



UNIVERSIDAD DE DEUSTO

NUEVOS PROTOCOLOS PARA EL  
DESCUBRIMIENTO Y LA COMPOSICIÓN  
AUTOMÁTICA DE SERVICIOS EN REDES  
MÓVILES AD HOC

Tesis doctoral presentada por Unai Aguilera Irazabal  
dentro del Programa de Doctorado en Ciencia de la Computación  
Dirigida por el Dr. Diego López de Ipiña González de Artaza





UNIVERSIDAD DE DEUSTO

NUEVOS PROTOCOLOS PARA EL  
DESCUBRIMIENTO Y LA COMPOSICIÓN  
AUTOMÁTICA DE SERVICIOS EN REDES  
MÓVILES AD HOC

Tesis doctoral presentada por Unai Aguilera Irazabal  
dentro del Programa de Doctorado en Ciencia de la Computación  
Dirigida por el Dr. Diego López de Ipiña González de Artaza

El doctorando

El director

Bilbao, enero de 2013

*Nuevos protocolos para el descubrimiento y la composición automática de servicios  
en redes móviles ad hoc*

Autor: Unai Aguilera Irazabal

Director: Dr. Diego López de Ipiña González de Artaza

Editado con  $\text{\LaTeX}$ 2e utilizando los tipos de letra *TeX Gyre Heros* para títulos y  
*Charter BT* para el contenido general.

Impreso en Bilbao

Primera edición, febrero 2013

---

*A mis padres Antonio y Mercedes  
y a mi hermano Borja.*



## Resumen

Las redes móviles ad hoc (MANET) son un tipo de red inalámbrica caracterizada por la movilidad de los dispositivos participantes y la comunicación salto a salto entre ellos. La creación de estas redes ocurre de una manera no planificada y respondiendo a la distribución de los dispositivos en un momento determinado. La topología de una red MANET puede cambiar rápidamente y se requieren mecanismos que posibiliten a los diferentes nodos comunicarse entre sí. Los dominios de aplicación de estas redes van desde las situaciones de emergencia hasta las operaciones militares, incluyendo aquellos otros dominios con necesidad de comunicación sin una infraestructura predeterminada.

En el enfoque orientado a servicios y aplicado a entornos distribuidos, los usuarios deben ser capaces de acceder no solamente a los servicios proporcionados de manera directa por los dispositivos, sino también a aquellos resultantes de la conexión y agregación de otros servicios más sencillos. La composición de servicios es un paradigma que puede proporcionar al usuario, y a sus aplicaciones, grandes beneficios en los entornos constituidos por dispositivos que forman redes MANET, ya que, debido a su movilidad, pueden cambiar tanto la topología como los recursos y servicios.

En esta tesis se propone un nuevo protocolo para la composición automática de servicios en redes MANET que se basa en la construcción y mantenimiento de un *grafo distribuido de servicios*. Dicho grafo permite determinar durante todo el tiempo de vida de la red, y en respuesta a los cambios de la misma, la dependencia entre los parámetros de entrada y salida de los diferentes servicios ofertados por los dispositivos. Posteriormente, el grafo de servicios es utilizado durante el proceso de construcción de composiciones para reducir el tiempo de búsqueda de las mismas.

Por otro lado, se propone también un nuevo mecanismo para el descubrimiento de servicios en redes móviles ad hoc que se fundamenta en la diseminación de los parámetros de entrada y salida de los servicios existentes en la red. Además, usando una ontología de

conceptos permite, mediante la aplicación de un proceso de *agrupamiento de mensajes* y *poda* de las búsquedas, reducir el número de mensajes transmitidos. El descubrimiento es parte fundamental del proceso de composición de servicios indicado anteriormente, definiendo un arquitectura que integra ambos protocolos.

Por último, tanto la propuesta de protocolo para el descubrimiento, como para la composición de servicios en redes MANET, han sido evaluadas de manera exhaustiva mediante la utilización de un simulador de red con objeto de conocer sus ventajas y limitaciones.



## Agradecimientos

La finalización de esta tesis doctoral ha sido solamente posible gracias a toda la experiencia en investigación, desarrollo de software y en el manejo de tecnologías, adquirida a lo largo de los años en diferentes grupos de la Facultad de Ingeniería (antiguo ESIDE) de la Universidad de Deusto.

Primero, gracias por la oportunidad que se me dio de participar en el grupo de multimedia MIRA, donde comencé los estudios de doctorado que han terminado en la realización de esta tesis doctoral.

Gracias también a todos los integrantes del antiguo grupo de Ingeniería Lingüística de la Universidad de Deusto (DELi), donde empecé a conocer los entresijos de la investigación científica.

Por supuesto, no puedo olvidarme de todas aquellas personas que, en el momento de finalización de esta tesis, son parte del laboratorio Smartlab, en el sentido más amplio del término, incluyendo todos los nombres, grupos y subgrupos que la unidad tenga en el presente, pasado, futuro o en cualquier otra línea temporal. Tampoco puedo dejar de recordar a todas aquellas personas con las que he coincidido en este grupo en todos los años en los que he trabajado en él. Son muchos años luchando juntos en la realización de proyectos de investigación y en la finalización *in extremis* de propuestas.

Por último, agradecer a mis padres y a mi hermano el apoyo recibido durante los momentos más complicados de la investigación. No hubiera podido conseguir esto sin vosotros.

*Eskerrik asko,*

Unai Aguilera

Bilbao, diciembre 2012



# Tabla de contenido

<b>Lista de figuras</b>	<b>xi</b>
<b>Lista de tablas</b>	<b>xv</b>
<b>Acrónimos</b>	<b>xvii</b>
<b>1 Introducción</b>	<b>1</b>
1.1 Motivación . . . . .	4
1.2 Hipótesis y objetivos . . . . .	5
1.3 Metodología de investigación . . . . .	7
1.4 Organización de la tesis . . . . .	9
<b>2 Breve introducción a las redes móviles ad hoc</b>	<b>11</b>
2.1 Redes móviles ad hoc . . . . .	11
2.1.1 Conceptos fundamentales . . . . .	13
2.1.2 Comunicación física y acceso al medio . . . . .	15
2.1.3 Protocolos de encaminamiento . . . . .	16
2.1.4 Aplicaciones . . . . .	17
2.2 Simulación y experimentación . . . . .	18
2.2.1 Simuladores de red . . . . .	19
2.2.2 Parámetros de simulación . . . . .	21
2.3 Conclusiones . . . . .	24
<b>3 Estado actual del descubrimiento y composición de servicios</b>	<b>25</b>
3.1 Descubrimiento de servicios . . . . .	25
3.1.1 Redes con infraestructura . . . . .	26
3.1.2 Clasificación de las soluciones . . . . .	28
3.1.3 Uso de directorios . . . . .	30
3.1.4 Difusión en la red . . . . .	32
3.1.5 Descripción de servicios . . . . .	33
3.1.6 Modo de operación . . . . .	34
3.1.7 Respuesta a la movilidad . . . . .	36

---

3.1.8	Integración de protocolos . . . . .	36
3.1.9	Mecanismo de selección . . . . .	37
3.1.10	Conclusiones . . . . .	37
3.2	Composición de servicios . . . . .	39
3.2.1	Clasificación de las soluciones . . . . .	39
3.2.2	Arquitectura . . . . .	41
3.2.3	Mecanismo de composición . . . . .	43
3.2.4	Especificación de la composición . . . . .	45
3.2.5	Descripción de servicios . . . . .	47
3.2.6	Participación del usuario . . . . .	48
3.2.7	Modo de operación . . . . .	48
3.2.8	Ejecución . . . . .	49
3.2.9	Respuesta a la movilidad . . . . .	50
3.2.10	Conclusiones . . . . .	51
<b>4</b>	<b>Arquitectura para la composición distribuida de servicios</b>	<b>53</b>
4.1	Definición del problema . . . . .	54
4.1.1	Caso de uso . . . . .	57
4.1.2	Fases de la composición . . . . .	58
4.1.3	Definición formal del problema . . . . .	59
4.2	Requisitos y limitaciones . . . . .	61
4.3	Arquitectura . . . . .	63
4.3.1	Detección de vecinos . . . . .	65
4.3.2	Diseminación . . . . .	65
4.3.3	Descubrimiento de servicios y gestión de rutas . . . . .	66
4.3.4	Emparejamiento distribuido de servicios . . . . .	67
4.3.5	Búsqueda de composiciones . . . . .	67
4.3.6	Validez de las soluciones . . . . .	68
4.4	Descripción de servicios . . . . .	69
4.4.1	Aspectos de un servicio . . . . .	70
4.4.2	Ontología de conceptos . . . . .	70
4.5	Gestión de red . . . . .	72
4.5.1	Detección de vecinos . . . . .	72
4.5.2	Difusión confiable . . . . .	75
4.5.3	Colas de mensajes . . . . .	78
4.6	Conclusiones . . . . .	79
<b>5</b>	<b>Descubrimiento de servicios basado en parámetros</b>	<b>81</b>
5.1	Visión general . . . . .	82
5.2	Diseminación . . . . .	85
5.2.1	Distancia estimada . . . . .	86
5.2.2	Tabla de parámetros . . . . .	88

---

5.2.3	Mensajes de actualización . . . . .	90
5.2.4	Actualización de la tabla de parámetros . . . . .	92
5.2.5	Uso de la ontología de conceptos . . . . .	96
5.2.6	Casos especiales de diseminación . . . . .	98
5.3	Búsqueda de servicios . . . . .	101
5.3.1	Mensajes de búsqueda . . . . .	101
5.3.2	Propagación de búsquedas . . . . .	102
5.3.3	Mantenimiento de búsquedas activas . . . . .	105
5.3.4	Cancelación de búsquedas . . . . .	107
5.4	Encaminamiento de mensajes . . . . .	109
5.4.1	Tabla de encaminamiento . . . . .	110
5.4.2	Mensajes de comunicación . . . . .	111
5.4.3	Gestión de las rutas de comunicación . . . . .	114
5.5	Evaluación . . . . .	115
5.5.1	Reducción de mensajes de diseminación . . . . .	117
5.5.2	Poda de mensajes de búsqueda . . . . .	119
5.5.3	Otros experimentos . . . . .	122
5.6	Conclusiones . . . . .	127
<b>6</b>	<b>Composición basada en un grafo distribuido de servicios</b>	<b>129</b>
6.1	Visión general . . . . .	130
6.2	Grafo distribuido de servicios . . . . .	132
6.3	Colisiones de parámetros . . . . .	134
6.3.1	Detección de colisiones . . . . .	135
6.3.2	Inhibición de colisiones . . . . .	139
6.3.3	Conexión de servicios compatibles . . . . .	140
6.3.4	Encaminamiento de mensajes en el grafo . . . . .	143
6.3.5	Desconexión de servicios . . . . .	145
6.4	Búsqueda de composiciones . . . . .	147
6.4.1	Servicios de especificación . . . . .	147
6.4.2	Registro de servicios de especificación . . . . .	149
6.4.3	Búsqueda en el grafo distribuido de servicios . . . . .	150
6.4.4	Expiración de búsquedas . . . . .	153
6.4.5	Ejecución de composiciones . . . . .	154
6.5	Evaluación . . . . .	155
6.5.1	Tiempo de pausa . . . . .	157
6.5.2	Número de nodos . . . . .	159
6.5.3	Frecuencia de búsqueda . . . . .	161
6.6	Conclusiones . . . . .	162

<b>7 Conclusiones</b>	<b>165</b>
7.1 Contribuciones . . . . .	165
7.2 Líneas abiertas de investigación . . . . .	169
7.2.1 Descubrimiento de servicios . . . . .	169
7.2.2 Composición de servicios . . . . .	170
<b>A Validación</b>	<b>173</b>
A.1 Metodología de validación . . . . .	173
A.1.1 Pruebas unitarias . . . . .	174
A.1.2 Pruebas de comportamiento . . . . .	175
A.2 Entorno de simulación y experimentación . . . . .	177
<b>B Lista de publicaciones</b>	<b>181</b>
<b>Bibliografía</b>	<b>185</b>

# Lista de figuras

1.1	Metodología de investigación . . . . .	8
2.1	Representación de una red móvil ad hoc . . . . .	14
4.1	Servicios atómicos que forman parte de un flujo de trabajo en un servicio compuesto . . . . .	56
4.2	Red ad hoc con diferentes tipos de dispositivos que ofertan servicios	57
4.3	Arquitectura propuesta para la composición automática de servicios en redes MANET . . . . .	63
4.4	Representación de las relaciones jerárquicas entre los conceptos de una ontología . . . . .	71
4.5	Contenido de las tablas de vecinos para una configuración determinada de la red . . . . .	74
4.6	Colas para el procesamiento desacoplado de los mensajes recibidos y de las respuestas enviadas . . . . .	78
5.1	Arquitectura para el descubrimiento de servicios basado en parámetros . . . . .	84
5.2	Diseminación en la red ad hoc de la información sobre los servicios provistos por un nodo . . . . .	86
5.3	Propagación del valor de <i>distancia estimada</i> sobre un tipo de parámetro de un servicio . . . . .	87
5.4	Tabla de parámetros de un nodo . . . . .	89
5.5	Envío de mensajes de actualización para la diseminación de parámetros . . . . .	91
5.6	Algoritmo para la diseminación de la información sobre parámetros	94
5.7	Estado de las tablas tras la diseminación de la información sobre los parámetros de los servicios . . . . .	96
5.8	Diseminación de parámetros utilizando las relaciones contenidas en una ontología de conceptos . . . . .	97
5.9	Diseminación de la información de parámetros en una configuración en bucle de los nodos de la red ad hoc . . . . .	98

5.10	Diseminación de parámetros en una red con información previa existente . . . . .	99
5.11	Rotura en la red ad hoc produciendo la actualización de la información anteriormente diseminada . . . . .	100
5.12	Propagación de los mensajes de búsqueda y posterior aplicación del proceso de <i>poda</i> usando la información diseminada . . . . .	105
5.13	Propagación de una búsqueda activa tras la aparición de un nuevo vecino . . . . .	106
5.14	Cancelación parcial de una búsqueda activa propagada . . . . .	108
5.15	Estado de las tablas de encaminamiento tras la propagación de los mensajes de búsqueda y respuesta . . . . .	111
5.16	Algoritmo para el procesado de los mensajes <i>multicast</i> . . . . .	113
5.17	Eliminación de rutas debido a la desaparición del vecino de un nodo . . . . .	115
5.18	Efecto en el número de mensajes de actualización y en el tiempo medio para el descubrimiento de la primera ocurrencia de un servicio compatible aplicando una ontología de conceptos . . . . .	119
5.19	Efecto de la poda de mensajes de búsqueda en el porcentaje medio de servicios descubiertos y en el tiempo medio de descubrimiento de servicios . . . . .	120
5.20	Efecto de la poda de mensajes de búsqueda en el número de mensajes enviados y en la sobrecarga introducida por el protocolo de descubrimiento . . . . .	121
5.21	Efecto de la variación del número de nodos en la cantidad de mensajes de actualización y en el porcentaje de servicios encontrados . . . . .	123
5.22	Efecto de la variación del número de nodos en el tiempo de descubrimiento y en la sobrecarga introducida por el protocolo de descubrimiento . . . . .	123
5.23	Efecto de la variación de la distancia de diseminación y búsqueda en el número de mensajes enviados y en el porcentaje de servicios encontrados . . . . .	124
5.24	Efecto de la variación de la distancia de diseminación y búsqueda en el tiempo de descubrimiento y en la sobrecarga introducida por el protocolo de descubrimiento . . . . .	125
5.25	Efecto de la variación de la frecuencia de búsqueda en el número de mensajes enviados y en el porcentaje de servicios encontrados . . . . .	126
5.26	Efecto de la variación de la frecuencia de búsqueda en el tiempo de descubrimiento y en la sobrecarga introducida por el protocolo . . . . .	126
6.1	Arquitectura interna para la composición de servicios basada en un grafo distribuido de servicios . . . . .	131



6.2	Grafo distribuido de servicios conectando servicios a través de la red . . . . .	133
6.3	Detección de una colisión entre parámetros de entrada y salida compatibles por parte de un nodo intermedio . . . . .	136
6.4	Algoritmo para la detección de colisiones tras la recepción de un mensaje de actualización . . . . .	137
6.5	Algoritmo para el cálculo de colisiones entre parámetros de entrada y salida . . . . .	138
6.6	Inhibición de colisiones duplicadas en los nodos intermedios . . .	139
6.7	Búsqueda de los servicios compatibles como resultado de la detección de una colisión y posterior recepción de las respuestas . .	141
6.8	Gestión de conexiones entre servicios compatibles tras la detección de una colisión entre parámetros . . . . .	142
6.9	Creación del grafo distribuido de servicios (GDS) . . . . .	143
6.10	Notificación de desconexión del grafo enviada por el nodo que ha detectado la colisión . . . . .	146
6.11	Eliminación de un servicio en un nodo y notificación a los servicios conectados . . . . .	147
6.12	Servicios de especificación creados a partir de la definición del servicio compuesto . . . . .	149
6.13	Conexión de los servicios de especificación de la composición en el grafo distribuido de servicios (GDS) . . . . .	150
6.14	Entradas de un servicio cubiertas por los parámetros de salida de sus antecesores . . . . .	151
6.15	Efecto del intervalo de pausa en el tiempo medio de búsqueda de una composición y en el número medio de saltos efectuados para longitudes de composición de 3, 5 y 7 servicios . . . . .	157
6.16	Efecto del intervalo de pausa en la sobrecarga debida al protocolo y en el porcentaje de composiciones encontradas para longitudes de composición de 3, 5 y 7 servicios . . . . .	158
6.17	Efecto de la variación del número de nodos en el tiempo medio de búsqueda de una composición y en el número medio de saltos efectuados para longitudes de composición de 3, 5 y 7 servicios .	160
6.18	Efecto de la variación de la cantidad de nodos en la sobrecarga debida al protocolo y en el porcentaje de composiciones encontradas para longitudes de composición de 3, 5 y 7 servicios . . . .	161
6.19	Efecto de la frecuencia de búsqueda en el tiempo medio de búsqueda de una composición y en el número medio de saltos efectuados para longitudes de composición de 3, 5 y 7 servicios . . . .	161
6.20	Efecto de la frecuencia de búsqueda en la sobrecarga debida al protocolo y en el porcentaje de composiciones encontradas para longitudes de composición de 3, 5 y 7 servicios . . . . .	162



# Lista de tablas

2.1	Características de los simuladores de red . . . . .	20
2.2	Configuración del simulador ns-2 para los experimentos . . . . .	22
5.1	Configuración de la evaluación del protocolo de descubrimiento .	116
6.1	Configuración de la evaluación del protocolo de composición . . .	156
A.1	Pruebas de comportamiento. Parte I . . . . .	178
A.2	Pruebas de comportamiento. Parte II . . . . .	179



# Acrónimos

<b>ACK</b>	Acknowledgement
<b>AODV</b>	Ad hoc On-Demand Distance Vector
<b>BPEL</b>	Business Process Execution Language
<b>CPU</b>	Central Processing Unit
<b>CSP</b>	Constraint Satisfaction Problem
<b>DAML</b>	DARPA Agent Markup Language
<b>DHCP</b>	Dynamic Host Configuration Protocol
<b>DHT</b>	Distributed Hash Table
<b>DL</b>	Description Logic
<b>DSDV</b>	Destination-Sequenced Distance Vector
<b>DSR</b>	Dynamic Source Routing
<b>GDS</b>	Grafo Distribuido de Servicios
<b>GNU</b>	GNU's Not Unix
<b>GOLOG</b>	Algol in Logic
<b>GPL</b>	GNU General Public License
<b>GPS</b>	Global Positioning System
<b>HTN</b>	Hierarchical Task Network
<b>HTTP</b>	Hypertext Transfer Protocol
<b>IA</b>	Inteligencia Artificial

<b>IEEE</b>	Institute of Electrical and Electronics Engineers
<b>IETF</b>	Internet Engineering Task Force
<b>IOPE</b>	Inputs, Outputs, Preconditions and Effects
<b>LDAP</b>	Lightweight Directory Access Protocol
<b>MAC</b>	Medium Access Control
<b>MANET</b>	Mobile Ad hoc Network
<b>OLSR</b>	Optimized Link State Routing
<b>OWL</b>	Web Ontology Language
<b>OWL-S</b>	Web Ontology Language for Services
<b>PAN</b>	Personal Area Network
<b>RDF</b>	Resource Description Framework
<b>REST</b>	Representational State Transfer
<b>RMI</b>	Remote Method Invocation
<b>SDP</b>	Service Discovery Protocol
<b>SHOP</b>	Simple Hierarchical Ordered Planner
<b>SLP</b>	Service Location Protocol
<b>STRIPS</b>	Stanford Research Institute Problem Solver
<b>TCP</b>	Transfer Control Protocol
<b>TTL</b>	Time To Live
<b>UDDI</b>	Universal Description, Discovery and Integration
<b>UDP</b>	User Datagram Protocol
<b>UPnP</b>	Universal Plug and Play
<b>URL</b>	Uniform Resource Locator
<b>UUID</b>	Universally Unique Identifier
<b>WPAN</b>	Wireless Personal Area Network

**WSDL** Web Service Definition Language

**WSMO** Web Service Modeling Ontology

**XML** Extensible Markup Language

**ZRP** Zone Routing Protocol





# Introducción

La aparición de las redes de computadores a finales de la década de los 60 supuso no solamente una revolución en el campo de las tecnologías de la comunicación, sino también en la concepción que hasta dicho momento se tenía de la propia ciencia de la computación. La conexión de diferentes computadores entre sí, que hasta ese momento se encontraban totalmente aislados en su funcionamiento, posibilitó, en un principio, compartir los recursos de computación e información dentro de las propias universidades y centros de investigación y, posteriormente, con cualquier otra entidad conectada. Conseguir que distintos computadores o, de forma más general, dispositivos que tengan capacidad de computación puedan comunicarse entre sí, supuso un gran esfuerzo de investigación para la resolución de los distintos problemas asociados: capacidad de la comunicación, fiabilidad y control de errores, gestión de los medios físicos a través de los que realizar la comunicación, etc.

Además, el auge de las comunicaciones inalámbricas proporcionó a los usuarios una mayor movilidad y libertad a la hora de acceder a este tipo de redes. Hoy en día, tanto el uso de las redes de computación como su acceso a través de dispositivos móviles inalámbricos se han convertido en aspectos fundamentales de la actividad diaria para millones de personas, todo ello mediante la utilización de servicios tales como el correo electrónico, el acceso a páginas web o las redes sociales.

Por otro lado, la propuesta del paradigma de la *computación ubicua*, realizada por Weiser (1991), supuso un cambio en el concepto de la computación distribuida: ésta ya no se encontraba limitada a una sola máquina o red de máquinas determinadas, sino que podía ser realizada por varios dispositivos más o menos complejos dispersos e integrados en el entorno en el que se encuentra situado el usuario. Con objeto de alcanzar la visión de la computación ubicua

los dispositivos deben conocerse y colaborar entre sí para intentar resolver los problemas del usuario, que simplemente utiliza el potencial de computación que se le proporciona sin preocuparse de cada uno de los dispositivos, o de la interrelación que pueda existir entre ellos. La computación ubicua, entre otros objetivos, busca que el usuario no perciba la tecnología que le rodea, pero sí que sienta los beneficios y servicios asociados a su despliegue. Es necesario, por lo tanto, que los dispositivos involucrados trabajen con el mínimo de intervención necesaria por parte del usuario, que no tiene que saber cómo conectar o configurar los dispositivos existentes. Deben ser los propios dispositivos los que determinen cuáles otros existen en sus cercanías y qué servicios pueden proporcionar cada uno de ellos, así como también cuáles de estos servicios son útiles para la realización de las tareas requeridas.

Cuando un usuario utiliza un entorno de computación ubicua debe poder acceder, tanto a los servicios que de forma individual le proporciona cada uno de los dispositivos, como a servicios complejos obtenidos del resultado de la composición de estos servicios más simples. Por otro lado, la movilidad del usuario en el entorno puede tener como consecuencia que servicios que anteriormente se encontraban disponibles dejen de estarlo, con el consiguiente inconveniente para la realización de las tareas o satisfacción de las necesidades del usuario. En estos casos, el sistema debería ser capaz de automatizar la configuración del entorno, así como de minimizar la intervención del usuario en el proceso de selección de los servicios disponibles. Así, la *composición de servicios* es un proceso mediante el cual se obtienen nuevas funcionalidades a partir de otros servicios disponibles en el entorno. Un ejemplo de esta visión es la computación orientada a tareas propuesta por Masuoka, Parsia y Labrou (2003), que pretende cubrir el hueco existente entre lo que quiere realizar el usuario y los servicios de los que dispone. El usuario especifica la tarea que quiere realizar en el entorno y este último es capaz de llevar a cabo la consecución de la tarea requerida utilizando los recursos y servicios disponibles.

Además de lo explicado anteriormente, la introducción de la web semántica por parte de Berners-Lee, Hendler y Lassila (2001) permitió nuevos caminos para la consecución de la visión de la computación ubicua. La web semántica propone utilizar descripciones semánticas para anotar los recursos de la web, extraer de forma automática el significado de la información disponible y, posteriormente, razonar sobre ella. Para poder lograr este objetivo la web semántica utiliza ontologías; según Castells (2002) *una ontología es una jerarquía de conceptos con atributos y relaciones que establece una terminología consensuada para definir redes semánticas de unidades de información interrelacionadas*. Se proporciona así un vocabulario de clases y relaciones que permite describir un dominio. A través de estas descripciones de conceptos, un programa informático puede conocer la relación entre diferentes términos y utilizarlos a la hora de procesar la información.

---

Una descripción semántica no solo puede ser aplicada a documentos de texto, sino también a cualquier otro recurso: imágenes, sonidos vídeos e incluso, como es el caso de esta investigación, a los servicios proporcionados por dispositivos. Como se mostrará a lo largo de esta tesis doctoral, gracias a las tecnologías semánticas es posible describir los servicios y dispositivos de un entorno ubicuo utilizando una ontología común, cuyo uso beneficia el proceso de descubrimiento, así como también permite llevar a cabo cierta automatización en el proceso de composición de servicios.

Por otro lado, hay que tener en cuenta que existen redes ubicuas con diferentes características en cuanto a la movilidad de sus dispositivos, mecanismos de comunicación y recursos de computación disponibles. Las redes de dispositivos pueden tener una *infraestructura* fija de comunicación que define cómo se organiza la topología de la red. Es el caso, por ejemplo, de la utilización de puntos de acceso Wi-Fi que gestionan y centralizan la comunicación entre dispositivos de computación inalámbricos. Sin embargo, existe también la posibilidad de que haya redes constituidas de una manera espontánea por los propios dispositivos que las integran. A este tipo pertenecen las *redes ad hoc*, que se caracterizan por no disponer de una infraestructura fija para la comunicación entre sus integrantes. Los dispositivos de las redes ad hoc participan activamente en el proceso de comunicación mediante su implicación en las tareas de *encaminamiento* de mensajes. Además, la comunicación en las redes ad hoc se realiza mediante el envío de mensajes que pueden requerir múltiples saltos a través de rutas que deben ser descubiertas en un proceso dinámico.

Los dispositivos integrantes de las redes ad hoc suelen ser dispositivos como teléfonos, ordenadores portátiles, sensores, etc., que debido a su posible movilidad tienden a cambiar su posición relativa los unos con respecto a los otros. El movimiento de los dispositivos integrantes de la red ad hoc supone una modificación de la topología que, a su vez, resulta en la necesidad de actualizar y gestionar las rutas de comunicación. Para diferenciarlas de las redes de comunicación con infraestructura tradicionales, este tipo de redes son conocidas como *redes móviles ad hoc* (*Mobile Ad hoc Network - MANET*).

El objetivo de esta tesis es el desarrollo de una solución que mejore la composición de servicios en redes móviles ad hoc mediante la introducción de un proceso de un grafo distribuido de servicios que es mantenido entre los nodos de la red. La construcción de este grafo hace uso de las relaciones de compatibilidad entre servicios, descubiertas a partir de su descripción mediante ontologías de conceptos comunes. Como parte de la investigación, se define también un proceso de descubrimiento de servicios para redes móviles ad hoc que es parte fundamental de la propuesta de composición de servicios. Este protocolo de descubrimiento de servicios constituye una aportación por sí mismo y puede ser utilizado independientemente de la solución para la composición.

## 1.1 Motivación

La composición de servicios es un paradigma que puede proporcionar al usuario, y a sus aplicaciones, grandes beneficios en los entornos constituidos por dispositivos que forman redes móviles ad hoc. Estas redes son utilizadas en situaciones en las que no existe posibilidad de desplegar una infraestructura de comunicación y que, debido a su movilidad, tienden a cambiar continuamente tanto su topología como los recursos y servicios que ofrecen: situaciones de emergencia, redes urbanas de dispositivos, etc. Es necesario, por lo tanto, un mecanismo que permita a los usuarios encontrar los servicios que satisfagan sus necesidades y, cuando esto no sea posible, componer otras funcionalidades disponibles en forma de servicios. Además, todo ello debe realizarse teniendo en cuenta las características de comunicación *multisalto* de este tipo de redes.

Tras la realización del estudio de la situación actual de la literatura científica sobre la composición de servicios, se ha comprobado que el problema ha suscitado un gran interés de la comunidad investigadora, tanto en su aplicación a redes con infraestructura como a redes móviles ad hoc. Sin embargo, como se constata en la revisión bibliográfica, las propuestas actuales para redes móviles ad hoc se centran en la aplicación de un *flujo de trabajo* predefinido que debe ser instanciado utilizando los servicios proporcionados por los dispositivos de la red. La creación y definición de este flujo de trabajo antes de comenzar el proceso de composición requiere la participación del usuario en tiempo de uso, o del desarrollador de la aplicación durante su creación. Además, en ambos casos se necesita tener un conocimiento de los tipos de servicios disponibles en el entorno donde se va a llevar a cabo la composición.

