en la que se conocen de antemano todas las rutas. Lo único que realizan las hormigas es elegir la ruta en función de la carga de tráfico.

En ese caso las rutas siempre son válidas, mientras que en redes móviles ad hoc esto no ocurre, porque la topología es desconocida y los nodos móviles. Por esa razón la aplicación directa no es factible.

Una topología dinámica implica que desconocemos las rutas para realizar una comunicación. Por eso el primer paso es hacer una exploración que capture la topología de la red.

La respuesta que se le pide a un protocolo de encaminamiento en la aplicación de ACO a redes móviles ad hoc es que haga una rápida captura de la topología, teniendo en cuenta los recursos que necesita. Si se destinan muchos recursos, porque aumenta el intercambio de información, se puede llegar a colapsar la red.

En las hormigas reales el borrado de feromona es accidental, en cambio, en una red con topología dinámica, es normal que la captura efectuada en un primer momento quede invalidada.

## 5.4. Modificación de ACO para su aplicación en MANETs

Existen aplicaciones de las ideas de Dorigo, pero todas en mayor o menor medida incurren en el mismo error: los algoritmos ACO pretenden que todas las soluciones estén disponibles utilizando la búsqueda en anchura.

En una red estática la topología no cambia y el algoritmo funciona bien, pero en una red móvil ad hoc esto no ocurre. La solución que adoptan otros autores es que realizan una primera búsqueda en anchura para tener todas las rutas disponibles; y una vez obtenida, aplican el ACO.

También hay autores que construyen un modelo de topología de la red, que simula la búsqueda en el proceso de exploración de rutas.

Señala Deborah Gordon en "*Ants at Work*" [22] que la exploración realizada por las hormigas recolectoras es dirigida. Y dice lo siguiente: un tipo de hormigas, que Gordon llama *patrulla* sale en diversas direcciones a explorar los alrededores de la colonia. Este tipo de hormigas, al volver a la colonia, indica si ha encontrado o no el alimento. En caso positivo, estimula de alguna manera a las hormigas recolectoras a salir hacia la comida en la dirección indicada por la patrulla, y crean así un flujo de tráfico entre el hormiguero, la comida y la vuelta al hormiguero. La obtención de la ruta mínima la realizan las hormigas recolectoras, pero lo hacen sólo en la zona marcada por la patrulla. El problema del cálculo de ruta se presenta después de haber realizado una exploración.

Este mismo comportamiento se puede aplicar en el ACO. Se crea un mecanismo para situar al objetivo en la zona. Así se consigue que la exploración que realiza el ACO se reduzca a una zona local donde se encuentra el objetivo. De esta forma se pueden hacer búsquedas en profundidad en dicha zona. Y para evitar que la zona local sea muy grande se mantiene el mecanismo de patrulla descrito por Gordon.

## **5.5. Ant-Net: el primer algoritmo ACO**

AntNet [10] es el primer algoritmo ACO que se aplicó en el encaminamiento de redes cableadas o estáticas (que no dinámicas). Es multicamino y se adapta al tráfico de la red, siendo idóneo en su aplicación al encaminamiento en el tipo de red para el que fue diseñado.

El encaminamiento en este algoritmo es emergente, y surge de la propia dinámica del algoritmo. Esto es muy atractivo porque ahorra recursos. No es necesario un exhaustivo cálculo de rutas, ni el mantenimiento de rutas e intercambio de información.

El cálculo de rutas se hace necesario porque las rutas se construyen a partir de otras ya conocidas, de manera que si vinculamos a las hormigas que realizan la exploración con los paquetes de datos que se transmiten, el resultado es que la propia transmisión de información se hace sobre la ruta óptima.

No es necesario un mantenimiento de rutas, porque la mantienen los saltos que se realizan de nodo a nodo. No se necesita actualizar una información global, porque en el caso de que una ruta sufra modificaciones el nodo que ha sufrido los cambios la actualiza.

El algoritmo AntNet-FA (*Flying Ants*) [9] es una versión mejorada de AntNet, porque hace que las hormigas se mueven más rápidamente sobre las colas que tienen más alta prioridad y porque tiene en cuenta los retardos fin-a-fin de acuerdo a modelos estadísticos que dependen de las colas locales.

El rendimiento de AntNet-FA mejora con el incremento del tamaño de la red y su eficiencia es similar o mejor que el de AntNet.

## **5.6. Protocolos de Encaminamiento ACO para MANETs**

Los protocolos de encaminamiento ACO para redes móviles ad hoc se clasifican, como cualquier otro protocolo de encaminamiento para este tipo de redes, en las siguientes clases: proactivos, reactivos e híbridos.

### **5.6.1. Protocolos de Encaminamiento ACO Proactivos**

#### **5.6.1.1. Probabilistic Emergent Routing Algorithm**

En *Probabilistic Emergent Routing Algorithm* (PERA) [2] el descubrimiento de ruta lo hacen dos clases de hormigas: *hacia delante* y *hacia atrás*. Estos agentes crean y ajustan la distribución de probabilidad en cada nodo respecto a sus vecinos. El nodo tiene la probabilidad de que cada uno de sus vecinos pueda recibir y reenviar el paquete de datos.

Cada hormiga *hacia delante* contiene las direcciones IP del nodo origen y destino, un número de secuencia, un campo contador de saltos y una pila que crece dinámicamente. La pila contiene la información sobre los nodos que la hormiga *hacia delante* visita y los tiempos asociados. Cuando un nodo no tiene un registro de una ruta a un destino se crea una hormiga *hacia delante*, donde el nodo pone su propia dirección IP en la pila de dicha hormiga, así como el tiempo en que la hormiga es creada. A partir de este momento el nodo guarda periódicamente las hormigas *hacia delante* enviadas a los destinos para cuando la ruta sea requerida. Cuando esta hormiga *hacia delante* alcanza el destino, el nodo destino crea un nuevo agente, una hormiga *hacia atrás*. Esta usa la información contenida en la hormiga *hacia delante* en el camino inverso, para actualizar la distribución de probabilidad en cada nodo y reflejar el estado actual de la red.

Como las hormigas *hacia delante* se envían en modo *broadcast* desde el origen, y en los nodos intermedios causa un múltiple *broadcast* al encontrar diferentes caminos al destino genera múltiples hormigas *hacia atrás*. Los paquetes agentes o las hormigas son de tamaño variable. Por lo tanto, este proceso de *broadcast* hace que haya mucha sobrecarga.

#### **5.6.1.2. Ant Routing Algorithm for Mobile Ad hoc networks**

*Ant Routing Algorithm for Mobile Ad hoc networks* (ARAMA) [26] es un algoritmo de encaminamiento proactivo en el que las hormigas *hacia delante* no sólo tienen en cuenta el factor de contador de saltos (como la mayoría de los protocolos), también valoran la heurística de enlace local a través de la ruta, como puede ser la energía de la batería del nodo y el retardo de la cola.

ARAMA define un valor llamado *grado*. Este valor es calculado por cada hormiga *hacia atrás* en función de la información del camino almacenado en la hormiga *hacia delante*. En cada nodo, la hormiga *hacia atrás* actualiza la cantidad de feromona de la tabla de rutas del nodo usando el valor *grado*. El protocolo usa el mismo *grado* para actualizar el valor de la feromona de todos los enlaces.

Los autores afirman que la sobrecarga del descubrimiento y mantenimiento de rutas se reduce por medio del control del número de hormigas *hacia delante*. Sin embargo no aclaran cómo controlar la generación de hormigas en un entorno dinámico.

## 5.6.2. Protocolos de Encaminamiento ACO Reactivos

### 5.6.2.1. Ad hoc Networking with Swarm Intelligence

*Ad hoc Networking with Swarm Intelligence* (ANSI) [47] es un protocolo de encaminamiento reactivo que utiliza dos conjuntos de agentes móviles llamados hormigas hacia delante y hormigas hacia atrás. La tabla de encaminamiento contiene una entrada por cada nodo que se alcanza y el siguiente mejor salto, mientras que las tablas de decisión de la hormiga almacenan los valores de feromona.

En ANSI las hormigas hacia delante sólo se generan cuando un nodo necesita transmitir datos a otro nodo. Estas hormigas son enviadas en modo *broadcast*, mientras que las *hacia atrás* lo hacen en modo *unicast*, siguiendo la pista que han dejado las hormigas *hacia delante* en el camino, y actualizan los valores de feromona de los nodos. Los paquetes de datos eligen el siguiente salto teniendo en cuenta el valor más grande de feromona.

ANSI es tan bueno o mejor que AODV con respecto a la entrega de paquetes y al retardo fin a fin.

### 5.6.2.2. Ant-Colony-Based Routing Algorithm

*Ant-Colony-Based Routing Algorithm* (ARA) [24] es un protocolo reactivo en el que las entradas de la tabla de encaminamiento contienen valores de feromona que facilitan la elección de vecino. Para conseguir un destino es preciso elegir un vecino que nos sirva de enlace y así sucesivamente hasta llegar al destino. En la tabla de encaminamiento los valores de feromona se degradan con el tiempo y los nodos entran en un modo *sleep* si ha alcanzado un umbral demasiado bajo.

En ARA el descubrimiento de rutas lo realiza un conjunto de dos agentes móviles, llamadas hormigas *hacia delante* y hormigas *hacia atrás*. Durante el descubrimiento de ruta, los paquetes de hormigas *hacia delante* y *hacia atrás* tienen un número de secuencia único, para evitar paquetes duplicados, siendo expandidos por los nodos fuente y por los nodos destino por medio de inundación. Las hormigas *hacia delante* y *hacia atrás* actualizan por inundación las tablas de feromonas en todos los nodos. La inundación tiene mayor alcance de transmisión de paquetes, porque los paquetes por inundación se transmiten a todos los nodos de la red por medio de multisalto, mientras que el *broadcast* se transmite a los vecinos que están a un salto. El problema de la inundación es que conlleva una alta sobrecarga.

Una vez que se ha realizado el descubrimiento de ruta para un destino determinado, el nodo emisor ya no genera un nuevo agente móvil hacia el destino, sino que el mantenimiento de la ruta lo realizan los paquetes de datos.

### **5.6.2.3. Position based Ant Colony Routing**

*Position based Ant Colony Routing* (POSANT) [33] es un algoritmo de encaminamiento reactivo basado en la localización de los nodos. POSANT encuentra las rutas óptimas o más cercanas en una red que contiene nodos de diferentes rangos de transmisión. Cada nodo asume su posición, la posición de sus vecinos y la del nodo destino. Sólo calcula la ruta bajo demanda en el momento de enviar datos desde un nodo origen a un nodo destino y se envían cuando la ruta se establece. Para minimizar el tiempo que POSANT necesita para encontrar una ruta la información sobre la posición de los nodos se usa como un valor heurístico.

El uso de la información de localización, como parámetro heurístico, reduce significativamente el tiempo necesitado para establecer rutas desde el origen al destino. Y también reduce el número de mensajes de control generados. Tiene un índice de entrega alto y un retardo medio de paquetes bajo, comparado con otros algoritmos de encaminamiento basados en posiciones. POSANT alcanza un comportamiento estable más rápido que AntNet.

#### 5.6.2.4. Ant colony based Multi-path QoS-aware Routing

*Ant Colony based Multi-path QoS-aware Routing* (AMQR) [35] es un algoritmo de encaminamiento reactivo. La mayoría de los protocolos de encaminamiento basados en hormigas para MANETs son esencialmente métodos de encaminamiento de ruta única, que tienden a tener una sobrecarga en los nodos, que se encuentran en el camino más corto del origen al destino. Esta sobrecarga es debida a que en los métodos de camino único no existe balanceo de carga. El encaminamiento multicamino de enlace disjunto es más robusto y puede soportar QoS mejor que el encaminamiento de camino único de las MANETs. AMQR combina inteligencia *swarm* y multicamino de enlace disjunto para balancear la carga. Establece y utiliza múltiples rutas de enlace disjunto para enviar paquetes de datos y adaptar la feromona para dispersar el tráfico de la comunicación.

#### 5.6.2.5. Distributed Ant Routing

*Distributed Ant Routing* (DAR) [49] crea las rutas bajo demanda con el fin de disminuir la sobrecarga de encaminamiento con respecto a las aproximaciones proactivas. Las hormigas *hacia delante* sólo recopilan información de los nodos que recorren. Las hormigas *hacia delante* sólo eligen el siguiente salto hacia el destino basándose en la cantidad de feromona. La cantidad de feromona depositada por las hormigas *hacia atrás* sobre cada enlace recorrido es constante.

La simplicidad del protocolo es útil para hacer un encaminamiento transparente en redes constituidas por elementos heterogéneos.

En DAR las tablas de encaminamiento en cada nodo son probabilísticas: el siguiente salto se selecciona de acuerdo con el mayor porcentaje de feromona dejado por las hormigas. Así, el reenvío de las hormigas *hacia delante* es probabilístico y permite la exploración de otras rutas disponibles en la red. Los paquetes de datos son enviados de forma *determinada* por los nodos intermedios que se encuentran en el camino desde el emisor al nodo destino. Este proceso crea una ruta global mediante el uso de información local.

#### **5.6.2.6. Ant-based Distributed Route Algorithm**

*Ant-based Distributed Route Algorithm* (ADRA) [55] es un algoritmo reactivo en el que las hormigas se mueven a través de la red entre pares de nodos elegidos aleatoriamente. Estas hormigas al moverse depositan feromona en función de varios parámetros: distancia en saltos desde su nodo origen, la calidad del enlace, la congestión encontrada en su viaje, la feromona actual que el nodo posee y la velocidad con que se mueven los nodos. Por supuesto, el mismo nodo, por la evaporación del feromona, cambia su valor de acuerdo con la edad del enlace. Una hormiga selecciona su camino en cada nodo intermedio según la feromona que tiene distribuida y para acelerar la elección del camino se dan parámetros con diferentes valores que actualizan la probabilidad en la tabla de encaminamiento.

El rendimiento del algoritmo se mide por *ratio* de paquetes perdidos, la sobrecarga de mensajes de control y el retardo fin a fin de paquetes. ADRA exhibe muchas características atractivas de control distribuido.

### **5.6.2.7. Ant-based Energy Aware Disjoint Multipath Routing Algorithm**

Investigaciones recientes han propuesto varios protocolos especiales de encaminamiento de un único camino. Este tipo de protocolos tiene el inconveniente de que, al haber un único camino, si se produce una ruptura de enlace y no disponer de otras alternativas, es necesario realizar de nuevo un proceso de descubrimiento de ruta con el consiguiente aumento del retardo, con la sobrecarga de mensajes de control y con la disminución del ratio de entrega de paquetes. Para mejorar estos problemas surge *Ant-based Energy Aware Disjoint Multipath Routing Algorithm* (AEADMRA) [54]. Este protocolo ha sido diseñado para permitir que haya más caminos en los paquetes de petición/respuesta de rutas y descubrirlos con una sobrecarga más baja.