A diferencia de lo señalado anteriormente, en esta tesis se propone un mecanismo de composición de servicios *automático* que está basado en la especificación de las características del servicio buscado, y no necesita de la representación abstracta del servicio compuesto en forma de flujo de trabajo. Se minimiza así la intervención del usuario o desarrollador de la aplicación en el proceso de composición. Además, esta solución es presentada de una forma adecuada a las características propias de las redes móviles ad hoc.

Existen propuestas para redes ad hoc que se basan en la construcción de una estructura virtual de servicios que es posteriormente utilizada con objeto de mejorar el proceso de composición en la red (Gu, Nahrstedt y Yu, 2004; Hu, Muthusamy, Li y Jacobsen, 2008; Herborn y Seneviratne, 2006; Hu, Tang, Wang y Ji, 2007; Huang, Bai, Zhang, Kong y Qian, 2007). Por lo tanto, la idea en la que se fundamenta esta investigación, utilizar un grafo precalculado de servicios para mejorar el proceso de composición, no es nueva y fue propuesta anteriormente por Gu, Li y Xu (2008). Sin embargo, dicha solución había sido desarrollada y evaluada únicamente para su aplicación a redes distribuidas con una infraestructura fija. La utilización de un grafo distribuido de servicios, y de

los procesos correspondientes de creación y mantenimiento del grafo, no han sido aplicados y evaluados hasta ahora a las características de las redes móviles ad hoc. Así, esta investigación propone un proceso nuevo para la construcción del grafo de servicios que se basa en la detección distribuida de colisiones entre parámetros por parte de otros nodos de la red ad hoc.

Por otro lado, el proceso de composición de servicios puede beneficiarse de las posibilidades que introducen las tecnologías semánticas para mejorar el descubrimiento de servicios. Hay que tener en cuenta que el descubrimiento es una parte esencial del proceso de composición, principalmente en los entornos distribuidos y dinámicos como las redes móviles ad hoc donde no existe un conocimiento perdurable de las características del entorno. La aplicación de tecnologías semánticas al proceso de descubrimiento ha sido realizada anteriormente en varios trabajos (Chakraborty, Joshi, Yesha y Finin, 2006; Klein, König-Ries y Obreiter, 2003; Ruta, Zacheo, Grieco, Di Noia, Boggia, Tinelli, Camarda y Di Sciascio, 2010; Said y Mahéo, 2008; Islam y Shaikh, 2012), sin embargo, estas soluciones se fundamentan en la descripción de los servicios en su totalidad o en la clasificación según su tipo. Además, durante la composición de servicios, uno de los aspectos que deben ser resueltos es la compatibilidad entre sus parámetros de entrada/salida (Mokhtar, Liu, Georgantas y Issarny, 2005; Paolucci, Kawamura, Payne y Sycara, 2002).

Se hace necesaria, por lo tanto, una solución para el descubrimiento de servicios que tenga en cuenta estos aspectos y que se adecue a las características de las redes móviles ad hoc. Además, se cree también que la utilización de la información contenida en la ontología puede servir para mejorar los procesos de búsqueda de servicios e incluso, debido a las relaciones de especificación y generalización contenidas en las ontologías mejorar el proceso de búsqueda de servicios a través de la reducción de la cantidad de información propagada.

## 1.2 Hipótesis y objetivos

De acuerdo a lo explicado en la sección previa, se desarrolla la hipótesis principal de la investigación, que puede expresarse de la siguiente manera:

**H(A):** *La composición de servicios en una red móvil ad hoc puede llevarse a cabo mediante la utilización de un grafo distribuido de servicios que es construido y mantenido en respuesta a los cambios que pueda sufrir la topología de la red. Este grafo debe permitir llevar a cabo la composición de servicios de una manera más rápida haciendo uso de la información de rutas y conexiones de servicios que contiene.*

Por otro lado, como también se ha explicado en la sección 1.1, el proceso de composición basado en la creación de un grafo distribuido de servicios necesita un mecanismo de descubrimiento con unas características adecuadas a las

redes móviles ad hoc. El descubrimiento de servicios constituye un problema propio y, por lo tanto, requiere la definición de una segunda hipótesis para su investigación:

**H(B):** *El descubrimiento de servicios en una red móvil ad hoc puede llevarse a cabo en base a una solicitud que contenga la descripción funcional (parámetros de entrada/salida) del servicio deseado y categorizados de acuerdo a una ontología de conceptos, permitiendo así mejorar el proceso de descubrimiento. Esta mejora se traduce tanto en un aumento de la expresividad durante las búsquedas como en la reducción del número de mensajes enviado.*

Para lograr la visión planteada en la investigación se ha definido una serie de objetivos, tanto para el descubrimiento de servicios como para la composición en redes móviles ad hoc. En el caso del descubrimiento de servicios, que se enuncia primero porque su consecución es necesaria para resolver el problema de composición, se han establecido los siguientes objetivos:

- **Objetivo B.1:** definición del proceso mediante el cual se lleva a cabo la diseminación de la información de los servicios en la red ad hoc. Hay que tener en cuenta que estas redes son de naturaleza completamente distribuida y, por lo tanto, es necesario un mecanismo mediante el cual los proveedores de servicios pongan su información a disposición de los clientes que quieran obtenerla. Los servicios deben ser descritos de la forma adecuada utilizando tecnologías semánticas.
- **Objetivo B.2:** definición del mecanismo de búsqueda de servicios que haga uso de la información sobre los servicios y permita encontrarlos en la red ad hoc frente a la movilidad de los dispositivos que la integran. El proceso de descubrimiento de servicios debe hacer uso de la información semántica con objeto de reducir el número de mensajes propagados por el protocolo.
- **Objetivo B.3:** la movilidad de los nodos tiene como consecuencia que las rutas de comunicación cambien durante el tiempo de vida de la red móvil ad hoc. Es necesario definir un proceso que permita integrar los procesos de creación y mantenimiento de rutas con el descubrimiento de servicios teniendo en cuenta además las características de las soluciones anteriores.
- **Objetivo B.4:** desarrollo de las soluciones planteadas para la diseminación de información y búsqueda de servicios, así como de la validación necesaria que permita comprobar que su funcionamiento se adecua a lo esperado.

- **Objetivo B.5:** evaluación de la solución de descubrimiento mediante el uso de un simulador de red y la realización de distintos experimentos orientados a conocer las ventajas y limitaciones de las propuesta.

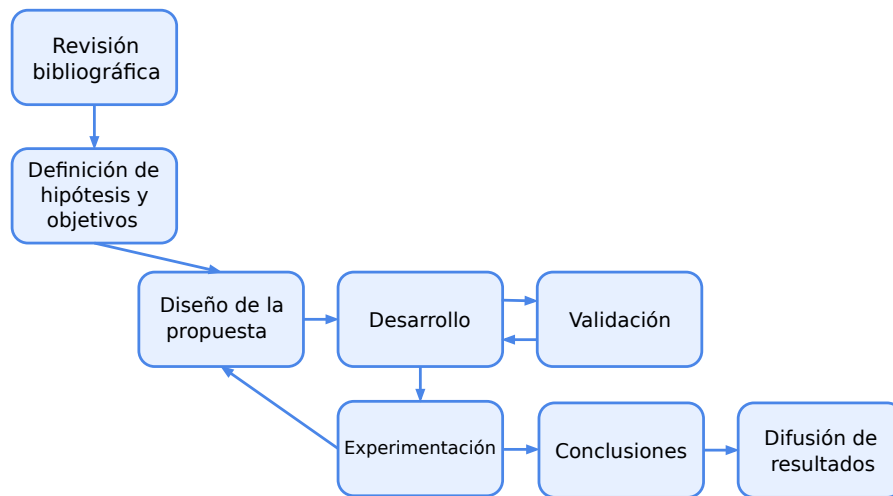
Por otro lado, el proceso de composición de servicios define también sus propios objetivos que, de acuerdo a la interrelación entre ambas propuestas, se basan en la consecución necesaria de los anteriores, y que son expuestos a continuación:

- **Objetivo A.1:** el proceso de construcción del grafo de servicios debe realizarse de una manera distribuida. Por lo tanto, es necesario definir un mecanismo que permita construir y mantener el grafo sin la necesidad de centralizar la información de servicios en uno o varios puntos de la red. El mecanismo debe hacer uso del protocolo de descubrimiento de servicios establecido anteriormente.
- **Objetivo A.2:** es necesario definir un proceso de búsqueda en el grafo distribuido de servicios que utilice la información contenida y que sea adecuado a la naturaleza dinámica y distribuida de las redes móviles ad hoc.
- **Objetivo A.3:** desarrollo de las solución de composición de servicios obtenida anteriormente y su validación para determinar que su funcionamiento es adecuado.
- **Objetivo A.4:** evaluación de la solución de composición propuesta mediante la utilización de un simulador de red con objeto de comprobar sus ventajas y limitaciones.

Como puede observarse, se distingue entre la validación por un lado, y la evaluación o experimentación por otro. La *validación* es el proceso mediante el cual se comprueba que la solución implementada es correcta desde un punto de vista funcional. Puede obtenerse más información acerca de la metodología seguida para la validación en el anexo correspondiente. Por otro lado, la *evaluación* o *experimentación* hace referencia a la obtención de medidas y resultados que permiten caracterizar el comportamiento y rendimiento de la propuesta en diferentes situaciones.

## 1.3 Metodología de investigación

Con objeto de conseguir los objetivos fijados en la sección anterior se ha establecido una estrategia de investigación que se representa de forma gráfica en la figura 1.1 y que ha consistido en la realización de las siguientes actividades.



**Figura 1.1:** Metodología de investigación

1. Revisión de la literatura científica sobre la composición de servicios de una forma general y, en concreto, en su aplicación a las redes móviles ad hoc. En esta fase se ha requerido también una revisión de las soluciones para el descubrimiento de servicios aplicadas a este tipo de redes con el objeto de decidir la posibilidad de adaptar o elaborar una solución propia. Por otro lado, la revisión bibliográfica ha permitido también definir la hipótesis de la investigación, compuesta por dos sub-hipótesis y presentada en la sección 1.2. Así mismo, en esta fase se ha llevado también a cabo un estudio de las propiedades de las redes móviles ad hoc para determinar cuáles son las características de estas redes y sus limitaciones a la hora de aplicar en ellas soluciones realizadas para las redes tradicionales.
2. La revisión bibliográfica ha dado lugar a una clasificación de las soluciones existentes con la finalidad de establecer sus limitaciones, ventajas y desventajas. Este trabajo ha sido realizado tanto para la composición de servicios como para el descubrimiento de los mismos, restringiéndose principalmente a su aplicación a los problemas y características de las redes móviles ad hoc.
3. Definición de los requisitos de la solución propuesta tanto en lo referente al descubrimiento como a la composición de servicios de forma adecuada a las características de las redes móviles ad hoc.
4. Diseño, desarrollo y pruebas de validación de las soluciones propuestas, teniendo en cuenta la dependencia que el protocolo de composición de servicios posee con respecto a la solución de descubrimiento. Las pruebas de validación han permitido, debido a la complejidad de las redes



móviles ad hoc, comprobar que el desarrollo de la solución se comporta correctamente en una serie de situaciones definidas y conocidas. Se puede consultar el anexo correspondiente para conocer más información a este respecto.

5. Evaluación de las soluciones propuestas, tanto en descubrimiento como en composición de servicios para redes móviles ad hoc, mediante la realización de distintos experimentos. La experimentación ha permitido comprobar las aportaciones, ventajas y limitaciones de las propuestas realizadas en la investigación.
6. Rediseño y mejora de los protocolos de descubrimiento y composición propuestos con los resultados obtenidos en las fases de prueba y experimentación. Este proceso ha sido *iterativo* y se ha realizado hasta obtener un funcionamiento correcto de la propuesta realizada.
7. Elaboración de las conclusiones y difusión de los resultados en foros científicos relacionados con la temática abordada en la tesis para su consideración y aprobación.

## 1.4 Organización de la tesis

En el capítulo 2 se lleva a cabo, tras la introducción a la problemática abordada en esta tesis, una breve presentación de las características básicas de las redes ad hoc con objeto de conocer las dificultades que plantean a la hora de trasladar las soluciones desarrolladas para las redes tradicionales. Además, dicho capítulo define varios conceptos sobre las redes móviles ad hoc y algunos aspectos relacionados con la utilización de herramientas de simulación.

En el capítulo 3 se incluye una revisión bibliográfica de las soluciones existentes en la actualidad para el descubrimiento y composición de servicios en su aplicación a las redes móviles ad hoc. En ambos casos dicho estudio se fundamenta en la realización de una clasificación de las soluciones existentes en la actualidad. Esto ha permitido compararlas con las características de las soluciones presentadas en esta investigación, así como las implicaciones de las decisiones que se han tomado durante su realización.

El capítulo 4 introduce de una manera formal cuál es el problema a resolver así como los requisitos y limitaciones de la solución. Dicho capítulo presenta también la arquitectura global de la propuesta para la composición de servicios en redes móviles ad hoc y su división en diferentes aspectos: *gestión de red*, *descubrimiento de servicios* y *composición*. Además, en este capítulo se presenta el mecanismo de descripción de servicios utilizado en la propuesta.

En el capítulo 5 se propone un protocolo para el descubrimiento de servicios en redes móviles ad hoc que hace uso del tipo de los parámetros de los

servicios categorizándolos de acuerdo a una ontología de conceptos. El capítulo contextualiza la solución propuesta, así como la definición de las principales aportaciones de la misma. Además de la descripción formal del protocolo, el capítulo incluye los resultados de la evaluación realizada que demuestran la validez de las aportaciones en el descubrimiento de servicios aplicado a redes móviles ad hoc.

El capítulo 6 describe las características de la solución de composición presentada y que se fundamenta en la construcción de un grafo distribuido de servicios. Nuevamente, dicho capítulo incluye una presentación formal del protocolo de composición aplicado a las redes móviles ad hoc, así como las aportaciones principales del mismo. Las evaluaciones realizadas, y cuyos resultados han sido también recogidos en el capítulo, permiten conocer cuál es el alcance de la propuesta realizada.

Por último, el capítulo 7 recoge las conclusiones extraídas tras la realización de la investigación y presenta las posibles líneas de investigación futuras que permitan continuar y mejorar el trabajo aquí expuesto.

## Breve introducción a las redes móviles ad hoc

Las redes de computadores pueden tener diferentes estructuras, dependiendo de la tecnología utilizada para la comunicación entre los dispositivos y de otros aspectos como los protocolos y servicios de comunicación usados en ellas. Este trabajo de investigación se centra en las redes inalámbricas ad hoc que se caracterizan, como su nombre indica, por la comunicación sin cables y el hecho de que operan sin la existencia de una infraestructura fija. Además, en este tipo de redes, la movilidad de los dispositivos de computación se traduce en cambios en la topología de la red, que a su vez afectan a los mecanismos de comunicación. La sección 2.1 lleva a cabo una introducción a las redes móviles ad hoc teniendo en cuenta sus aspectos y características fundamentales, proporcionando una visión general de los aspectos de comunicación subyacentes y de las problemáticas que este tipo de redes introducen. Por otro lado, en la sección 2.2 se introduce la necesidad del uso de simuladores para la experimentación con redes móviles ad hoc, así como los parámetros que permiten la realización de simulaciones correctas.

### 2.1 Redes móviles ad hoc

Las *redes móviles ad hoc* (*Mobile Ad hoc Network - MANET*) son un tipo de redes inalámbricas que se basan en la propagación de mensajes mediante múltiples saltos. La locución latina *ad hoc*, que significa literalmente *para esto*, hace referencia al hecho de que este tipo de redes carece de una infraestructura de

comunicación predefinida. La comunicación en estas redes ocurre de una manera no planificada y respondiendo a la distribución de los dispositivos que exista en un momento determinado. Estas redes no disponen de elementos cuyo cometido sea gestionar las tareas de comunicación, siendo el proceso de envío de mensajes una tarea distribuida entre todos los participantes de la red. Son los diferentes elementos con capacidades de computación los que llevan a cabo todos aquellos procesos relacionados con la comunicación. Para una información más detallada sobre las redes móviles ad hoc, e inalámbricas en general, puede consultarse el libro de Murthy y Manoj (2004).

Las redes MANET tienen una serie de características que deberán ser tenidas en cuenta por los protocolos desarrollados para las mismas:

- **Operación distribuida:** los nodos de este tipo de redes no pueden confiar en una red de nodos fija que le proporcione la funcionalidad de encaminamiento, que debe ser proporcionada de una manera distribuida por los diferentes nodos que integran la red.
- **Topología dinámica:** los nodos que integran la red se conectan y desconectan aleatoriamente o cambian su posición dentro de la misma. Por lo tanto, los protocolos deben realizar las operaciones necesarias para garantizar que la conectividad entre los nodos pueda ser mantenida, que dependerá de las características del protocolo utilizado.
- **Variaciones en la capacidad de enlace:** debido a que la comunicación en estas redes es inalámbrica y se realiza mediante saltos, los errores introducidos por los enlaces tienden a producirse con mayor frecuencia. Los protocolos creados para este tipo de redes deben ayudar a controlar y minimizar los errores que puedan aparecer.
- **Requisitos de baja energía:** los dispositivos de estas redes suelen estar operados con baterías, lo que impone una serie de límites a todos los recursos del dispositivo (CPU, memoria, comunicaciones, ...). Por esta razón los protocolos de comunicación desarrollados para estas redes deben estar diseñados para reducir el consumo de recursos en los dispositivos.

Las redes MANET se crean de forma espontánea cuando dispositivos con capacidades de computación se encuentran en las proximidades de otros dispositivos con los que pueden establecer una comunicación mediante cierta tecnología inalámbrica. Esto sucederá cuando los dispositivos se sitúen dentro del rango de comunicación que posibilite la tecnología de comunicación utilizada. Dos o más dispositivos entre los que existe una comunicación directa pueden, en teoría, intercambiar información o compartir recursos entre ellos. Sin embargo, permitir la comunicación solamente entre dispositivos que se encuentren conectados de esta forma limitaría enormemente las aplicaciones de este tipo

de redes. Las redes ad hoc posibilitan también la *comunicación indirecta* entre dispositivos que no se encuentran a distancia directa de comunicación. Cuando esto sucede, la información es propagada mediante una serie de saltos consecutivos de dispositivo a dispositivo de la red, hasta que se alcanza el destino requerido. El descubrimiento y mantenimiento de estas rutas de comunicación indirecta es un proceso complejo que requiere la participación de los distintos dispositivos que conforman la red.

La otra característica fundamental de estas redes es la movilidad de sus participantes. Los dispositivos de computación que forman parte de una red MANET pueden moverse durante su participación en la red de comunicación. En las redes de infraestructura fija el movimiento de los elementos de computación no suele tener un gran impacto, más allá de la imposibilidad de acceder a cierto dispositivo porque éste ya no se encuentra conectado a la red. Esto es debido a que en las redes tradicionales existe una separación entre los elementos que conforman la red de comunicación (*encaminadores, conmutadores y puentes*) y los dispositivos de computación que hacen uso de ella (*ordenadores personales, servidores, etc.*). En las redes tradicionales la movilidad suele aplicarse al último grupo y, por lo tanto, no suele afectar a la comunicación en su totalidad. Así, la movilidad de un servidor dentro de la red únicamente afectará a aquellos otros dispositivos que quieran acceder a los recursos del mismo, pero normalmente no tendrá un impacto en las posibilidades de comunicación entre otros dispositivos. Sin embargo, esto no es así en el caso de las redes MANET debido a que en estas redes no existe una separación clara entre elementos de comunicación o de computación. Así, la movilidad de un elemento de la red puede afectar seriamente a la posibilidad de comunicación entre otros dispositivos.

### 2.1.1 Conceptos fundamentales

Cualquier red de comunicación, y en concreto las redes ad hoc, puede representarse como un grafo no dirigido en el que los nodos representan los dispositivos con capacidades de comunicación y las aristas un enlace de comunicación entre dos nodos. Los enlaces de comunicación se establecen entre aquellos nodos que se encuentren dentro del radio de transmisión uno del otro. En las figuras que se encuentran en esta tesis se utilizará esta convención para representar una red.

Con objeto de clarificar el uso que se hace en esta investigación de ciertos términos usuales en las redes ad hoc, se introducen ahora las definiciones y convenciones que se utilizarán a lo largo del presente capítulo y posteriores.

- **Nodo:** cada uno de los dispositivos con capacidades de computación que participan en una red ad hoc. Pueden ser dispositivos con características muy variadas, pero se supone que todos ellos poseen la misma tecnología

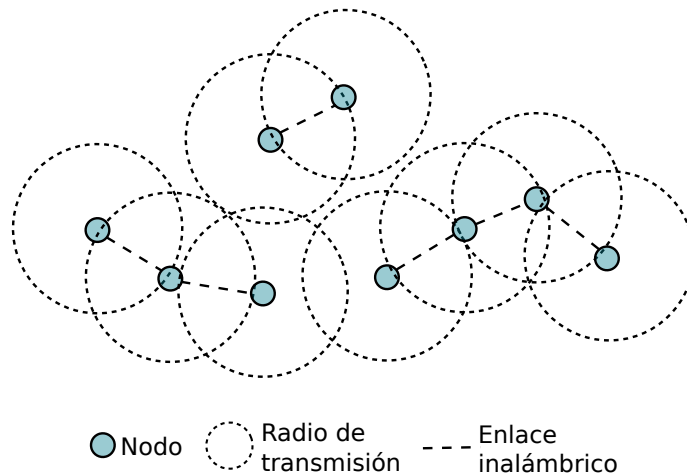


Figura 2.1: Representación de una red móvil ad hoc

de comunicación inalámbrica y, por lo tanto, pueden formar parte de la misma red ad hoc.

- **Enlace:** si dos nodos se encuentran a una distancia tal, dependiente de las características de la tecnología inalámbrica utilizada, que es posible su intercomunicación, se dirá que existe un enlace entre dichos nodos.
- **Vecino:** se utiliza este término para referirse a aquellos nodos que se encuentran en rango de comunicación directa entre ellos. En la literatura puede encontrarse una generalización de este término para referirse a: vecinos de grado  $n$ , es decir nodos que se encuentran a una distancia de  $n$  saltos del nodo actual.
- **Distancia:** se mide en *saltos* y hace referencia al número de enlaces de comunicación que la información tiene que realizar para llegar a cierto nodo destino.
- **Mensaje:** es la unidad de información que se transmite de un nodo a otro a través de un enlace. El tamaño máximo del mensaje dependerá, nuevamente, de las características de la tecnología inalámbrica utilizada.
- **Encaminamiento:** es el proceso por el cual dos nodos de la red ad hoc pueden comunicarse entre ellos mediante el paso de uno o más mensajes a través de diferentes enlaces y nodos intermedios. Los caminos existentes entre dos nodos, formados por enlaces consecutivos, pueden variar durante la movilidad de la red ad hoc.

En la figura 2.1 se muestra una red ad hoc con sus distintos elementos representada mediante un grafo con los diferentes nodos y enlaces de comunicación disponibles.

Además de los aspectos que se explican en las siguientes secciones, las redes ad hoc plantean otros desafíos en el desarrollo de soluciones que puedan ser aplicadas a sus características: la gestión de la energía en los nodos, la seguridad de las comunicaciones y los protocolos, la calidad del servicio, etc.

### 2.1.2 Comunicación física y acceso al medio

Las redes ad hoc son redes de naturaleza inalámbrica y la comunicación entre los nodos se lleva a cabo mediante la utilización de ondas electromagnéticas que se propagan en el medio. Debido a que no se utiliza un canal de transmisión dedicado, las ondas electromagnéticas sufren diferentes problemas durante la transmisión: *atenuación, interferencias y bloqueo*.

La atenuación se define como la relación entre la potencia transmitida y la potencia recibida. Debido a la naturaleza de la radiación electromagnética, la potencia recibida disminuye de forma inversamente proporcional al cuadrado de la distancia recorrida, en el caso de utilizar el modelo más simple para la transmisión en el *espacio libre*. Cuando existe transmisión a través de materiales, o teniendo en cuenta otros factores físicos, las pérdidas debidas al fenómeno físico de la atenuación serán mayores.

En el caso de la transmisión de información mediante el uso de ondas electromagnéticas, la interferencia tiene como resultado la pérdida de la misma. Como el medio de transmisión es compartido, los dispositivos inalámbricos no pueden transmitir siempre que quieran o se producirían interferencias al utilizar la misma frecuencia de transmisión. Para solucionar este problema existen diferentes técnicas para controlar el acceso al canal compartido.

Por último, el bloqueo de la señal se produce por la existencia de objetos físicos intermedios que atenúan la señal transmitida de tal forma que no puede ser captada por los dispositivos receptores. En los entornos reales en los que se despliegan las redes ad hoc existirán objetos (muebles, paredes, etc.) que producirán no solo el bloqueo de la señal, sino también otros fenómenos como la reflexión, la difracción y la dispersión de las ondas electromagnéticas.

Por todas las razones anteriores, la comunicación mediante el uso de ondas electromagnéticas requiere mecanismos de control que permiten solucionar o reducir el impacto de estos problemas. El *Control de Acceso al Medio (Medium Access Control - MAC)* tiene como objetivo, como su nombre indica, determinar cuándo un dispositivo puede realizar una transmisión en el medio compartido para maximizar su eficiencia. Esta tesis se ha basado en la utilización del protocolo *Wi-Fi IEEE 802.11* (Crow, Widjaja, Kim y Sakai, 1997) para la comunicación entre los dispositivos de la red ad hoc. Sin embargo, las redes móviles ad

hoc podrían estar constituidas por otros mecanismos de comunicación como el estándar *Zigbee* (Zigbee Alliance, 2006) o *6LoWPAN* (Mulligan, 2007).

### 2.1.3 Protocolos de encaminamiento

Los protocolos de encaminamiento para redes MANET pueden clasificarse en dos grandes grupos: *proactivos* y *reactivos*.

Los protocolos proactivos incluyen aquellas soluciones en las que los nodos de la red mantienen tablas con la información sobre las rutas existentes incluso antes de que esta información sea necesaria. Normalmente, estos protocolos no son adecuados para redes grandes, debido a que el mantenimiento de la información y el envío de mensajes de actualización suponen un coste adicional. Sin embargo, en redes de número de nodos más reducido pueden suponer una ventaja a la hora de reducir el tiempo de búsqueda de rutas. Ejemplos clásicos de protocolos proactivos son *Optimized Link State Routing* (OLSR) (Clausen y Jacquet, 2003) y *Destination Sequenced Distance-Vector* (DSDV) (Perkins y Bhagwat, 1994).

Por el contrario, en el caso de los protocolos reactivos, las rutas de comunicación se buscan únicamente en los momentos en los que un nodo quiere establecer una comunicación con otro. La búsqueda de las rutas ocurre mediante la inundación de la red con mensajes de solicitud de comunicación. Tras el descubrimiento de una ruta de comunicación con un nodo, los protocolos pueden aplicar diferentes mecanismos para mantener su validez durante un periodo de tiempo frente a los cambios en la topología de la red ad hoc. Por ejemplo, *Dynamic Source Routing* (DSR) (Johnson, Maltz y Broch, 2001) y *Ad hoc On-Demand Distance-Vector* (AODV) (Perkins, Royer y Das, 2003).

Existen, por otro lado, soluciones *híbridas* que combinan los dos enfoques anteriores con el objeto de intentar obtener las ventajas de ambos tipos de protocolos, como es el caso de *Zone Routing Protocol* (ZRP) (Haas, Pearlman y Samar, 2002).

Los protocolos se pueden clasificar también en *jerárquicos*, que utilizan una descomposición jerárquica de la ruta para llegar al destino; *geográficos* en los que el encaminamiento tiene en cuenta la situación física mediante el uso de dispositivos GPS, de tal forma que los mensajes son enviados a aquellos nodos más cercanos al lugar geográfico que se quiere alcanzar; *conscientes de la energía*, que tienen en cuenta las limitaciones de consumo de los nodos e intentan repartir la carga de encaminamiento entre ellos de forma que el consumo medio de cada nodo sea similar; y protocolos *multicast*, que crean y gestionan grupos para realizar el encaminamiento. Por supuesto, existen protocolos de encaminamiento para redes ad hoc que hacen uso de varias de las características anteriormente indicadas.



### 2.1.4 Aplicaciones

Las redes ad hoc son útiles en aquellas situaciones en las que hay una necesidad de desplegar una red de dispositivos y no existe una infraestructura de comunicaciones anterior. Además, su naturaleza distribuida las hace adecuadas para tareas como la computación colaborativa. Los dominios de aplicación de las redes ad hoc van desde las *situaciones de emergencia* hasta las *operaciones militares*, pasando por las *redes de sensores*, que pueden considerarse un tipo de red ad hoc orientada a la captura de información. Estas últimas se aplican en numerosos ámbitos como las *ciudades inteligentes*, la monitorización de bosques, en la agricultura y ganadería, etc.

Dependiendo de la gravedad de una situación de emergencia, causada, por ejemplo, por una catástrofe natural (terremotos, inundaciones, tornados, . . .), es posible que las infraestructuras de comunicación de una zona hayan quedado inutilizadas. Las redes ad hoc proporcionan la posibilidad a los equipos de rescate de llevar a cabo un despliegue rápido de una red de comunicación formada por los propios dispositivos de computación, así como ser utilizadas durante las tareas de búsqueda o reparación con el objeto de mejorar la eficiencia de estas tareas. Así, servicios o recursos que ya no están disponibles debido a la caída de la red de comunicación tradicional pueden ser puestos a disposición a través de la red ad hoc desplegada.

Compartiendo características con el dominio de las situaciones de emergencia se encuentran también las operaciones militares. Al igual que en las situaciones de rescate, en el ámbito militar suele existir la necesidad de desplegar una red de comunicaciones de forma rápida sobre un terreno desconocido y haciendo frente al movimiento que pueda producirse en el campo de batalla.

Por otro lado, las redes ad hoc pueden utilizarse también en situaciones más cotidianas, por ejemplo, conferencias o reuniones puntuales de personas en lugares sin acceso a una red tradicional. En dichas situaciones puede existir la necesidad de distribuir información entre las personas asistentes, y el uso de una red ad hoc puede hacer más fácil esta tarea sin la necesidad de desplegar y gestionar una red con infraestructura.