### **5.6.2.8. ImProved Ant Colony Optimization Algorithm for Mobile Ad Hoc Networks**

En [42] se propone *imProved Ant Colony Optimization algorithm for mobile ad hoc NETworks* (PACONET), un protocolo de encaminamiento reactivo que utiliza dos clases de agentes: hormigas *hacia delante* (FANT) y hormigas *hacia atrás* (BANT). Las FANT exploran los caminos de la red en busca de rutas desde un origen a un destino en modo *broadcast* restrictivo y las BANT establecen la información del camino adquirido por las FANT.

Los paquetes de datos son enviados probabilísticamente hacia los nodos que tienen la concentración de feromona más alta. Las FANT viajan hacia los nodos no visitados, pero si no los encuentran siguen la ruta de los nodos con más alta concentración.

Las filas de la tabla de rutas representan a los vecinos de un nodo y las columnas representan a los nodos de la red. Cada par (fila, columna) en la tabla de encaminamiento tiene dos valores: (a) un valor binario que indica si el nodo ha sido visitado, y (b) la concentración de feromona.

Las FANT sólo tendrán en cuenta la concentración de feromona después de que todos los vecinos de una columna han sido visitados. El propósito de esto es asegurar que todos los caminos son explorados para encontrar el mejor camino hacia el destino. El nodo con la feromona más alta es elegido como el siguiente salto, después de que la FANT ha determinado que el nodo no ha sido visitado antes.

### **5.6.3. Protocolos de Encaminamiento ACO Híbridos**

#### **5.6.3.1. Ant-AODV**

La técnica Ant-AODV [37] constituye una forma híbrida de encaminamiento basado en ACO y en el protocolo de encaminamiento AODV.

Para superar algunos de los inconvenientes de AODV, como es la sobrecarga generada por el aumento de mensajes de control, se emplea esta técnica híbrida que destaca la conectividad de los nodos y disminuye el retardo extremo a extremo, así como la latencia de descubrimiento de ruta.

Los agentes de hormigas Ant-AODV trabajan independientemente y proporcionan rutas a los nodos. Los nodos también tienen la capacidad de realizar un descubrimiento de ruta bajo demanda para los destinos que no tienen una entrada de ruta lo suficientemente actualizada. El uso de hormigas con AODV incrementa la conectividad de los nodos, que está asociada al número de destinos que tiene dicho nodo y a los que se llega por medio de la correspondiente entrada actualizada en la tabla de encaminamiento.

La latencia de descubrimiento de ruta es más reducida, porque se reduce el proceso de descubrimiento de ruta. Ante la petición de ruta RREQ, la probabilidad de recibir una respuesta más rápida es mayor al haber más nodos conectados. Como las hormigas actualizan las rutas continuamente, un nodo emisor puede seleccionar una nueva ruta y más corta. Esto conduce a una disminución considerable en el retardo medio extremo a extremo, comparado con AODV y con el encaminamiento basado en ACO.

Además Ant-AODV usa mensajes de error de ruta (RERR) para informar en cadena a otros nodos de un fallo de enlace local similar a AODV.

### **5.6.3.2. A hybrid ant colony optimization routing algorithm for mobile ad hoc network**

*A hybrid ant colony optimization routing algorithm for mobile ad hoc network* (HOPNET) [52] es un algoritmo de encaminamiento híbrido, basado en hormigas que saltan de una zona a otra. El algoritmo tiene características extraídas de los protocolos ZRP y DSR, siendo altamente escalable, comparado con otros protocolos híbridos.

Este algoritmo consiste en el descubrimiento proactivo de ruta dentro de la zona de vecindad de un nodo, y la comunicación entre zonas se hace de forma reactiva bajo demanda cuando se trata de enviar paquetes de una zona a otra. El tamaño de la zona lo determina la longitud del radio medido en saltos y no el nodo. Por consiguiente, la zona de encaminamiento la constituyen los nodos que están dentro de la longitud de radio especificado. Un nodo puede pertenecer a múltiples zonas que se solapen, y las zonas pueden variar de tamaño. Los nodos pueden ser clasificados como interiores o fronterizos (o periféricos). Los nodos fronterizos son los que están a la distancia del radio; los que se encuentran a una distancia menor del radio son interiores.

Cada nodo tiene dos tablas de encaminamiento: tabla de encaminamiento intrazona (*Intrazone Routing Table* - IntraRT) y tabla de encaminamiento interzona (*Interzone Routing Table* - InterRT). La zona IntraRT es mantenida proactivamente por lo que un nodo puede obtener rápidamente un camino hacia cualquier otro nodo dentro de esta zona. Esto se realiza enviando periódicamente hormigas *hacia delante* que detectan los caminos dentro de la zona y el cambio en la topología (nodos que se salen de la zona, fallos de enlace, nuevos nodos que entran, etc.) Recordamos que una vez que una hormiga *hacia delante* alcanza un destino, se envía una hormiga de retorno (*hacia atrás*) a través del camino descubierto. La zona InterRT almacena el camino a un nodo más allá de su zona. Esta tabla de rutas se establece bajo demanda y los nodos periféricos son encargados de enlazar las zonas.

Cuando el número de nodos es pequeño el movimiento continuo de los nodos periféricos hace que constantemente haya que descubrir nuevas rutas, lo que provoca más retardo que en otros protocolos de encaminamiento híbridos.

## 6. ANTHOCNET

---

AntHocNet [8] [11] [12] es un algoritmo de encaminamiento ACO híbrido. Data de 2004 y en estos cinco años ha tenido numerosas extensiones y variaciones para mejorar su rendimiento.

AntHocNet sigue una estructura parecida a AntNet-FA, pero difiere en sus características. AntNet-FA se aplica a topologías de redes estáticas, en las que se conocen las rutas y la convergencia es lenta. Por tanto, lo único que tienen que hacer las hormigas es elegir el camino. AntHocNet, por su parte, tiene en cuenta la topología dinámica y demás características de las redes ad hoc. Cuando cambia la topología de la red es necesario restablecerla rápidamente y se hace por medio de un nuevo proceso de descubrimiento de ruta. Si se destinan muchos recursos para acelerar este proceso, aumenta el intercambio de información, que puede llegar a colapsar la red.

Nos encontramos, por tanto, con un problema: si no queremos sobrecargar la red, aumentamos el tiempo de convergencia del algoritmo ACO; si queremos disminuir el tiempo de convergencia, sobrecargamos la red.

La problemática anterior hace que no se pueda aplicar directamente el algoritmo AntNet-FA a las redes móviles ad hoc por lo que necesita una modificación que acelere su tiempo de convergencia sin sobrecargar la red. De esta forma surge AntHocNet como un algoritmo reactivo y proactivo (híbrido), multicamino y adaptativo. Es reactivo porque tiene agentes que actúan bajo demanda para establecer los caminos hacia los destinos.

Es proactivo porque posee agentes que actúan con la información recogida y tienen la posibilidad de descubrir nuevas rutas para tener alternativas ante fallos de enlace.

Es multicamino porque establece diferentes caminos para enviar la información al destino. Finalmente, es adaptativo porque se adecua a las condiciones del tráfico y de la red.

## 6.1. Estructuras de datos

En AntHocNet se distinguen las siguientes estructuras de datos:

- Tabla de encaminamiento: donde se almacenan los valores de los registros de la tabla de rutas. Estos valores son:
  - Feromona regular ( $\tau_{ij}^d$ ): Indica el camino por donde viajan los datos.
  - Feromona virtual ( $w_{ij}^d$ ). Indica el camino que posiblemente puede ser bueno.
  - Número medio de saltos ( $h_{ij}^d$ ): Utilizado en el proceso de reparación de rutas.
- Tabla de vecinos: que nos proporciona información de los enlaces/vecinos, que contiene el tiempo de escucha del vecino  $t_{n_j}$ .

Se utiliza para indicar la presencia de los vecinos y detectar posibles fallos de enlace.

## 6.2. Funcionamiento

En el funcionamiento de AntHocNet se distinguen las siguientes etapas o fases:

- Establecimiento de la información de encaminamiento: El nodo fuente envía agentes reactivos para descubrir la primera ruta disponible al destino.
- Encaminamiento de los datos: Los datos son enviados a través de los nodos hasta el destino valiéndose de la información de las rutas, pudiendo utilizar la técnica multisalto, que consiste en el envío de datos a través de nodos intermedios que funcionan como encaminadores.
- Mantenimiento de las rutas establecidas y exploración de nuevas: Se actualiza proactivamente la información de las rutas existentes y la posibilidad de descubrir otras nuevas.
- Gestión de los fallos de enlace: Gestión de los fallos que acontecen porque un nodo se sale del alcance de la red o porque no recibe los mensajes de control, que se encargan de informar a un nodo de sus vecinos más próximos (que están a un salto), etc.

### 6.2.1. Establecimiento del camino

En un principio los nodos no disponen de información de encaminamiento para enviar los datos, pero sí tienen las aplicaciones para empezar: generador de tráfico, *ftp*, *ping*, ..., los interfaces de red, la pila de protocolos (IP, UDP/TCP, etc.). La aplicación genera los datos en el nodo, pero al no tener ruta disponible no los puede enviar. Por lo tanto, el nodo necesita enviar unos agentes reactivos (hormigas reactivas) que descubran las rutas al destino. Estas hormigas, llamadas *hacia delante* (*forward*), van desde el nodo fuente al nodo destino, siendo enviadas en modo *broadcast*. Los nodos intermedios que reciben estas hormigas las reenvían en el proceso de búsqueda de ruta hasta llegar al destino. Esta difusión de la hormiga hace que un mismo nodo pueda recibir la información varias veces. Para evitar la sobrecarga en este proceso se sigue la siguiente estrategia:

Cuando un nodo recibe varias hormigas, que proceden de la misma hormiga generada en el origen, se comparan los caminos de las hormigas recibidas y sólo reenvía la hormiga que tiene el mejor factor. Este factor está determinado por el número de saltos y el tiempo empleado en el viaje por la mejor hormiga de la generación. También lo determina un valor *time to live* que se establece de acuerdo con el tamaño de la red.

Según la versión de AntHocNet, este proceso de descubrimiento de ruta puede ser multi-camino o no. Cuando una hormiga *hacia delante* llega al destino la procesa, y genera y envía una hormiga reactiva *hacia atrás* (*backward*) de retorno al origen. Si recibe varias hormigas, toma dos decisiones: a) Se queda con la primera hormiga y descarta las restantes, b) se queda con todas las hormigas recibidas. En este caso se produce un descubrimiento de ruta multicamino.

Como se explicó en ACO, la idea de las hormigas naturales se mejora con el concepto de hormigas artificiales. Aquellas depositan feromona en la ida y en la vuelta para conocer el camino, mientras que las artificiales disponen de memoria interna, que contiene la información de los nodos recorridos. Esta información la utilizan las hormigas *hacia atrás* para la vuelta. Por esta razón el retorno de la hormiga al origen lo hace en modo *unicast*. En este recorrido la hormiga *hacia atrás* se encarga de actualizar o crear un registro en la tabla de encaminamiento. Esta tabla de encaminamiento almacena un *valor* que representa la media de la inversa de un coste en términos del tiempo estimado y del número de saltos, que supone el ir del nodo destino al origen pasando por los nodos intermedios.

En la versión de AntHocNet que ha servido de referencia para este trabajo [17] se utilizan las siguientes fórmulas y heurísticas.

La hormiga reactiva *hacia delante* guarda una lista  $P$  de nodos que ha visitado. Esta lista es utilizada por las hormigas reactivas *hacia atrás* para marcar el camino de retorno. La hormiga *hacia atrás* computa incrementalmente una estimación  $\hat{T}_p$  del tiempo que tardaría un paquete de datos en viajar a través de esa lista  $P$  de nodos hacia el destino, actualizando las tablas de encaminamiento.

Como se ha dicho anteriormente,  $\hat{T}_p$  es la suma de estimaciones del tiempo que tarda en alcanzar el siguiente salto en cada nodo de la lista  $P$ :

$$\hat{T}_p = \sum_{i=1}^{n-1} \hat{T}_{i \rightarrow i+1}$$

Ecuación 6.1. Estimación de tiempo utilizado en el establecimiento del camino.

El valor de la estimación local  $\hat{T}_{i \rightarrow i+1}$  se define como el producto de dos términos:

- El número actual de paquetes en la cola que están listos para enviarse en la capa MAC más uno ( $Q_{mac}^i + 1$ ).
- Tiempo medio necesitado para enviar un paquete  $\hat{T}_{mac}^i$  :

$$\hat{T}_{i \rightarrow i+1} = (Q_{mac}^i + 1)\hat{T}_{mac}^i$$

Ecuación 6.2. Valor de estimación local en AntHocNet.

Si tenemos en cuenta el tiempo real  $t_{mac}^i$  que tarda un nodo en enviar un paquete, la fórmula es la siguiente:

$$\hat{T}_{mac}^i = \alpha \hat{T}_{mac}^i + (1 - \alpha)t_{mac}^i \quad \alpha \in [0,1]$$

Ecuación 6.3. Tiempo medio necesitado para enviar un paquete en AntHocNet.

En los experimentos  $\alpha$  se establece a 0,7. Con este parámetro se quiere indicar que tiene más prioridad  $\hat{T}_{mac}^i$  que  $t_{mac}^i$ , concretamente un 70%.

En cada nodo  $i$  intermedio perteneciente a la lista  $P$  la hormiga *hacia atrás* establece una ruta hacia el origen de la sesión, creando y actualizando los registros de la tabla de encaminamiento. El valor de la feromona  $\tau_{id}$  representa una media de la inversa del coste en términos del tiempo estimado y del número de saltos. Se define como se indica en la siguiente ecuación, donde  $h$  es el número de saltos y  $T_{hop}$  es un valor fijo, que representa el tiempo que tarda un salto en condiciones ideales (sin carga).

$$\tau_{id} = \left( \frac{\hat{T}_{i \rightarrow d} + hT_{hop}}{2} \right)^{-1}$$

Ecuación 6.4. Valor de la feromona en AntHocNet.

Se utiliza la inversa porque el valor de la feromona es inversamente proporcional al coste, es decir, cuánto más tarda en ir desde un origen a un destino, menor valor de la feromona se obtiene.