Por último, las *redes de sensores* pueden clasificarse como un tipo de red ad hoc en la que los dispositivos conectados tienen capacidades de computación muy limitadas y se dedican exclusivamente a las tareas de captura de información a través de los sensores incorporados, así como a la posterior propagación de la misma a una estación central. Este tipo de redes suele tener una movilidad más limitada que las redes ad hoc en su sentido más amplio, por lo que no comparten los problemas asociados al mantenimiento de las rutas de encañamiento. Sin embargo, introducen problemas nuevos relacionados principalmente con la cantidad de sensores que pueden participar en una misma red (escalabilidad), el ancho de banda en la transmisión y con la gestión de la energía. Normalmente, estos sensores serán desplegados sobre una zona y deberán

monitorizar el entorno, sin la posibilidad de ser recargados, durante un largo periodo de tiempo.

## 2.2 Simulación y experimentación

La naturaleza dinámica de las redes MANET complica su experimentación en entornos reales. Las pruebas deben realizarse con un número variable de dispositivos inalámbricos que deben moverse en un entorno con dimensiones ajustables siguiendo uno o varios patrones y con distintos rangos de velocidades. Es, por lo tanto, muy común en la investigación con redes móviles ad hoc la utilización de simuladores de red que permiten la experimentación en un entorno controlado. Un simulador de red proporciona modelos del comportamiento de los diferentes elementos de una red de nodos móviles: transmisión en el medio físico, protocolos de encaminamiento, generación de tráfico, movilidad, dimensiones del escenario de prueba, etc.

La credibilidad y fiabilidad de los simuladores de red, y su aplicación a las redes MANET, es un aspecto discutido en la comunidad científica (Cavin, Sasson y Schiper, 2002; Andel y Yasinsac, 2006). El principal problema argumentado por sus autores es que las mismas configuraciones y protocolos evaluados en distintos simuladores ofrecen resultados muy dispares. Estas diferencias de resultados tienen su razón en que los modelos e implementaciones utilizados por los distintos simuladores de red disponibles no son equivalentes. Los autores indican que únicamente la realización de pruebas reales puede proporcionar resultados fiables sobre una solución particular. Sin embargo, como se ha argumentado anteriormente, la realización de pruebas con dispositivos reales requiere una inversión considerable, tanto en tiempo, para la realización de las pruebas con las condiciones ideales, como en dinero por el desembolso en el equipamiento necesario para obtener el número de dispositivos que pueden llegar a participar en una configuración de red determinada.

Por otro lado, los mismos autores de dichas investigaciones consideran que las pruebas realizadas mediante la utilización de simuladores pueden resultar significativas para determinar la validez de una determinada propuesta, siempre que estas lleven a cabo con una configuración correcta de los parámetros correspondientes. En la siguiente sección se discuten brevemente las características de los parámetros de una simulación y sus implicaciones.

En el caso de esta tesis, se ha optado por la realización de simulaciones de red para conocer las características de las soluciones presentadas en un entorno *de laboratorio* que pueda proporcionar una mayor facilidad a la hora de modificar las configuraciones de prueba. Las simulaciones realizadas en el capítulo 5 y capítulo 6 utilizan redes constituidas por un número de dispositivos que puede variar entre los 20 y los 100 nodos y que se mueven a distintas velocidades en escenarios de unos 350 x 350 m a 700 x 700 m. La realización de experimentos

reales y la posterior recolección de los datos resultantes utilizando dispositivos reales hubieran requerido un esfuerzo muy difícil de realizar con los medios disponibles para la realización de la investigación.

### 2.2.1 Simuladores de red

Para la realización de la investigación se han estudiado los siguientes simuladores de red para la experimentación con redes MANET: ns-2 (The VINT Project, 2011), GloMoSim (UCLA Parallel Computing Laboratory, 1999; Zeng, Bagrodia y Gerla, 1998) y OPNET (OPNET Technologies, Inc., 2012).

Los simuladores estudiados comparten el hecho de que están basados en la simulación de *eventos discretos*; esto significa que el comportamiento de la red de nodos es modelado mediante el procesamiento, en el orden de aparición, de los eventos que se producen. Los eventos se crean como respuesta a los sucesos que ocurren en la red ad hoc: envío de un mensaje, recepción de mensaje, detección de colisión en las transmisiones; incluso pueden ser generados por el usuario mediante el uso de guiones para definir el comportamiento de la simulación. Suelen proporcionar modelos para simular la capa física del medio de transmisión, los protocolos de acceso al medio, de encaminamiento y la generación de tráfico en la red. Además, incluyen mecanismos para generar y simular la movilidad de los nodos que participan en la simulación.

- **ns-2:** es un simulador de red de código abierto distribuido bajo licencia *GNU GPLv2* y que se desarrolla de forma colaborativa. Es utilizado principalmente con fines educativos y de investigación. Es una variación del trabajo iniciado en 1989 en la *Cornell University* y continuado posteriormente por el *Departamento de Defensa de Estados Unidos* a través de *Xerox PARC*. En la actualidad incluye aportaciones de diferentes centros de investigación y universidades. Las extensiones del simulador y la implementación de nuevos protocolos se realizan en el lenguaje de programación C++. Por otro lado, la configuración de los escenarios de simulación y los distintos eventos a simular durante la experimentación se lleva a cabo utilizando el lenguaje Tcl. El principal problema de ns-2 es su escalabilidad para simular grandes redes debido al aumento de los recursos de computación necesarios. Actualmente, se está trabajando en una nueva versión del simulador denominada NS-3 (NS-3 Development Group, 2011) que incluye un rediseño completo de la herramienta para facilitar su uso y mantenimiento así como la inclusión de futuras extensiones.
- **GloMoSim:** es otro simulador de red basado también en la utilización de eventos discretos y utiliza un lenguaje de programación paralela denominado *Parsec*, y desarrollado por el *UCLA Parallel Computing Laboratory*. Todas las extensiones y protocolos nuevos que quieran incluirse en una

Tabla 2.1: Características de los simuladores de red

	ns-2	GloMoSim	OPNET
<b>Lenguaje</b>	C++	C / Parsec	C++
<b>Redes inalámbricas</b>	Si	Si	Si
<b>Escenarios</b>	Tcl	Parsec	Interfaz gráfica
<b>Cliente Java</b>	Si (AgentJ)	No	No
<b>Licencia</b>	GNU GPLv2	Uso educativo	Edición limitada
<b>Otros</b>	Más utilizado y bien documentado	Posibilidad de ejecución en paralelo	Simplicidad de uso

simulación deben implementarse en dicho lenguaje. Proporciona únicamente la posibilidad de simular redes inalámbricas. Su principal ventaja es que tiene la posibilidad de ejecutarse de forma paralela en varios procesadores y reducir de esta forma el tiempo de ejecución de una simulación. La última versión del simulador fue realizada en el año 2000, a partir de ese momento la empresa desarrolladora dejó de trabajar en la versión de uso libre. Sin embargo el trabajo ha dado lugar a nuevo simulador de licencia comercial denominado QualNet (SCALABLE Network Technologies, Inc., 2011).

- **OPNET:** es un simulador comercial desarrollado por *OPNET Technologies*. Tiene un amplio uso en el ámbito empresarial y educacional y dispone de una edición limitada para uso no comercial. Es también un simulador basado en eventos y su principal ventaja radica en el hecho de que integra los procesos de modelado, simulación y análisis de redes. Además, a diferencia de las herramientas anteriores, todo el proceso se realiza de forma gráfica, lo que simplifica enormemente su uso y aprendizaje.

La tabla 2.1 resume y compara algunas características de los simuladores de red comentados anteriormente.

Para llevar a cabo la experimentación correspondiente a la investigación presentada en esta tesis, se ha seleccionado el simulador ns-2 porque permite la ejecución de código Java mediante la extensión con AgentJ (Taylor, 2010). Tanto el protocolo de descubrimiento presentado en el capítulo 5, como el de composición de servicios del capítulo 6, se han desarrollado utilizando el len-

guaje de programación Java y mecanismos de comunicación básicos como los *sockets UDP*.

El uso de las librerías estándar del entorno Java, así como de mecanismos de comunicación de red básicos, han permitido que la solución pueda ser validada mediante una implementación real aunque no haya sido desplegada en dispositivos reales. Además, el hecho de que el simulador ns-2 está distribuido con una licencia abierta permite una mayor facilidad a la hora de conocer su funcionamiento y su extensión con los protocolos necesarios.

### 2.2.2 Parámetros de simulación

La configuración de los escenarios de prueba y de los distintos parámetros de un simulador de redes ad hoc, como es el caso del simulador ns-2, se lleva a cabo mediante la especificación de un gran número de parámetros. La variación de estos parámetros puede tener un gran impacto en los resultados obtenidos en una simulación. Los parámetros que definen una simulación pueden definirse en tres grupos principales: los relativos de la herramienta de simulación, los relacionados con características del escenario, y aquellos que son propios a la solución que se está evaluando.

Una misma herramienta de simulación de redes MANET puede ser configurada de muchas maneras posibles. Kurkowski, Navidi y Camp (2007b) identifican distintos aspectos que deben ser configurados correctamente en el simulador para obtener resultados fiables: el tipo de antena, la distancia de transmisión, el protocolo MAC utilizado, la gestión de la cola de mensajes y el modelo de propagación utilizado. Muchas investigaciones consultadas en la revisión bibliográfica no especifican cuál ha sido la configuración de estos parámetros en la herramienta lo que dificulta, o incluso imposibilita, en muchos casos, la realización de comparaciones entre diferentes soluciones.

La tabla 2.2 muestra los parámetros seleccionados, que son comunes para la realización de los experimentos del capítulo 5 y capítulo 6, y que han sido determinados teniendo en cuentas las configuraciones más típicas para las redes inalámbricas móviles ad hoc.

Se han definido algunos parámetros físicos comunes, como la elección de una antena omni-direccional y un modelo de propagación de dos rayos. El modelo de propagación *2-Ray Ground* seleccionado se basa en el uso de un camino directo y otro rayo reflejado en el suelo, y según indica la especificación del simulador ns-2, ofrece resultados más precisos, para transmisiones a largas distancias, que el modelo de propagación basado en la transmisión en *espacio vacío*.

Por otro lado, existen parámetros que son propios del escenario de simulación: la topología del escenario, el número de nodos y la movilidad de los mismos. Hay que tener en cuenta que las dimensiones del escenario con res-

**Tabla 2.2:** Configuración del simulador ns-2 para los experimentos

Parámetro simulador	Valor
Tipo de antena	Omnidireccional
Modelo de propagación	2-Ray Ground
Protocolo MAC	IEEE 802.11
Cola de mensajes	5 máx. (descartar)
Rango de transmisión	100 m
Velocidad de transmisión	[11, 54] Mb/s
Máx. paquete UDP	1500 bytes

pecto a la movilidad de los nodos y a su distancia de transmisión puede hacer que escenarios, que en un principio parecen muy distintos entre sí, sean realmente muy similares en cuanto a sus elementos característicos. Por ejemplo, teniendo en cuenta que la movilidad de los nodos suele estar limitada normalmente a la superficie determinada por la topología del escenario, si la distancia de transmisión es  $T$  y el área de pruebas es un rectángulo de dimensiones  $(a, b)$  se obtiene que

$$n_{max} = \frac{\sqrt{a^2 + b^2}}{T}$$

donde  $n_{max}$  es el número de nodos máximo que puede cubrir la diagonal del área de simulación con la distancia de transmisión dada. Lo que significa que, si los dos parámetros relacionados se varían sin tener en cuenta esta relación, se pueden obtener escenarios que realmente no prueben las características del protocolo evaluado. Por ejemplo, en el caso de que se quiera comprobar cómo reacciona un protocolo de encaminamiento al aumentar el número de saltos a realizar, puede ser que todos los nodos sean vecinos durante las pruebas y no se haga uso, por lo tanto, de las características de encaminamiento del protocolo. Problemas similares se producen con la relación entre la velocidad de los nodos y el área de simulación.

Por estos motivos, existen propuestas para la construcción de escenarios que permitan una evaluación más rigurosa de las soluciones para redes MANET realizadas por Kurkowski, Navidi y Camp (2007a), existiendo una versión más reciente de este trabajo desarrollada por Munjal, Camp y Navidi (2010). Estas investigaciones proponen la utilización de dos métricas fundamentales a la hora de determinar las características de los escenarios, y son las que han sido las seleccionadas para llevar a cabo la experimentación en esta investigación.

- **Partición media de la red.** Esta característica hace referencia al porcentaje de nodos de la red que de media se encuentran desconectados en un instante determinado. La selección de valores bajos permite garantizar que la red utilizada en la simulación está construyendo caminos de comunicación entre la mayoría de los nodos de la red. Una red en la que muchos nodos se encuentran desconectados no permitiría comprobar de forma adecuada la funcionalidad de un protocolo para redes móviles ad hoc.
- **Camino medio más corto.** Representa la distancia media más corta que puede existir entre dos nodos de la red. Determina el número medio de saltos que los mensajes deben realizar en la red para llegar de un nodo hasta otro. Configuraciones incorrectas de los parámetros de simulación pueden resultar en redes en las que los nodos están casi todos a distancia de un salto entre sí. En estos casos, las características de la solución evaluada podrían no estar siendo medidas correctamente.

Por otro lado, la decisión del modelo para la movilidad de los nodos de la red ad hoc es otro aspecto significativo de la evaluación de protocolos. Los modelos definen el comportamiento de los nodos durante la simulación, esto es, su velocidad, el tiempo que los nodos permanecen en un punto determinado hasta moverse de nuevo, si se mueven en grupo, etc. Camp, Boleng y Davies (2002) identifican algunos de estos modelos y las implicaciones que tiene el uso de cada uno de ellos. La mayoría de ellos son modelos estadísticos que se basan en la especificación de distintos parámetros de configuración. Existe, sin embargo, la posibilidad de utilizar trazas reales de movilidad capturadas de un uso real. La selección del modelo de movilidad para la experimentación depende del dominio de aplicación para el protocolo evaluado.

En la investigación presentada en esta tesis doctoral se ha optado por utilizar el modelo de movimiento aleatorio denominado *Random Waypoint*, que es un modelo sencillo basado en el movimiento de los nodos con velocidades y en direcciones aleatorias. Se ha seleccionado este modelo por su simplicidad de uso y porque se ha querido proporcionar una visión general de las características de la solución planteada sin entrar en escenarios más específicos de aplicación.

Por último, la evaluación de protocolos, por ejemplo, para el descubrimiento y composición de servicios, plantea problemas a la hora de configurar adecuadamente los experimentos. Es muy difícil encontrar estándares para la configuración de las pruebas que permitan comparar entre sí distintas propuestas de una forma sencilla, aunque existen algunas, como es el caso de la realizada por Abou El Saoud, Kunz y Mahmoud (2006) para el descubrimiento de servicios, aunque no han tenido aceptación para su uso sistemático en otras investigaciones posteriores.

## 2.3 Conclusiones

Se ha realizado una pequeña introducción a las características fundamentales de las redes MANET con el objetivo de que pueda servir para comprender la aplicabilidad y la problemática que presentan este tipo de redes. Se han resumido las características principales de la transmisión inalámbrica, el control de acceso al medio, así como conceptos fundamentales sobre los protocolos de encaminamiento usados en estas redes. Por otro lado, se ha razonado el motivo por el que se ha utilizado un simulador de red durante el desarrollo de la investigación para llevar a cabo la evaluación de la propuesta realizada.

Asimismo, el capítulo incluye una comparativa de los simuladores de red más utilizados y expone los motivos por los que se ha elegido ns-2 como la herramienta utilizada, por su posibilidad de realizar simulaciones con código real desarrollado en Java y utilizando librerías básicas de comunicación.

Por último, se han mostrado cuáles son los parámetros que afectan a la validez de una simulación, así como el conjunto de parámetros seleccionados que son utilizados en la realización de la experimentación en esta tesis doctoral.



# Estado actual del descubrimiento y composición de servicios

La investigación presentada en esta tesis se centra en dos aspectos principales: el descubrimiento y la composición de servicios, aplicados ambos a las características de las redes móviles ad hoc. Por lo tanto, la revisión bibliográfica se presenta dividida a su vez en estos dos temas principales. El objetivo de este capítulo es ofrecer una visión de las posibles alternativas existentes en la actualidad para la resolución de problemas relacionados con el descubrimiento y la composición de servicios en redes MANET, que han sido proporcionadas por trabajos previos en el área. El resultado de la revisión bibliográfica es una clasificación que puede ayudar a comprender mejor las características de estas soluciones. Durante la realización del proceso de revisión se han seleccionado los trabajos más representativos en el área y aquellos que pueden servir como ejemplo de los aspectos que se quieren destacar.

## 3.1 Descubrimiento de servicios

El descubrimiento de servicios es el proceso que permite, a los distintos participantes de una red, conocer cuáles son los servicios que se encuentran disponibles, así como obtener la información necesaria que posibilite su posterior utilización. Se pueden distinguir dos procesos fundamentales: primero, la publicación de las características de los servicios ofertados por parte de los proveedores de servicios y, segundo, las consultas de búsqueda realizadas por los

clientes potenciales. La distinción entre *servidores* y *clientes* de servicios suele ser difusa en el caso de las redes MANET, ya que es muy usual que los nodos que constituyen la red lleven a cabo las dos funciones de manera simultánea, convirtiéndose en lo que en la literatura relacionada suele denominarse *prosumer*. En las secciones siguientes se presenta una clasificación, según diferentes aspectos, de las soluciones existentes en la actualidad para el descubrimiento.

### 3.1.1 Redes con infraestructura

Antes de presentar las soluciones para el descubrimiento de servicios que han sido propuestas específicamente para su aplicación en redes MANET, se realiza un breve resumen de aquellas otras que fueron elaboradas para redes de dispositivos con una infraestructura fija. Estas propuestas no son aplicables a las redes ad hoc ya que no tienen en cuenta sus características fundamentales, presentadas en el capítulo 2: la movilidad de los nodos, la posibilidad de comunicación mediante múltiples saltos, los cambios en la topología de la red y la inexistencia de un mecanismo de comunicación fiable entre los dispositivos. Sin embargo, estas propuestas fueron adoptadas como estándar para el descubrimiento de servicios y definieron aspectos básicos del proceso que también han sido utilizados posteriormente en las redes móviles ad hoc.

*Universal Plug and Play* (UPnP) (UPnP Forum, 2008) es un estándar cuya finalidad es permitir la comunicación entre diferentes dispositivos reduciendo al máximo la configuración necesaria. La arquitectura de UPnP está orientada totalmente hacia el concepto de *dispositivo* y de *servicio*. Un dispositivo expone una serie de servicios que pueden ser descubiertos y utilizados. Cada dispositivo puede a su vez estar compuesto por otros, siendo el dispositivo principal conocido como *dispositivo raíz*. Estos servicios pueden ser descubiertos e invocados desde los *puntos de control*. Cuando un nuevo dispositivo es añadido a la red envía una serie de mensajes *multicast* informando de los dispositivos y servicios que proporciona. Por otro lado, los puntos de control pueden buscar activamente los servicios y dispositivos ofertados en la red mediante el empleo de mensajes *multicast*. UPnP permite además suscribirse a un sistema de notificaciones que avisa de cambios en variables de estado internas a cada servicio. Proporciona, además, auto-configuración de los dispositivos para la obtención automática de direcciones IP, incluso sin un servicio DHCP disponible en la red, y la comunicación entre dispositivos se realiza mediante el envío de mensajes HTTP. Cada dispositivo se describe mediante un documento en formato XML que contiene, entre otras aspectos, información sobre el dispositivo y su fabricante, así como sobre todos los dispositivos que lo componen, una URL de presentación y control del dispositivo y la lista completa de los servicios que proporciona. Para cada uno de los servicios se incluye una lista con sus acciones, así como los parámetros que necesita cada una de ellas para poder ser

utilizada correctamente. Además, en cada servicio se puede incluir una lista de variables que modelan el estado del servicio en ejecución.

*Service Discovery Protocol* (SDP) (Bluetooth Special Interest Group, 2004) es parte de la arquitectura de Bluetooth y permite buscar servicios en una red basada en dicha tecnología. Su aplicación está limitada a la comunicación con dispositivos vecinos. Es un protocolo muy simple ya que está pensado para ser utilizado en dispositivos con poca capacidad. Al contrario que otros protocolos de descubrimiento, SDP no proporciona ningún mecanismo para acceder a los servicios encontrados ni tampoco características avanzadas como notificación. Existen dos entidades diferenciadas en este protocolo: el cliente y el servidor SDP. Cuando una aplicación quiere buscar un servicio pide a su cliente que interroge a los servidores SDP para obtener aquellos servicios que reúnen las características buscadas. Los servidores SDP mantienen una lista de *registros* de servicios, conteniendo cada uno información sobre un servicio particular. Los clientes SDP pueden recuperar información sobre un servicio enviando una petición específica al servidor y que contenga los identificadores de los servicios requeridos. Además, SDP proporciona un mecanismo para buscar servicios en una red Bluetooth mediante el uso de atributos con un determinado valor. Como respuesta, el servidor SDP devuelve aquellos registros de servicio que se corresponden con los identificadores indicados en el patrón de búsqueda.

*Service Location Protocol* (SLP) (Guttman y Perkins, 1999) es un protocolo de descubrimiento de servicios propuesto por el IETF. Los elementos básicos de SLP son el *agente de usuario*, el *directorio de servicios* y el *agente de servicios*. El agente de usuario se encuentra en el lado del cliente y realiza la búsqueda de servicios. Por su parte, el agente de servicios es utilizado por las aplicaciones para exponer los servicios que ofrecen a la red. Cuando un agente de usuario quiere buscar un servicio puede enviar sus peticiones a todos los agentes de servicios presentes en la red utilizando *multicast*, y estos devolverán las respuestas mediante envío *unicast*. En redes grandes se puede utilizar más de un directorio para gestionar el registro de los servicios ofertados. La finalidad de estos directorios de servicios es registrar los servicios expuestos por diferentes agentes, reduciendo así el envío de paquetes. En presencia de un directorio de servicios, la búsqueda se realiza enviando mensajes de forma directa a los directorios. SLP permite agrupar los servicios mediante el uso de ámbitos que son cadenas de texto que identifican servicios que tienen algo en común: localización, administrador, precio del servicio, fiabilidad, etc. La búsqueda de servicios en SLP se puede realizar tanto a través del tipo del servicio como mediante un patrón de búsqueda de atributos mediante LDAP. Por último, todos los directorios y agentes de servicios tienen un identificador de ámbito asignado de tal forma que un directorio solo puede registrar servicios de los agentes de servicio que pertenezcan a su mismo ámbito.

*Salutation* (Salutation Consortium Inc., 1998) es un protocolo para el descubrimiento de servicios, dispositivos y aplicaciones mediante la publicación de sus capacidades que se expresan como conjuntos de atributos. Estas descripciones son gestionadas de forma local en cada dispositivo mediante un gestor de servicios. Define también un protocolo para la comunicación entre los gestores locales de cada dispositivo con el fin de permitir el descubrimiento de servicios proporcionados por dispositivos remotos. La disponibilidad de los servicios descubiertos puede ser comprobada mediante la realización de consultas periódicas. Una de las características más importantes de este protocolo de descubrimiento es su independencia de la tecnología de comunicación subyacente.

*Jini* (Sun Microsystems, 2003) es una arquitectura para el descubrimiento de servicios en dispositivos que utilicen la tecnología Java. Se fundamenta en la utilización de uno o varios elementos centralizadores que registran los servicios disponibles en los diferentes dispositivos. Los directorios de servicios permiten que los proveedores lleven a cabo el registro de los servicios que proporcionan, y que los clientes puedan conocer dónde llevar a cabo la búsqueda de servicios. Para ello, los diferentes dispositivos de la red realizan primero un descubrimiento de los directorios de servicios disponibles. Los servicios son registrados en los directorios por sus proveedores mediante conjuntos de atributos que los describen. Cada una de estas descripciones de servicio está acompañada de un *proxy* Java que permite la invocación remota del servicio utilizando RMI, una vez descargada su descripción. Las solicitudes de búsqueda de servicios pueden incluir el identificador del servicio buscado y otros atributos que lo describan. El protocolo Jini proporciona, además, mecanismos para suscribirse a eventos en los servicios y ser notificado de los cambios que se producen.

### 3.1.2 Clasificación de las soluciones

El problema del descubrimiento de servicios en redes móviles ad hoc ha sido objeto de numerosos trabajos de investigación que han tratado de adecuar las características de las soluciones tradicionales, presentadas anteriormente, a este tipo de redes. Las soluciones propuestas pueden clasificarse de acuerdo a los diferentes elementos que constituyen el proceso de descubrimiento de servicios y las técnicas que se utilizan para llevarlo a cabo. Así, Ververidis y Polyzos (2008) proponen la utilización de los siguientes aspectos para su clasificación:

- **Uso de directorios:** se distinguen dos tipos principales de propuestas, aquellas que hacen uso de directorios de servicios, que centralizan la información referente a los servicios ofertados en la red ad hoc en distintos nodos y, por otro lado, las soluciones que se basan en la disseminación de la información a través de los nodos de la red sin la existencia de que algunos sean designados específicamente como directorios de servicios.

- **Descripción de servicios:** la tecnología utilizada para la construcción de las publicaciones de los servicios ofertados, así como de las solicitudes de búsqueda, determina cómo se establece la compatibilidad entre ambas. Existen soluciones *sintácticas*, basadas en el uso de parejas *clave-valor* o expresiones regulares, y *semánticas*, que hacen uso de razonamientos más avanzados como la aplicación de ontologías.
- **Modo de operación:** las soluciones pueden funcionar en modo *reactivo* o *proactivo*, dependiendo de si los procesos relacionados con el descubrimiento de servicios se activan solamente en respuesta a una solicitud de usuario o de forma autónoma, respectivamente.
- **Respuesta a la movilidad:** debido a la movilidad y dinamismo de los dispositivos que constituyen la red ad hoc, es muy probable que los servicios ofertados cambien durante el tiempo de vida de la red o, simplemente, dejen de ser accesibles. Las soluciones propuestas para el descubrimiento de servicios suelen ofrecer dos mecanismos para hacer frente a este problema: el *sondeo periódico (polling)* y la utilización de *notificaciones*.
- **Integración de protocolos:** el proceso de descubrimiento de servicios puede llevarse a cabo de forma integrada con la capa de encaminamiento de mensajes de la red ad hoc o de una forma independiente.
- **Mecanismo de selección:** una solicitud de búsqueda puede resultar en varias respuestas de diferentes servicios compatibles con la misma. La selección de servicios se aplica con el objeto de filtrar entre todas las soluciones encontradas y obtener aquella que es más adecuada a las necesidades del cliente solicitante.
- **Seguridad y privacidad:** la naturaleza completamente distribuida y dinámica de una red MANET plantea numerosos problemas de seguridad durante el descubrimiento de servicios: servicios maliciosos, uso no autorizado de servicios, control de la información transmitida, etc.

En el estudio bibliográfico realizado sobre las propuestas para el descubrimiento de servicios en redes MANET, se han tenido en cuenta únicamente aquellos aspectos directamente relacionados con la propuesta central a esta investigación. Así, se omiten de dicho estudio los aspectos referentes a la seguridad y privacidad porque quedan fuera del ámbito de la investigación. Además, el protocolo de descubrimiento explicado en el capítulo 5 no aplica ningún proceso de selección de servicios tras llevar a cabo su descubrimiento. Sin embargo, en este caso, sí que se incluye una pequeña introducción a dicho aspecto porque podría dar lugar, como una línea de trabajo futuro, a su uso en combinación con la propuesta aquí presentada.

En las siguientes secciones se realiza una clasificación de las soluciones para el descubrimiento de servicios en redes MANET existentes en la actualidad y que, además, se ha determinado que son más relevantes para la investigación.

### 3.1.3 Uso de directorios

Como se explica en la sección 3.1.1, desde las primeras soluciones para el descubrimiento de servicios se ha hecho uso de directorios donde los proveedores pueden registrar sus servicios y los clientes realizar búsquedas para obtener las descripciones. Normalmente, estos directorios de servicios son designados por un administrador en una o varias máquinas dedicadas, según las necesidades de la red de dispositivos. Sin embargo, las características de las redes MANET hace que la aplicación de estas soluciones no sea viable. Primero, en este tipo de redes los directorios de servicios deben estar localizados en dispositivos que forman parte de la red ad hoc, ya que no es posible suponer la existencia de otros elementos de infraestructura que lleven a cabo estos procesos. Además, las rutas de comunicación con los directorios de servicios pueden variar durante la movilidad de los nodos que la constituyen, dando como resultado que dichos directorios puedan volverse inaccesibles. Por otro lado, las rutas de comunicación no suelen ser directas y requieren la realización de múltiples saltos a través de los enlaces entre nodos de la red.

Se hace necesario, por lo tanto, un mecanismo que lleve a cabo la selección, de forma dinámica y reaccionando ante los cambios en la red, de aquellos nodos más adecuados para funcionar como directorios de servicios. Además, debido a la naturaleza distribuida de la red ad hoc, estos directorios se suelen comunicar entre sí con el objeto de intercambiar información y hacer accesible a las distintas zonas de la red las descripciones de servicios que contienen (Zhu, Mutka y Ni, 2003). Por ejemplo, Kozat y Tassiulas (2003) proponen que los nodos que se constituyen en directorios sean seleccionados mediante la construcción de un *conjunto mínimo dominante* entre todos los nodos de la red. En teoría de grafos, se define el *conjunto mínimo dominante* como aquel que está constituido por aquellos vértices del grafo tales que, el resto de vértices que no pertenecen al conjunto están unidos, al menos, a un miembro de dicho conjunto.