Finalmente el valor de un registro  $F_{nd}^i$  en la tabla de encaminamiento del nodo  $i$  se actualiza con la siguiente ecuación, donde  $n$  representa el siguiente salto y  $d$  el destino que se quiere alcanzar:

$$F_{nd}^i = \gamma F_{nd}^i + (1 - \gamma)\tau_{id} \quad \gamma \in [0,1]$$

Ecuación 6.5. Actualización de un registro de la tabla de encaminamiento en AntHocNet.

El valor  $\gamma$  se establece experimentalmente a 0,7.

Cuando no hay información de encaminamiento (por fallo de enlace) se repite el proceso de establecimiento de ruta. La diferencia con el primer proceso de descubrimiento está en que ya pueden existir otras rutas, pues el fallo de enlace no altera toda la información de encaminamiento, sino que sólo afecta a una parte. Por esta razón, las hormigas reactivas *hacia delante* vuelven a ser reenviadas en modo *broadcast*, como se vio anteriormente, o en modo *unicast*. Este modo de transmisión sólo se utiliza cuando hay información disponible de ruta al destino.

El proceso de envío se realiza probabilísticamente utilizando la siguiente ecuación donde  $N_d^i$  es el conjunto de vecinos del nodo  $i$  con una ruta disponible a  $d$  y  $\beta_1$  es un parámetro que puede disminuir el comportamiento exploratorio de las hormigas (en los experimentos se estableció a 1).

$$P_{nd} = \frac{(F_{nd}^i)^{\beta_1}}{\sum_{j \in N_d^i} (F_{jd}^i)^{\beta_1}} \quad \beta_1 \geq 1$$

Ecuación 6.6. Proceso de envío en el descubrimiento de ruta en AntHocNet.

Si se utiliza un valor alto de  $\beta_1$  se obtiene una mayor posibilidad de mejorar una ruta alternativa. En cambio, si le damos un valor más bajo la posibilidad de encontrarla disminuye, porque las rutas son más parecidas.

### 6.2.2. Encaminamiento de datos

El establecimiento de ruta crea un número de caminos entre origen y destino, que se indica en la tabla de encaminamiento de los nodos. Los datos pueden ser reenviados de modo multisalto de acuerdo a una política probabilística basada en las tablas de encaminamiento. La estrategia consiste en hacer que la carga de los datos se expanda mediante el balanceo de carga. Esto es importante en las redes móviles ad hoc porque el ancho de banda del canal inalámbrico es muy limitado. A continuación mostramos la ecuación para el encaminamiento de los datos.

$$P_{nd} = \frac{(F_{nd})^2}{\sum_{i \in N_d} (F_{id})^2}$$

Ecuación 6.7. Encaminamiento de los datos en AntHocNet.

Esta ecuación significa que cuando un nodo tiene la posibilidad de hacer el salto a varios nodos para llegar al destino  $d$ , selecciona aleatoriamente uno de los saltos como el cuadrado del valor del registro de feromona con probabilidad  $P_{nd}$  de un siguiente salto  $n$ . Se toma el cuadrado para aumentar la posibilidad de elegir los mejores caminos. De acuerdo con esta estrategia, no se elige a priori las rutas que se van a utilizar, sino que las selecciona automáticamente en función de su calidad. El proceso para calcular la ruta alternativa es el siguiente:

- Se genera un número aleatorio  $rand$  entre 0 y 1 con probabilidad uniforme.
- Para cada una de las alternativas posibles de encaminamiento de datos se calcula  $P_{nd}$ .
- Si  $P_{nd}$  es mayor que  $rand$  se selecciona esta alternativa para encaminar los datos y se hace el siguiente salto  $n$  para reenviar los datos al destino  $d$ .

### 6.2.3. Mantenimiento de las rutas establecidas y exploración de nuevas

Mientras se ejecuta una sesión de datos el nodo origen de la sesión envía *hormigas proactivas de mantenimiento* según la tasa de datos (una cada  $n$  paquetes de datos). Estas hormigas siguen el mismo camino que los datos, pero pueden hacerlo en modo *broadcast* si no tiene ruta, y si tiene ruta lo hace probabilísticamente. En otras palabras, si una hormiga alcanza el destino sin realizar ningún *broadcast* entonces esta hormiga sólo muestra una ruta existente. Recoge estimaciones de la calidad de este camino y actualiza los valores de la feromona del camino. En el otro caso, si la hormiga realizó un *broadcast* en cualquier punto, dejará el camino que conoce por la cantidad de feromona y explorará nuevos caminos. Para evitar la proliferación de estos agentes, el número de hormigas proactivas se limita a dos.

Al hacer la búsqueda de nuevos caminos se busca mejorar la ruta con la elección de otra. Para hacerla más eficiente los nodos también hacen uso de *hormigas de difusión de feromona*: los nodos envían mensajes cortos (parecidos a los mensajes HELLO) en modo *broadcast* cada  $t$  segundos. Si un nodo recibe un agente de difusión de feromona de otro nodo  $n$ , entonces añade  $n$  en su tabla de encaminamiento como un nuevo destino. Después espera recibir respuesta del nodo  $n$  cada  $t$  segundos. Si no recibe respuesta (espera hasta 2 mensajes) y si no la encuentra  $n$  será eliminado de la tabla de rutas. Usando estos mensajes los nodos conocen a sus vecinos inmediatos y tienen información de su feromona en su tabla de encaminamiento.

Las hormigas de difusión de feromona también sirven para detectar enlaces rotos. Así, cuando falla un enlace los nodos pueden actualizar sus tablas de encaminamiento.

#### **6.2.4. Gestión de los fallos de enlace**

Los nodos pueden detectar fallos de enlace cuando falla una transmisión *unicast* o cuando un *agente de difusión* de feromona se espera y no se recibe. Cuando un enlace falla, el nodo podría perder la ruta a uno o más destinos. Un ejemplo de fallo de enlace se produce cuando un vecino se mueve más allá del rango de transmisión. En el fallo de enlace se consideran dos clases de problemas:

- Si el nodo tiene otras alternativas al destino o si el destino se ha perdido porque no se ha usado regularmente, se tiene que notificar con mensaje de fallo de enlace.

En este caso el nodo actualizará su tabla de encaminamiento y enviará una hormiga de *notificación de fallo* en modo *broadcast*. Esta hormiga contiene una lista de los destinos que perdieron el camino, el nuevo retardo estimado extremo a extremo y el número de saltos a este destino.

Todos sus vecinos reciben la notificación y actualizan su tabla de feromona usando las nuevas estimaciones. Si los vecinos, por su parte, pierden su mejor o su único camino a un destino debido al fallo, generarán y enviarán una hormiga de fallo en modo *broadcast* hasta que todos los nodos de los diferentes caminos hayan sido notificados de la nueva situación.

- Si se perdió la ruta a un destino regularmente usado por los datos y era la única alternativa del nodo, la pérdida es importante y el nodo debería intentar reparar localmente el camino.

En AntHocNet el nodo sólo repara el camino si descubre que el enlace perdido es debido al fallo de una transmisión de paquetes de datos.

El proceso de reparación de ruta es muy similar a un establecimiento de ruta: el nodo envía una hormiga de reparación de ruta en modo *broadcast* y los nodos intermedios reenvían esta hormiga en modo *broadcast* (con un límite máximo) o *unicast*, dependiendo de si hay información de ruta disponible o no.

El nodo que inicia el proceso de reparación espera un cierto tiempo a que le llegue una hormiga de reparación de ruta *hacia atrás*. Si no la recibe, da por terminado el proceso al no encontrar un camino alternativo al destino. Después actualiza su tabla de rutas eliminando este destino inalcanzable, y se genera una hormiga de notificación de fallo que anuncia la nueva situación.

## 6.3. Variantes del protocolo

### 6.3.1. Ant Colony Based Node Disjoint Hybrid Multi-path Routing

En [32] se propone *Ant Colony based Node Disjoint Hybrid Multi-path Routing* (ACNDHMR), una variante de AntHocNet que incluye la propiedad de multicamino de nodos disjuntos. ACNDHMR facilita el balanceo de carga, la tolerancia de fallos y la reducción del retardo extremo a extremo. Analizando las siguientes métricas de rendimiento: retardo medio extremo a extremo, ratio de entrega de paquetes y balanceo de la carga para varios tiempos de pausa, se comprueba que las dos primeras métricas no varían significativamente. En cambio, se aprecia que la carga se distribuye mejor por rutas de nodos disjuntos.

Como se ha comentado, las rutas múltiples generadas por el AntHocNet son no-disjuntas. Estas rutas pueden tener nodos y enlaces en común. Pero tiene desventajas respecto a las rutas disjuntas que proporciona el algoritmo ACNDHMR. Rutas disjuntas de enlaces o nodos son aquellas que para una misma sesión de datos no comparten nodos o enlaces respectivamente.

### 6.3.2. An Efficient Ant-based Routing Algorithm for MANETs

[53] describe una variación más eficiente de AntHocNet. Este algoritmo reduce la sobrecarga porque maneja de forma eficiente las hormigas (agentes móviles). Las simulaciones demuestran que se reduce la sobrecarga y el retardo extremo a extremo, mientras que el ratio de entrega de paquetes se mantiene igual comparado con la versión de AntHocNet. Este algoritmo introduce varias modificaciones a AntHocNet en la fase de establecimiento de ruta para controlar el número de hormigas que se mueven por la red y actualizar eficientemente los valores del feromona en todos los nodos intermedios cuando se ha establecido la ruta. Las otras fases siguen los procedimientos dados en AntHocNet.

### 6.3.3. Integrated Routing Algorithm

*Integrated Routing Algorithm* (IRA) [18] es un algoritmo híbrido con una parte proactiva que se ejecuta en segundo plano y ofrece un servicio de encaminamiento *best effort* y otra parte reactiva porque puede ser llamado bajo demanda y ofrecer un servicio orientado a conexión. La propiedad más importante de IRA es que permite elegir por separado entre el encaminamiento proactivo y reactivo para cada sesión de datos. Los paquetes de datos pueden seguir tanto rutas proactivas como reactivas. Existe una sinergia (cooperación interactiva) porque la parte reactiva del algoritmo confía en la información de encaminamiento proactiva. Este algoritmo resulta ventajoso en una serie de escenarios.

## 6.4. Evaluación

En [13] los autores realizan una gran cantidad de experimentos para evaluar AntHocNet. Diseñan un escenario real con obstáculos y patrones de tráfico reales. Las simulaciones se realizan con QualNet. Los experimentos demuestran que el rendimiento de AntHocNet es mejor que AODV y OLSR. AntHocNet tiene más ratio de entrega de datos, menor retardo y *jitter*. Además, en la mayoría de las pruebas, AntHocNet tiene menos sobrecarga que AODV y OLSR.

En [14] Di Caro *et al.* realizan el estudio del rendimiento de AntHocNet y AODV en un entorno urbano utilizando aplicaciones en tiempo real. Se realiza una simulación realista en términos de radio de propagación, restricción de movilidad de los nodos y tráfico de los datos. Para las simulaciones se usa QualNet. En la mayoría de los escenarios AntHocNet mejora a AODV en términos de ratio de entrega y retardo. En escenarios urbanos, AntHocNet tiene la ventaja de que la densidad local (número de vecinos) experimentada por cada nodo es relativamente baja y crece lentamente. Se observa que la cantidad de nodos influye mucho en el ratio de entrega, mientras que el movimiento de los nodos no parece que tenga tanta influencia. También se observa que la densidad de nodos tiene un fuerte impacto en el ratio de entrega, mientras que la velocidad de los nodos parece tener relativamente una menor repercusión.



## 7. ANTOR: PROTOCOLO DE ENCAMINAMIENTO ACO HÍBRIDO

---

El protocolo propuesto es un protocolo de encaminamiento ACO híbrido que toma como punto de partida el algoritmo de Ducatelle [17] e introduce los siguientes aspectos:

- Protocolo de enlace/nodo disjunto.
- Separación del proceso de difusión de feromona.
- Utilización de la métrica *distancia* en la exploración de caminos.

### 7.1. Protocolo de enlace / nodo disjunto

En este tipo de protocolos existen dos clases de rutas: disjuntas de nodo y disjuntas de enlace. Las primeras corresponden a rutas en las que no se comparten nodos y las segundas hacen referencia a rutas que no comparten enlaces.

Se cumple la propiedad de que toda ruta de nodo disjunto es también de enlace disjunto, pero no al contrario. Ambos tipos de rutas disjuntas presentan las siguientes ventajas:

1. Un fallo en un nodo sólo afecta a una ruta.
2. El balanceo de carga es mejor porque no se repiten rutas por la propiedad disjunta.

No obstante, la utilización de este tipo de rutas presenta los siguientes inconvenientes:

1. Se necesitan más recursos porque no comparten enlaces y nodos.
2. Estas rutas son más difíciles de descubrir, porque limitamos los nodos que pueden ser visitados.

### 7.1.1. Rutas de nodo disjunto

Como se ha señalado anteriormente, una ruta de nodo disjunto es aquella en la no se comparten nodos en una misma sesión de datos. La Figura 7.1 muestra un ejemplo de rutas de nodo disjunto.

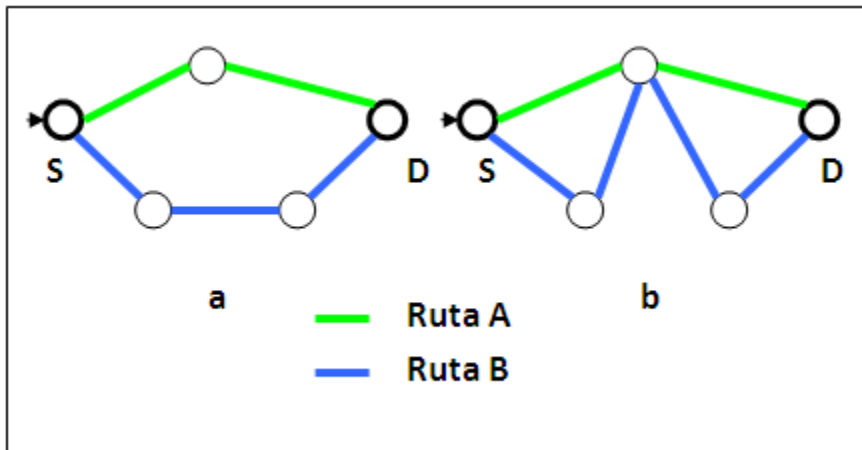


Fig. 7.1. Rutas de nodos disjuntos (a) versus no disjuntos (b).

El procedimiento para calcular rutas de nodo disjunto es diferente que el de enlace disjunto. En este caso se *marca* cada nodo intermedio que se visita para no repetir nodos. Esta información de nodos ya visitados se almacena localmente en la tabla de vecinos del nodo que pretende realizar un proceso proactivo de exploración de nueva rutas.