Una solución similar al uso del *conjunto mínimo* de nodos es propuesta por Sailhan y Issarny (2005), donde la selección de los directorios se realiza también con la intención de que todos los miembros de la red se encuentren cubiertos por al menos otro. Sin embargo, en este caso, los directorios intercambian información sobre su contenido de una manera resumida mediante el empleo de filtros de Bloom (Bloom, 1970). Además, las solicitudes de búsqueda son dirigidas por los directorios hacia aquellos otros que contienen el servicio buscado, de acuerdo a la información contenida en los filtros. El proceso de selección de los nodos que forman parte del conjunto de directorios puede realizarse me-

diante el empleo de un algoritmo que tenga en cuenta aspectos muy variados como, por ejemplo, la velocidad del nodo, su energía restante, el número de sus vecinos, sus capacidades de procesamiento o una combinación de éstas, como es el caso del trabajo de Al Mallah y Quintero (2009).

Existe otra posibilidad a la hora de utilizar directorios de servicios y es mediante la creación de agrupaciones de nodos que tienen alguna relación entre ellos. Además, como en el caso de Klein y König-Ries (2002), estas relaciones pueden establecerse en varios niveles, creando así agrupaciones jerárquicas. Un ejemplo de esto es realizado por Klein, König-Ries y Obreiter (2003), donde los autores proponen la creación de *anillos* de proveedores de servicios entre aquellos que se encuentren físicamente cerca o que provean servicios semánticamente relacionados. Cada agrupación tiene un punto de acceso en un nodo seleccionado mediante un algoritmo, a través del cual se dirigen las solicitudes de búsqueda, existiendo agrupaciones de anillos de mayor nivel. El agrupamiento de nodos relacionados, esta vez únicamente por cercanía, se utiliza también en varias soluciones (Seno, Budiarto y Wan, 2007; Artail, Safa, Hamze y Mershad, 2007; Artail, Hamze, Mershad y Safa, 2011; Wister y Torres, 2009).

Siguiendo con la agrupación de directorios, Schiele, Becker y Rothermel (2004) aplican este concepto para reducir la energía consumida, de tal forma que solamente los puntos de acceso se encuentran disponibles en todo momento. El resto de nodos que forma parte de un grupo están *dormidos* y se despiertan periódicamente para responder a las solicitudes de búsqueda enviadas por el punto de acceso.

El proceso de creación de los agrupamientos y la selección de directorios puede combinarse dando lugar a distintas regiones que se comunican entre sí a alto nivel, como proponen Deepa y Swamynathan (2010). Estas regiones pueden crearse de acuerdo a las características geográficas de los nodos utilizando información de localización proporcionada por un receptor GPS incluido en el nodo, según el trabajo de Hwang, Talipov y Cha (2011).

Además de la utilización de los filtros de Bloom comentada anteriormente, la distribución de la información entre los diferentes nodos de la red ad hoc puede conseguirse mediante la utilización de *tablas de hash distribuidas* (*Distributed Hash Tables - DHT*) en diferentes nodos seleccionados de la red. En este caso, los directorios son actualizados con la información propagada desde otros nodos y las solicitudes se dirigen mediante la utilización de las claves distribuidas. Se evita así la replicación innecesaria de la información en todos los directorios de la red (Seada y Helmy, 2004; Pirrò, Talia y Trunfio, 2012). Por su parte, Sivavakeesar, Gonzalez y Pavlou (2006) aplican esta misma solución pero mediante la división de la red en diferentes regiones geográficas, utilizando nuevamente para ello la información de localización.

Por último, los fallos en los directorios designados pueden resolverse mediante la elección incremental de nuevos directorios, tal y como llevan a cabo

Raychoudhury, Cao, Wu, Lai, Chen y Ma (2011). Aunque no se produzca el fallo total de un directorio, en el trabajo de Cui, Li y Jiang (2011) se muestra que es posible llevar a cabo la migración del contenido del mismo a un nodo que disponga de mejores recursos computacionales para mantenerlo.

### 3.1.4 Difusión en la red

Al contrario que las soluciones basadas en el uso de directorios, este tipo de propuestas se fundamenta en la difusión de las publicaciones y de las solicitudes de búsqueda a través de los nodos de la red. Se evita así la necesidad de centralizar la información de los servicios en un conjunto de directorios seleccionados. Los desafíos en este tipo de propuestas están relacionados, por un lado, con las frecuencias de difusión y, por lo tanto, de la información mantenida sobre las publicaciones. Por otro lado, se encuentran relacionados con las distancias a las que los mensajes de publicación y solicitud son propagados y cómo se realiza esta propagación.

Una de las primeras propuestas de este tipo de soluciones es realizada por Nidd (2001), donde la diseminación de la información sobre los servicios proporcionados por un nodo se lleva a cabo salto a salto a través de la red ad hoc. Cada nodo propaga a sus vecinos, que se encuentran a distancia de un salto en la red, la información sobre sus servicios locales y sobre aquellos que fueron recibidos, a su vez, desde sus propios vecinos. La frecuencia con la que la información sobre servicios es propagada por la red es controlada por un *decaimiento exponencial*. Una solución similar a la anterior puede encontrarse en el trabajo de Campo, Munoz, Perea, Mann y Garcia-Rubio (2005), donde se tiene en cuenta la gestión de las capacidades de los dispositivos priorizando las respuestas desde aquellos menos limitados.

La gestión de la distancia a la que las publicaciones de servicios son difundidas a través de la red es controlado mediante el uso de contadores de *tiempo de vida* contenidos en los mensajes y que son decrementados en cada salto (Ratsimor, Chakraborty, Joshi y Finin, 2002; Chakraborty, Joshi, Yesha y Finin, 2006). En estas soluciones los nodos forman grupos que cachean la información proporcionada por otros nodos de la red, de tal forma que, eventualmente, cualquier nodo de la red puede ser capaz de responder a cualquier búsqueda de servicio. Otra solución, propuesta por Meier, Cahill, Nedos y Clarke (2005), es que los nodos propaguen información sobre los servicios que proporcionan a cierta zona de la red, cuando un cliente compatible entra en dicha zona es notificado con aquellos servicios compatibles con sus requerimientos.

Por otro lado, Crespo y Garcia-Molina (2002) proponen como técnica para controlar la propagación de las solicitudes de búsqueda la utilización del *reenvío selectivo*, donde los nodos que reciben una petición determinan hacia qué vecino concreto deben reenviar dicha petición. Esta solución es utilizada tam-



bién en varios trabajos posteriores (Ratsimor, Chakraborty, Joshi y Finin, 2002; Chakraborty, Joshi, Yesha y Finin, 2006; Gao, Wang, Yang y Yang, 2006; Ververidis y Polyzos, 2009) con el fin de reducir el número de mensajes propagados. Para dirigir los mensajes hacia los proveedores es posible utilizar información geográfica a partir de la localización de los nodos, tal y como llevan a cabo Weng, Wen, Jin y Zhang (2009).

Un enfoque completamente distinto para el descubrimiento de servicios ha sido realizado por Lenders, May y Plattner (2005), donde los autores proponen un mecanismo para el descubrimiento de servicios que aplica un modelo similar a la teoría del campo electrostático. En dicho trabajo los servicios son modelados como cargas eléctricas y los mensajes de búsqueda son dirigidos hacia ellos mediante un mecanismo similar a la fuerza de atracción entre cargas.

Existen soluciones en las que son los nodos intermedios los que responden a las solicitudes de búsqueda. En el trabajo de Palmieri (2012) cualquier nodo de la red, al recibir las publicaciones de los proveedores de los servicios, las almacena y puede devolverlas como respuesta a un cliente que realiza una búsqueda, si existe compatibilidad. Se evita así la propagación de las peticiones de búsqueda hasta los proveedores de servicios, tal y como también proponen otros trabajos posteriores (Motegi, Yoshihara y Horiuchi, 2002; Kang, Im y Kim, 2008). Un enfoque similar es utilizado por Tchakarov y Vaidya (2004), donde la propagación de las publicaciones de servicios crea rutas inversas que apuntan hacia los proveedores, mientras que los nodos intermedios almacenan cierta información sobre los servicios diseminados. Así, cuando una solicitud de búsqueda se encuentra en la intersección con uno de estos caminos es propagada directamente hacia el proveedor del servicio siguiendo dicha ruta.

### 3.1.5 Descripción de servicios

La tecnología utilizada para la descripción de los servicios determina la expresividad que puede tener una solicitud de búsqueda y limita en cierta forma la posibilidad de obtener los resultados adecuados. Los servicios pueden ser clasificados de acuerdo a un identificador universal único (UUID) que permite la localización de aquellos elementos cuyo identificador es contenido exactamente en la petición. La utilización de un UUID requiere una categorización previa para clasificar los diferentes servicios y asignarles los códigos representativos. Esta solución es utilizada, por ejemplo, en el protocolo SDP que ha sido comentado anteriormente en la sección 3.1.1.

El uso de identificadores universales es adecuado para aquellos mecanismos de comunicación de capacidad reducida porque el tamaño de estos identificadores puede estar limitado a unos pocos *bytes* (Ververidis y Polyzos, 2009; Arias-Torres y García-Macías, 2005; Outay, Kaisser, Veque y Bouallegue, 2010). Sin embargo, este enfoque no es lo suficientemente flexible ya que requiere el

conocimiento *a priori* de estos identificadores, tanto por parte de los proveedores como de los clientes.

Un enfoque que permite una mayor expresividad a la hora de realizar las búsquedas de servicio se encuentra en la utilización de parejas *clave-valor* para la descripción de los atributos del servicio, como por ejemplo en el trabajo de Serhani y Gadallah (2010). Esta solución ha sido utilizado en UPnP (UPnP Forum, 2008) y, primeramente, en los directorios de servicios UDDI. Las descripciones de los atributos de los servicios pueden expresarse en cualquier formato de texto, si bien, en la mayoría de los casos, se utiliza XML o lenguajes para la descripción de servicios web como WSDL (Noh y Shin, 2007).

Como se explica en el trabajo de Paolucci, Kawamura, Payne y Sycara (2002), o en el de Aguilera, Abaitua, Diaz, Bujan y Lopez de Ipina (2007), las posibilidades del enfoque anterior son limitadas. La búsqueda sintáctica basada en la comparación de cadenas o en la utilización de expresiones regulares no es adecuada en la mayoría de los casos porque únicamente permite el uso de palabras clave combinada con un sistema de clasificación. Frecuentemente, este tipo de búsqueda no devuelve ningún resultado porque no es capaz de reconocer las similitudes y diferencias entre los conceptos empleados.

Por la razón anteriormente comentada, existen soluciones que utilizan las posibilidades que ofrecen tecnologías semánticas como OWL (W3C OWL Working Group, 2009) para la descripción de los servicios y las consultas de búsqueda. Así, en varios trabajos (Chakraborty, Joshi, Yesha y Finin, 2006; Klein, König-Ries y Obreiter, 2003) se utilizan ontologías para la clasificación de los servicios y su emparejamiento con las búsquedas realizadas. Este mismo enfoque para la clasificación de los servicios ha sido seguido posteriormente por otras propuestas (Ruta, Zacheo, Grieco, Di Noia, Boggia, Tinelli, Camarda y Di Sciascio, 2010; Said y Mahéo, 2008; Islam y Shaikh, 2012).

Obviamente, la utilización de una ontología supone que los distintos nodos de la red tienen conocimiento de la misma para poder aplicarla, y por lo tanto, debe ser compartida. Sin embargo, la utilización de una ontología posibilita llevar a cabo procesos que permiten una mayor expresividad y que no son posibles con el uso de consultas sintácticas.

### 3.1.6 Modo de operación

Los protocolos de descubrimiento de servicios pueden ser clasificados, de acuerdo a cómo los clientes o directorios adquieren la información sobre los servicios disponibles, en *proactivos* y *reactivos*, de una forma muy parecida a cómo se clasifican los protocolos de encaminamiento de la sección 2.1.3.

En el enfoque *proactivo* la propagación de la información es llevada a cabo de forma activa por el protocolo, sin la necesidad de responder a las acciones que pueda producirse en el entorno. En el caso del descubrimiento de servicios,

el enfoque proactivo se aplica a la publicación de las descripciones de servicios a través de la red o en los directorios. La información sobre los servicios se diseña de forma periódica con objeto de mantener la consistencia y la precisión a pesar de los cambios que puedan producirse debido a la movilidad de la red ad hoc (Arias-Torres y García-Macías, 2005; Nedos, Singh y Clarke, 2005).

En el caso de un protocolo *reactivo*, la información sobre los servicios es obtenida únicamente cuando los clientes realizan una petición de búsqueda y ésta es procesada por los proveedores o los directorios de servicios. Para hacer frente a los cambios que supone la movilidad de la red, el proceso de descubrimiento por parte de un cliente debe producirse de forma periódica (Klein, König-Ries y Obreiter, 2003; Seada y Helmy, 2004; Chakraborty, Yesha y Joshi, 2004; Artail, Safa, Hamze y Mershad, 2007; Deepa y Swamynathan, 2010; Cui, Li y Jiang, 2011).

Como es de suponer, cada modo de operación tiene sus ventajas y desventajas, y el trabajo de Mohan, Almeroth y Belding-Royer (2004) presenta una comparación entre ambos enfoques. Cuando aumenta el número de servidores, el enfoque proactivo tiene una mayor latencia para el descubrimiento de servicios e incrementa la congestión en la red. Por el contrario, el enfoque reactivo se comporta peor cuando crece el número de clientes. Tanto en los protocolos reactivos como proactivos es necesario tener en cuenta la periodicidad con la que los clientes o los servidores realizan las búsquedas o publican las descripciones de los servicios, respectivamente. Además, en ambos casos, al aumentar la periodicidad de envío de mensajes, se produce un mayor envío de mensajes y la posible congestión de la red ad hoc.

Por estos motivos, existen soluciones que combinan los modos de operación proactivo y reactivo para el descubrimiento de servicios en una misma solución, dando lugar a los llamados enfoques *híbridos*. Existen distintas propuestas (Herral, Desai, Verma y Lee, 2003; Macker y Taylor, 2011) que permiten seleccionar el modo de operación de la red dependiendo del tamaño de la red o de otros factores, como pueden ser el número de servicios o servidores esperados.

En el caso de que el descubrimiento se lleve a cabo dividido en zonas, es posible aplicar un modo de operación a cada una de ellas. Por ejemplo, Noh y Shin (2007) utilizan la diseminación proactiva de la información sobre servicios dentro de una zona, mientras que el enfoque reactivo se utiliza para el descubrimiento de servicios fuera de dicha zona. En el trabajo de Seno, Budiarto y Wan (2007) y en el de Liao y Grigoras (2009) se sigue también una propuesta similar utilizando diferentes modos de operación para cada una de las zonas, que se basan en la extensión del protocolo de encaminamiento ZRP, propuesto originalmente por Haas, Pearlman y Samar (2002).

### 3.1.7 Respuesta a la movilidad

Los nodos de la red ad hoc pueden cambiar sus posiciones relativas durante el tiempo de vida de la red; esto significa que los servicios descubiertos pueden dejar de ser accesibles. En la literatura científica existen diferentes soluciones a la hora de hacer frente al dinamismo de la red.

Una solución es que los clientes puedan llevar a cabo un proceso periódico de envío de solicitudes de búsqueda para comprobar continuamente mediante un *sondeo activo* si el servicio descubierto sigue siendo utilizable o ha sufrido variaciones en su estado, como aplican Varshavsky, Reid y de Lara (2005). Otra opción es la utilización de *notificaciones* enviadas por los proveedores de servicios para indicar que un servicio ha dejado de estar disponible o ha cambiado su estado. Esta última solución es aplicada por Dabrowski, Mills y Elder (2002). En este mismo trabajo se prueba además que, si bien el uso de notificaciones es menos eficiente respecto al número de mensajes utilizados que la aplicación del sondeo periódico, es posible obtener una respuesta más inmediata a los cambios que se producen en los servicios de la red.

Por otro lado, el aumento de la frecuencia en el caso de la búsqueda periódica de servicios tiene el problema de que, aunque disminuye la posibilidad de trabajar con información obsoleta, puede tener un impacto considerable en la congestión de la red si el número de clientes es elevado. Por último, en las redes que mantienen estructuras entre los nodos, por ejemplo para la creación de agrupaciones de directorios u otras configuraciones, es necesario un procedimiento que actualice dicha información de forma continua frente a los cambios de la red (Klein, Konig-Ries y Obreiter, 2003; Seno, Budiarto y Wan, 2007; Artail, Safa, Hamze y Mershad, 2007).

### 3.1.8 Integración de protocolos

El envío de los mensajes de descubrimiento puede integrarse como parte del mecanismo de obtención de rutas y encaminamiento con el fin de reducir el número de mensajes enviados a través de la red y de permitir, además, conocer los cambios producidos en su topología.

Por ejemplo, en varias propuestas (García-Macias y Torres, 2005; Obaid, Khir y Mili, 2007; Arias-Torres y García-Macias, 2005) se extiende el protocolo de encaminamiento reactivo AODV para incluir información sobre los servicios de la red y posibilitar su descubrimiento. Esta solución también puede aplicarse para la integración del descubrimiento de servicios en protocolos de encaminamiento proactivos como puede ser OLSR (Heni y Bouallegue, 2011).

La integración puede llevarse a cabo también en el caso de protocolos de descubrimiento híbridos, como proponen Noh y Shin (2007), existiendo incluso la posibilidad de combinar múltiples protocolos de encaminamiento para diferentes zonas de la red. Se puede, por ejemplo, utilizar OLSR o DSDV para

la comunicación dentro de una zona, donde los nodos mantienen de manera proactiva información sobre los servicios que hay en ella, y DSR o AODV para la comunicación reactiva entre diferentes zonas de la red, extendiendo, por ejemplo, el protocolo de encaminamiento ZRP (Ververidis y Polyzos, 2009; Liao y Grigoras, 2009). Por su parte, Outay, Kaisser, Veque y Bouallegue (2010) añaden, además de la integración entre el protocolo ZRP y el descubrimiento de servicios, la utilización de filtros de Bloom para la compartición de información entre las diferentes zonas de la red ad hoc.

### 3.1.9 Mecanismo de selección

La selección es el proceso que se inicia cuando un cliente recibe varias respuestas a una misma solicitud de búsqueda de servicios y necesita escoger aquella que mejor se adecúe a sus necesidades, de acuerdo a Tyan y Mahmoud (2004). Por otro lado, Ververidis y Polyzos (2008) clasifican la selección en *asistida* o *automática*, dependiendo de si ésta es realizada con ayuda de la intervención del usuario o no, respectivamente.

En el caso de la selección automática se puede llevar a cabo la clasificación de los servicios recibidos de acuerdo a algún criterio, como puede ser el número de saltos en la red u otro aspecto del servicio. En algunas ocasiones el proceso de selección de servicio es parte del propio proceso de búsqueda y el filtrado se realiza antes de llevar la respuesta al usuario. Así, Lenders, May y Plattner (2005) tienen en cuenta el número de saltos y las capacidades del servicio buscado para determinar si debe ser devuelto como una respuesta al cliente.

Casi toda la bibliografía consultada sobre descubrimiento de servicios en redes ad hoc no tiene en cuenta la selección de servicios en la formulación de sus propuestas y se centran en la definición del descubrimiento a través de la red. Como se ha explicado anteriormente, este aspecto tampoco ha sido objeto de estudio en la investigación presentada en esta tesis, dejando este problema como una posible línea de investigación futura.

### 3.1.10 Conclusiones

Se ha presentado una revisión bibliográfica de la situación actual en el descubrimiento de servicios en redes móviles ad hoc que ha permitido conocer y categorizar las diferentes decisiones que pueden tomarse para el desarrollo de soluciones de estas características.

Con respecto a la selección de una arquitectura basada en la utilización de directorios o, por el contrario, en la difusión de la información, no existe un consenso claro que permita determinar si un enfoque es mejor que otro. Ververidis y Polyzos (2008) exponen que el motivo principal para esta indecisión es que existen muchos factores a tener en cuenta cuando se evalúa un protocolo

para una red MANET: movilidad de los nodos, densidad de servidores y clientes, la frecuencia de actualizaciones, etc. Así, en el caso de una red con alta movilidad de los nodos, puede ser preferible un enfoque sin directorios para evitar los problemas asociados con la necesidad de realizar una actualización constante y, de esta manera, reducir la posibilidad de que contengan información obsoleta. Sin embargo, si en la misma red se aumenta el número de peticiones de búsqueda de servicio, el uso de directorios puede empezar a suponer una ventaja al centralizar las peticiones hacia unos ciertos nodos y reducir el número de mensajes transmitidos a través de toda la red.

A pesar de estas cuestiones, Engelstad y Zheng (2005) concluyen que la utilización de un enfoque sin directorios puede ser mejor que un enfoque en el que estos sean usados. En el protocolo de descubrimiento propuesto en el capítulo 5 no se utilizan directorios y, además, el descubrimiento se basa en la propagación de la información sobre los servicios salto a salto a través de los nodos de la red ad hoc. En dicho capítulo se exponen los motivos para la elección del enfoque utilizado y cuáles son las consecuencias de su utilización.

Sobre la tecnología utilizada para la descripción de los servicios se ha comprobado que la mayoría de las propuestas estudiadas no tienen en cuenta este aspecto. Solamente algunas de ellas proporcionan información específica sobre la utilización de una descripción basada en UUID, en atributos o semántica. Sin embargo, es en este punto donde la introducción de la tecnología semántica en el proceso de descubrimiento de servicios puede proporcionar otros beneficios. Esto es así no solamente por el hecho de, como ya se ha visto en el estudio bibliográfico, permitir una mayor expresividad, sino también porque como se expone en el capítulo 5, la utilización de relaciones entre los conceptos permite aplicar una serie de mejoras en el proceso de descubrimiento orientadas a la reducción de los mensajes enviados durante los procesos de publicación y búsqueda de servicios.

Por otro lado, de acuerdo a Mohan, Almeroth y Belding-Royer (2004), donde se realiza una comparación entre el enfoque proactivo y reactivo para la disseminación de la información, las soluciones proactivas tienen un mejor rendimiento cuando existe un mayor número de clientes que de proveedores de servicios y, además, el número de peticiones de búsqueda no es muy elevado. En dicho caso, el coste de mantener la información sobre los servicios ofertados puede resultar en una disminución de la latencia de búsqueda. En la propuesta para el protocolo de descubrimiento de servicios realizada en el capítulo 5 se utiliza un enfoque proactivo para la disseminación de la información sobre los servicios en la red MANET.

Por último, el protocolo para el descubrimiento de servicios propuesto en esta investigación se encuentra integrado con un mecanismo para el encaminamiento de mensajes a través de la red MANET que es también de naturaleza proactiva. Esta integración permite el uso de notificaciones para la comunica-

ción de los cambios que puedan producirse en los servicios y rutas descubiertas y, por otro lado, posibilita una reducción de los mensajes necesarios para gestionar ambos aspectos. En el capítulo 5 se proporciona más información acerca de esta integración y de las características del protocolo de encaminamiento.

## 3.2 Composición de servicios

La composición de servicios se define como el proceso mediante el cual se lleva a cabo la creación de *servicios complejos* a partir de otros más simples y sencillos, según la definición dada por Chakraborty, Joshi, Finin y Yesha (2005). El concepto de crear nuevas funcionalidades a partir de otros componentes no es algo nuevo y es la base de la ingeniería del software. Sin embargo, la composición de servicios en su visión más general, propone mecanismos que posibilitan resolver las necesidades de los usuarios, frente a la solicitud de servicios no existentes y de una forma dinámica. En estos casos, en los cuales no existe un servicio que satisfaga directamente las necesidades de un cliente, usuario o aplicación, la respuesta a una búsqueda de un servicio concreto quizá pueda obtenerse mediante la composición de otros servicios disponibles.

Al igual que en las soluciones para el descubrimiento de servicios presentadas en la sección 3.1, en el caso de la composición se supone que existen dispositivos que proveen ciertos servicios en la red. Dichos servicios pueden ser usados para crear los servicios compuestos de acuerdo a las solicitudes de otros dispositivos que actúan como clientes.

### 3.2.1 Clasificación de las soluciones

La composición de servicios es un problema que ha recibido una gran atención por parte de la comunidad científica en los últimos años, principalmente en su aplicación a las redes con infraestructura. Las soluciones propuestas pueden clasificarse según los diferentes elementos que definen el proceso de la composición de servicios, siendo varios de estos aspectos comunes tanto a las soluciones para redes con infraestructura como a las redes móviles ad hoc. Así, en la literatura revisada sobre la composición de servicios se distinguen los siguientes aspectos fundamentales para su clasificación (Urbietta, Barrutieta, Parra y Uribarren, 2008; Brønsted, Hansen y Ingstrup, 2010):

- **Arquitectura:** las soluciones para la búsqueda de composiciones pueden estar *centralizadas* o *distribuidas*, siendo el segundo tipo más adecuado para las características de las redes MANET, ya que no depende de un proceso que se base en la existencia de uno o varios repositorios de servicios.

- **Mecanismo de composición:** el proceso de composición utilizado para encontrar una solución a una petición de composición define las aplicaciones y limitaciones de cada propuesta, y determina si puede ser usada de una forma distribuida en una red de dispositivos.
- **Especificación de la composición:** la solicitud de la composición a obtener puede llevarse a cabo de una forma explícita y a bajo nivel mediante la utilización de un *flujo de trabajo* o, por el contrario, mediante la definición de una forma implícita del *objetivo* a cumplir.
- **Descripción de servicios:** los servicios simples ofertados por los dispositivos de la red deben ser descritos mediante un lenguaje que posibilite el proceso de composición. Además, también es necesario el uso de otro lenguaje para especificar la solicitud de composición a resolver. Existen dos tipos principales de propuestas para la descripción de los servicios y las solicitudes de composición: *sintácticas* y *semánticas*.
- **Participación del usuario:** las propuestas se pueden clasificar de acuerdo a la participación del usuario en el proceso de composición. Se distinguen así tres modos de participación: *manual*, *semi-automático* y *completamente automático*.
- **Ejecución:** una vez que se ha localizado una composición de servicios compatible con una solicitud dada puede llevarse a cabo su ejecución. Existen dos técnicas principales para la ejecución de los servicios compuestos: *orquestración* y *coreografía*.
- **Calidad de servicio:** durante el proceso de composición de servicios pueden existir varias soluciones posibles para una misma solicitud. La aplicación de técnicas de calidad de servicio permite elegir aquellas soluciones que mejor se adecuen a las necesidades del usuario y a las características de la red: energía, movilidad, ancho de banda, etc.
- **Seguridad:** la naturaleza distribuida de las redes ad hoc plantea también problemas de seguridad durante los procesos de composición de servicios: servicios maliciosos, usos indebidos, denegación de servicio, etc.
- **Aspectos económicos:** la utilización de servicios proporcionados por otros dispositivos, y de sus recursos asociados, puede introducir la necesidad de llevar a cabo ciertas transacciones económicas o acuerdos de uso entre los clientes y los proveedores.

Además de la clasificación anterior, es necesario tener en cuenta otros aspectos que son propios a las características de las redes MANET:



- **Modo de operación:** las soluciones pueden funcionar en modo *reactivo* o *proactivo*, dependiendo de si los procesos relacionados con la búsqueda de composiciones se activan únicamente en respuesta a una solicitud de usuario o se realizan de forma continua durante el tiempo de vida de la red MANET, respectivamente. Este aspecto es similar a la clasificación de los protocolos de encaminamiento de la sección 2.1.3 y de las soluciones de descubrimiento de la sección 3.1.6.
- **Respuesta a la movilidad:** la composición en redes con infraestructura y que no tienen una gran movilidad de sus dispositivos participantes presenta una mayor facilidad para las soluciones existentes. Por el contrario, cuando la topología de la red es dinámica las soluciones deben hacer frente a los cambios que se produzcan con objeto de poder mantener las soluciones encontradas o buscar otras existentes.

En la revisión bibliográfica realizada se han tenido en cuenta únicamente aquellos aspectos relacionados de forma directa con la propuesta para la composición de servicios en redes MANET desarrollada en el capítulo 6. Por lo tanto, se omiten los aspectos relativos a la *calidad de servicio*, la *seguridad* y los *aspectos económicos*, que no han sido tenidos en cuenta en el desarrollo la investigación.

Por otro lado, la investigación realizada se centra principalmente en la búsqueda de composiciones y no en la posterior ejecución de las soluciones encontradas. Sin embargo, en esta sección se recogen algunas soluciones existentes en la actualidad que pueden servir para llevar a cabo la ejecución de las composiciones encontradas en un entorno distribuido, como es el caso de las redes ad hoc. Las soluciones existentes en la actualidad, y que se han considerado más relevantes para la propuesta realizada en el capítulo 6, son presentadas en las siguientes secciones.

### 3.2.2 Arquitectura

Las soluciones tradicionales para la composición de servicios se basan en el uso de *repositorios centralizados* donde los proveedores registran los servicios disponibles. Cuando un cliente tiene necesidad de llevar a cabo una composición de servicios inicia un proceso de búsqueda en dichos repositorios. La principal ventaja de la centralización del proceso de composición es que el mecanismo de búsqueda de soluciones puede tener un conocimiento de todos los servicios disponibles así como de las relaciones que se establecen entre ellos. En este caso, la búsqueda de composiciones no se lleva a cabo de una manera parcial, como ocurre con las soluciones distribuidas, y, por lo tanto, podrá obtener todas las soluciones disponibles en cada momento.

El uso de un servidor para centralizar la información de los servicios disponibles es utilizado por Mokhtar, Liu, Georgantas y Issarny (2005). En dicha solución se usa un algoritmo basado en la propuesta para el emparejamiento semántico de servicios realizada por Paolucci, Kawamura, Payne y Sycara (2002), donde se utilizan las operaciones de entrada/salida de los servicios. Otros ejemplos clásicos que han servido para definir los aspectos básicos del proceso de composición de servicios, y que también utilizan un enfoque centralizado, pueden encontrarse en las propuestas de Román, Ziebart y Campbell (2003) y de Ponnekanti, Lee, Fox, Hanrahan y Winograd (2001).

Existen, por otro lado, algunas soluciones descentralizadas para la composición de servicios que a pesar de no utilizar un gestor centralizado para la resolución de las composiciones se basan principalmente en resolver los problemas de heterogeneidad durante la comunicación entre los servicios de los dispositivos (Grimm, 2004; Edwards, Newman, Seciivy y Smith, 2005). Además, la composición de servicios puede ser vista también como un enfoque orientado a tareas en el cual, el sistema, a partir de una definición del usuario, determina cuáles son los servicios que deben ser ejecutados para obtener el resultado deseado (Masuoka, Parsia y Labrou, 2003; Sousa, Poladian, Garlan, Schmerl y Shaw, 2006).

Sin embargo, como ocurre en el caso del descubrimiento de servicios tratado en la sección 3.1, el uso de uno o más repositorios que centralicen las descripciones de servicios no es adecuado para la naturaleza dinámica de las redes MANET. La ausencia de una infraestructura fija en este tipo de redes hace que sea imposible confiar en la existencia de un repositorio que pueda ser continuamente accesible desde cualquier punto de la red. Así, las soluciones anteriormente comentadas están orientadas hacia entornos de computación ubicua que disponen de una infraestructura fija de comunicación. Esto ha hecho que sea necesario el desarrollo de propuestas específicas para la composición de servicios que tengan en cuenta las características de las redes MANET.

Para evitar el uso de uno o más repositorios fijos existen soluciones que se basan en la *selección de un gestor* dentro de la red que lleva a cabo el proceso de composición de servicios de una forma completamente distribuida. Así, en el caso del trabajo de Chakraborty, Yesha y Joshi (2004), la selección del gestor se realiza, para cada una de las peticiones de composición, mediante un proceso de *arbitraje* que selecciona el mejor dispositivo para cada tarea.

Por su parte, el trabajo de Ruta, Zacheo, Grieco, Di Noia, Boggia, Tinelli, Camarda y Di Sciascio (2010) propone un proceso similar al anterior para la selección del gestor de cada composición, pero en este caso se tienen en cuenta diferentes aspectos como la energía restante, sus capacidades de computación, la memoria libre disponible u otros aspectos relacionados. La desventaja de este tipo de soluciones es que los gestores de composición llevan a cabo el proceso de composición mediante el envío de mensajes de descubrimiento y selección

de servicios a través de la red ad hoc, lo que supone un aumento en el número de mensajes que deben ser enviados por cada uno de los nodos seleccionados.

Por último, existen otras soluciones que se basan en el uso de una arquitectura de tipo de *red entre pares (Peer to Peer - P2P)* en las que todos los nodos de la red ad hoc poseen el mismo comportamiento potencial y el proceso de composición se encuentra distribuido entre ellos de forma natural (Gu, Nahrstedt y Yu, 2004; Zhu, Du y Zhu, 2006; Mandreoli, Perdichizzi y Penzo, 2007; Gharzouli y Boufaida, 2009). Este tipo de soluciones evita la necesidad de llevar a cabo la selección de un gestor para la realización de los procesos de composición y, por lo tanto, reduce las posibilidades de aparición de puntos de fallo, así como el número de mensajes enviados para su gestión.

### 3.2.3 Mecanismo de composición

A lo largo de los años se han propuesto diferentes mecanismos para resolver el problema de la composición de servicios a partir de la especificación del servicio o tarea a componer. Así existen, por un lado, varias propuestas que basan el proceso de búsqueda de soluciones en técnicas de *planificación* de IA. La composición de servicios basada en planificación consiste, como su nombre indica, en utilizar un algoritmo que permite resolver un problema que, partiendo de un estado inicial determina los pasos a realizar para llegar a un estado final o meta. Para resolver un problema de planificación es necesario realizar una representación de los estados posibles del entorno, incluyendo el estado inicial, el final y las operaciones permitidas para progresar. Normalmente, un problema de planificación se suele representar mediante una tupla de cinco elementos ( $S, S_o, G, A, \Gamma$ ):

- $S$  es el conjunto de todos los estados posibles.
- $S_o$  es el estado inicial, aquel del que parte el problema de composición.
- $G$  es el estado final, el objetivo de la composición.
- $A$  es el conjunto de acciones permitidas que el planificador puede realizar para progresar de un estado al siguiente.
- $\Gamma$  define las precondiciones y efectos en la ejecución de cada acción. Las *precondiciones* de una acción o servicio definen el estado en el que se puede llevar a cabo correctamente la ejecución del mismo. Los *efectos* determinan el estado que se obtiene tras la ejecución de la acción o servicio.

En el caso de la planificación para la composición de servicios,  $S_o$  y  $G$  son estados especificados mediante la solicitud del servicio compuesto.  $A$  es el conjunto de los servicios disponibles, y  $\Gamma$  denota la función de cambio de estado asociado a cada uno de los servicios.

Así, Kalofonos y Wisner (2007), aplican un algoritmo de planificación mediante una búsqueda en profundidad para encontrar soluciones a un problema de composición planteado, mientras que, tanto el trabajo de Sheshagiri, desJardins y Finin (2003) como el de Ni (2005), emplean el planificador automático *STRIPS* (Fikes y Nilsson, 1972).

Existen soluciones que utilizan algoritmos de planificación más complejos y que se basan en la descomposición de la tarea o servicio planteado en problemas más sencillos. Este es el caso de la planificación *HTN* (*Hierarchical Task Network*) utilizada por Qasem, Heflin y Muñoz-avila (2004), haciendo uso del planificador *SHOP2* (Nau, Ilghami, Kuter, Murdock, Wu y Fusun, 2003). Este enfoque es aplicado también en otras soluciones propuestas (Sirin, Parsia, Wu, Hendler y Nau, 2004; Vukovic y Robinson, 2004).

Además, se pueden aplicar otras técnicas de IA para resolver el problema de la composición de servicios. Por ejemplo, éste puede ser llevado a cabo mediante la utilización de un lenguaje de programación lógico, como es el caso de la solución planteada por McIlraith y Son (2002), en la cual los autores proponen la utilización del lenguaje *GOLOG* (Levesque, Reiter, Lesperance, Lin y Scherl, 1997). Por otro lado, Sheshagiri, desJardins y Finin (2003) utilizan un motor de reglas basado en Rete (Forgy, 1982) para encontrar las soluciones a una solicitud de composición de servicios.

Las técnicas anteriores se utilizan en entornos centralizados en los que es posible tener un conocimiento de los diferentes servicios disponibles en el entorno. Su aplicación a las redes MANET plantea problemas debido a la necesidad de registrar los servicios disponibles en un único punto y al dinamismo de la topología de la red. Son necesarias, por lo tanto, soluciones adaptadas a las necesidades de las redes de este tipo. Por ejemplo, la descomposición HTN puede utilizarse en estas redes para descomponer una tarea en subtareas más sencillas y que pueden ser distribuidas a otros dispositivos de la red. Las soluciones a cada subproblema de composición son agrupadas de forma jerárquica para obtener la solución final, tal y como aplican Basu, Ke y Little (2002). Otra solución, como es el caso de Zhang y Kowalczyk (2006), es utilizar una versión distribuida de un algoritmo de un algoritmo de planificación tradicional, o de un algoritmo *CSP* (*Constraint Satisfaction Problem*) distribuido, tal y como proponen Karmouch y Nayak (2010).

Por otro lado, existen soluciones creadas específicamente para las redes móviles ad hoc que tienen en cuenta la naturaleza multisalto de dichas redes y que se basan en la difusión de mensajes a través de los enlaces de la red. Así, en el caso de Chakraborty, Yesha y Joshi (2004), el nodo que gestiona el proceso de composición, que es elegido mediante un mecanismo de arbitraje, lleva a cabo un proceso para descubrir aquellos servicios que se encuentran en las inmediaciones y que pueden formar parte de la composición requerida. Un procedimiento similar es seguido por Ruta, Zacheo, Grieco, Di Noia, Boggia, Tinelli,

Camarda y Di Sciascio (2010), sin embargo, en este caso el proceso de búsqueda de los servicios que participan en la composición es llevado a cabo por el propio nodo que la ha iniciado mientras que la zona de búsqueda se incrementa salto a salto desde el nodo inicial hasta llegar a una distancia máxima.

Además de lo anterior, la propuesta de Han y Zhang (2010) utiliza un mecanismo de composición de servicios basado en el envío de mensajes a través de la red para la localización de las posibles soluciones y que está integrado con el protocolo de encaminamiento DSR. Por último, la localización de los servicios que forman parte de la composición puede utilizar un proceso de búsqueda en la red combinado con su selección aplicando *lógica difusa*, tal y como han propuesto Prochart, Weiss, Schmid y Kaefer (2007).

Por otra parte, hay soluciones que basan el proceso de composición en la creación de *estructuras virtuales (overlay networks)* que establecen una red lógica sobre la topología física formada por los dispositivos. Estas redes virtuales mantienen información sobre las relaciones entre los diferentes servicios disponibles. Por ejemplo, este proceso puede ser aplicado a redes con infraestructura, como es el caso de diferentes trabajos (Gu, Nahrstedt y Yu, 2004; Hu, Muthusamy, Li y Jacobsen, 2008) o a redes de área personal (*Personal Area Network - PAN*), como es llevado a cabo por Herborn y Seneviratne (2006).

En el caso de las redes MANET, la creación de una red virtual de servicios es aplicada por Hu, Tang, Wang y Ji (2007), donde determinados nodos de la red son seleccionados para constituir la red virtual. Los vecinos de estos nodos registran sus servicios en los nodos seleccionados y se descubren relaciones de compatibilidad entre ellos. Estas relaciones de compatibilidad son utilizadas durante la búsqueda de composiciones para encontrar posibles conexiones. Los servicios también pueden agruparse para formar una red virtual según el dominio de aplicación al que pertenecen, de acuerdo al trabajo de Huang, Bai, Zhang, Kong y Qian (2007).

La propuesta realizada por Hu, Tang, Wang y Ji (2007) puede tener algunos aspectos comunes con la realizada en esta tesis, sin embargo, en dicho trabajo previo, la detección de compatibilidades se realiza en nodos directorio y, además, no se profundiza en cuál es el proceso concreto que se lleva a cabo para llevar a cabo esta detección. Por el contrario, y como se explica en el capítulo 6, en la solución propuesta en esta tesis el proceso de creación del grafo de servicios es completamente distribuido.

### 3.2.4 Especificación de la composición

Para poder iniciar el proceso de composición es necesario poder especificar de alguna forma cuál es el servicio o tarea compleja que se quiere resolver. Se distinguen, según Rao y Su (2005), dos maneras de llevar a cabo este proceso: mediante la especificación de un *flujo de trabajo* que define de forma explícita

las relaciones entre los servicios o, por el contrario, mediante la especificación de un *objetivo* final a conseguir.

La composición de servicios basada en la especificación de un *flujo de trabajo* consiste en indicar el servicio compuesto que se desea obtener mediante la definición de un *modelo abstracto* del mismo. El modelo de proceso abstracto incluye los tipos de los servicios a ejecutar y las dependencias de datos entre ellos. La construcción del modelo abstracto debe llevarse a cabo antes de que comience el proceso de composición en sí. Dicho modelo se suele representar mediante la utilización de un grafo en el cual los nodos representan los servicios simples que constituyen el servicio compuesto y las aristas las dependencias de datos entre ellos. En este tipo de composición, el sistema se encarga de realizar un proceso denominado *instanciación*, que consiste en seleccionar dispositivos y servicios reales en la red que satisfagan los requisitos contenidos en la especificación en forma de modelo abstracto.

La utilización de un *flujo de trabajo* para la búsqueda de composiciones ha sido ampliamente utilizada en soluciones centralizadas y en redes tradicionales con infraestructura (Song, Labrou y Masuoka, 2004; Hesselman, Tokmakoff, Pawar y Iacob, 2006; Vallée, Ramparany y Vercouter, 2005; Lee, Ko, Lee, Lee y Helal, 2007). Además, la instanciación de un flujo de trabajo también ha sido aplicada a las redes MANET en distintos trabajos (Basu, Ke y Little, 2002; Coloberti, Lombriser, Roggen, Tröster, Guarneri y Riboni, 2008; Huang, Bai, Zhang, Kong y Qian, 2007; Sen, Roman y Gill, 2008; Groba, 2012; Luo, Yang, Tang, Zhang y Xiong, 2012).

Sin embargo, el principal problema con la composición basada en la especificación de un *flujo de trabajo* es que el modelo abstracto que define al servicio compuesto debe ser construido *a priori*, sin conocer cuáles son los servicios que están realmente disponibles en la red, lo que complica el proceso y que puede resultar en que algunas soluciones disponibles sean descartadas porque no son adecuadas según la solicitud.

Por otro lado, la especificación del servicio o tarea a componer como un *objetivo* está relacionada con las soluciones basadas en técnicas de planificación de IA. Como se ha comentado en la sección 3.2.3 el objetivo final debe acompañarse siempre de la especificación del estado actual y de un conjunto de posibles acciones a realizar.

El enfoque orientado a *objetivo* ha sido utilizado tanto de manera centralizada (Qasem, Heflin y Muñoz-avila, 2004; Sirin, Parsia, Wu, Hendler y Nau, 2004; Vukovic y Robinson, 2004; McIlraith y Son, 2002; Sheshagiri, desJardins y Finin, 2003), como para su aplicación en redes distribuidas o de naturaleza ad hoc (Basu, Ke y Little, 2002; Zhang y Kowalczyk, 2006; Karmouch y Nayak, 2010). Además, la especificación del servicio a componer puede realizarse en estos casos mediante la descripción de las *entradas*, *salidas*, *precondiciones* y

*efectos (Inputs, Outputs, Preconditions and Effects - IOPEs)* de la composición buscada, tal y como han propuesto Hu, Muthusamy, Li y Jacobsen (2008).

### 3.2.5 Descripción de servicios

Los servicios provistos por los dispositivos que constituyen la red ad hoc deben ser descritos utilizando un lenguaje que permita determinar si pueden formar parte de una composición requerida. La composición de servicios hace uso de los mecanismos de descubrimiento explicados en la sección 3.1 y, por lo tanto, puede aplicarse el mismo criterio para la clasificación de las descripciones de servicios que en dicho caso, existiendo así descripciones de tipo *sintáctico* o *semántico*. Esta clasificación puede utilizarse tanto en aquellas soluciones aplicadas a redes con infraestructura, centralizadas y a las redes MANET.

De nuevo, las diferencias entre el uso de tecnologías sintácticas o semánticas para la descripción de los servicios depende de las necesidades de expresividad y el posterior razonamiento a aplicar durante la composición. Como ocurre con el descubrimiento de servicios, el uso de lenguajes semánticos permite mayores posibilidades a la hora de establecer relaciones entre los servicios y seleccionar los más adecuados para la composición.

Por otro lado, es usual que la solicitud de composición sea expresada utilizando el mismo lenguaje en el que se encuentran descritos los servicios disponibles, ya sea este sintáctico o semántico. Si la especificación de la solicitud de composición se lleva a cabo mediante el uso de un *flujo de trabajo* existen lenguajes tanto sintácticos como semánticos su representación. Así, en el caso de las propuestas que usan WSDL, como la realizada por Christensen, Curbera, Meredith y Weerawarana (2001), es usual utilizar WS-BPEL (OASIS Group, 2007) como lenguaje para especificar la solicitud en forma de *flujo de trabajo*; mientras que, en el caso de las descripciones semánticas, OWL-S (W3C Member Submission, 2004) es un lenguaje que permite representar también relaciones de flujo de datos y ejecución entre servicios.

En cuanto a la revisión bibliográfica realizada, existen soluciones que hacen uso de descripciones simples de los servicios mediante parejas de propiedades *clave-valor*, como el realizado por Lee, Ko, Lee, Lee y Helal (2007). Aunque más común que el enfoque anterior es la utilización un lenguaje para la descripción de servicios como WSDL, utilizada en varios trabajos Vukovic y Robinson (2004); Song, Labrou y Masuoka (2004); Kalofonos y Wisner (2007). Por otro lado, son comunes también las propuestas que utilizan descripciones semánticas expresadas en lenguajes como OWL-DL (Ruta, Zacheo, Grieco, Di Noia, Boggia, Tinelli, Camarda y Di Sciascio, 2010), DAML-S u OWL-S (Chakraborty, Joshi, Yesha y Finin, 2006; Hu, Muthusamy, Li y Jacobsen, 2008).

### 3.2.6 Participación del usuario

El usuario puede participar en el proceso de composición seleccionando qué servicios participan en una solución dada. Si el proceso de selección es completamente dirigido por el usuario mediante el ensamblado de los servicios disponibles se habla de un proceso *manual*. En este caso, el sistema se encarga únicamente de la posterior ejecución del servicio compuesto. Más común en la bibliografía revisada es la utilización del enfoque *semi-automático* en el cual la selección de los servicios participantes es todavía realizada por el usuario pero ayudado por el sistema con el filtrado y selección de los servicios compatibles. La composición semi-automática de servicios es utilizado principalmente en los enfoques centralizados y en aquellos enfocados a redes con infraestructura (Kalofonos y Wisner, 2007; Ranganathan y Campbell, 2004; Ni, 2005).

Por otro lado, cuando se trata de una composición *automática*, el usuario participa únicamente en el proceso mediante la especificación del servicio a componer, como se aplica en los trabajos de Ponnekanti, Lee, Fox, Hanrahan y Winograd (2001) y Mokhtar, Liu, Georgantas y Issarny (2005). En otros casos, la solicitud de búsqueda es realizada directamente por la aplicación y ocultada al usuario de la misma (Carey, Lewis, Higel y Wade, 2004; Sirin, Parsia, Wu, Hendler y Nau, 2004; Vukovic y Robinson, 2004). Además, de acuerdo a la revisión bibliográfica realizada, la composición automática de servicios es la más adecuada en el caso de las redes MANET, tal y como se observa en diferentes trabajos (Chakraborty, Yesha y Joshi, 2004; Ruta, Zacheo, Grieco, Di Noia, Boggia, Tinelli, Camarda y Di Sciascio, 2010; Han, Zhang, Li y Zhang, 2007; Han y Zhang, 2010; Prochart, Weiss, Schmid y Kaefer, 2007; Hu, Tang, Wang y Ji, 2007).

La participación del usuario en el proceso de composición requiere que el nodo que comienza el mismo tenga conocimiento de las diferentes alternativas que existen en cada paso. Esto supone que la información sobre los diferentes servicios disponibles debe transmitirse hasta dicho nodo, con el consiguiente incremento en el número de mensajes en la red. Sin embargo, en estos casos, sí que puede ser posible la participación del usuario para seleccionar la composición más adecuada a sus necesidades una vez que se han obtenido las diferentes soluciones disponibles.

### 3.2.7 Modo de operación

Al igual que ocurre con los protocolos de encaminamiento y descubrimiento para redes móviles ad hoc, o aquellas con infraestructura pero de naturaleza completamente distribuida, las soluciones para la composición de servicios también pueden ser clasificadas en mecanismos *proactivos* o *reactivos*.

Entre las soluciones proactivas se incluyen aquellas propuestas que se basan en la construcción de una estructura de información que es mantenida por los



nodos que constituyen la red. La propia existencia de los servicios provistos por los nodos de la red ad hoc resulta en el envío de mensajes que tienen como finalidad la actualización de la estructura. La información mantenida por los nodos es utilizada durante los procesos de búsqueda de composiciones para encontrar las soluciones de una forma más directa. Estas soluciones tienen como desventaja el hecho de que pueden suponer un mayor número de mensajes enviados durante las tareas de mantenimiento. Se incluyen en este grupo aquellas soluciones comentadas anteriormente en otros apartados y que hacen uso de una estructura virtual de información sobre la red de dispositivos (Gu, Nahrstedt y Yu, 2004; Hu, Muthusamy, Li y Jacobsen, 2008; Herborn y Seneviratne, 2006; Hu, Tang, Wang y Ji, 2007).

Por otro lado, las soluciones reactivas se basan en realizar las comunicaciones necesarias para la búsqueda de las composiciones únicamente en aquellos momentos en las que se inicia el proceso. La ventaja de este tipo de soluciones sobre las anteriores es que se minimizan los mensajes transmitidos a través de la red ad hoc cuando no se lleva a cabo ninguna solicitud de composición (Chakraborty, Yesha y Joshi, 2004; Ruta, Zacheo, Grieco, Di Noia, Boggia, Tineili, Camarda y Di Sciascio, 2010; Han, Zhang, Li y Zhang, 2007; Han y Zhang, 2010; Prochart, Weiss, Schmid y Kaefer, 2007).

### 3.2.8 Ejecución

Cuando se obtiene una composición de servicio puede llevarse a cabo su ejecución. En la ejecución de servicios compuestos existen dos enfoques tradicionales: *orquestración* y *coreografía*.

Por un lado, la orquestración de un servicio compuesto se lleva a cabo mediante el intercambio de mensajes entre un elemento centralizador y aquellos otros que participan en la composición. Incluye aspectos relacionados con la lógica del proceso y el orden de las acciones que deben llevarse a cabo. Por el contrario, la coreografía de servicios es un proceso más colaborativo y define la forma en la que cada participante toma parte en el proceso conjunto. De acuerdo a Peltz (2003), cada uno de los elementos de una coreografía conoce como responder a los mensajes que se producen sin la necesidad de que un coordinador central deba indicarle cómo hacerlo.

En el caso de las soluciones centralizadas revisadas, es usual que el resultado de la composición sea un proceso ejecutable que representa al servicio compuesto. Dicho proceso representa mediante orquestración las relaciones entre los servicios seleccionados participantes (Sheshagiri, desJardins y Finin, 2003; Carey, Lewis, Higel y Wade, 2004; Vukovic y Robinson, 2004; Vallée, Ramparany y Vercouter, 2005; Pourreza y Graham, 2006).

Por otro lado, en las soluciones aplicadas a redes MANET, y dada su naturaleza distribuida, la aplicación de la orquestración plantea problemas a la hora

de centralizar su ejecución. Hay que tener en cuenta que la orquestación necesita que la entidad coordinadora envíe y reciba las solicitudes y respuestas de ejecución. En el caso de una red ad hoc, esto supone un mayor envío de mensajes hacia el coordinador central que puede suponer la congestión de la red o la disminución drástica de su batería. Así, la ejecución de los servicios en este caso suele consistir en un proceso que se lleva a cabo de manera distribuida a través de la red ad hoc. El control de la ejecución es gestionado en cada momento por el servicio que recibe la composición a ejecutar hasta que ésta es realizada completamente (Walzer, Anke, Loser y Wu, 2007; Sun, Zhang, Chen, Peng y Xu, 2008; Zhou, Ge, Chen, Jing y Sun, 2011).

Por lo tanto, se sigue que la aplicación de la coreografía es más adecuada para este tipo de redes, debido a su naturaleza distribuida, como es el caso del trabajo realizado por Zhang, Sun, Chen, Mao y Xu (2008). Los problemas que puedan surgir durante el proceso de ejecución son comunicados a los nodos que la iniciaron o incluso se lleva a cabo una reparación de la misma buscando y seleccionando otra composición alternativa disponible, como es el caso de las propuestas de (Chen, He, Ren y Sun, 2008; Jiang, Xue y Schmidt, 2009).

### 3.2.9 Respuesta a la movilidad

La naturaleza dinámica de las redes móviles ad hoc plantea problemas para la composición de servicios. La movilidad de los nodos, que se traduce en la desaparición de rutas de comunicación entre ellos, hace que las composiciones encontradas puedan dejar de ser utilizables. Evidentemente, las soluciones que han sido desarrolladas para entornos con infraestructura y, por lo tanto, con posibilidad de utilizar servidores o gestores centralizados, no tienen estos aspectos en cuenta y no son adecuadas para las redes móviles ad hoc.

En el caso de las redes MANET, las soluciones *proactivas* que llevan a cabo la construcción de estructuras, o que mantienen información sobre la situación de la red, deben responder a estos cambios (Gu, Nahrstedt y Yu, 2004; Hu, Muthusamy, Li y Jacobsen, 2008; Herborn y Seneviratne, 2006; Hu, Tang, Wang y Ji, 2007). Como se ha comentado anteriormente, actualizar esta información supone el envío de mensajes que se traduce en un mayor número de transmisiones entre los nodos.

Por otro lado, en las propuestas de naturaleza *reactiva*, debido a que la búsqueda de composiciones se realiza únicamente cuando es necesaria, no existe, o se minimiza, la necesidad de responder a los cambios que se producen en la red ad hoc durante la movilidad (Chakraborty, Yesha y Joshi, 2004; Ruta, Zacheo, Grieco, Di Noia, Boggia, Tinelli, Camarda y Di Sciascio, 2010; Han, Zhang, Li y Zhang, 2007; Han y Zhang, 2010; Prochart, Weiss, Schmid y Kaefer, 2007). Sin embargo, tanto en uno como en otro caso pueden llevarse a cabo procedimientos para detectar si las composiciones encontradas siguen siendo válidas

y llevar a cabo las reparaciones necesarias. Estas reparaciones pueden consistir simplemente en la substitución parcial de uno o más servicios afectados que participan en la composición o, como se ha comentado en la sección anterior, la búsqueda de una solución alternativa completa (Chen, He, Ren y Sun, 2008; Jiang, Xue y Schmidt, 2009; Ruta, Zacheo, Grieco, Di Noia, Boggia, Tinelli, Camarda y Di Sciascio, 2010; Zhou, Ge, Chen, Jing y Sun, 2011).

### 3.2.10 Conclusiones

En esta sección se ha realizado una revisión de las soluciones existentes en la actualidad para la composición de servicios, tanto en su aplicación a redes con infraestructura como, principalmente, a las redes móviles ad hoc. Este estudio ha permitido categorizar las diferentes decisiones que pueden tomarse en el desarrollo de una solución de composición para redes MANET.

Respecto a la arquitectura de las propuestas, es obvio que una solución que se base en la utilización de un servidor centralizado para los servicios disponibles, no es adecuada a las características de una red MANET. En el caso concreto de la solución presentada en el capítulo 6, se propone la utilización de una arquitectura distribuida de naturaleza P2P en la que todos los nodos de la red participan de la misma forma en el proceso de composición.

Por otro lado, acerca del mecanismo de composición utilizado puede decirse que únicamente aquellos que tienen en cuenta la naturaleza multisalto de las redes ad hoc son adecuados para las mismas. Las soluciones que se basan en procesos de planificación o búsqueda en un repositorio de servicios deben ser modificadas para utilizar algoritmos distribuidos que tengan en cuenta el hecho de que el conocimiento sobre los servicios disponibles no está centralizado. La solución propuesta en el capítulo 6 de esta tesis se basa en la construcción de una estructura distribuida que debe ser mantenida en la red ad hoc. Las características de esta estructura y el proceso por el cual es construida y mantenida se detallan en dicho capítulo.

La mayoría de las soluciones revisadas basan la especificación de la solicitud de composición de servicios en un *flujo de trabajo* preconstruido. Como se ha visto, esto requiere que la especificación del grafo sea elaborada *a priori*, ya sea por el usuario o por el desarrollador de la aplicación. Sin embargo, la solución propuesta en esta investigación se basa en la definición de los estados inicial y final del servicio buscado, especificados ambos mediante los tipos de sus parámetros de entrada y salida. Una aplicación que usa un servicio debe conocer al menos esta información para poder ejecutarlo. Por lo tanto, en el capítulo 6 se propone cómo utilizar esta información para resolver el problema de composición de servicios en una red MANET, minimizando de esta forma la participación del usuario o desarrollador de la aplicación en el proceso de composición y consiguiendo un proceso más *automático*.

Por otra parte, al igual que ocurre con el descubrimiento de servicios, la descripciones semánticas proporcionan mayores posibilidades a la hora de seleccionar servicios para ser utilizados en una composición. La utilización de una ontología de conceptos para categorizar los servicios, o sus entradas y salidas como es el caso de esta tesis doctoral, permite encontrar relaciones que, de utilizar un lenguaje sintáctico, podrían quedar ocultas. Además, debido a que el mecanismo de composición propuesto en esta investigación utiliza el protocolo de descubrimiento explicado en el capítulo 5, es natural que se utilice la misma descripción para los servicios que participan en ambos procesos. Por estas razones, la solución propuesta en esta investigación para la composición de servicios en redes MANET utiliza una descripción de servicios semántica basada en la categorización de las entradas y salidas de acuerdo a una ontología de conceptos conocida por los diferentes nodos de la red.

Por último, la propuesta realizada en esta tesis se centra en el mecanismo para la búsqueda de las composiciones y no en los procedimientos de ejecución de los resultados obtenidos. Como se explica en el capítulo 6, las composiciones obtenidas como respuesta a una solicitud pueden ser ejecutadas de una forma distribuida mediante la propagación de un mensaje a través de los nodos de la red ad hoc que participan en la composición. En tal caso se estaría utilizando un enfoque cercano a la *coreografía* de servicios, debido a que las soluciones basadas en *orquestración* de servicios no son adecuadas para la naturaleza distribuida de las redes ad hoc.

# Arquitectura para la composición distribuida de servicios

El objetivo de las redes de computadores es que los dispositivos que las integran puedan compartir información y recursos entre ellos. Por ejemplo, la composición de servicios hace uso de las funcionalidades proporcionadas por diferentes dispositivos con el objeto de obtener, mediante el acceso a los recursos disponibles en un orden determinado, una funcionalidad no disponible directamente. La composición de servicios es, como se puede ver en la sección 3.2 del estado actual, un proceso complejo que requiere resolver distintas problemáticas para poder ser llevado a cabo con éxito. Su aplicación a las redes móviles ad hoc no hace sino aumentar dicha complejidad debido a las características propias de este tipo de redes: movilidad, comunicación inalámbrica, ausencia de infraestructura fija, etc., explicadas con mayor detalle en la sección 2.1.

El objetivo principal de esta investigación es proporcionar una solución al proceso automático de composición de servicios en su aplicación al ámbito de las redes MANET. En la sección 4.1 de este capítulo se define cuál es el problema de la composición de servicios que se quiere resolver, mientras que en la sección 4.2 se expone de una manera más formal cuáles deben ser los requisitos que han sido identificados y que debe cumplir una solución para que pueda ser aplicada al dominio de las redes MANET.

A partir de la definición de los requisitos, en la sección 4.3 se propone una arquitectura para la composición colaborativa y dinámica de servicios en redes MANET. Las particularidades de esta arquitectura son recogidas en el capítulo 5

y el capítulo 6. Por otro lado, en la sección 4.4 se presentan los aspectos relativos a los servicios utilizados en esta investigación para su uso en el proceso de composición definido. Se presenta además un procedimiento para la anotación de los recursos disponibles mediante la aplicación de una ontología de conceptos compartida por los diferentes nodos de la red.