En la Figura 7.2 el nodo emisor A realiza un proceso proactivo hacia el destino E. La ruta A constituye la ruta creada en el proceso reactivo. El proceso proactivo se tiene que realizar por la ruta B a través de los nodos B y D, porque el nodo C ya pertenece al proceso reactivo y no se pueden repetir nodos. El nodo A *marca* en su tabla de vecinos al nodo C como ya visitado.

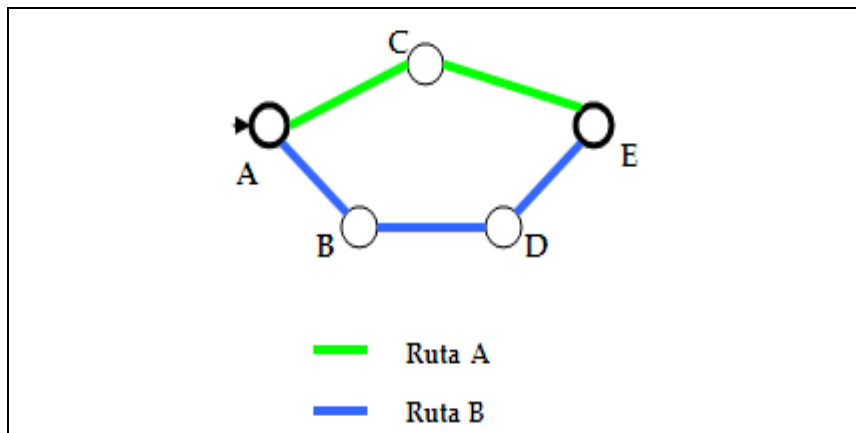


Fig. 7.2. Esquema de rutas de nodo disjunto.

### 7.1.2. Rutas de enlace disjunto

Como se ha señalado anteriormente, una ruta de enlace disjunto es aquella en la que no se comparten enlaces en una misma sesión de datos. La Figura 7.3 muestra un ejemplo de rutas de enlace disjunto, donde se observa cómo se constituyen las rutas de enlace disjunto con respecto a las de enlace no disjunto. La idea básica para hallar y representar rutas de enlace disjunto consiste en *marcar* cada enlace disjunto con una etiqueta que indique cuál es el origen de la sesión de datos.

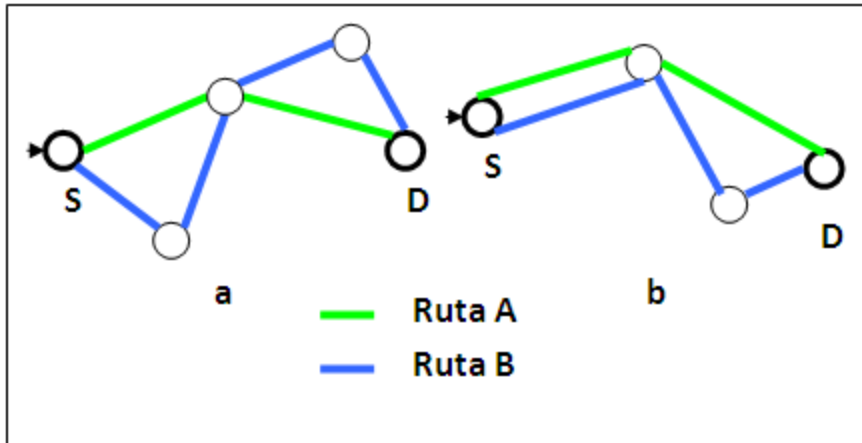


Fig. 7.3. Rutas de enlaces disjuntos (a) versus no disjuntos (b).

En la Fig. 7.4 mostramos un ejemplo de cálculo de rutas disjuntas donde sólo tenemos una sesión de datos formada por el nodo origen A y el nodo destino D. Primero se calcula la ruta disjunta  $S_1$  marcando cada enlace disjunto con el origen de la sesión S. Después se calcula la ruta alternativa  $S_2$  teniendo en cuenta de no repetir enlaces que ya pertenecen al origen S. La característica de poder compartir nodos hace que se puedan visitar nodos intermedios por rutas alternativas, pero no los enlaces que pertenecen a otras rutas.

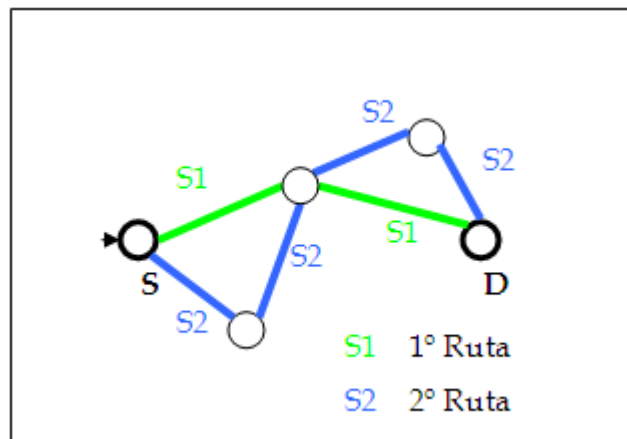


Fig. 7.4. Esquema de rutas de enlaces disjuntos.

Cuando trabajamos con varias sesiones de datos, varios pares origen-destino, la propiedad disjunta también se cumple porque el marcado de las rutas es único para cada ruta disjunta perteneciente a cada sesión de datos. La Figura 7.5 muestra un ejemplo en el que se solapan rutas disjuntas pertenecientes a sesiones de datos diferentes. Puede observarse que hay dos sesiones de datos formadas por los pares (B-C) y (A-D). Se comprueba que aunque haya solapamiento al ser sesiones distintas de datos no altera el comportamiento del protocolo.

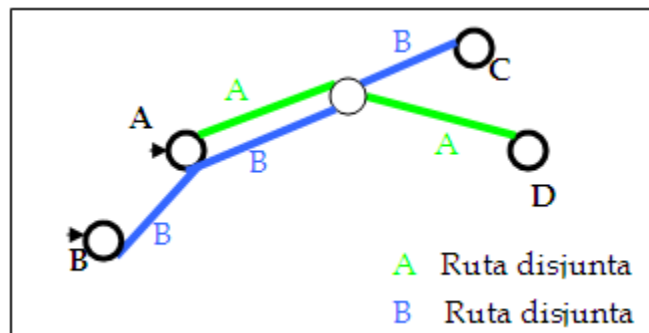


Fig. 7.5. Esquema de rutas disjuntas para dos sesiones de datos.

## 7.2. Separación del proceso de difusión de feromona

En la propuesta de Ducatelle una misma ruta puede tener simultáneamente valores de feromona *regular* y *virtual*. En el protocolo que se propone una ruta no puede tener simultáneamente un valor de feromona *regular* y otro de feromona *virtual*. Para llevar a cabo esta separación se ha modificado la ecuación de exploración de rutas de Ducatelle, resultando la siguiente expresión:

$$P_{in}^d = \frac{(\psi_{in}^d)^{\beta_2}}{\sum_{j \in N_i^d} (\psi_{ij}^d)^{\beta_2}}, \quad \psi \in \begin{cases} k & \text{virtual} \\ \tau & \text{regular} \end{cases}$$

### 7.3. Utilización de la métrica distancia en la exploración de caminos

En nuestra propuesta se tiene en cuenta el número de saltos de las mejores rutas halladas. De esta forma se controla que una hormiga proactiva no pueda recorrer más nodos de los establecidos por el límite del número de saltos. Este límite de saltos se establece según las mejores rutas (menor distancia en número de saltos) calculadas anteriormente.

### 7.4. Funcionamiento

Para una mejor comprensión del protocolo propuesto nos apoyamos en el ejemplo de la Figura 7.6 donde se observa que tenemos una única sesión de datos formada por los pares (A, D), esto es, A envía la información al destino D.

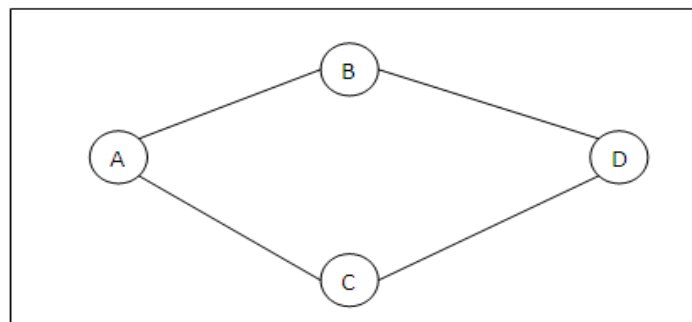


Fig. 7.6. Ejemplo de escenario del funcionamiento del protocolo.

Se entiende por iniciar la sesión de datos al hecho de que se quiere enviar datos desde el origen A al destino D. Aunque no hay sesión de datos se crean rutas entre pares de vecinos que están a un salto. Se originan 8 rutas (A-C, C-A, A-B, B-A, B-D, D-B, C-D, D-C).

El proceso de marcado y asentamiento de ruta para el cálculo de  $\tau$  (véase Figura 7.7) y  $h$  es idéntico al utilizado por Ducatelle. En este caso el valor de feromona virtual  $k$  vale 0 para todas las rutas.

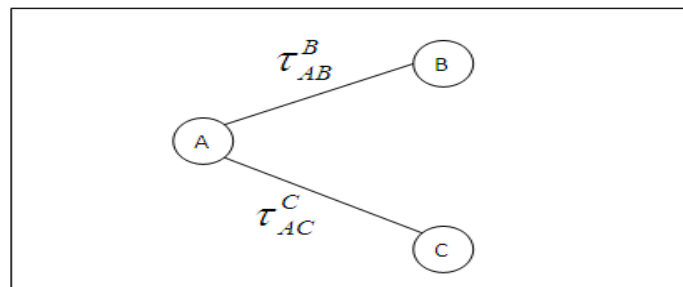


Fig. 7.7. Esquema del cálculo de los valores de feromona regular.

Las tablas de rutas quedan de la siguiente forma:

Rutas para Nodo A	Destino	Siguiente Salto	$\tau$	$k$	$h$
Vecinos	B	B	0,3	0	0,3
	C	C	0,3	0	0,3

Tabla 7.1. Tabla de rutas para el nodo A.

Rutas para Nodo D	Destino	Siguiente Salto	$\tau$	$k$	$h$
-------------------	---------	-----------------	--------	-----	-----

Vecinos	B	B	0,3	0	0,3
	C	C	0,3	0	0,3

Tabla 7.2. Tabla de rutas para el nodo D.

Rutas para Nodo B	Destino	Siguiente Salto	$\tau$	$k$	$h$
Vecinos	A	A	0,3	0	0,3
	D	D	0,3	0	0,3

Tabla 7.3. Tabla de rutas para el nodo B.

Rutas para Nodo C	Destino	Siguiente Salto	$\tau$	$k$	$h$
Vecinos	A	A	0,3	0	0,3
	D	D	0,3	0	0,3

Tabla 7.4. Tabla de rutas para el nodo C.

Otro proceso que se realiza independientemente de la sesión de datos es la *difusión de encaminamiento*. Concretamos un caso: el nodo A todavía no tiene ruta a D a través de C, así que el nodo C difunde la información de sus destinos al nodo A. El nodo C informa a A de la mejor ruta al destino D, en este caso sólo tiene una. Si hubiera otras alternativas elegiría el mejor valor *regular* o *virtual* al destino D. En la Figura 7.8 se ve el proceso de difusión de feromona donde la línea discontinua representa la feromona *virtual* cuyo valor está función del mejor destino de C a D (0,3).

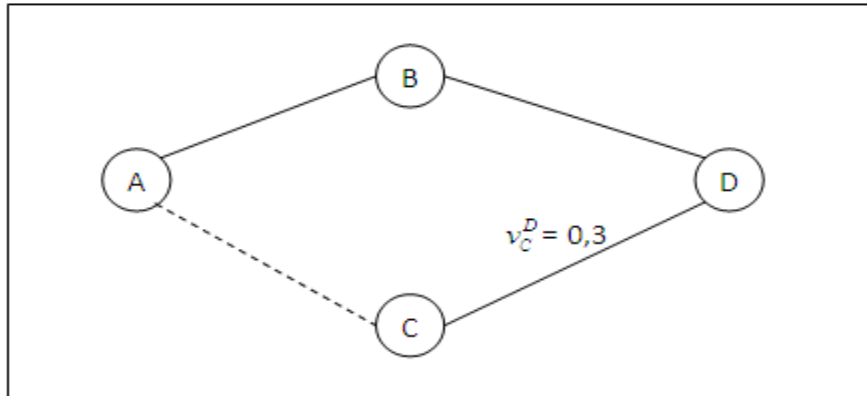


Fig. 7.8. Esquema de difusión de encaminamiento.

El nodo A que recibe la información de C aplica la ecuación de difusión de feromona de Ducatelle, quedando la nueva tabla de rutas de este nodo tal y como se indica en la Tabla 7.5.

Rutas para Nodo A	Destino	Siguiente Salto	$\tau$	$k$	$h$
Vecinos	B	B	0,3	0	0,3
	C	C	0,3	0	0,3
	D	C	0	0,23	0

Tabla 7.5. Tabla de rutas para A en el proceso de difusión.

Si se diera el caso de que A tuviera un valor *regular* el valor *virtual* no se actualizaría en el nodo A.

Este proceso de *difusión de encaminamiento* se repite constantemente.

El resto de procesos del protocolo que se propone es análogo al establecido por Ducatelle.



## 8. SIMULACIONES Y RESULTADOS

---

Para evaluar el comportamiento del protocolo se ha empleado el simulador de redes Network Simulator 3 (NS-3) [51]. En el Anexo B de esta memoria se recogen los detalles del escenario de simulación. Las métricas de rendimiento más importantes que se han utilizan para evaluar la propuesta han sido:

- *ThroughPut*: Volumen de trabajo o de información que fluye a través de un sistema. Se calcula dividiendo el total de bits entregados al destino por el tiempo de entrega de paquetes.
- *Delivery Packet Ratio*: Relación del número de paquetes enviados y el número de paquetes correctamente entregados.
- *Overhead in number of Packets*: Relación entre el número total de paquetes de control transmitidos por los nodos de la red y el número de paquetes de datos entregados a sus destinos.
- *Overhead in number of Bytes*: Relación entre el número total de bytes de control transmitidos y los bytes de datos entregados.
- *Average End-To-End Delay*: Medida de efectividad acumulativa de los retardos experimentados por los paquetes que van desde el origen al destino.

## 8.1. Resultados

Los resultados de las gráficas que se muestran a continuación son orientativos, debido a que la capacidad de procesamiento de NS-3 es mucho menor que la de Qualnet. A continuación comentamos las diferentes gráficas según las métricas de rendimiento planteadas. Se comparan OLSR, AntHocNet y AntOR en la versión de ruta de enlace disjunto.