La aportación principal de este capítulo es la definición de una arquitectura integral para la composición de servicios en redes ad hoc de una forma distribuida y dinámica, así como la introducción de la ontología de conceptos para la anotación de los distintos parámetros de los servicios. Como se explica en el capítulo 5, la utilización de una ontología de conceptos para la descripción de los parámetros de los servicios beneficia los procesos de búsqueda de servicios en la red ad hoc, así como su composición.

## 4.1 Definición del problema

En los entornos de computación ubicua, los usuarios deben ser capaces de acceder no solamente a los servicios proporcionados de manera directa por los dispositivos, sino también a aquellos resultantes de la conexión y agregación de estos servicios más sencillos. El entorno, constituido por los diferentes dispositivos disponibles y comunicados entre sí, debe ayudar al usuario en el proceso de obtener dicha composición de servicios. Entendiendo las redes ad hoc como una particularización de los entornos ubicuos, el objeto principal de esta investigación es proporcionar un nuevo enfoque para resolver el problema de la composición de servicios en este tipo de redes de comunicación. La motivación de la investigación, así como la definición de la hipótesis de partida y los objetivos marcados son explicados en la sección 1.1 y sección 1.2, respectivamente.

Casati, Sayal, Shan, Dittrich, Geppert y Norrie (2001) presentan la composición de servicios como *un proceso similar a la definición de un flujo de trabajo constituido por diferentes servicios. Para definir un servicio compuesto es necesario especificar los servicios utilizados, el flujo de datos de entrada salida y sus dependencias de ejecución. Es decir, un servicio compuesto incluye los servicios individuales que lo constituyen junto con el flujo de control y datos entre ellos, tal y como también es propuesto en el caso de Rao y Su (2005).*

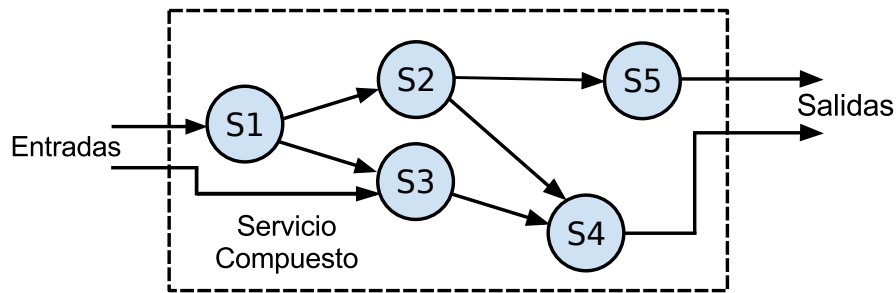
Se introducen a continuación una serie de términos que son utilizados en el ámbito de la composición de servicios:

- **Dispositivo:** cada uno de los elementos con capacidades de computación y comunicación que forman parte de la red ad hoc. De acuerdo a la introducción realizada a este tipo de redes en la sección 2.1, estos dispositivos se corresponden con el concepto de *nodo* utilizado en dichas redes. Los dispositivos pueden ser de naturaleza muy variada: ordenadores portátiles, *smartphones*, sensores, etc.

- **Servicio:** los dispositivos pueden proporcionar funcionalidades que podrán ser accedidas desde otros dispositivos. Cada una de las funcionalidades expuestas se entiende en esta investigación que se corresponde con un *único* servicio. Las particularidades de acceso a los servicios dependerán de la tecnología utilizada. En la sección 4.4 se explica con mayor detalle cuáles son las características de los servicios, tal y como se utilizan en esta investigación. Se distingue entre *servicio atómico* y *servicio compuesto*. El primero hace referencia a aquellos servicios individuales e indivisibles que participan en una composición de servicios, mientras que el segundo término se refiere a los servicios resultantes del proceso de composición. Por supuesto, es posible que los servicios compuestos resultantes puedan participar, a su vez, en futuras composiciones.
- **Flujo de trabajo:** define las relaciones que se establecen entre los servicios que forman parte de una composición. Los datos de salida de un servicio podrán ser utilizados como datos de entrada por otro. Además del flujo de datos puede definir un flujo de control que determine las condiciones para la ejecución de los servicios: *alternativas, bucles, excepciones, etc.*
- **Entradas/salidas:** cada servicio tiene un conjunto de parámetros de entrada y un conjunto de parámetros de salida. Los parámetros de entrada definen el número y tipo de los datos que el servicio necesita para llevar a cabo la ejecución de su funcionalidad. Por su parte, los parámetros de salida determinan cuál será el resultado de la ejecución del servicio en forma de datos.
- **Precondiciones y efectos:** aunque se disponga del conjunto completo de parámetros de entrada de un servicio, no siempre puede llevarse a cabo la ejecución del mismo. Las precondiciones de un servicio definen una serie de condiciones lógicas que deben ser ciertas para que la ejecución del servicio pueda ser llevada a cabo de forma correcta. Por su parte, los efectos de un servicio determinan el conjunto de condiciones lógicas que son ciertas tras la *correcta ejecución* del servicio.

La figura 4.1 muestra una representación de un flujo de trabajo que conecta distintos servicios atómicos. Las flechas entre servicios representan el flujo de datos entre dos servicios. Los datos de salida de un servicio pueden ser utilizados como datos de entrada por otros. El servicio compuesto resultante puede verse como un nuevo servicio con sus propias entradas y salidas.

De acuerdo a lo presentado en la sección 3.2.6 del estado actual, la selección de los servicios que participan en la composición así como de los flujos de control que se creen entre ellos puede realizarse de varias formas, *manual*,



**Figura 4.1:** Servicios atómicos que forman parte de un flujo de trabajo en un servicio compuesto

*semi-automática* o completamente *automática*, dependiendo del grado de participación del usuario en el proceso. Esta investigación se centra en el proceso de creación de un servicio compuesto de forma automática, es decir, aquel que, partiendo de una especificación del servicio a componer, lleva a cabo una selección de los servicios participantes y crea el flujo de datos sin la participación del usuario (Sheshagiri, desJardins y Finin, 2003; Constantinescu, Faltings y Binder, 2004; Peer, 2004; Pistore, Barbon, Bertoli, Shaparau y Traverso, 2004; Vukovic y Robinson, 2004).

Como se explica con más detalle en capítulo correspondiente al estado actual, y en concreto en la sección 3.2, el problema de la composición de servicios ha supuesto en la última década un gran esfuerzo de investigación para conseguir su aplicación en las redes con infraestructura tradicionales. En este tipo de redes es posible aplicar soluciones que se basen en la existencia de repositorios centrales de servicios (Brønsted, Hansen y Ingstrup, 2010). Sin embargo, estas soluciones no pueden ser aplicadas directamente a las redes MANET, redes sin infraestructura fija, debido a los cambios que pueden producirse continuamente en la topología de la red. Por ello, ha sido necesario la creación de soluciones específicas para este tipo de redes (Chakraborty, Yesha y Joshi, 2004; Basu, Ke y Little, 2002; Huang, Bai, Zhang, Kong y Qian, 2007; Hu, Tang, Wang y Ji, 2007; Ruta, Zacheo, Grieco, Di Noia, Boggia, Tinelli, Camarda y Di Sciascio, 2010; Prochart, Weiss, Schmid y Kaefer, 2007; Karmouch y Nayak, 2010; Sadiq, Kumar, Passarella y Conti, 2011).

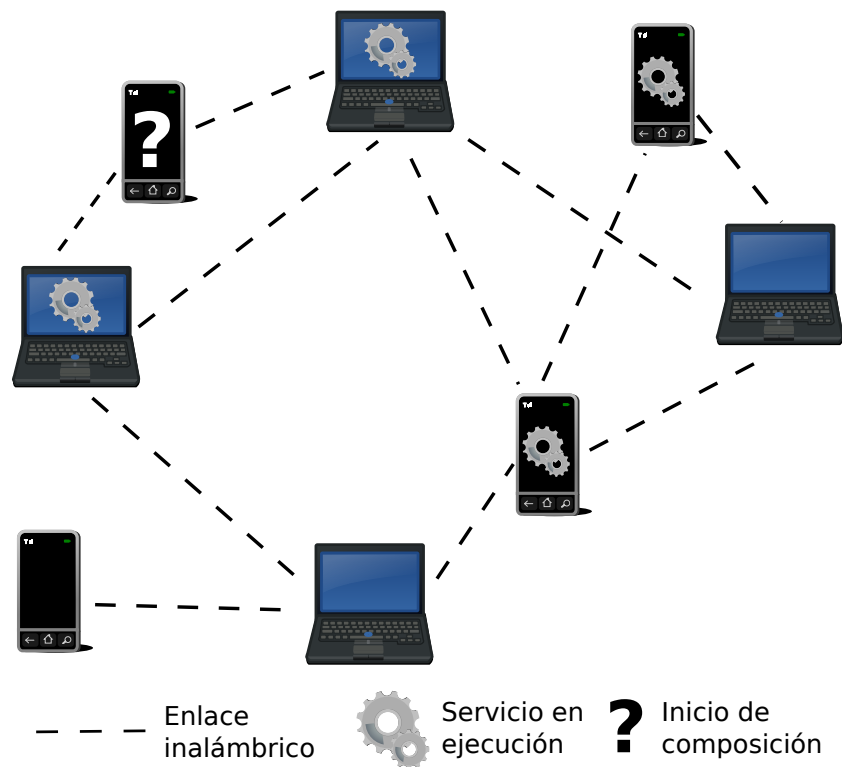
Sin embargo, las propuestas existentes en la actualidad para llevar a cabo la composición de servicios en redes móviles ad hoc se basan en la definición de un flujo de trabajo por parte del usuario. Este flujo de trabajo define los tipos y relaciones entre los servicios que forman parte del servicio compuesto de una manera abstracta. El proceso de composición consiste en la instanciación de dicho flujo de trabajo utilizando los servicios disponibles en la red móvil ad hoc. La definición de este flujo de trabajo requiere una participación del usuario, cuya minimización, es el objetivo la investigación presentada en esta tesis



doctoral. Así, la propuesta que se hace en esta investigación para la composición se basa únicamente en la especificación formal del servicio que se quiere componer mediante sus entradas y salidas.

#### 4.1.1 Caso de uso

La figura 4.2 ilustra una situación en la que diferentes dispositivos conforman una red ad hoc. Estos dispositivos ofertan diferentes servicios que pueden ser accedidos por otros dispositivos de la red. El objetivo de la composición de servicios aplicado a este tipo de redes es buscar un flujo de trabajo que utilice los servicios disponibles en la red, teniendo en cuenta la ausencia de un controlador central y la aplicación de la comunicación indirecta mediante múltiples saltos.



**Figura 4.2:** Red ad hoc con diferentes tipos de dispositivos que ofertan servicios

Por ejemplo, se supone la existencia de una aplicación que se está ejecutando en un dispositivo, como puede ser el caso de un teléfono inteligente, y que necesita acceder a cierta funcionalidad proporcionada por un servicio que está localizado en otro dispositivo de la red. Además, ambos dispositivos se encuentran participando en la misma red ad hoc, por lo que la comunicación entre ellos puede ser posible.

La aplicación puede necesitar obtener, para llevar a cabo su funcionalidad, la temperatura del entorno en grados *Celsius*. Debido a la naturaleza distribuida de las redes MANET, la aplicación no posee a *a priori* un conocimiento de todos los servicios disponibles, ni como acceder a ellos. Por esta razón es necesario utilizar primero los mecanismos de descubrimiento de servicios. Si existe un dispositivo que oferte un servicio para obtener la temperatura adecuado a las restricciones de parámetros de salida aplicadas, dicho servicio puede ser consumido por la aplicación.

Por el contrario, puede darse el caso de que no exista ningún dispositivo accesible a través de la red MANET que proporcione la temperatura en el formato buscado. Sin embargo, puede ser también posible que en el entorno exista algún dispositivo que proporcione un servicio para calcular la temperatura, pero que la salida esté en otra escala, por ejemplo, en grados *Fahrenheit*. Este servicio no podría ser utilizado directamente por la aplicación. Si la búsqueda de servicios estuviera limitada únicamente a los servicios disponibles, la aplicación no podría continuar con su ejecución al no disponer de un componente necesario para la realización de la misma.

Supongamos, sin embargo, que en la misma red de comunicación existe un dispositivo que proporciona un servicio para llevar a cabo la conversión de unidades en grados Celsius a Fahrenheit. Este servicio, podría ser utilizado en conjunción con el anterior, que proporciona la temperatura en grados Fahrenheit, para obtener un servicio compuesto que resolvería las necesidades de la aplicación. El proceso de composición, que será iniciado en el nodo que quiere acceder al servicio, posibilitará, en este caso, obtener una solución al problema, aunque esto solo será posible si tanto el servicio de temperatura como el servicio conversor pueden ser conectados a través de algún camino existente en la red MANET.

### 4.1.2 Fases de la composición

A partir del caso de uso anterior, y del estudio del estado actual llevado a cabo en la sección 3.2 del capítulo 3, pueden identificarse las siguientes fases en el proceso de composición:

1. Se supone la existencia de una red ad hoc que comunica los diferentes dispositivos de computación. Debido a las posiciones relativas entre los dispositivos, y a las características de la tecnología de comunicación inalámbrica utilizada, la topología de esta red puede ser muy diferente en cada caso. Además, dicha topología puede cambiar debido a la posible movilidad relativa entre los nodos que la constituyen.
2. Cada dispositivo puede poner a disposición del resto de nodos de la red uno o más servicios que podrán ser accedidos a través de la misma.

3. Uno o más dispositivos de la red pueden indicar, en cualquier momento, su necesidad de acceder a cierto servicio mediante la descripción de las características del mismo.
4. Se inicia un proceso de descubrimiento que localizará el servicio requerido en caso de que exista. Si esto no es posible, la composición automática de servicios ensambla los servicios disponibles en el orden adecuado para obtener un servicio compatible con las necesidades del usuario.
5. El servicio encontrado, ya sea un servicio simple o compuesto, puede ser ejecutado por el dispositivo que inició la búsqueda. En el caso de un servicio compuesto esto supone la existencia de un proceso de ejecución distribuida.
6. Los cambios en la topología de la red pueden hacer que una composición resulte inservible porque los dispositivos que proporcionan los servicios implicados ya no se encuentran accesibles. Sin embargo, dada la naturaleza de las redes ad hoc es posible que aparezcan rutas alternativas que permitan seguir accediendo a dichos dispositivos.

La ejecución continúa de servicios compuestos, comentada en el último punto, ha sido explorado anteriormente por Aguilera, Almeida, Orduña, López-de Ipiña y de las Heras (2010) en dispositivos móviles, pero sin llevar su aplicación a las redes MANET. Esta investigación se centra en la resolución de los problemas asociados a todos los puntos anteriores excepto al punto 5. Aunque queda fuera del ámbito de la investigación presentada en esta tesis, en la sección 6.4.5 del capítulo 6 se tratan algunos aspectos relaciones con los mecanismos para la ejecución de las composiciones encontradas.

### 4.1.3 Definición formal del problema

La composición de servicios consiste, en esta investigación, en la obtención de un flujo de trabajo que represente las relaciones de entrada/salida entre los servicios seleccionados. El flujo de trabajo puede representarse mediante un grafo dirigido en el que los vértices del grafo representan los servicios y las aristas el flujo de datos entre dos servicios.

En su forma más general, y planteando la composición de servicios como un problema de planificación, tal y como hacen Pistore, Barbon, Bertoli, Shaparau y Traverso (2004), se puede representar el problema de una manera más formal, mediante una tupla de cinco elementos, de una forma similar a como se ha presentado en la sección 3.2.3. En el caso de la composición de servicios, la tupla que define al problema es  $(E, S, P_{re}, E_{ff}, S_{erv})$  donde

- $E$  define la situación inicial del problema de composición a partir del conjunto de parámetros de entrada. Para que la solución encontrada sea válida es necesario que los parámetros de entrada  $E(C)$  del servicio compuesto resultante sean un subconjunto de los parámetros de entrada requeridos,  $E(C) \subseteq E$ . Puede ser posible que se encuentren soluciones que requieran menos parámetros de los especificados. Esto no se considera un problema, ya que dichas soluciones podrán ser utilizadas con el conjunto de valores de entrada disponibles.
- $P_{re}$  representa el conjunto de pre-condiciones del servicio compuesto deseado. Para que la composición encontrada sea válida es necesario que las precondiciones  $P_{re}(C)$  para el servicio compuesto sean satisfechas por las requeridas,  $P_{re}(C) \subseteq P_{re}$ . Esto significa que el servicio puede ser ejecutado en la situación actual del sistema.
- $S$  define la situación final del problema de composición a partir del conjunto de parámetros de salida. La composición cumplirá los requisitos especificados si los parámetros de salida  $S(C)$  del servicio compuesto son un superconjunto de los requeridos,  $S \subseteq S(C)$ . Es decir, el servicio compuesto debe proporcionar, al menos, el conjunto de los parámetros requeridos. Es posible que el servicio proporcione más parámetros de salida, pero estos parámetros extra se consideran un efecto secundario de la ejecución del servicio y no interfieren con la solución buscada.
- $E_{ff}$  es el conjunto de efectos que el servicio tendrá como resultado de su ejecución. En este caso la solución encontrada será válida si el servicio compuesto produce el total de los efectos deseados  $E_{ff} \propto E(C)$ . El servicio compuesto podría producir más efectos que los requeridos pero esto no siempre será una situación deseable y dependerá de la aplicación o del caso concreto que se esté contemplando.
- $S_{erv}$  es el conjunto de todos los servicios  $S_i$  disponibles en el entorno. Cada servicio  $S_i$  es a su vez una tupla  $(E_i, S_i, P_{rei}, E_{ffi})$  que define las entradas, salidas, pre-condiciones y efectos de cada uno de los servicios.

En el caso concreto de su aplicación a las redes MANET, el proceso de composición debe tener en cuenta la naturaleza distribuida de la red. Así, Hu, Muthusamy, Li y Jacobsen (2008) definen la composición de servicios en una red distribuida como la instanciación de un subgrafo de las posibles conexiones entre los servicios existentes en la red. Si  $G = (N, C)$  es un grafo tal que

- $N$  es el conjunto conformado por los nodos de la red ad hoc y que constituyen los vértices del grafo. Es decir, aquellos dispositivos con capacidades de comunicación inalámbrica que puede ofrecer uno o más servicios  $S_i$  al resto de nodos de la red.

- $C$  es el conjunto de las aristas que determinan los enlaces de comunicación existentes en un momento determinado entre dispositivos vecinos.

Así, el problema de composición, aplicado a una red MANET, consiste en encontrar un sub-grafo  $G_s \subseteq G$  tal que, los servicios  $S_i$  proporcionados por los dispositivos  $N_i$  satisfagan las restricciones de la composición y, además, el conjunto de enlaces  $C_i$  existente entre ellos permita la invocación del flujo de trabajo correspondiente.

## 4.2 Requisitos y limitaciones

De acuerdo a la revisión bibliográfica realizada en la sección 3.2, así como a partir de la definición formal del problema presentada en la sección anterior, se ha obtenido el siguiente conjunto de requisitos para la obtención de una solución adecuada al problema que se quiere resolver:

- **Descripción de servicios:** es necesario un lenguaje que permita llevar a cabo la descripción de los diferentes servicios ofrecidos por los dispositivos de la red ad hoc. Este lenguaje debe ser lo suficientemente expresivo como para capturar las características funcionales del servicio y permitir determinar, durante la composición, si un servicio es compatible con una petición de búsqueda dada.
- **Descripción de la composición:** la búsqueda de un servicio se inicia con la descripción de las necesidades del servicio a localizar. Se necesita, al igual que en el caso anterior, un lenguaje que permita determinar las características que el servicio a buscar debe cumplir. El lenguaje para la descripción formal de la composición buscada y para la descripción de los servicios individuales puede ser el mismo, esto simplificará su gestión y facilitará el posterior uso de las composiciones resultantes.
- **Registro de servicios:** cada dispositivo debe ser capaz de registrar las descripciones de los servicios que proporciona y ponerlas a disposición del resto de la red ad hoc. Este debe ser un proceso dinámico, ya que es posible que, durante su tiempo de pertenencia a la red, los servicios provistos por un dispositivo pueden sufrir cambios.
- **Gestión de la búsqueda:** aquellos dispositivos que estén interesados en realizar búsquedas, tanto de servicios simples como compuestos, deben ser capaces de iniciar y detener el proceso en cualquier momento. La búsqueda de servicios debe ser automática, en el sentido de que, si bien es iniciada por un dispositivo interesado en resolver un problema, se lleva a cabo sin la intervención de dicho nodo y del usuario o aplicación que resida en el mismo.

El proceso de composición debe tener, por lo tanto, las siguientes características:

- **Automático:** la composición de servicios debe minimizar la participación del usuario en el proceso. El proceso debe comenzar con la definición de las características del servicio buscado y, a partir de éstas, obtener el flujo de trabajo. La solución no debe utilizar una definición de flujo de trabajo *a priori* que obligue a los usuarios, o a los desarrolladores de las aplicaciones, a su definición concreta.
- **Exhaustivo:** el proceso de composición tiene como objetivo buscar todas las soluciones compatibles con una búsqueda iniciada por un dispositivo. Dichas soluciones son entregadas al nodo que inició la búsqueda según vayan siendo encontradas por el mecanismo de búsqueda. Obviamente, solamente pueden encontrarse aquellas soluciones que la topología de la red permita construir; si existen nodos que no se encuentran conectados al resto de la red no podrán formar parte de las posibles soluciones.
- **Distribuido:** se busca que el proceso de composición sea completamente distribuido, con el objeto de minimizar los posibles fallos que puedan existir debido al uso de gestores centralizados ante posibles cambios en la topología de la red.
- **Dinámico:** debido a la movilidad de la red, es posible que las soluciones encontradas ya no sigan siendo válidas. La solución propuesta debe tener en cuenta la movilidad de los nodos de la red ad hoc con el objeto de responder a dichos cambios y posibilitar que las soluciones encontradas sigan siendo accesibles.

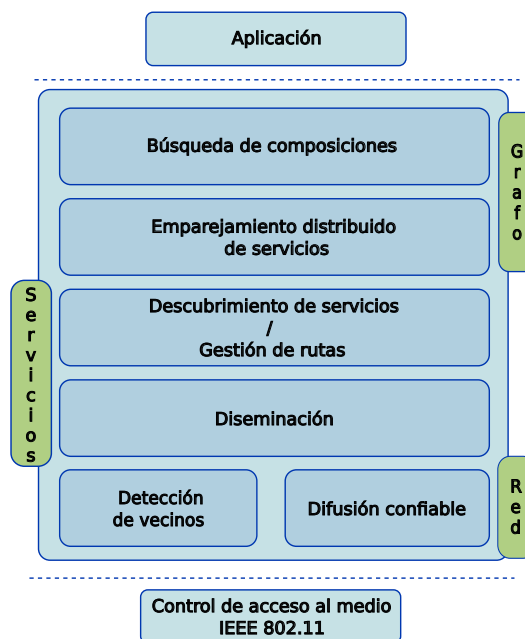
Así mismo, existen otros aspectos y características que no han sido contemplados en la realización de la solución:

- **Solución óptima:** la propuesta no tiene como objetivo la búsqueda de la *mejor solución*. Como se ha comentado anteriormente, el proceso de búsqueda debe ser exhaustivo y permitir encontrar todas las soluciones compatibles con la búsqueda realizada. Sin embargo, es responsabilidad del nodo que inició la composición determinar, por otros procedimientos, cuál es la solución que mejor se adecua a sus necesidades.
- **Solución correcta:** como en el caso anterior, la propuesta realizada no tiene en cuenta si la composición obtenida es válida desde un punto de vista semántico, es decir, si realmente una solución obtenida tiene el efecto requerido o, si por el contrario, realiza otras acciones asociadas y no deseadas. Los aspectos relacionados con la validez de las soluciones obtenidas se discuten en la sección 4.3.6.

- **Monitorización de la ejecución:** aunque las soluciones obtenidas pueden ser ejecutadas directamente sobre la red ad hoc, la propuesta realizada se centra únicamente en la búsqueda de composiciones. Los aspectos relacionados con la ejecución de servicios compuestos en una red de nodos distribuida han sido recogidos en la sección 3.2.8 del estado actual y en la sección 6.4.5 del capítulo 6.

## 4.3 Arquitectura

A partir de los requisitos recogidos en la sección 4.2 se propone la arquitectura esquematizada en la figura 4.3. Esta arquitectura está constituida por 5 capas que trabajan en conjunción con el objetivo de resolver el problema de la composición de servicios en una red MANET. Como es habitual en el diseño por niveles, cada una de las capas proporciona una funcionalidad que es utilizada por los niveles.



**Figura 4.3:** Arquitectura propuesta para la composición automática de servicios en redes MANET

Para posibilitar la visión de la composición de servicios, propuesta en esta solución, todos los nodos de la red ad hoc deben contener la implementación de la arquitectura completa. Cada uno de los niveles de la arquitectura se corresponde con un protocolo que resuelve una problemática particular, definiendo una serie de mensajes y funcionalidades. Con la finalidad de reducir el número

de transmisiones desde un nodo, los mensajes producidos por un nivel pueden ser incluidos en el *área de datos* de los mensajes de las capas inferiores.

Debido a que todos los nodos de la red implementan la misma arquitectura para la composición de servicios, no existe una distinción entre las funcionalidades que cada nodo puede realizar ni jerarquías de control entre los diferentes nodos. La solución propuesta requiere la colaboración de todos los nodos implicados y es igualitaria en el reparto que hace de las tareas realizadas. Las actuaciones concretas de cada nodo dependen únicamente de las situaciones que se vayan produciendo en la red, y no de una configuración previa que selecciona un comportamiento específico.

La arquitectura propuesta ha sido construida directamente sobre la capa de acceso al medio, tal y como se explica en la sección 2.1.2. En concreto, se ha tomado como base para el diseño el protocolo *MAC IEEE 802.11*. Por lo tanto, la arquitectura no utiliza la funcionalidad proporcionada por ninguno de los protocolos de encaminamiento existentes en la actualidad, que han sido resumidos en la sección 2.1.3 de la introducción a las características de las redes MANET. La razón para haber tomado esta decisión es que, tras estudiar las propuestas de protocolos actuales, ninguna de las soluciones existentes en la actualidad está orientada a la búsqueda de recursos de la forma en la que es necesaria para la composición de servicios, tal y como se quiere llevar a cabo en esta propuesta. Por ello, se ha decidido que es necesaria la propuesta de una arquitectura integral para la composición de servicios que resuelva las situaciones de una forma adecuada. Las características de esta solución se discuten en el capítulo 5.

Las aplicaciones que necesiten llevar a cabo el proceso de composición de servicios deben situarse en la parte superior de la arquitectura. Así, podrán hacer uso de la funcionalidad proporcionada por las diferentes capas inferiores. Con objeto de abstraer a los desarrolladores de aplicaciones de los aspectos relacionados con la búsqueda de composiciones, y también con el descubrimiento de los mismos, la arquitectura proporciona un conjunto de funciones de gestión muy simples. Estas funciones son detalladas en el capítulo 5 y el capítulo 6.

Como se representa en la figura 4.3, existen tres partes claramente diferenciadas en la arquitectura propuesta. Primeramente, la *gestión de red*, que se encarga de aquellos aspectos más básicos de la comunicación usando el medio inalámbrico. A este nivel se resuelven aspectos como la *detección de vecinos* y la *difusión confiable*, que son explicados con mayor detalle en la sección 4.5. Por encima de la gestión básica de la red se encuentra la funcionalidad relacionada con la *administración de servicios*. Esta funcionalidad se concreta en las capas de *diseminación*, *descubrimiento de servicios* y *gestión de rutas*, explicadas en el capítulo 5. Por último, en el tercer nivel de la arquitectura propuesta se sitúan los aspectos relacionados con la *gestión del grafo distribuido de servicios*. Esta última parte de la arquitectura realiza el *emparejamiento distribuido de servicios* y la *búsqueda de composiciones*, tratadas en mayor detalle en el capítulo 6.



### 4.3.1 Detección de vecinos

La solución propuesta requiere que los dispositivos sean capaces de conocer en cada momento cuáles son los vecinos con los que pueden establecer una comunicación. Es decir, con aquellos dispositivos que se encuentran a un salto de distancia según las capacidades de la tecnología de comunicación inalámbrica. Así, la solución propuesta se basa en la detección de los eventos de *aparición* y *desaparición* de vecinos para llevar a cabo sus procesos. Normalmente, los cambios en los vecinos de un nodo supondrán una actualización en la configuración interna de dicho nodo y la posible propagación de mensajes a través de la red.

El proceso de detección de vecinos no es necesario únicamente en el caso de que la red posea cierta movilidad. Si la red ad hoc es estática, por ejemplo en el caso de una red de sensores, la detección automática de vecinos evita un proceso inicial de configuración manual de la topología de la red. Además, como se ha comentado anteriormente, durante el tiempo de vida de cualquier red ad hoc, se pueden producir cambios en su estructura, debido a las apariciones de nuevos vecinos o a las desapariciones de los existentes. Esto puede producirse por motivos tan diversos como la movilidad relativa de los nodos de la red, fallos temporales en las comunicaciones o, incluso, su desconexión definitiva.

Por lo tanto, en esta investigación se propone, como mecanismo para la detección de los vecinos de un nodo, la utilización de *balizas periódicas* (*beaconing* o *heartbeating* en inglés). Así, los nodos propagan mediante difusión mensajes especiales de forma periódica. Siempre que un nodo recibe un mensaje, ya sea una baliza o de otro tipo, que provenga de otro nodo de la red, concluirá que dicho nodo es su *vecino*. Por el contrario, cuando un nodo no recibe, durante un periodo de tiempo determinado, mensajes originados en un vecino detectado anteriormente, se supondrá que ya no puede establecerse una comunicación con dicho nodo. La detección de la aparición y desaparición de los nodos vecinos produce *eventos* que son utilizados por las capas superiores para responder en consecuencia. Las implicaciones del uso de mensajes periódicos en la solución propuesta son discutidas con mayor detalle en la sección 4.5.1.

### 4.3.2 Diseminación

La información sobre los servicios que proporciona cada dispositivo es, inicialmente, conocida únicamente por el propio nodo. El cometido de esta capa es diseminar la información referente a los servicios de un dispositivo a través de la red MANET. La solución propuesta no depende de la existencia de un repositorio central en el que se lleve a cabo el registro de los diferentes servicios ofertados por cada dispositivo. Por el contrario, la información sobre los servicios de cada dispositivo es propagada, mediante mensajes, a sus vecinos. La diseminación de mensajes es iniciada por cada nodo que los proporciona y difundida salto a salto a través de la red ad hoc.

Uno de los objetivos en la propuesta de esta capa ha sido la reducción en el número de mensajes que deben transmitirse a través de la red para llevar a cabo la diseminación de la información de los servicios. Por este motivo, se ha propuesto un protocolo que, aplicando una serie de agrupamientos de la información a diseminar, consigue reducir el número de mensajes necesarios para llevarla a cabo. El funcionamiento de la capa de diseminación se materializa en un protocolo basado en los parámetros de los servicios y que es explicado con mayor profundidad en el capítulo 5.

### 4.3.3 Descubrimiento de servicios y gestión de rutas

Este nivel ofrece funcionalidad a los nodos para localizar aquellos servicios que tengan unas características determinadas. El proceso de búsqueda puede ser iniciado por una capa superior o por una aplicación que quiera hacer uso de esta funcionalidad. Para iniciar una búsqueda es necesaria una especificación del servicio que se quiere localizar, que será utilizada, en conjunción con las descripciones de los servicios ofertados por cada dispositivo, para determinar si existe compatibilidad entre ellos. Las particularidades de la descripción utilizada para los servicios, así como las especificaciones en las búsquedas son detalladas en la sección 4.4.

El inicio de una búsqueda produce la propagación, desde el nodo que la ha comenzado, de un mensaje que contiene la especificación del servicio a localizar. Este mensaje es recibido, y propagado, por los vecinos de dicho nodo que deben determinar si ofertan servicios compatibles con la especificación recibida. Si el nodo provee servicios compatibles se produce un mensaje de respuesta que es enviado al nodo inicial. Dicho mensaje contiene la dirección del nodo que oferta el servicio localizado, junto la descripción completa de este último.

Los mensajes de búsqueda son propagados a través de la red utilizando la información que ha sido difundida por la capa de *Diseminación* explicada anteriormente. Cada nodo de la red, que ha recibido información sobre los servicios ofertados por otro nodo gracias al funcionamiento de la capa de diseminación, contiene una tabla con información de las distancias estimadas, en número de saltos, a los nodos que ofrecen servicios compatibles. Esta información es utilizada en conjunción con un contador de tiempo de vida (*Time To Live - TTL*), que determina si un mensaje de búsqueda debe seguir siendo propagado a través de la red. Como se muestra en el capítulo 5, esto se traduce en una reducción en el número de mensajes propagados durante las tareas de búsqueda en la red. Los detalles del funcionamiento de esta capa son explicados en dicho capítulo.

Esta capa gestiona también la creación y el mantenimiento de las rutas de comunicación disponibles entre los diferentes nodos durante la movilidad de la red MANET. Cada vez que un mensaje de búsqueda, o de respuesta a una búsqueda, es recibido por un nodo, se llevan a cabo las actualizaciones de las

tablas de encaminamiento locales. Las tablas de encaminamiento son actualizadas también en respuesta a los eventos producidos por la capa de *Detección de Vecinos*. Esta capa proporciona, asimismo, la funcionalidad para enviar mensajes de *unicast* o *multicast*.

#### 4.3.4 Emparejamiento distribuido de servicios

La composición automática necesita determinar las relaciones que se establecen entre diferentes servicios, con el fin de determinar qué conexiones pueden crearse entre ellos. Por emparejamiento se entiende el proceso por el cual se determina si dos servicios pueden ser conectados entre sí. En el caso de las soluciones centralizadas vistas en la sección 3.2.2, el emparejamiento de servicios puede llevarse a cabo de una forma exhaustiva comprobando las relaciones que existen entre todos los servicios registrados en el mismo repositorio. Sin embargo, en el caso de las redes ad hoc, los servicios se encuentran distribuidos a través de toda la red. Por esta razón, no existe una forma directa de llevar a cabo el emparejamiento de servicios en este tipo de redes.

En la propuesta realizada en el capítulo 6, el emparejamiento de servicios es realizado de forma colaborativa por los diferentes nodos de la red. En concreto, serán los nodos intermedios, es decir aquellos nodos que se encuentren en una ruta entre dos servicios compatibles los que llevarán a cabo la detección del emparejamiento. El objeto final de esta capa es crear un grafo distribuido a través de la red MANET que represente las conexiones que pueden establecerse entre los diferentes servicios ofertados. Dicho grafo distribuido de servicios es mantenido frente a los cambios en la topología de la red y utilizado durante el proceso de búsqueda de composiciones. La aplicación de los grafos de dependencia de servicios a las redes MANET es otra de las aportaciones de esta investigación y es explicado con mayor detalle en el capítulo correspondiente.

#### 4.3.5 Búsqueda de composiciones

Cada vez que un nodo quiere llevar a cabo la búsqueda de un servicio compuesto debe especificar las características de dicho servicio. Esta especificación del servicio requerido es realizada tal y como se explica en la sección 4.4. La descripción es utilizada por esta capa para obtener una solución que, en base a los servicios disponibles en la red, proporcione un servicio compuesto que satisfaga los requisitos indicados.

El proceso de composición consiste en una búsqueda a través del grafo distribuido de servicios. El nodo que inicia la búsqueda de una composición debe registrar dos servicios especiales que sirven como inicio y final de una composición y que, utilizando los procedimientos para el emparejamiento de servicios llevados a cabo por la capa inferior, pueden dar como resultado conexiones con

otros servicios compatibles. Obviamente, esto depende de que en la red existan servicios compatibles con la búsqueda que se está llevando a cabo. La búsqueda de la solución a la solicitud de composición comienza como un mensaje que es propagado desde el nodo que inicia el proceso. El objetivo es encontrar un sub-grafo, dentro del grafo distribuido de servicios, que satisfaga las necesidades de la composición. El proceso concreto por el cual se lleva a cabo la búsqueda de composiciones es explicado en el capítulo 6.

### **4.3.6 Validez de las soluciones**

La propuesta realizada en esta investigación para la composición de servicios en redes ad hoc se centra únicamente en la utilización de la información funcional de las entradas y salidas de los servicios. Es decir, la composición obtenida garantiza que si, al servicio compuesto obtenido, se le proporcionan el número y tipo adecuado de las entradas se obtendrá, tras la ejecución de la composición, las salidas requeridas. Por lo tanto, deben ser los nodos que reciben las composiciones los que determinen si el resultado obtenido se adecua a otros requisitos antes de llevar a cabo su ejecución.

Existen otros aspectos que pueden tenerse en cuenta a la hora de seleccionar una determinada composición entre todas las disponibles, como por ejemplo, el número total servicios participantes en la composición, el costo en saltos dentro de la red cuando se ejecute, el uso de ciertos nodos que tienen un mayor consumo energético, etc. Sin embargo, hay que tener en cuenta que muchos de estos aspectos únicamente pueden ser calculados con fiabilidad cuando se lleva a cabo un recorrido del grafo de la solución. Por lo tanto, la solución propuesta se basa en la construcción de un grafo distribuido en la que cada servicio únicamente tiene conocimiento de aquellos otros que se encuentran conectados directamente con él. Desde cada uno de los servicios que forman parte del grafo no es posible calcular los aspectos anteriores, ya que requieren llevar a cabo un recorrido del grafo en parte o en su totalidad. Se incluyen más detalles sobre esta limitación en la sección sección 7.2.2 de las líneas abiertas de investigación de esta tesis doctoral.

Una solución más adecuada para este tipo de redes es llevar a cabo, durante el propio proceso de búsqueda de las composiciones, el cálculo de los indicadores para la validez de la solución. Por ejemplo, el usuario puede indicar que no quiere soluciones que tengan una distancia en saltos a través de la red mayor que cierto límite especificado o limitar el número de servicios que participan en una composición. Las búsquedas cuya solución no cumple estos requisitos son detenidas y no se continúa el proceso. Este es el enfoque seguido en la solución propuesta, tal y como se explica en el capítulo 6.

## 4.4 Descripción de servicios

Es necesario definir cuáles son las características de un servicio tal y como se entienden en el ámbito de esta investigación y, además, qué aspectos del mismo deben ser modelados para que la propuesta pueda llevarse a cabo de forma satisfactoria. Se entiende por servicio cualquier funcionalidad que un dispositivo pueda ofrecer a otros que se encuentran en la red. Esta funcionalidad debe poder ser accedida de forma remota y permitir la obtención de información o la realización de acciones sobre el dispositivo que la ofrece. La visión de servicio en esta investigación es cercana a la de un método remoto que recibe ciertos entradas y proporciona ciertos parámetros de salida como resultado.

En la definición anterior no se ha hecho referencia a la tecnología mediante la cual se proporcionan dichos servicios. Por ejemplo, existen tecnologías concretas, como es el caso de los *servicios web*, en las cuales se entiende que el acceso y uso de los servicios se lleva a cabo mediante el empleo de mecanismos como *HTTP* sobre protocolos de comunicación como *IP*. También existen otras soluciones como *REST*, en los que también se usa *HTTP* para el acceso a servicios/recursos, pero en este caso de una forma mucho más simple que en la anterior, ya a que no requiere la gestión de mensajes XML ni el procesamiento de contenedores de mensajes complejos.

Con objeto de abstraer el desarrollo de la investigación de aspectos como la comunicación con los servicios disponibles en la red, no se supone la existencia de una tecnología concreta para llevar a cabo el acceso a los servicios. Únicamente se tendrán en cuenta, para cada servicio, los aspectos relacionados con el número y tipo de los parámetros de entrada y salida que cada servicio define.

En esta investigación, la definición del tipo de los parámetros de entrada y salida para un servicio está tomada del concepto de *servicio web semántico*. Estos son servicios web que han sido descritos semánticamente aplicando los conceptos introducidos por la Web Semántica. Para más información sobre la descripción de los servicios, se puede consultar la sección 3.1.5 y la sección 3.2.5 de la revisión bibliográfica contenida en el capítulo correspondiente.

La principal ventaja de un servicio web semántico es que su descripción se encuentra en un lenguaje que puede ser procesado fácilmente de forma automática y que, además, permite determinar las relaciones entre conceptos utilizados en la descripción. Esta facilidad posibilita mejorar la localización de servicios web, la creación de servicios compuestos, así como lograr la interoperabilidad entre los servicios. Por último, hay que hacer destacar que, los lenguajes de anotación semántica de servicios web, como OWL-S y WSMO (W3C Member Submission, 2005), permiten describir no solamente las entradas y salidas de un servicio, sino también sus precondiciones y sus efectos.

### 4.4.1 Aspectos de un servicio

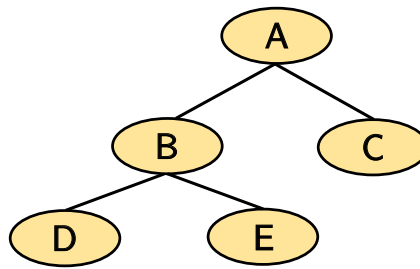
Los aspectos mínimos que un servicio debe satisfacer para poder ser utilizado en la composición de servicios, tal y como se define en esta investigación, son los siguientes:

- **Identificador único:** cada servicio debe estar identificado de forma única mediante un identificador. Es decir, dos servicios que tengan el mismo identificador se considera que representan al mismo servicio. Además, con objeto de identificar los servicios de forma única a través de la red, el nombre de un servicio es utilizado en conjunción con el identificador único del nodo. Así, un dispositivo no puede ofrecer más de un servicio con un mismo identificador completo, constituido por el servicio y su nodo.
- **Parámetros de entrada:** un servicio puede definir  $n$  parámetros de entrada. Cada uno de estos parámetros de entrada tiene un tipo que viene definido según la ontología de conceptos utilizada.
- **Parámetros de salida:** la ejecución de un servicio puede dar como resultado  $n$  parámetros de salida. Al igual que en el caso de los parámetros de entrada, su tipo viene determinado también por una ontología de conceptos.
- **Funcionalidad:** cada servicio dispone de una funcionalidad que puede ser ejecutada cuando se satisfaga el número y tipo de los parámetros de entrada requeridos por el servicio. La tecnología concreta para el despliegue y posterior ejecución del servicio por parte de los dispositivos de la red no es un aspecto crucial y, por lo tanto, no se incluye ninguna limitación en este punto.

### 4.4.2 Ontología de conceptos

En esta investigación se propone la aplicación de una ontología de conceptos con el objeto de clasificar los tipos de los parámetros de los diferentes servicios existentes en la red ad hoc. Todos los parámetros de un servicio, ya sean de entrada o salida, deben tener un tipo asociado y contenido en dicha ontología. El uso de clasificaciones de servicios no es nuevo en la literatura informática, como puede verse en la sección 3.1.5 y sección 3.2.5 del estado actual, sin embargo, sí que lo es la forma en la que esta información es utilizada para mejorar los procesos de diseminación y búsqueda de los servicios en una red MANET.

Una ontología puede representarse de diferentes formas para su procesamiento automático por parte de un computador. Existen lenguajes para la representación de las relaciones entre diferentes conceptos, como pueden ser RDF u OWL (W3C OWL Working Group, 2009). Algunos de estos lenguajes, como es



**Figura 4.4:** Representación de las relaciones jerárquicas entre los conceptos de una ontología

el caso de los dos nombrados anteriormente, permiten establecer no solamente relaciones de especialización entre conceptos, sino también otras relaciones más complejas entre los mismos, tal y como exponen Gruninger, Bodenreider, Olken, Obrst y Yim (2008). Sin embargo, en esta solución únicamente se hace uso de dichas relaciones de especialización entre conceptos.

Según el trabajo de Paolucci, Kawamura, Payne y Sycara (2002), mediante el uso de una ontología de conceptos es posible determinar si existen diferentes relaciones entre los tipos de los parámetros de dos o más servicios. La figura 4.4 muestra las relaciones de jerarquía que establece una ontología entre los términos que la constituyen. Por ejemplo, supongamos que tenemos dos parámetros  $P_1$  y  $P_2$ , de acuerdo a información contenida en la ontología utilizada pueden darse las siguientes relaciones entre conceptos:

- **Igualdad:** ocurre cuando  $P_1$  y  $P_2$  tienen el mismo tipo,  $T(P_1) \equiv T(P_2)$ .
- **Subsunción:** sucede cuando el tipo del parámetro  $P_1$  es más general que el tipo del parámetro  $P_2$ . Es decir,  $P_1$  se corresponde con un concepto que engloba a  $P_2$ , representado por  $T(P_2) \sqsubseteq T(P_1)$ . Por ejemplo, en una ontología de medios de transporte, el concepto *vehículo* subsume al concepto *automóvil*, ya que es un concepto más general. Por ejemplo, en la figura 4.4 esto puede suceder cuando el tipo de  $P_1$  es A y el tipo de  $P_2$  es B, o incluso D.
- **No relacionados:** ocurre cuando la ontología de conceptos no establece ninguna relación de *jerarquía* directa entre el tipo de los parámetros. En la figura esto sucede, por ejemplo, si el tipo de los parámetros es B y C.

Cada uno de los dispositivos involucrados en la composición de servicios debe ser capaz de interpretar la ontología de tal forma que pueda responder a las relaciones anteriormente establecidas. Con objeto de no depender de una tecnología concreta para la realización de los experimentos realizados en el capítulo 5 y capítulo 6, se ha hecho uso de un lenguaje muy sencillo para la

representación de las ontologías que se limita a establecer las relaciones padre-hijo entre diferentes conceptos.

El proceso para determinar la compatibilidad entre dos servicios se explica en la sección 6.3. La existencia de una ontología de servicios conocida por los diferentes dispositivos que participan en la composición es uno de los requisitos de la solución propuesta. Todos los dispositivos deben conocer *a priori* dicha ontología, ya que se supone que todos los dispositivos comparten una misma ontología de conceptos con el objeto de formar parte de la red de dispositivos. Aquellos nodos que no compartan la misma ontología con el resto no podrán participar en el proceso de composición de servicios. Una solución a este problema puede ser el compartir, mediante un proceso de diseminación a través de la red, la ontología utilizada por dichos dispositivos. El modo en el que los nodos de la red comparten la ontología de conceptos utilizada se deja como una línea de trabajo futuro y es explicado, con algo de mayor detalle, en la sección 7.2.1.

## 4.5 Gestión de red

En este apartado se presentan los aspectos relativos a la gestión de red en la solución propuesta. Sobre estos mecanismos se construye la solución para la composición de servicios en una red MANET, que es presentada en los siguientes capítulos. Sin embargo, ya que dichos elementos son comunes a toda la arquitectura, son definidos en esta sección. Se incluyen tres aspectos fundamentales en la gestión de la red: la *detección de vecinos*, la *difusión confiable* y la *gestión de los mensajes*.

### 4.5.1 Detección de vecinos

Como se explica en el capítulo 5 y el capítulo 6, el protocolo propuesto para la composición de servicios es *proactivo*. Esto significa que necesita reaccionar frente a los cambios que se producen en la red con el objeto de proveer su funcionalidad. Por ejemplo, durante la fase de diseminación de la información sobre los servicios proporcionados por un dispositivo, se hace uso del conocimiento que un dispositivo tiene sobre sus *vecinos* en la red. El protocolo propuesto disemina y mantiene información sobre los servicios existentes durante la movilidad de la red haciendo frente a las apariciones y desapariciones de nodos que puedan producirse. Esto también es aplicable a los procesos de búsqueda de servicios y mantenimiento del grafo de composición. En ambos casos es necesario que cada nodo conozca en todo momento qué nuevos dispositivos aparecen en las cercanías y cuáles han desaparecido, todo ello con el objeto de añadir nueva información o eliminar aquella que ha quedado obsoleta.



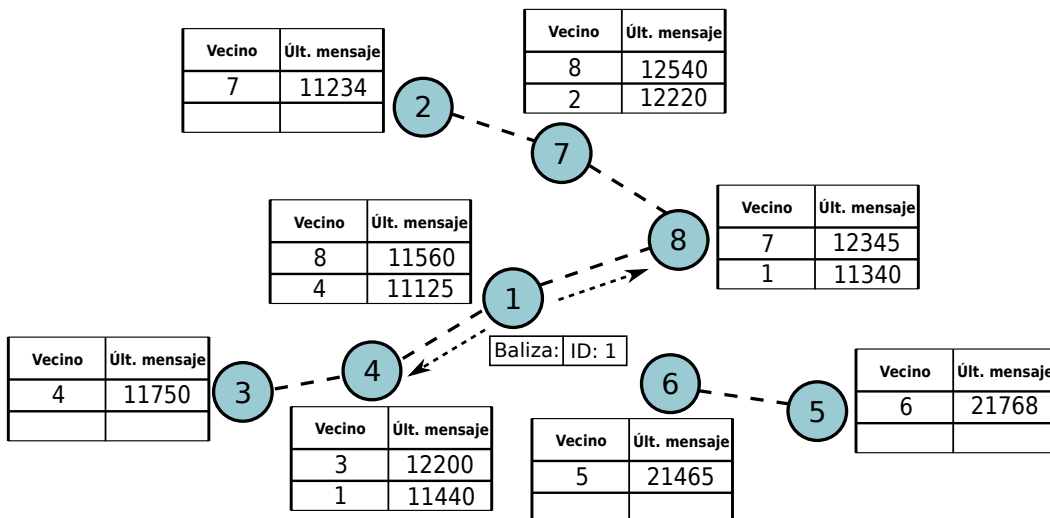
Como se ha explicado anteriormente, por *vecino* de un nodo se entiende aquellos otros nodos que se encuentran a distancia de un salto, de acuerdo a las características de la tecnología de comunicación utilizada. La detección de vecinos es el proceso por el cual los nodos de la red determinan qué dispositivos se encuentran a distancia de un salto (comunicación directa) con ellos, y es uno de los mecanismos mediante el cual los nodos pueden obtener información sobre la topología de la red.

Para llevar a cabo la detección de vecinos es necesario el mantenimiento de una tabla que contenga información sobre la topología de la red. Existen dos aspectos fundamentales en estas tablas: qué información se mantiene y cuál es la frecuencia de actualización. La gestión de los vecinos directos requiere que cada nodo mantenga una tabla con los identificadores únicos de los nodos vecinos detectados. La frecuencia de actualización de la información de la tabla viene determinada por la movilidad de los nodos de la red ad hoc.

El mecanismo para la detección de vecinos, propuesto para su utilización en esta investigación, se basa en la utilización de *balizas periódicas* que son enviadas por cada nodo de la red. Hay que tener en cuenta que el medio inalámbrico es compartido y, por lo tanto, todos los nodos que se encuentren en un determinado radio del emisor pueden recibir un determinado mensaje. Así, la recepción de un mensaje por parte de un nodo indica que dicho nodo y el emisor se encuentran a distancia de comunicación de un salto entre sí, de tal forma que el nodo receptor podrá considerar al nodo emisor como su vecino tras la recepción del mensaje. Una vez que un nodo ha detectado un vecino espera la recepción periódica de nuevos mensajes de baliza provenientes de dicho vecino. En caso de que no se reciban nuevos mensajes en un tiempo determinado se supondrá que ya no es posible mantener la comunicación con dicho nodo y que, por lo tanto, el nodo ya no es vecino del nodo actual.

Cada nodo mantiene una tabla en la que cada entrada tiene dos valores: el identificador único de los vecinos detectados y el instante, relativo al tiempo local de cada nodo, en el que se recibió el último mensaje proveniente de dicho vecino. Esta tabla representa el conjunto de los vecinos que son conocidos actualmente por un nodo. Todos los mensajes enviados por los nodos, ya sean mensajes de baliza periódicos, o cualquier otro tipo de mensaje, contienen siempre el identificador único de dicho nodo. Cada vez que un mensaje proveniente de un vecino desconocido es escuchado por un nodo, se añadirá el identificador del *nodo origen* del mensaje a la tabla y se creará una entrada nueva que incluirá el identificador único del nuevo nodo y el instante en el cual dicho mensaje ha sido escuchado.

En la figura 4.5 se muestra un ejemplo del contenido de las tablas de vecinos. Cada tabla mantiene el identificador único de aquellos nodos detectados, así como el instante en el que se ha recibido el último mensaje desde dicho vecino. El instante de recepción es local al nodo y, debido a que no se realiza



**Figura 4.5:** Contenido de las tablas de vecinos para una configuración determinada de la red

ninguna sincronización de los relojes de cada nodo, solamente tiene significado de forma interna a cada uno. En el caso del ejemplo, los nodos han sido iniciados más o menos en el mismo instante, por lo que sus tiempos de recepción se muestran como próximos entre sí. Sin embargo, en una aplicación real, aunque dos nodos reciban un mismo mensaje de manera simultánea, podrán tener tiempos internos de recepción muy diferentes. Por otro lado, en el ejemplo se muestra también como el nodo 1 lleva a cabo el envío de un mensaje de baliza. Este mensaje será recibido por aquellos nodos que se encuentren a distancia de transmisión del emisor de la baliza, en este caso, los nodos 4 y 8.

Los nodos envían mensajes de baliza con una periodicidad de  $T_b$  segundos. Con objeto de reducir el número de mensajes de baliza enviados por un nodo se aplican una serie de mejoras. Primero, cualquier mensaje enviado por un nodo puede actuar como un mensaje de baliza. Esto significa que los mensajes de baliza específicos solamente son enviados por un nodo si dicho nodo no había enviado otro mensaje en el tiempo  $T_b$ . Como se ha comentado anteriormente, la naturaleza compartida del medio inalámbrico hace que cualquier mensaje enviado por un nodo sea escuchado por los vecinos que se encuentren a un salto según la distancia de comunicación. Así, todos los mensajes, sean del tipo que sean, que hayan sido escuchados por otros nodos, habrán servido para descubrir que existe una comunicación directa con los emisores.

Cada vez que un nodo recibe un mensaje proveniente de un vecino que ya es conocido anteriormente, es decir, que se encuentra en su tabla de vecinos, se procede a actualizar la entrada correspondiente. Se utiliza para ello el tiempo de recepción del último mensaje recibido para actualizar la entrada correspon-

diente. Mientras se reciban mensajes de un vecino, dentro de una periodicidad máxima, no habrá cambios en la tabla de vecinos. Sin embargo, si un nodo no escucha ningún mensaje, de un vecino determinado, durante un periodo de tiempo  $T \geq 2 \cdot T_b$ , se considera que dicho nodo es inaccesible y que, por lo tanto, deberá ser eliminado de la tabla de vecinos.

El valor concreto de  $T_b$  puede ajustarse según las características de la red MANET. La problemática en el uso de mensajes de baliza para la detección de vecinos tiene sus ventajas e inconvenientes, los cuales han sido estudiados anteriormente por Heissenbüttel, Braun, Wälchli y Bernoulli (2007), y por Giruka y Singhal (2005). Valores muy pequeños del periodo de actualización tienen como consecuencia una reducción de la cantidad de información obsoleta que es mantenida en las tablas. Los nodos tendrán un conocimiento más exacto de los cambios que se produzcan debido a la movilidad de la red, sin embargo, esto repercute en el consumo de energía de cada nodo, ya que se aumenta el número de mensajes transmitidos en un periodo de tiempo determinado.

Por el contrario, el aumento del periodo de transmisión de las balizas hace que se reduzca el número de mensajes enviados, aunque resulta en el mantenimiento de información inexacta en las tablas de vecinos. Cuanto mayor es el periodo de actualización mayor es también la posibilidad de que un nodo contenga en su tabla vecinos que ya no se encuentran accesibles. El periodo de envío de mensajes de baliza debe estar relacionado con la movilidad de la red. Así, redes con mayor movilidad deben tener un periodo de actualización reducido, mientras que puede reducirse en aquellas más estáticas.

Cada vez que se detecta un nuevo vecino, o se elimina un vecino existente de la tabla de vecinos, se produce un evento que se notifica a las capas superiores. Estos eventos son utilizados por las distintas partes de la solución propuesta para llevar a cabo su funcionalidad. En concreto, se notifica la lista de identificadores de aquellos nodos que han sido detectados recientemente o aquellos para los que se ha determinado que ya no es posible establecer comunicación con ellos. El movimiento de los nodos en la red puede resultar en la aparición o desaparición simultánea, dentro de un corto intervalo de tiempo, de varios vecinos. Los cambios en la tabla de vecinos son notificados con un pequeño retraso, lo que permite agrupar distintos eventos que se producen casi simultáneamente en uno solo, que será notificado a las capas superiores, reduciendo así el número de eventos producidos como resultado.

### 4.5.2 Difusión confiable

A pesar del control de acceso al medio llevado a cabo por el protocolo *IEEE 802.11*, no es posible garantizar la recepción de todos los mensajes enviados a través del medio inalámbrico. Problemas como las interferencias, la disminución de la potencia de la señal con la distancia, así como los fenómenos de

reflexión, refracción y bloqueo, tienen un gran impacto en la transmisión de la señal. La intensidad de la señal puede resultar tan disminuida, o su contenido afectado de tal forma, que esta no pueda ser interpretada correctamente por los dispositivos receptores.

Por otro lado, existen problemas que no pueden ser gestionados en una red ad hoc desarrollada sobre el protocolo *IEEE 802.11*. Es el caso de los *terminales ocultos* (*hidden terminals*) que se refiere a aquellos nodos que no son accesibles desde el emisor, pero sí son accesibles desde el receptor del mensaje. Es decir, dos posibles emisores no alcanzables entre sí pueden intentar de forma simultánea transmitir a un nodo vecino común a ambos. Ya que los nodos emisores no se conocen entre ellos, no pueden llevar a cabo un control eficiente de la transmisión, pudiendo resultar en colisiones no detectadas de mensajes. Como explican Li, Blake, De Couto, Lee y Morris (2001), la existencia de *terminales ocultos* reduce considerablemente el rendimiento de una red ad hoc debido a las colisiones producidas en los mensajes transmitidos.

En un medio inalámbrico compartido no es posible saber por parte del emisor, sin un mecanismo de control más avanzado, si los mensajes han sido recibidos por los vecinos destinatarios. Esta investigación se basa en la existencia de un mecanismo que permita la difusión confiable de mensajes a los nodos vecinos. Es decir, que un nodo pueda tener garantías de que su transmisión ha sido recibida por todos aquellos nodos vecinos y destinatarios del mismo.

Por ejemplo, el descubrimiento de servicios del capítulo 5 necesita del uso de la *difusión confiable* debido al hecho de que los mensajes de actualización son incrementales. Estos mensajes requieren la garantía de que se ha llevado a cabo una propagación de cierta información previa en una transmisión anterior. La información es solamente enviada cuando se producen cambios y estos deben ser transmitidos a los vecinos. Una vez que la información ha sido difundida de forma confiable se considera que es conocida por todos los vecinos. Se puede consultar una comparación de diferentes técnicas de difusión para las redes móviles ad hoc en el trabajo realizado por Williams y Camp (2002).

En la propuesta de *difusión confiable* realizada en esta investigación, los vecinos que se encuentran a distancia de 1 salto y que reciben un mensaje, deben contestar con otro que indique la recepción correcta del primero. Se utiliza para ello un mensaje de tipo *ACK* (*Acknowledgement*), que contiene el identificador único del mensaje recibido. Los identificadores únicos para los mensajes se construyen utilizando el identificador del nodo emisor junto con un contador secuencial que se incrementa con cada transmisión.