### 8.1.1. Volumen de trabajo

En la Figura 8.1 puede observarse cómo AntOR mejora sensiblemente a los otros dos protocolos en el caso de redes poco densas. En el caso de redes más densas, el volumen de trabajo se iguala. También puede observarse como AntOR supera a OLSR en todos los casos.

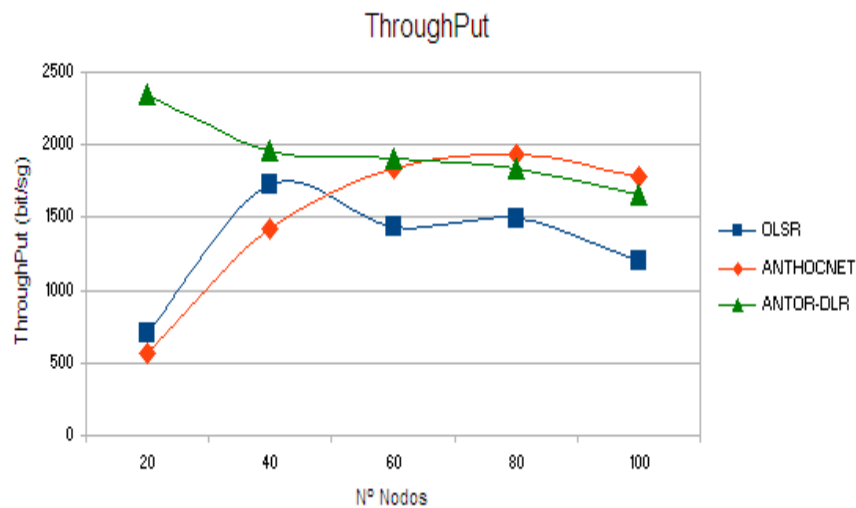


Fig. 8.1. Volumen de trabajo (*ThroughPut*).

### 8.1.2. Ratio de entrega de paquetes

En la Figura 8.2 se observa que el ratio de entrega de paquetes en AntOR en situaciones extremas (redes poco densas o muy densas) es superior al ratio en los otros dos protocolos. En situaciones intermedias, los valores se igualan. Asimismo, se observa la robustez del protocolo ya que la curva muestra una mayor estabilidad frente a variaciones en las condiciones de red.

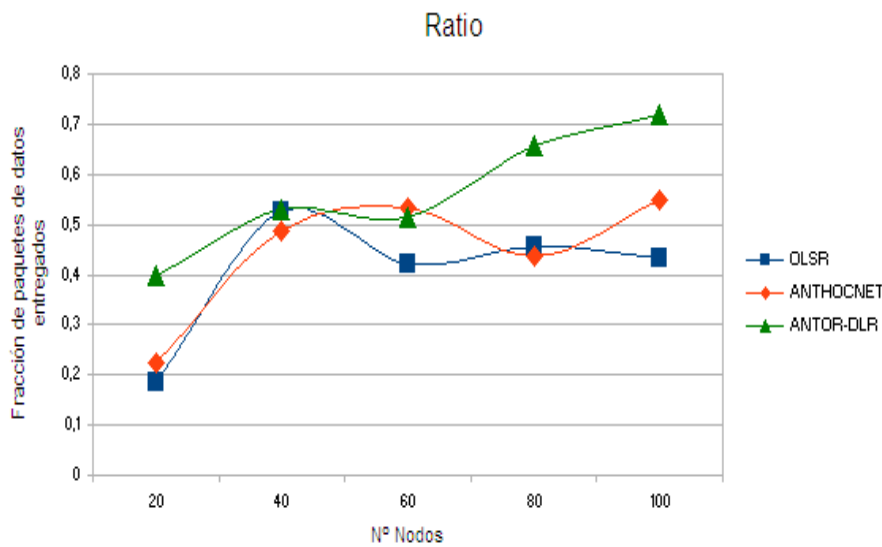


Fig. 8.2. Ratio de entrega de paquetes.

### 8.1.3. Sobrecarga en el número de paquetes

En la Figura 8.3 se observa que la sobrecarga en el número de paquetes de AntOR es inferior a la de AntHocNet, el trabajo de partida. Esto certifica que las modificaciones realizadas han sido correctas. Así, la capacidad disjunta de enlaces hace que se envíen menos PFA. Al tolerar mejor los fallos de enlace por la existencia de rutas alternativas también se necesita enviar menos mensajes RRFA en el proceso de reparación de rutas.

En la citada Figura también se observa que la sobrecarga de AntOR es mayor que en OLSR pero debe tenerse en cuenta que el intervalo utilizado de los mensajes HELLO es el doble en OLSR (cada 2 segundos) que en AntHocNet y AntOR (que ha sido de 1 segundo), con lo cual ambas gráficas no son directamente comparables.

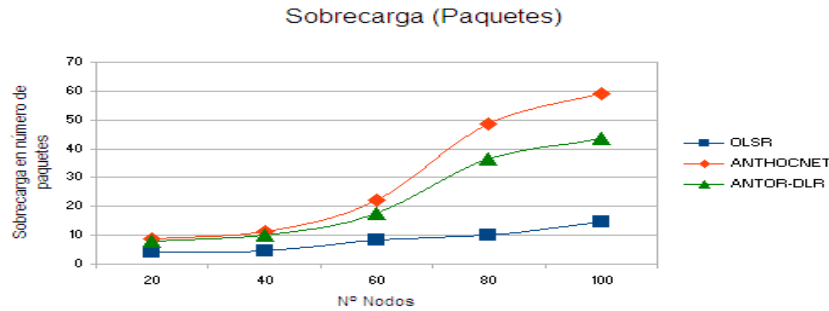


Fig. 8.3. Sobrecarga en el número de paquetes.

#### 8.1.4. Sobrecarga en el número de bytes

En la Figura 8.4 se aprecia que la sobrecarga en el número de bytes (que es un parámetro más representativo) es similar en los tres protocolos en el caso de redes poco densas, mientras que en el caso de redes más densas AntOR presenta una menor sobrecarga que AntHocNet y las diferencias respecto a OLSR se reducen.

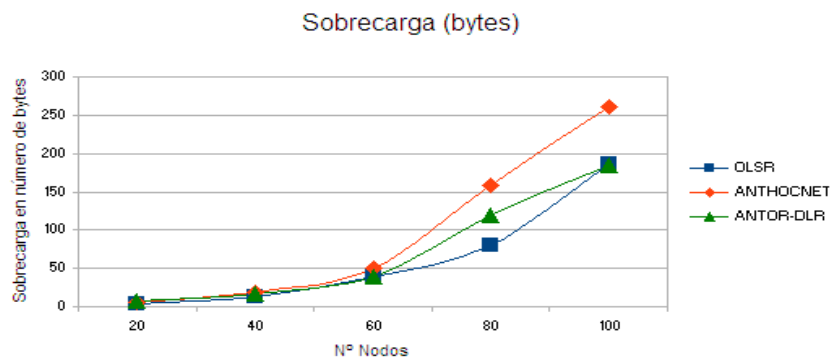


Fig. 8.4. Sobrecarga en el número de bytes.

### 8.1.5. Retardo medio extremo a extremo

En la Figura 8.5 se observa que el retardo medio extremo a extremo de AntOR en redes poco densas es superior al encontrado en los otros dos protocolos. No obstante, el valor de este retardo no es muy elevado. Para redes más densas, AntOR supera a OLSR, que se muestra bastante inestable, y se iguala prácticamente a AntHocNet. Aunque este último presenta mejores valores de retardos en redes poco densas parece mostrarse más inestable frente al incremento de nodos que AntOR.

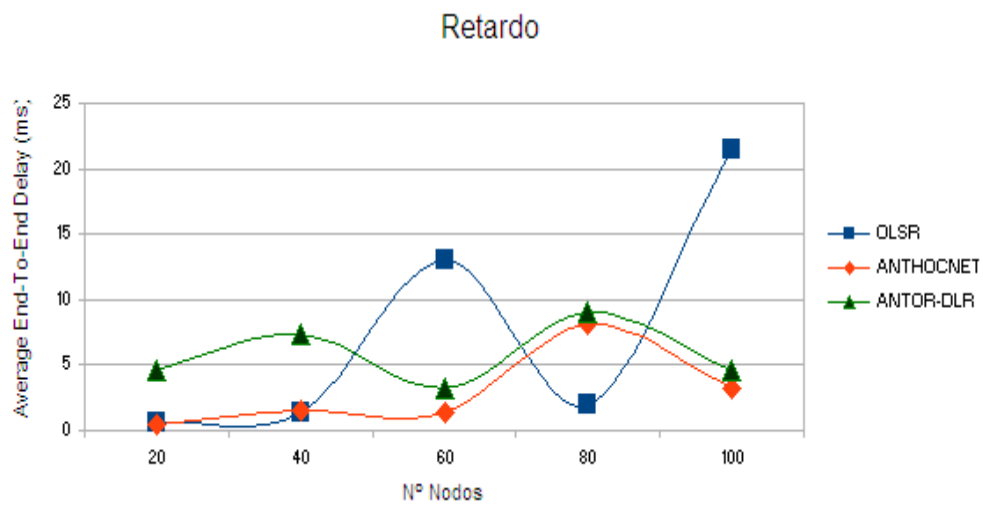


Fig. 8.5. Retardo medio extremo-a-extremo.



## 9. CONCLUSIONES Y TRABAJO FUTURO

---

Se ha presentado un protocolo de encaminamiento para redes móviles ad hoc denominado AntOR (*Ant Optimized Routing*). Este protocolo se clasifica como un protocolo de encaminamiento ACO (basado en el algoritmo de la colonia de hormigas) híbrido y puede considerarse como una variante del protocolo AntHocNet que mejora las prestaciones de éste en parámetros tan relevantes como el ratio de entrega de paquetes, la sobrecarga en el número de paquetes o la sobrecarga en el número de bytes. El protocolo se muestra estable en las simulaciones realizadas lo que permite pensar en su escalabilidad.

Como características más relevantes de esta variante del AntHocNet propuesta pueden señalarse las siguientes:

- La utilización de rutas de enlace/nodo disjuntas.
- La separación del proceso de difusión de las feromonas.
- El proceso de exploración de rutas tiene en consideración el número de saltos de las mejores rutas encontradas anteriormente.

Características éstas últimas orientadas a la disminución de la sobrecarga en el número de bytes y al aumento del ratio de entrega de paquetes de AntHocNet, aspectos que se han conseguido con la presente propuesta.

## 9.1. Trabajo Futuro

Existen numerosas líneas futuras de trabajo posibles, entre las que cabe señalar las siguientes:

- Especificación de nuevos escenarios de simulación para validar los resultados obtenidos.
- Analizar otras métricas de rendimiento como puede ser el *jitter*.
- Implementar la versión de nodo disjunto (se ha implementado la de enlace disjunto).
- Realizar una comparativa con más protocolos de referencia.

## REFERENCIAS

---

- [1] Norman Abramson: "The ALOHA system-another alternative for computer communications". *Proceeding of the Fall 1970 AFIPS Computer Conference*, pp. 281-285, 1970.
- [2] John S. Baras, Harsh Mehta: "A Probabilistic Emergent Routing Algorithm for Mobile Ad Hoc Networks", *WiOpt' 03: Modeling and optimization in Mobile Ad hoc wireless networks*, Mar. 2003.
- [3] Elizabeth M. Belding-Royer: "Routing Approaches in Mobile Ad hoc Networks". *Mobile Ad hoc Networking*, Stefano Basagni, Marco Conti, Silvia Giordano, Ivan Stojmenovic (ed). Wiley Inter-Science. 1st Edition. New Jersey. ch.10, pp. 275-300. 2004.
- [4] I. Chakeres, Charles E. Perkins: "Dynamic MANET On-demand (DYMO) Routing". *Internet Draft*, March 2009 (work in progress).  
<http://tools.ietf.org/html/draft-ietf-manet-dymo-17>.
- [5] Tsu-Wei Chen, Mario Gerla: "Global State Routing: A New Routing Scheme for Ad-hoc Wireless Networks". *Proceedings of IEEE International Conference on Communications (ICC'98)*, June 1998.
- [6] T. Clausen, P. Jacquet: "Optimized link state routing protocol (OLSR)". *IETF RFC3626*, October 2003.  
[www.ietf.org/rfc/rfc3626.txt](http://www.ietf.org/rfc/rfc3626.txt)

- [7] M.S. Corson, A. Ephremides: "A distributed routing algorithm for mobile wireless networks". *ACM/Baltzer Wireless Networks*, vol. 1, no. 1, pp. 61-81, February 1999.
- [8] G. Di Caro: "Ant Colony Optimization and its application to adaptive routing in telecommunication networks". *PhD thesis in Applied Sciences*, Polytechnic School, Université Libre de Bruxelles, Brussels, Belgium, 2004.
- [9] G. Di Caro, M. Dorigo: "Two ant colony algorithms for best-effort routing in datagram networks". In *Proceedings of the Tenth IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS'98)*, pages 541-546. IASTED/ACTA Press, 1998.
- [10] G. Di Caro, M. Dorigo: "AntNet: Distributed Stigmergetic Control for Communications Networks" (reprinted by permission from JAIR), *Vivek, A Quarterly in Artificial Intelligence*, Vol. 12(3-4), Pag. 2-37, 1999.
- [11] G. Di Caro, F. Ducatelle, L.M. Gambardella: "AntHocNet: probabilistic multi-path routing in mobile ad hoc networks using ant-like agents". *Technical Report 16-04, IDSIA, Lugano, Switzerland, April 2004*.
- [12] G. Di Caro, F. Ducatelle, L.M. Gambardella: "AntHocNet: an ant-based hybrid routing algorithm for mobile ad hoc networks" *Proceedings of PPSN VIII - Eight International Conference on Parallel Problem Solving from Nature*, Birmingham, UK, September 18-22, 2004, Springer-Verlag, Lecture Notes in Computer Science, Vol. 3242 (BEST PAPER AWARD).
- [13] G.A. Di Caro, F. Ducatelle, L.M. Gambardella: "Studies of routing performance in a city-like testbed for mobile ad hoc networks", *Technical Report IDSIA-07-06, March 2006, IDSIA, Lugano, Switzerland*.

- [14] G.A. Caro, F. Ducatelle, L. M. Gambardella: "A Simulation Study of Routing Performance in Realistic Urban Scenarios for MANETs". In *Proceedings of the 6th international Conference on Ant Colony Optimization and Swarm intelligence* (Brussels, Belgium, September 22 - 24, 2008).
- [15] M. Dorigo, V. Maniezzo, A. Coloni: "Ant System: Optimization by a colony of cooperating agents". *IEEE Transactions on Systems, Man, and Cybernetics – Part B*, 26(1), 29–41. (1996)
- [16] Falko Dressler: "Self-Organization in Ad Hoc Networks: Overview and Classification". *Technical Report, Autonomic Networking Group, University of Erlangen*, February 2006.
- [17] F. Ducatelle: "Adaptive Routing in Ad Hoc Wireless Multi-hop Networks", *PhD thesis*, Università della Svizzera Italiana, Istituto Dalle Molle di Studi sull'Intelligenza Artificiale, 2007.
- [18] F. Ducatelle, G. Di Caro, L. Gambardella: "A New Approach for Integrating Proactive and Reactive Routing in Mobile Ad Hoc Networks", *Proceedings of the 5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, Atlanta, Georgia, USA, 29 Sep - 2 Oct, 2008
- [19] James A. Freebersyser, Barry Leiner: "A DoD perspective on mobile ad hoc networks". *Ad Hoc Networking*, Charles E. Perkins (ed.). Addison-Wesley, ch. 2, pp. 29-51, 2001.
- [20] L. M. Feeney: "An Energy Consumption Model for Performance Analysis of Routing Protocols for Mobile Ad Hoc Networks", *Mobile Networks and Application*, vol. 6, 3, pp. 239 - 249, 2001.

- [21] Mario Gerla, Xiaoyan Hong, Guangyu Pei: "Fisheye State Routing Protocol (FSR) for Ad Hoc Networks". *Internet Draft*, June 2002.
- <http://tools.ietf.org/html/draft-ietf-manet-fsr-03>
- [22] D.M. Gordon: "Ants at Work: How an Insect Society Is Organized". *WW Norton & Company*, 2000.
- [23] S. Goss, S. Aron, J.L. Deneubourg, J.M. Pasteels: "Self-organized shortcuts in the argentine ant". *Naturwissenschaften*, 76(12):579-581, 1989.
- [24] M. Günes, U. Sorges, and I. Bouazizi. ARA - The ant-colony based routing algorithm for MANETs. In *Proceedings of the ICPP International Workshop on Ad Hoc Networks (IWAHN)*, 2002.
- [25] Z. J. Haas, M. R. Pearlman: "The zone routing protocol (ZRP) for ad hoc networks", *IETF Draft* (July 2002).
- [tools.ietf.org/id/draft-ietf-manet-zone-zrp-04.txt](http://tools.ietf.org/id/draft-ietf-manet-zone-zrp-04.txt)
- [26] C. Hedrick: "Routing Information Protocol", Rutgers University, RFC 1058 June 1998.
- <http://tools.ietf.org/html/rfc1058>
- [27] O. Hossein, T. Saadawi: "Ant routing algorithm for mobile ad hoc networks (ARAMA)", *Proceedings of the 22nd IEEE International Performance, Computing, and Communications Conference*, Phoenix, Arizona, USA, April 2003, pp. 281-290.

- [28] IEEE 802.11-1999: "IEEE Standard for Local and Metropolitan Area Networks – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications". *IEEE Standard Association, The Working Group for Wireless LAN*, Edición de 1999, ratificada en Junio 2003.
- [29] Sushant Jain: "Energy Aware Communication in Ad – Hoc Networks". *Technical Report UW-CSE 03-06-03*, Computer Science and Engineering, University of Washington, Seattle, January 2003.  
<ftp://ftp.cs.washington.edu/tr/2003/06/UW-CSE-03-06-03.pdf>.
- [30] Geetha Jayakumar, G. Gopinath: "Ad Hoc Mobile Wireless Networks Routing Protocols – A Review". *Journal of Computer Science* 3(8). 2007.
- [31] D. Jonson, Y. Hu, D. Maltz: "The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4". *RFC4728*, February 2007.  
<http://www.ietf.org/rfc/rfc4728>.
- [32] B. Kalaavathi, K. Duraiswamy: "Ant Colony Based Node Disjoint Hybrid Multi-path Routing for Mobile Ad Hoc Network". *Journal of Computer Science* 4 (2): 80-86, ISSN 1549-3636 1- Feb- 2008 (JCR F.I: 0,441).
- [33] Shahab Kamali, Jaroslav Opatrny: "A Position Based Ant Colony Routing Algorithm for Mobile Ad-hoc Networks", *JOURNAL OF NETWORKS*, VOL. 3, NO.4, Academy Publisher, Apr-08.
- [34] Young-Bae Ko, Nitin H. Vaidya: "Locationaided routing (LAR) in mobile ad hoc networks". *Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networkings*. pp. 66-75, 1998.

- [35] Lianggui Liu, Guangzeng Feng: "A Novel Ant Colony Based QoS-Aware Routing Algorithm for MANETs", ICNC 2005, LNCS 3612, Springer-Verlag Berlin Heidelberg, 2005, pp. 457 - 466.
- [36] Joseph P. Macker, M. Scott Corson: "Mobile Ad hoc Networks (MANETs): Routing Technology for Dynamic, Wireless Networking". *Mobile Ad hoc Networking*, Basagni Stefano, Conti Marco, Giordano Silvia, Stojmenovic Ivan (ed). Wiley Inter-Science. 1st Edition. New Jersey. ch.9, pp. 255-274. 2004.
- [37] Shivanajay Marwaha, Chen Kong Tham, Dipti Srinivasan: "Mobile Agents based Routing Protocol for Mobile Ad Hoc Networks", *IEEE Global Telecommunications Conference (GLOBECOM'02)*, Taipei, Taiwan. 2002.
- [38] Mobile Ad-hoc Networks Work Group (manet).  
<http://tools.ietf.org/wg/manet/>
- [39] Shree Murthy, J. J. Garcia-Luna-Aceves: "A routing protocol for packet radio networks". *Proceedings of First Annual ACM International Conference on Mobile Computing and Networking*. pp. 86-95, 1995.
- [40] R. Ogier, F. Templin, M. Lewis: "Topology Dissemination Based on Reverse-Path Forwarding (TBRPF)". *RFC 3684*, February 2004.  
<http://www.ietf.org/rfc/rfc3684>
- [41] Open Shortest Path First IGP Work Group (OSPF).  
<http://tools.ietf.org/wg/autoconf/>

- [42] Eseosa Osagie, Parimala Thulasiraman, Ruppa K. Thulasiram: "PACONET: imProved Ant Colony Optimization routing algorithm formobile ad hoc NETworks", *22nd International Conference on Advanced Information Networking and Applications*, 2008, pp.204-211
- [43] V. Park, S. Corson: "Temporally-Ordered Routing Algorithm (TORA) Version 1 Functional Specification". *Internet Draft*, July 2001.  
<http://tools.ietf.org/html/draft-ietf-manet-tora-spec-04>
- [44] Charles E. Perkins, Elizabeth M. Belding-Royer, S. Das: "Ad hoc On-Demand Distance Vector (AODV) Routing". *RFC3561*, July 2003.  
<http://tools.ietf.org/html/rfc3561>
- [45] Charles Perkins, Pravin Bhagwat: "Highly dynamic destination-sequenced distance-vector routing (dsv) for mobile computers". In *ACM SIGCOMM'94 Conference on Communications Architectures, Protocols and Applications*, pages 234-244, 1994.
- [46] Ricardo Puttini, Maíra Hanashiro, Fábio Miziara, Rafael de Sousa, L. Javier García Villalba, C. J. Barenco: "On the Anomaly Intrusion-Detection in Mobile Ad Hoc Network Environments". *Lecture Notes in Computer Science*, Springer-Verlag, LNCS 4217, Vol. 4217/2006, pp. 182-193, 2006.
- [47] S. Rajagopalan, C. Shen: "ANSI: a unicast routing protocol for mobile ad hoc networks using swarm intelligence", *Proceedings of the International Conference on Artificial Intelligence*, 2005, pp.24-27.
- [48] Jyoti Raju, J. J. Garcia-Luna-Aceves: "A New Approach To On-Demand Loop-Free Multipath Routing". *Proceedings of 8th Annual IEEE International*

*Conference on Computer Communications and Networks (ICCCN)*, pp. 522–527, October 1999.

- [49] Laura Rosati, Matteo Beriolo, Gianluca Reali: “Distributed Ant Routing ant routing – On Ant routing algorithms in ad hoc networks with critical connectivity”, *Ad Hoc Network Journal*, Volume 6, Issue 6, August 2008.
- [50] J. Silberholz, B. Golden: “The generalized traveling salesman problem: A new genetic algorithm approach”, 2007, pp. 165-181.
- [51] The ns-3 network simulator.  
<http://www.nsnam.org>
- [52] Jianping Wang, Eseosa Osagie, Parimala Thulasiraman, Ruppia K. Thulasiram: “HOPNET: A hybrid ant colony optimization routing algorithm for mobile ad hoc network”, *Ad Hoc Netw.* Elsevier Science Publishers 7 (4): 690-705, ISSN 1570-8705 (2009).
- [53] Miae Woo, Ngo Huu Dung, Woo Jong Roh: "An Efficient Ant-based Routing Algorithm for MANETs", *Advanced Communication Technology*, 2008. ICACT 2008. *10th International Conference on*, vol.2, no., pp.933-937, 17-20 Feb. 2008
- [54] Zhengyu Wu, Hantao Song, Shaofeng Jiang, Xiaomei Xu: “Ant based Energy Aware Disjoint Multipath Routing Algorithm in MANETs”, *International Conference on Multimedia and Ubiquitous Engineering*, 2007. Volume 1, Issue 1, 26-28 April 2007, pp: 674 - 679.
- [55] Xiangquan Zheng, Wei Guo, Renting Liu: “An ant-based distributed routing algorithm for ad-hoc networks (ADRA)”, *International Conference on Communications, Circuits and Systems*, ICCCAS 2004, Volume 1, Issue 1 27-29 June 2004, pp: 412 - 417

## ANEXO A: SIMULADOR DE REDES NS-3

---

NS-3 es un simulador de red de eventos discretos para sistemas de Internet, dedicado a la investigación y al uso educativo. NS-3 es de código libre y utiliza la licencia *GNU GPLv2*, y su uso está públicamente disponible para la investigación y el desarrollo. NS-3 realmente se puede considerar un conjunto de librerías para realizar simulaciones de redes.

NS-3 es un simulador completamente nuevo y reemplaza al popular ns-2 simulator.

NS-3 está implementado en C++. El utilizar este lenguaje de programación tan extendido, tiene las siguientes ventajas:

- Expresividad respecto a otros lenguajes.
- Proporciona nuevas características.
- Portabilidad.
- Eficiencia.

NS-3 realmente se puede considerar un conjunto de librerías para realizar simulaciones de redes. Tiene implementadas diferentes capas del modelo TCP/IP, y en ellas se encuentran diferentes protocolos y estructuras de datos contempladas en los protocolos de redes actuales.

Como es un simulador que se desarrolla en comunidad y con código abierto, las implementaciones van aumentando gracias a la colaboración desinteresada de desarrolladores de todo el mundo.

## Características NS-3

Las principales características de este simulador de redes son:

- Fácilmente extensible
- Código abierto y libre para el estudio o modificaciones
- Multiplataforma
- Desarrollo en comunidad

## Implementación

NS-3 está formado de las siguientes partes:

- Nodos: Un nodo puede ser visto como la carcasa de una computadora, en la que se puede poner Dispositivos de Red, protocolos y aplicaciones.
- Dispositivos de red: Corresponde a un hardware concreto que comunica el *host* con el canal. Un mismo nodo puede tener varios dispositivos, pero todos con direcciones IP únicas.
- Canales: Corresponden al medio por el que se envía los datos desde el emisor al receptor.
- Pilas de Internet: Lista de protocolos que se añaden al nodo.
- Aplicaciones: Gestionan el tráfico de la red.
- Contenedores: Almacena varios elementos de la misma clase.

- *Helpers*: Son elementos que ayudan a la creación y sirve de enlace de las partes del simulador.

## Protocolos y estructuras implementadas

A continuación se muestran los diferentes componentes de NS-3 según las capas. Como se puede comprobar NS-3 está todavía en fases de desarrollo.

Aplicación	Aplicación OnOff, PacketSink, UdpEcho, V4ping, CBR, API sockets asíncronos, Sockets de paquetes
Capa de Transporte	UDP, TCP
Capa de Red	Unicast: Ipv4, global static routing Multicast: static routing MANET: OLSR, ANTHOCNET, ANTOR
Capa de Enlace	Punto-A-Punto, CSMA, 802.11 MAC low / high Control de Tasa de Envío
Capa Física	802.11a, 802.11b, Modelos de de retardo de propagación y pérdida de propagación.
Soporte	Generador de números aleatorios, trazas, Logging, Callbacks, visualizadores de movilidad, modelos de error

Fig. A.1. Modelos de posición en NS-3

## Estructuras de clases de NS-3

Esta estructura de clases facilita la tarea del programador, ya que es fácil encontrar los elementos deseados para una simulación, así como las configuraciones requeridas para la estructura de la red recreada. Estos son los principales módulos:

- Simulator: se centran las clases involucradas en la simulación en sí, como son *Time* y *Scheduler*. Objetos de estas clases son necesarias para cualquier simulación en un escenario del simulador NS-3, ya que se usan para simular todos los eventos que se suceden en una simulación, como la entrada o salida de nodos de una red.
- Node: encontramos todas las clases usadas para la configuración de nodos, como la dirección IP, el canal de transmisión, los diferentes dispositivos de red o *sockets*.
- Devices: modelan diferentes accesos a un medio de transmisión, como un puente (*bridge*), el modelado del protocolo punto a punto (PPP), el protocolo CSMA o el medio WIFI.
- InternetStack: se centra en la implementación de los protocolos necesarios para Internet, como son TCP, UDP o ARP.
- Helpers: se engloban todos los Helpers creados hasta el momento por el equipo NSNAM. Encontramos Helpers destinados a aplicar diferentes configuraciones a los dispositivos de la red como *InternetStackHelper*; pero también encontramos otros que son útiles para crear un conjunto de objetos con una propiedad en común como *CsmaHelper* (el cual crea dispositivos capacitados con el protocolo CSMA).

- *Applications*: en este módulo existen un conjunto de clases que simulan el comportamiento de la capa de aplicación del modelo TCP/IP, como son los mensajes UDP Echo.
- *Mobility*: se describen una serie de modelados de movilidad, tales como la posición en ejes cartesianos, la velocidad o la dirección que van tomando diferentes nodos.
- *Routing*: corresponde con los protocolos de encaminamiento implementados en este simulador.

### Investigaciones Adicionales NS-3

NS-3 está todavía en fases de desarrollo, por lo tanto se ha tenido que investigar en esta área.

NS-3 no disponía de la aplicación *Constant Bit Rate (CBR)*, y se ha tenido que implementar valiéndonos de la aplicación *OnOff* ya creada.

Se ha implementado *AntHocNet* y *AntOR* desde cero y adaptarlos para NS-3, porque *AntHocNet* estaba implementado para *Qualnet*, que tiene sus características totalmente diferentes a NS-3.

Se han hecho muchas pruebas con diferentes escenarios, utilizando varios modelos de posición y movilidad. Se ha comprobado que el modelo de posición "*ns3::ListPositionAllocator*" combinado con el modelo de movilidad "*ns3::RandomWaypointMobilityModel*" es muy útil en la realización de la pruebas y para la obtención de buenos resultados.

## El encaminamiento en NS-3.

Se utilizan dos métodos principales:

- *RouteOutput*: Petición de una ruta que ya existe para enviar un paquete. Este método es usado por los protocolos de la capa de transporte. Además no causa el reenvío de ningún paquete y es síncrono. Puede ser usado para *multicast* y *unicast*. Es equivalente a *ip\_route\_output()* de Linux.
- *RouteInput*: Encamina un paquete que se recibe (puede ser reenviado o entregado localmente). Los paquetes están controlados por el protocolo de encaminamiento. Existen 4 casos para el procesamiento de los paquetes: Reenvío *unicast*, *multicast*, entrega local y error de reenvío. El equivalente en Linux es *ip\_route\_input()*.

## Tipos de Escenario

Los escenarios están compuestos por nodos a los que se les aplican dos modelos:

- *Modelo de posición*: Posiciona los nodos en el área de trabajo.
- *Modelo de movilidad*: Proporciona un patrón de movilidad, desde el más simple (estático) hasta otros más sofisticados.

## Modelos de Posición

Los modelos de posición que incluye NS-3 son los siguientes:

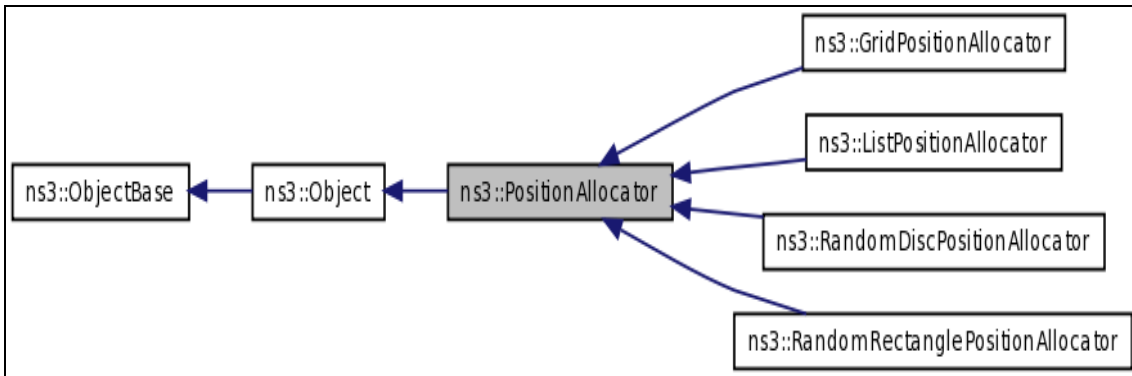


Fig A.2. Modelos de posición en NS-3

- *Grid Position*: Asigna posiciones según una red 2d rectangular.
- *List Position*: Asigna posiciones de una lista especificada por el usuario.
- *Random Disc Position*: Asigna posiciones aleatorias dentro de un área en forma de disco de acuerdo a un par de variables aleatorias.
- *Random Rectangle Position*: Asigna posiciones aleatorias dentro de un rectángulo de acuerdo a un par de variables aleatorias.

## Modelos de Movilidad

Los modelos de movilidad son estos:

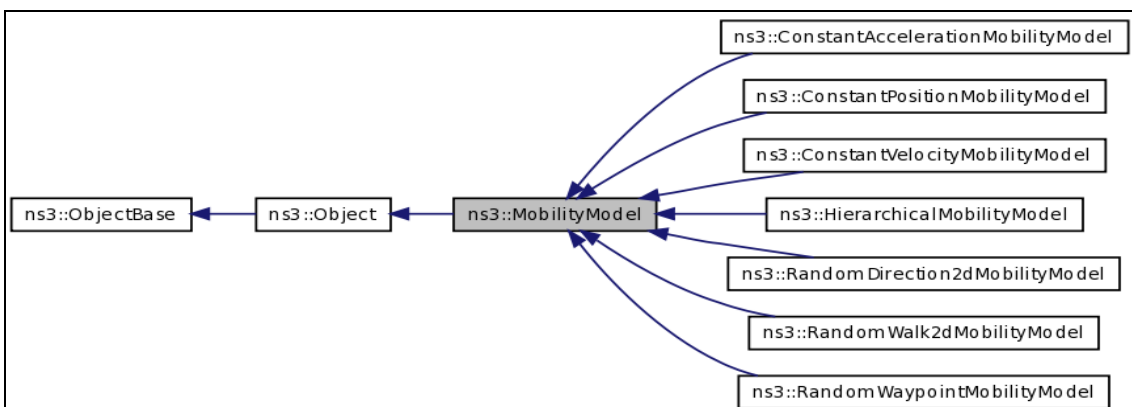


Fig. C.3. Modelos de movilidad en NS-3

- Constant Acceleration Mobility: Posee una posición constante. La posición de un modelo no cambia mientras que no se le aplique un nuevo valor.
- Constant Position Mobility: Tiene una posición constante. La posición de un modelo no varía hasta que no se le cambie explícitamente el valor.
- Constant Velocity Mobility: Tiene una velocidad constante. La velocidad de un modelo no cambia hasta que no se le aplique un nuevo valor.
- Hierarchical Mobility: Un modelo jerárquico permite especificar posiciones a los objetos “hijos” en relación a los objetos “padres”. Básicamente este modelo combina dos tipos de modelos de movilidad: Un modelo “padre” y un modelo “hijo”. La posición del modelo jerárquico es siempre el vector suma de las posiciones del padre y del hijo, de tal forma que, si el modelo padre se “mueve”, informa del movimiento relativo. Este modelo es útil, si se quiere simular un nodo dentro de otro nodo que se mueve, como en un vehículo.
- Random Direction 2d Mobility: El movimiento de los objetos está basado en direcciones aleatorias: Cada objeto se para cada tiempo específico, elige una dirección y velocidad aleatoria y viaja en la dirección especificada, hasta que alcanza una de las fronteras del modelo. Cuando alcanza la frontera, se para, y selecciona una nueva dirección y velocidad. Así sucesivamente hasta que finaliza la simulación.

- Random Walk 2d Mobility: En cada instante los nodos se mueven con una velocidad y dirección elegidos al azar con variables aleatorias proporcionadas por el usuario hasta que o bien se ha recorrido una distancia fija o a transcurrido un tiempo. Si los nodos chocan con la frontera del modelo (especificada por un rectángulo), ellos rebotan con un ángulo y velocidad reflexiva.
- Random Waypoint Mobility: En este modelo cada objeto elige un destino aleatorio, con una velocidad aleatoria, y tiempo de pausa aleatoria: Los objetos paran según el tiempo de pausa especificado, y empiezan a moverse hacia el destino con la velocidad especificada. Una vez que el destino es alcanzado, el proceso comienza otra vez. La implementación de este modelo no sirve para un modelo 2d. Trabaja, si se proporciona un modelo de posiciones aleatorias 3d.



## ANEXO B: CARACTERÍSTICAS DEL PROTOCOLO

---

En esta sección se expone el protocolo de la propuesta: AntOR (*Ant Optimized Routing*). Tiene dos variantes: Ruta de enlace disjunto (*Disjoint Link Route, DLR*) y Ruta de nodo disjunto (*Disjoint Node Route, DNR*)

### Estructura de datos

- Tabla de vecinos: Esta tabla almacena las direcciones y los últimos tiempos en que el nodo ha escuchado a sus vecinos.

	Nodo A	Dirección	Tiempo (seg.)
Vecinos	B	10.0.0.2	0,25
	C	10.0.0.3	0,31

Tabla B.1 Vecinos

- Tabla de encaminamiento: Contiene la información necesaria para transmitir los paquetes de datos desde un origen a un destino. Esta tabla contiene los valores de encaminamiento. Los valores de encaminamiento son: feromona regular, virtual y número medio de saltos.

Nodo A	Destino	Siguiente Salto	Regular ( $\tau$ )	Virtual ( $k$ )	Numero Medio de saltos ( $h$ )
Registros	10.0.0.2	10.0.0.2	0,30	0	0,30
	10.0.0.3	10.0.0.3	0,30	0	0,30

Tabla B.2 Encaminamiento

## Estructura de los mensajes de control

Los mensajes de control han tenido que ser adaptados a NS-3.

Un paquete es una estructura (contenedor) donde se almacenan varios mensajes. El formato general de los paquetes en AntOR se muestra a continuación:

La primera fila (4 bytes) representa la cabecera del paquete donde *Packet Length* es la longitud total del paquete en bytes y *Packet Sequence Number* es un identificador único del paquete. Los posibles número de identificadores con 16 bits permite 65536 valores ( $2^{16}$ ), cuando se alcanza el límite se empieza otra vez desde el comienzo. Este número identificador puede ser utilizado para comprobar si es la correcta y está en orden la entrega de los paquetes al destino.

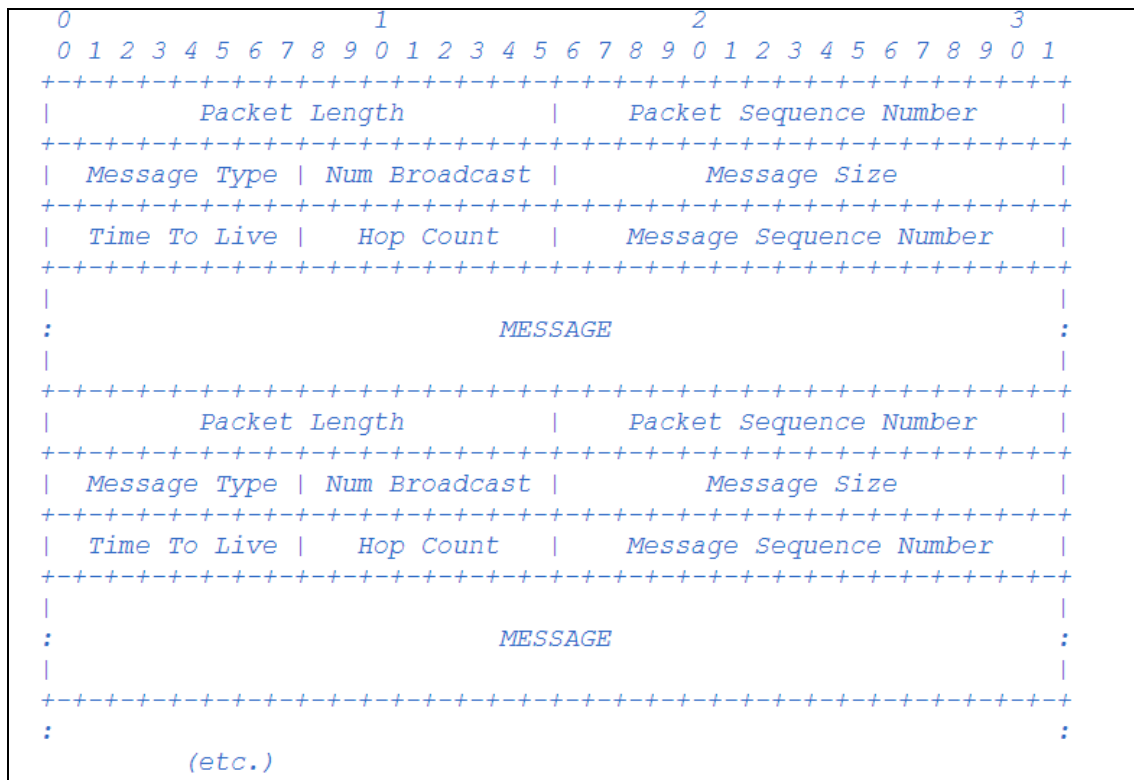


Fig. B.1 Formato general de los paquetes AntOR



A continuación se muestran los formatos de los diferentes mensajes de control.

## Mensaje HELLO

Este mensaje se utiliza para conocer cuáles son los vecinos de un nodo y en el proceso de difusión de la feromona.

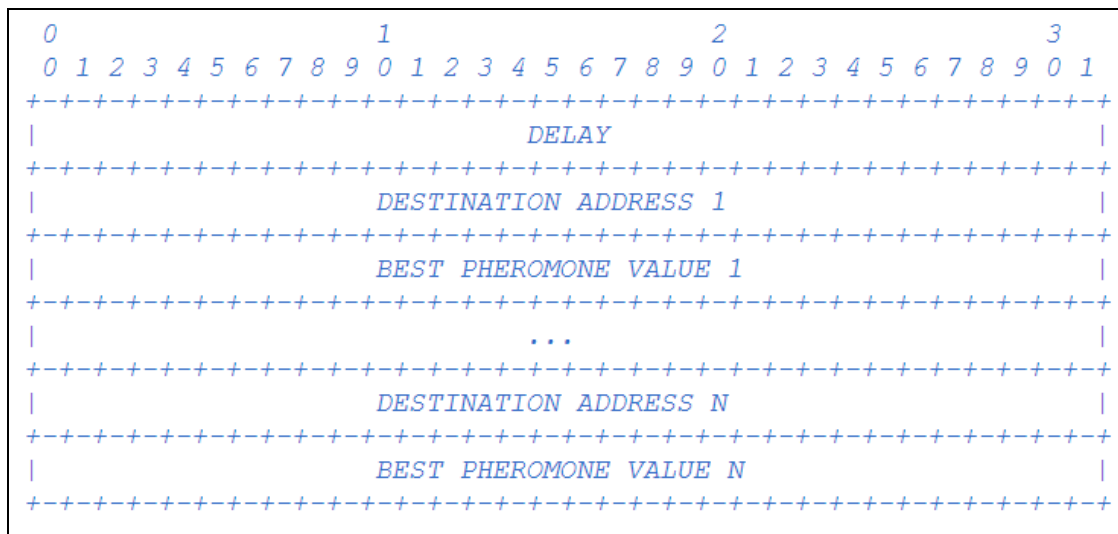


Fig. B.3 Formato del mensaje HELLO

El mensaje HELLO contiene los siguientes campos:

- Delay: Indica el tiempo en segundo de envío del Hello.
- Destination Address: Son los destinos de los que el nodo tiene información de encaminamiento.
- Best Pheromone Value: Indica el mejor valor (regular o virtual) que tiene una ruta con varias alternativas.

## Mensaje RFA

Se emplea en el proceso de descubrimiento de ruta.

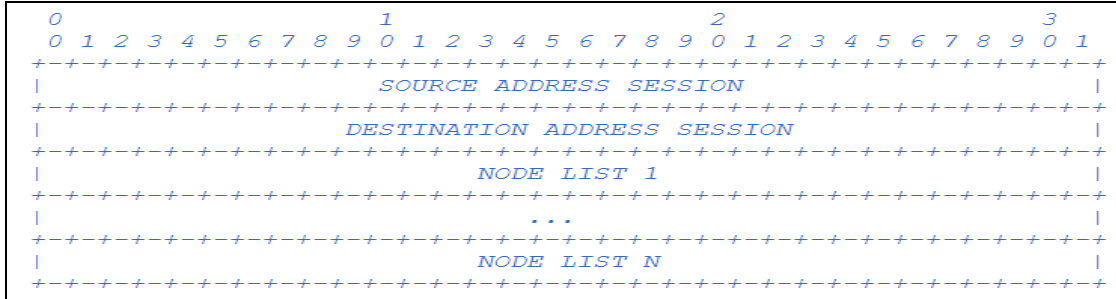


Fig. B.4 Formato del mensaje Reactive Forward Ant (RFA)

El mensaje RFA contiene los siguientes campos:

- *Source Address Session*: Representa el nodo que envía este mensaje.
- *Destination Address Session*: Es la dirección del nodo que recibe este mensaje.
- *Node List*: Incluye los nodos que han sido visitados por el mensaje de control. Sirve para evitar bucles.

## Mensaje RBA

Se emplea el *marcado* del camino de retorno en el proceso de descubrimiento de ruta.

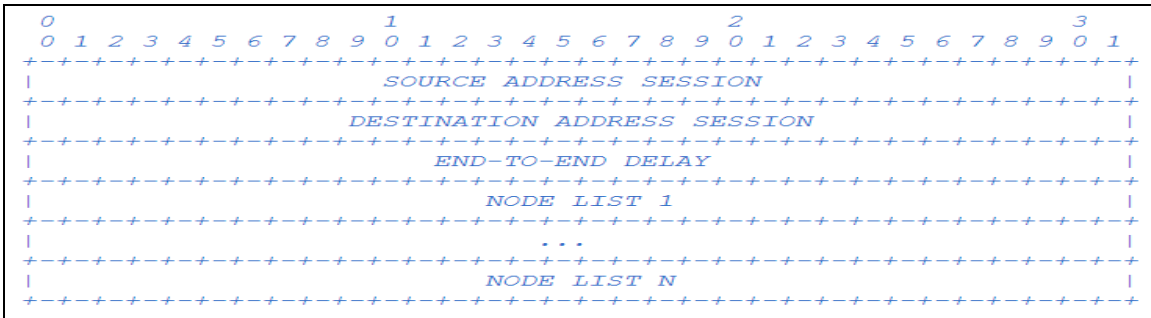


Fig. B.5 Formato del mensaje Reactive Backward Ant (RBA)

El mensaje RBA contiene los siguientes campos:

- Source Address Session: Representa la dirección el nodo que envía este mensaje.
- Destination Address Session: Es la dirección del nodo que recibe este mensaje.
- End-To-End Delay: Tiempo acumulado por el mensaje en el camino de retorno.
- Node List: Lista de nodos que debe recorrer este mensaje para llegar a su destino.

### Mensaje PFA

Se usa en el proceso de exploración de nuevas rutas. Los campos de este mensaje con los mismos que los de RFA, por lo que se omiten.





## Mensaje RRFA

Este mensaje es utilizado para la reparación local de ruta ante el fallo de enlace. El formato de este mensaje es el mismo que RFA.

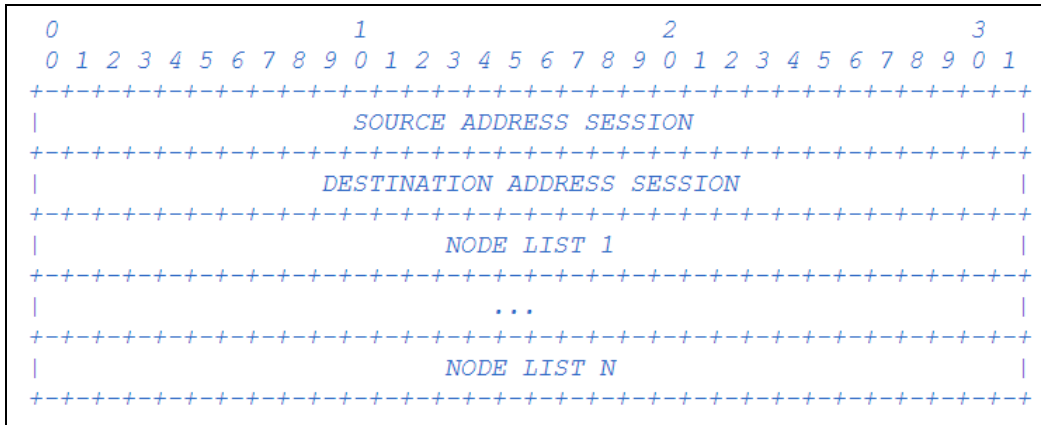


Fig. B.9 Formato del mensaje Route Repair Forward Ant (RRFA)

## Mensaje RRBA

Este mensaje se utiliza para *marcado* de las rutas localmente reparadas. El formato de este mensaje es el mismo que RBA

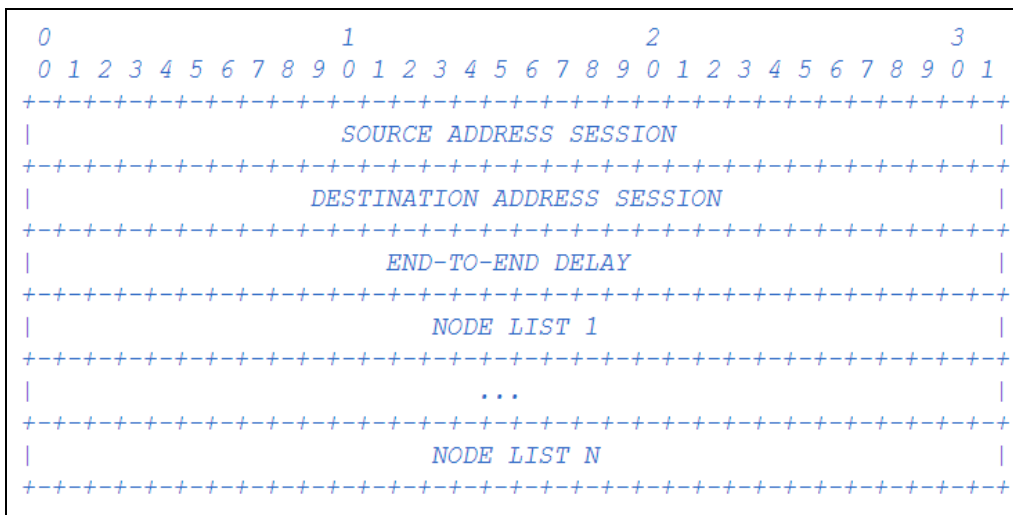


Fig. B.10 Formato del mensaje Route Repair Backward Ant (RRBA)

## Métricas

AntOR utiliza las siguientes métricas para su funcionamiento.

- Retardo extremo-a-extremo: Retardo entre el nodo origen y destino de la sesión de datos. Esta métrica servirá para calcular el nuevo valor de feromona regular de la ruta desde el origen al destino. Es una métrica adaptativa pero no estable, por las condiciones del tráfico de la red.
- Número de saltos (distancia). Hace referencia al número de nodos intermedios que hay desde un origen a un destino. En cambio, esta métrica es estática y estable.

## Funcionamiento de AntOR

### Difusión de encaminamiento

Este proceso se realiza cada cierto intervalo de tiempo (un segundo), que consiste en difundir el valor de feromona virtual en otros nodos. Para ello los nodos envían mensajes HELLO en modo *broadcast* a sus vecinos. En estos mensajes se indican los destinos con sus mejores valores de feromona regular o virtual de las rutas.

Un nodo que recibe el mensaje HELLO, lo primero que hace es actualizar su tabla de vecinos indicando que el emisor del mensaje es su vecino. Después crea una entrada en la tabla de encaminamiento indicando la ruta al vecino. En esta entrada se añade un valor de feromona regular y el número medio de saltos.

Por último actualiza su feromona virtual hacia los destinos que contiene el mensaje HELLO, siendo el emisor el encaminador hacia esos destinos.

Sólo se actualiza el valor de feromona virtual si no tiene un valor de feromona regular establecido anteriormente.

### Exploración de nuevas rutas

Este proceso se realiza pro-activamente cada cierto intervalo de tiempo (2 segundos) por los nodos que son origen de la sesión. Se crean mensajes PFA que son enviados en modo *unicast* hacia el destino de la sesión. Estos mensajes nunca son enviados en modo *broadcast*.

Estos mensajes sólo pueden visitar un número limitado de nodos, que lo establece la distancia mínima de las rutas determinadas anteriormente. Esta distancia permitida queda indicada en el campo TTL de la cabecera del mensaje PFA. Los nodos intermedios que reciben un PFA con un valor TTL igual a 0 descartan el mensaje.

Otra característica es que se limitan los nodos intermedios a los que puede ir el mensaje, dependiendo de si se trata de un protocolo de ruta de enlace disjunto o nodo disjunto.

Cuando un mensaje llega al destino, se procesa y se envía un mensaje PBA de retorno al origen de la sesión. En este proceso se actualiza la tabla de encaminamiento de forma similar al que se utiliza en el descubrimiento de ruta.

## **Implementación**

La codificación del algoritmo ha supuesto un arduo trabajo de investigación, porque había que adaptarlo a las librerías del Simulador de red NS-3. Había que entender cómo estaba estructurado el formato de los mensajes y la implementación de la tabla de encaminamiento en NS-3.

## Características

Se han tenido en cuenta diferentes aspectos a la hora de implementar AntOR:

- Para que no haya sincronismo en el envío de paquetes entre nodos, se utiliza un valor *jitter*, siendo la cuarta parte del intervalo de emisión de los mensajes HELLO (1 sg.)
- Evita los bucles en el proceso de descubrimiento de ruta a la hora de visitar los nodos intermedios, para que no haya la repetición, por medio de una lista de nodos visitados.
- Se ha utilizado una estructura para clasificar y agrupar estos mensajes según el destino.
- El número máximo de mensajes por paquete son 64 para no sobrecargar la red.
- Incluye información sobre el origen y destino de la sesión de datos, para saber en todo momento los nodos intermedios a la hora de reparar la ruta.

## Desarrollo de los experimentos

Los experimentos que se han realizado han servido para comparar AntOR con OLSR y AntHocNet. La versión de AntOR que se ha utilizado para comparar es la de ruta de enlace disjunto, porque se ha visto que es más representativa. Se ha tenido en cuenta que en todas las pruebas o experimentos se utiliza siempre el mismo escenario.

En la siguiente tabla se exponen las características generales de las simulaciones.

<b>Número de Nodos:</b> Entre 20 y 100
<b>Dimensiones del área:</b> 1400x1400 $m^2$
<b>Rango de transmisión (área abierta):</b> 300 $m$
<b>Capa Física:</b> Configurada para IEEE 802.11b
<b>Tasa de envío de datos:</b> Constante Wifi-6 $mps$
<b>Tiempo de simulación:</b> 30 seg.
<b>Número de pruebas o experimentos:</b> 3

Tabla B.3 Características generales de las simulaciones

Los parámetros de configuración interna del protocolo AntOR se muestran a continuación.

$\gamma = 0,7$
$\alpha = 0,7$
$\beta_1 = 20$
$\beta_2 = 2$
$\beta_3 = 20$
<b>Número Máximo de Destinos de mensajes HELLO:</b> 10
<b>Intervalo emisión HELLO:</b> 1 seg.
<b>Intervalo emisión PFA:</b> 2 seg.
<b>Número máximo de intentos de establecimiento de ruta:</b> 5
<b>Intervalo emisión RFA:</b> 5 seg.
<b>Número máximo de broadcast de los mensajes RRFA:</b> 2

Tabla B.4 Parámetros de configuración

El parámetro  $\beta_1$  es utilizado en el proceso reactivo. Se utiliza un valor alto para dar mejor prioridad a la ruta con mejor valor de “feromona regular” y así descubrir la ruta lo más rápido posible.

El parámetro  $\beta_2$  se emplea en el proceso pro-activo de exploración de nuevas rutas. En este caso se ha utilizado un valor muy bajo para que todas las rutas alternativas a un destino se puedan elegir con la misma probabilidad.

Por último,  $\beta_3$  es usado para el encaminamiento de los datos, como queremos que lleguen por la mejor ruta y lo más rápido posible, este parámetro se establece al mismo valor que el de  $\beta_1$ .

Se ha utilizado la aplicación *Constant Bit Rate* (CBR) para generar el tráfico de cada sesión de datos.

A continuación se muestra la tabla con la configuración del escenario.

<b>Tiempo de comienzo CBR cliente:</b> 0sg.
<b>Tiempo de finalización CBR cliente:</b> 30sg.
<b>Tiempo de comienzo CBR servidor:</b> 0sg.
<b>Tiempo de finalización CBR servidor:</b> 30sg.
<b>Nº de sesiones de datos:</b> 4
<b>Modelo de posición:</b> Lista
<b>Modelo de movilidad:</b> <i>Random WayPoint</i> (RWP)

Tabla B.5 Parámetros del escenario

El número de sesiones se ha establecido a 4 sesiones de datos independientes. Por independientes queremos decir que los pares origen-destino son distintos por cada sesión. Un mismo emisor no puede tener varios receptores (el protocolo que proponemos es *unicast*), y un mismo receptor no tiene varios emisores (esta característica es utilizada para no congestionar el receptor al sobrecargarlo con varios paquetes de datos procedentes de varios emisores).

En cada sesión de datos se envían 4 paquetes de datos de 64 bytes cada segundo.

Para el modelo de posición que se ha utilizado hemos utilizado una lista de nodos colocados aleatoriamente (siempre se usa el mismo escenario para las diferentes pruebas o experimentos, y para los diferentes protocolos que se ejecutan).

El modelo de movilidad utilizado es RWP con un tiempo de pausa de 10sg y una velocidad variable entre 0 y 10sg. Se han utilizado estas características para que sea más dinámico el escenario. Las posiciones destinos de este modelo se establecen aleatoriamente e indican a dónde se mueven los nodos.