El proceso de difusión confiable comienza cuando un nodo quiere transmitir un mensaje a todos o a un subconjunto de sus vecinos. El nodo emisor debe indicar la lista de destinatarios del mensaje mediante los identificadores únicos de dichos nodos. Esta lista puede obtenerse de la tabla de vecinos explicada en la sección 4.5.1. Si algún vecino destinatario del mensaje desaparece durante la

transmisión será eliminado de la lista de destinatarios. Un mensaje se considera entregado si ha sido recibido por todos los nodos destinatarios del mensaje.

Debido a las características del medio inalámbrico, todos los nodos que se encuentren a distancia de comunicación del nodo emisor, es decir, que sean sus vecinos, reciben el mensaje transmitido. Aquellos nodos vecinos cuyo identificador esté contenido en la lista de destinatarios aceptarán el mensaje recibido, mientras que los que no se encuentren en ella, no serán destinatarios del mensaje y, por lo tanto, deberán descartarlo. Todos los nodos que acepten el mensaje deben contestar al mismo utilizando un mensaje *ACK*. Con objeto de reducir el uso del medio compartido de transmisión, los mensajes de respuesta *ACK* pendientes de ser enviados son transmitidos de forma conjunta en una única transmisión que los agrupa.

Cada vez que un nodo recibe una respuesta *ACK* de un destinatario procede a eliminar su identificador de la lista de destinos de los que se espera una respuesta. Cuando la lista de destinatarios se encuentra vacía, ya sea porque se ha recibido respuesta de todos ellos o debido a la desaparición de todos los destinatarios, se considera que el mensaje ha sido entregado.

El tiempo  $T_R$  que un nodo espera a la recepción de las respuestas de los destinatarios viene determinado por la expresión

$$T_R = T_t \cdot (\text{destinos} + 1) + t + \text{jitter}$$

donde  $T_t$  es el tiempo estimado de transmisión en el medio, sin suponer la existencia de posibles colisiones o interferencias. El tiempo de espera tiene en cuenta el tiempo de transmisión inicial, el tiempo de transmisión de todos los mensajes de respuesta esperados y un tiempo  $t$  que es un tiempo estimado y utilizado por cada nodo para procesar los mensajes entrantes. El tiempo  $T_R$  incluye también un factor aleatorio (*jitter*), que permite resolver las posibles sincronizaciones que puedan surgir entre los diferentes nodos.

Si un nodo no recibe todas las respuestas *ACK* esperadas dentro del tiempo correspondiente se procede a repetir el envío del mensaje. Hay que tener en cuenta que el número de destinatarios puede haber disminuido si se han recibido algunos mensajes de respuesta desde ciertos vecinos y se han eliminado los nodos correspondientes de la lista de destinatarios. Entre distintas retransmisiones de un mensaje se aplica un tiempo de espera, que tiene como objeto reducir las posibles colisiones producidas por varios nodos que vuelvan a transmitir a la vez. El tiempo de espera entre dos retransmisiones sucesivas se calcula aplicando la expresión

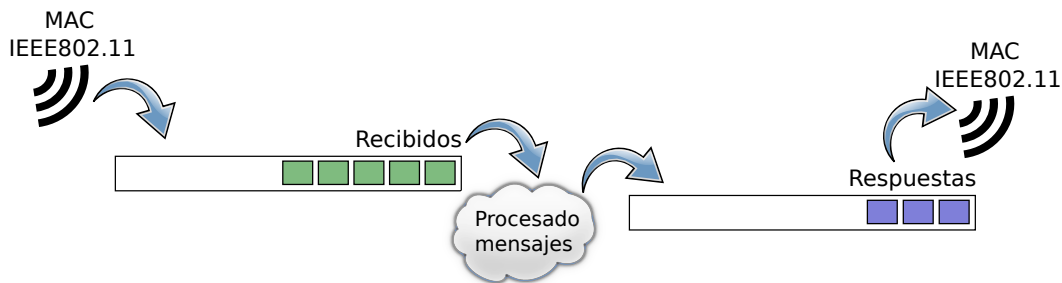
$$T_E = t \cdot n^2 - \text{jitter}$$

donde  $t$  es el tiempo medio de procesamiento utilizando anteriormente,  $n$  es el intento actual de retransmisión e incluye un tiempo aleatorio para evitar sincronizaciones. Como puede observarse, el tiempo de espera aumenta de forma

cuadrática con el objeto de evitar que, debido a la desaparición de los vecinos destinatarios de un mensaje, se realice un número elevado de transmisiones hasta detectar que el vecino ha desaparecido. Este enfoque es similar al decaimiento exponencial (*backoff*) seguido utilizado en el protocolo TCP. Se puede consultar más información sobre el envío de mensajes y el uso de *jittering* en el trabajo realizado por Clausen, Adamson y Dearlove (2008).

### 4.5.3 Colas de mensajes

El envío y recepción de los mensajes se gestiona mediante la utilización de colas que permiten desacoplar su procesamiento de la gestión de su transmisión. En concreto, cada nodo hace uso de dos colas de mensajes: *recibidos* y *respuestas*.



**Figura 4.6:** Colas para el procesamiento desacoplado de los mensajes recibidos y de las respuestas enviadas

El proceso por el cual los nodos envían y reciben mensajes está representado en la figura 4.6 y se explica a continuación:

1. Cuando un nodo recibe un mensaje a través de la interfaz inalámbrica, éste es añadido a la cola de mensajes *recibidos*. Como suele ser habitual en la gestión de las interfaces de red, la adquisición de mensajes entrantes se lleva a cabo mediante la utilización de un hilo de control dedicado. Se posibilita así la recepción de mensajes y la liberación del *buffer de recepción* aún cuando el dispositivo se encuentra realizando otras tareas.
2. Cada cierto periodo de tiempo  $t$  se procesan los mensajes contenidos en la cola de *recibidos*. Estos mensajes pueden corresponder a distintas capas de la arquitectura presentada en la sección 4.3. Por lo tanto, cada mensaje debe ser procesado por el nivel correspondiente de la arquitectura. Como resultado del procesamiento de cada mensaje recibido se producirán cambios en el estado interno del nodo, que a su vez pueden dar como resultado nuevos mensajes que deben ser propagados a sus vecinos.

3. Los mensajes propagados por el nodo son añadidos a la cola de *respuestas*, donde esperan a ser enviados en la siguiente transmisión. Hay que tener en cuenta, que cada nodo de la red hace uso del mecanismo de difusión confiable explicado en la sección 4.5.2 para el envío de sus mensajes. El proceso de difusión confiable es bloqueante, esto significa que no se pueden enviar nuevos mensajes hasta que el mensaje anterior haya sido *entregado* completamente a sus destinatarios.
4. Con objeto de reducir el número de transmisiones realizadas los mensajes pueden ser agrupados. Para ello se define un *mensaje de grupo* que contiene una lista de uno o más mensajes a transmitir. Los mensajes contenidos en la lista tienen un orden de procesamiento que debe ser respetado por los receptores del mensaje de grupo. A efectos prácticos, los mensajes son procesados uno a uno como si hubieran sido recibidos individualmente. Las transmisiones se realizan cada cierto periodo  $t$  con objeto de agrupar los mensajes y reducir el número de paquetes enviados.

## 4.6 Conclusiones

En este capítulo se ha presentado el problema de la composición automática de servicios en una red MANET. Primeramente, se ha realizado una introducción al problema mediante la presentación de un caso de uso y, posteriormente, se han establecido los requisitos que la solución presentada debe cumplir. Por otro lado, se han indicado cuáles son las principales aportaciones de la solución propuesta. El objetivo de esta presentación es sentar las bases para el resto de aspectos de esta investigación, que son desarrollados en los siguientes capítulos de la tesis doctoral.

La contribución principal de este capítulo es la definición de una arquitectura integral para la composición automática de servicios en redes MANET. La arquitectura propuesta, que está estructurada en niveles, se fundamenta en tres bloques principales: gestión de red, de servicios y del grafo distribuido. En este capítulo se ha explicado el fundamento de la gestión de red que está constituida por el módulo de auto-configuración encargado de la detección de vecinos y por el mecanismo para permitir la difusión confiable. Como se ha explicado, la detección de vecinos, que se basa en el uso de balizas periódicas, es necesaria debido a la naturaleza *proactiva* de la solución propuesta.

Por otro lado, la difusión confiable, que garantiza el envío de mensajes a los vecinos de un nodo, se ha desarrollado porque las propuestas realizadas en el capítulo 5 y capítulo 6 necesitan hacer uso de esta funcionalidad, ya que se basan en el envío incremental de información.

El resto de los niveles que han sido definidos en la arquitectura propuesta son presentados y validados en los próximos capítulos de esta tesis. En concre-

to, en el capítulo 5 se propone, y posteriormente se evalúa experimentalmente, un protocolo para el descubrimiento de servicios basado en parámetros. Por su parte, en el capítulo 6 se explica la propuesta para la composición de servicios utilizando un grafo de servicios distribuido, que es también evaluado experimentalmente en dicho capítulo.

Por último, este capítulo también ha definido las características que los servicios provistos por los dispositivos de la red deben tener para poder ser utilizados en las tareas de composición. Se han justificado las decisiones tomadas para la descripción de los servicios, las cuales son necesarias para utilizar el tipo de los parámetros de entrada y salida de los servicios como información para la composición. Se han presentado, así mismo, los motivos por los que no se han utilizado aquellos aspectos que pueden definir a los servicios desde un aspecto no funcional. De nuevo, la descripción de los servicios es utilizada en los siguientes capítulos para definir las propuestas de descubrimiento y composición de servicios en redes MANET.



## Descubrimiento de servicios basado en parámetros

En este capítulo se propone una solución para el descubrimiento de servicios en redes móviles ad hoc que se fundamenta en la disseminación, a través de la red, de información sobre los tipos de los parámetros de entrada y salida de los servicios. Posteriormente, los clientes podrán utilizar la información disseminada para realizar búsquedas de servicios. Por otro lado, el mecanismo para la composición de servicios, explicado en el capítulo 6, utiliza la funcionalidad proporcionada por este protocolo de descubrimiento para localizar servicios compatibles que pueden formar parte de una solución de composición. Sin embargo, la propuesta realizada en este capítulo para el descubrimiento de servicios puede utilizarse de una forma totalmente independiente a la composición de servicios para móviles ad hoc. Por esta razón, es presentada en este capítulo como una aportación independiente.

Las aportaciones principales de la propuesta para el descubrimiento de servicios presentada en este capítulo son:

- Definición del proceso de disseminación de la información acerca de los parámetros de entrada y salida, utilizando para ello las relaciones entre los tipos de los parámetros, de acuerdo a una ontología, con objeto de agrupar la información y reducir el número de mensajes enviados.
- Mecanismo de búsqueda de servicios basado en el tipo de los parámetros de entrada/salida y que utiliza la información disseminada para aplicar un proceso de *poda* durante la propagación de los mensajes. Este proceso permite reducir la transmisión de mensajes de búsqueda innecesarios.

- Integración del proceso de diseminación y búsqueda de servicios con el mecanismo de encaminamiento y gestión de rutas entre los nodos.
- Evaluación de la solución propuesta mediante la utilización de un simulador de red para determinar sus ventajas y limitaciones.

La sección 5.1 proporciona una visión general de los diferentes aspectos implicados en el proceso de búsqueda. A continuación, en la sección 5.2 se presentan las particularidades del proceso de diseminación propuesto. Por su parte, la sección 5.3 presenta el mecanismo de búsqueda de servicios, que se basa en la información de parámetros diseminada. Además, como se explica en la sección 5.4, la búsqueda de servicios está relacionada directamente con la gestión de las rutas de comunicación entre los nodos de la red. Por último, en la sección 5.5 se lleva a cabo una evaluación del protocolo propuesto utilizando un simulador de red.

## 5.1 Visión general

El descubrimiento de servicios es un aspecto necesario para realizar la composición. Aunque el objetivo final de la investigación presentada en esta tesis es una nueva propuesta para la composición de servicios en redes MANET, se propone aquí un nuevo protocolo para el descubrimiento de servicios en este tipo de redes. La finalidad de cualquier protocolo para el descubrimiento es que los clientes puedan localizar servicios, proporcionados por los servidores de la red, que posean una serie de características adecuadas a sus necesidades. Se puede consultar la sección 3.1 para conocer más aspectos sobre el estado actual de descubrimiento de servicios en general y, de forma particular, en su aplicación a las redes móviles ad hoc.

La principal característica del protocolo propuesto es que permite la localización de servicios basándose en el tipo de sus parámetros de entrada y salida, estando dichos parámetros categorizados de acuerdo a una ontología de conceptos. Como se explica en la sección 3.2.3 del estado actual, uno de los aspectos a resolver durante la composición de servicios es la compatibilidad entre los parámetros de *entrada/salida* de los distintos servicios disponibles. Para ello, es necesaria la existencia de un mecanismo que permita a los nodos de la red determinar cuáles son los parámetros de entrada/salida que proporcionan los distintos servicios ofertados en la red y, además, cuáles son las relaciones de compatibilidad que se establecen entre ellos. Se propone, por lo tanto, una solución basada en la diseminación a través de la red de información sobre el tipo de los parámetros de entrada/salida de los distintos servicios.

Como puede observarse en la sección 3.1.5 de la revisión bibliográfica realizada, los protocolos de descubrimiento de servicios en redes MANET se basan,

generalmente, en la utilización de otros aspectos de la descripción de los servicios como puede ser su identificador (UUID), su nombre, la categorización del servicio completo, etc. Sin embargo, en esta tesis doctoral se propone la utilización de la información relativa a sus parámetros de entrada y salida por varios motivos:

- Propagar información acerca de los parámetros de un servicio es más adecuado para el descubrimiento cuando se intenta resolver tareas más complejas, como es el caso de la composición de servicios, en las que es necesario determinar la compatibilidad funcional entre parámetros de entrada/salida de los servicios.
- Cuando se lleva a cabo la diseminación de la información sobre los parámetros de los servicios, los nodos únicamente necesitan compartir información acerca de los diferentes tipos de datos utilizados. Esto permite la especificación de múltiples servicios distintos reutilizando los mismos tipos. Por el contrario, la diseminación de la información relativa al tipo o identificador de los servicios requiere que todos los nodos de la red tengan conocimiento de los diferentes servicios que pueden existir en el resto de nodos de la red ad hoc.
- La utilización de una ontología de conceptos para categorizar los parámetros de entrada/salida de los servicios permite aplicar, como se explica en este capítulo, mejoras en los procesos de diseminación y posterior búsqueda de los servicios. Estas mejoras tienen como resultado una reducción en el número de mensajes durante ambos procesos.

El protocolo propuesto para el descubrimiento de servicios en redes MANET se encuentra dividido en tres funcionalidades principales: la *diseminación de parámetros*, la *búsqueda de servicios* basada en la información diseminada y la *gestión de las rutas* de comunicación descubiertas. El objetivo de la diseminación de parámetros es permitir que los nodos de la red puedan tener conocimiento de las características de los parámetros de entrada/salida que proporcionan los servicios disponibles. Por otro lado, el cometido de la búsqueda de servicios es permitir la localización en la red de los servicios que satisfagan los requisitos de búsqueda especificados. Por último, la gestión de rutas es el proceso mediante el cual, durante la búsqueda de servicios, se establecen y gestionan los caminos dentro de la red MANET que permiten la comunicación entre los clientes y los proveedores de los servicios.

La figura 5.1 muestra las funcionalidades anteriores como diferentes capas de una arquitectura. El protocolo propuesto utiliza las funcionalidades proporcionadas por la *detección de vecinos* y la *difusión confiable*, que son explicadas en la sección 4.5.1 y sección 4.5.2, respectivamente.

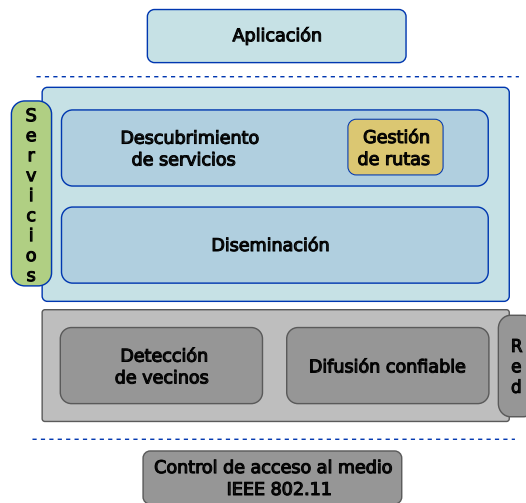


Figura 5.1: Arquitectura para el descubrimiento de servicios basado en parámetros

La propuesta realizada en este capítulo puede clasificarse de acuerdo a la categorización presentada en la sección 3.1.2 para las soluciones de descubrimiento de servicios en redes MANET. En cada apartado se incluye una referencia a las secciones de la revisión bibliográfica en las que se puede consultar más información referente a cada aspecto.

- **Uso de directorios:** la solución propuesta no utiliza directorios para centralizar las descripciones de los servicios. En cambio, los nodos que proveen servicios propagan la información salto a salto a través de la red mediante el uso de una difusión controlada. Esto evita la necesidad de mantener directorios de servicios y reduce los posibles puntos de fallo al evitar la centralización de la información en unos nodos (sección 3.1.3 y sección 3.1.4).
- **Descripción de servicios:** los servicios se describen mediante el tipo de sus parámetros de entrada y salida categorizados en base a una ontología de conceptos. Durante la búsqueda de servicios se hace uso de las relaciones de generalización y especificación para localizar los servicios compatibles (sección 3.1.5 y sección 4.4).
- **Modo de operación:** el proceso de diseminación de la información sobre los parámetros de un servicio es *proactivo* y se lleva a cabo con independencia de que se realicen o no solicitudes de búsqueda en la red (sección 3.1.6).
- **Respuesta a la movilidad:** la información diseminada y las búsquedas realizadas por los nodos son mantenidas en la red frente a los posibles

cambios que se producen en su topología. Se utilizan *notificaciones*, tanto por parte de la capa de descubrimiento como de encaminamiento, para actualizar la información de servicios y las rutas de comunicación (sección 3.1.7).

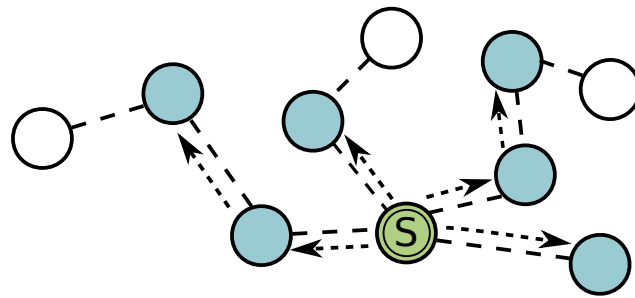
- **Integración de protocolos:** el descubrimiento y encaminamiento se encuentran integrados de tal forma que la búsqueda de servicios produce la creación de las rutas de comunicación entre los nodos de la red. El objetivo es minimizar el número de mensajes empleados (sección 3.1.8).
- **Mecanismo de selección:** los aspectos relativos a la selección de servicios entre las múltiples respuestas recibidas durante una búsqueda no son contemplados en esta solución. Sin embargo, pueden aplicarse alguno de los mecanismos explicados en la revisión bibliográfica como, por ejemplo, la selección de aquellos servicios que se encuentren a una distancia, en saltos, más cercana al cliente que ha iniciado la búsqueda (sección 3.1.9).

## 5.2 Diseminación

Los servicios ofrecidos por cada nodo de la red son, inicialmente, solamente conocidos por el propio nodo que los provee. Cuando un cliente en otro nodo quiere utilizar un servicio no dispone de ningún conocimiento previo que le permita localizarlo. Así, el cometido de la capa de *diseminación* es propagar información a través de la red ad hoc con el propósito de facilitar la localización de los servicios.

El proceso de diseminación comienza en aquellos nodos que proveen los servicios; a partir de ellos, mediante la realización de saltos sucesivos a través de la red ad hoc, se produce la propagación de la información a otros nodos de la red. La diseminación se lleva a cabo mediante la utilización de mensajes de actualización que se generan como respuesta a los eventos que se producen durante cambios en la topología de la red. La figura 5.2 representa, de un modo general, el proceso de diseminación. Desde el nodo que provee el servicio la información se difunde salto a salto a través de la red MANET hasta una cierta distancia máxima que depende de la configuración.

Como se ha explicado anteriormente, tanto el proceso de diseminación como las búsquedas posteriores se fundamentan en la utilización de la información sobre el tipo de los parámetros de entrada y salida de los servicios. Estos parámetros se encuentran categorizados en base a una ontología de conceptos compartida y adecuada al dominio de aplicación de los servicios. Se supone que dicha ontología es conocida *a priori* por todos los nodos de la red ad hoc. En la sección 4.4 se explica con mayor detalle cómo se describe un servicio y en qué consiste una ontología de conceptos.



**Figura 5.2:** Diseminación en la red ad hoc de la información sobre los servicios provistos por un nodo

### 5.2.1 Distancia estimada

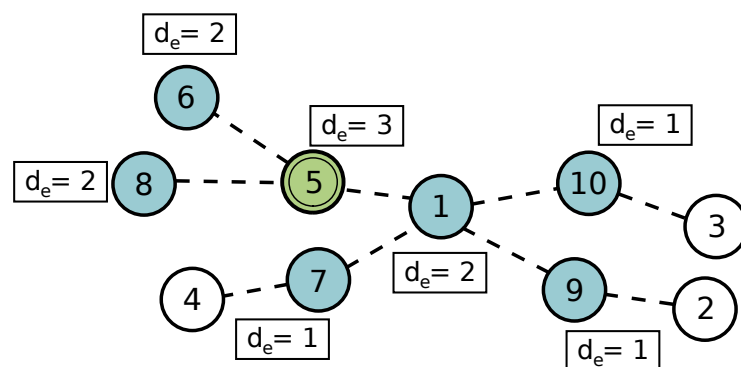
Cada nodo de la red mantiene una tabla, denominada *tabla de parámetros*, que contiene información acerca de los tipos de los parámetros de entrada/salida que han sido diseminados por nodos cercanos. A partir del contenido de esta tabla es posible conocer a qué distancia del nodo actual, en saltos, se puede encontrar un parámetro perteneciente a un determinado tipo. Como se explica con mayor detalle en la sección 5.2.2, la tabla mantiene para cada tipo de parámetro un valor numérico denominado *distancia estimada*, que se representa como  $d_e$ . Este valor es utilizado durante el proceso de búsqueda para determinar, en función de la distancia en saltos a la que se encuentra un parámetro buscado, si el mensaje de búsqueda debe continuar su propagación a través de la red. El uso del valor de *distancia estimada* durante el proceso de búsqueda se detalla en la sección 5.3.2.

Los nodos que proveen servicios son los que inician la diseminación de los tipos de sus parámetros de entrada/salida a través de la red MANET. Para ello, estos nodos difunden el contenido de sus tablas de parámetros a sus vecinos utilizando *mensajes de actualización*. Cada vez que un nodo recibe una actualización y la procesa puede dar como resultado la modificación de su tabla de parámetros. Estas modificaciones pueden producir, a su vez, nuevos mensajes de actualización que son propagados a los vecinos del nodo actual. Los mensajes de actualización se comunican mediante *difusión (broadcast)* a través del medio inalámbrico. Como el medio de transmisión es compartido, los mensajes emitidos pueden ser recibidos por todos aquellos nodos que se encuentren a distancia de comunicación del emisor. El proceso de diseminación y actualización de las tablas de parámetros continúa de nodo en nodo hasta que se alcanza la *distancia máxima de diseminación*,  $D_d$ . Dicho valor determina hasta qué distancia en la red se realiza, desde el nodo que la origina, la propagación para un determinado tipo de parámetro.

Cuanto mayor sea el valor del indicador  $d_e$  para un parámetro concreto más cerca se encontrará el nodo actual de un nodo o nodos que proveen un servicio con dicho tipo de parámetro. El valor de *distancia estimada* para un parámetro disminuye en una unidad cada vez que la información sobre dicho parámetro se propaga un salto a través de la red. Así, aquellos parámetros que pertenecen a un servicio localizado en el propio nodo tienen el valor máximo posible para su *distancia estimada*,  $d_e = D_d$ , es decir, igual a la distancia máxima de diseminación. Los nodos que se encuentren a una distancia en saltos sucesivamente mayor van decrementando el valor de  $d_e$  asociado al tipo de parámetro.

El proceso de diseminación se detiene, para un tipo de parámetro, cuando el valor para su *distancia estimada* llega a la unidad en un nodo. Cuando esto ocurre, dicho nodo ya no lleva a cabo ninguna propagación de la información sobre el parámetro a sus vecinos. Cada tipo de parámetro se gestiona de manera independiente durante su diseminación en los mensajes de actualización; es posible, por lo tanto, que unos parámetros dejen de ser diseminados tras alcanzar la distancia máxima, mientras que otros continúen el proceso.

Con objeto de aclarar el proceso de diseminación, se incluye un ejemplo del mismo en la figura 5.3, donde se muestra el proceso de diseminación para la información de parámetros proporcionada por el nodo 5, representado mediante una línea doble. El número contenido dentro de cada nodo es su identificador único, mientras que asociado a ellos se representa el valor para la *distancia estimada*  $d_e$  correspondiente a la diseminación del parámetro.



**Figura 5.3:** Propagación del valor de *distancia estimada* sobre un tipo de parámetro de un servicio

En el ejemplo se ha utilizado un valor para la distancia máxima de diseminación  $D_d = 3$ . Así, en el nodo inicial 5 la *distancia estimada* tiene un valor de  $d_e = D_d = 3$ , mientras que aquellos nodos situados a una distancia de  $n$  saltos poseen un valor tal que  $d_e = D_d - n$ . La propagación se ha producido en la red hasta que  $d_e = 1$ , momento en el que el proceso de diseminación se ha detenido. Los nodos que se encuentran a una distancia mayor de ese pun-

to no tienen en su tabla una entrada para los parámetros diseminados, lo que significa que dichos nodos no tienen conocimiento sobre la existencia de los parámetros propagados.

La *distancia estimada* no representa la distancia exacta en saltos a la que se puede encontrar un determinado tipo de parámetro, de ahí la utilización del adjetivo *estimada*. Es posible que diferentes nodos de la red proporcionen servicios cuyos parámetros de entrada/salida tengan tipos idénticos o relacionados de acuerdo a la ontología de conceptos utilizada. Estos nodos pueden encontrarse entre sí a distancias tales que la información propagada por ellos se solape en los mismos nodos intermedios. Además de esto, y como se explica en la sección 5.2.2, los tipos de parámetros relacionados son agrupados en una misma entrada de la tabla de parámetros. Cuando se agrupa la información de distintos orígenes para la distancia a un parámetro se utiliza el mayor valor (el más cercano) de todos los disponibles.

Por los motivos anteriores, la *distancia estimada* representa realmente la distancia a la que se encuentra el parámetro *más cercano* del tipo indicado en la entrada correspondiente de la tabla, aunque pueden existir otros parámetros más alejados para los que no se conoce su distancia real en saltos a través de la red. Por lo tanto, la existencia de una entrada en la tabla de parámetros no proporciona información *exacta* sobre la distancia a la que se encuentran todos los posibles parámetros de ese tipo o relacionados. Sin embargo, sí que garantiza la existencia de un parámetro compatible en las cercanías a una distancia máxima determinada por la distancia de diseminación  $D_d$ .

## 5.2.2 Tabla de parámetros

Cada nodo  $n_i$  de la red mantiene una *tabla de parámetros*  $P_i$  que contiene la *distancia estimada* a cada uno de los tipos de parámetros conocidos. La tabla de parámetros se define como una función

$$P_i : O \rightarrow E$$

que relaciona tipos de parámetros pertenecientes a la ontología  $O$  con el conjunto de los valores de *distancia estimada*  $E$ . En concreto, cada tabla de parámetros de un nodo mantiene un conjunto de entradas  $E_p$  definidas de tal forma que

$$E_p = \{(p, L) \mid p \in O\}$$

donde  $p$  es el tipo del parámetro y  $L = \{e_1, e_2, \dots, e_k\} \mid e \in E\}$  es un conjunto finito de elementos donde cada uno de ellos contiene la información acerca de la *distancia estimada* de un parámetro y está definido a su vez como

$$E = \{(d_e, v) \mid 1 \leq d_e \leq D_d, v \in N\}$$



Tipo de Parámetro	Lista
$P_1$	$(d_e, v)_1, (d_e, v)_2, \dots, (d_e, v)_i$
$P_2$	$(d_e, v)_1, (d_e, v)_2, \dots, (d_e, v)_j$
$\dots$	$\dots$
$P_n$	$(d_e, v)_1, (d_e, v)_2, \dots, (d_e, v)_k$

Figura 5.4: Tabla de parámetros de un nodo

siendo  $d_e$  un valor para la *distancia estimada* y  $v$  el identificador del nodo vecino que ha proporcionado la información, sobre la distancia al parámetro  $p$  correspondiente, al nodo  $n_i$  actual.

Debido a que la lista  $L$  puede tener más de un elemento, se define como *valor efectivo*  $d_{eff}(p_i)$ , para una entrada de tipo de parámetro  $p_i$  de la tabla, aquél elemento de la lista  $L_i$  que tenga el mayor valor para la *distancia estimada* con respecto a los otros elementos de la entrada

$$d_{eff}(p_i) = e_j \in L_i \mid \forall e_i \in L_i : d_{e_j} \geq d_{e_i}$$

El indicador de distancia efectiva  $d_{eff}$  es el valor realmente propagado, para cada tipo de parámetro, durante los mensajes de actualización enviados a los nodos vecinos. En una lista  $L$  no puede existir más de un elemento  $e_i$  con un mismo valor de  $v$ . Es decir, un nodo no mantiene, para un parámetro  $p_i$  dado, dos valores de *distancia estimada* que provengan de un mismo vecino. Por lo tanto, si un nodo, tras haber recibido de un vecino un valor de *distancia estimada*, recibe otro valor del mismo vecino, el nuevo valor es considerado una sustitución del valor antiguo y no una nueva inserción.

El valor  $v$  contenido en el elemento de *distancia estimada* permite evitar la propagación hacia atrás de los valores diseminados. Esto ocurre cuando un nodo recibe, desde un vecino, información sobre la distancia a un tipo de parámetro que él mismo había proporcionado anteriormente. Así, en el protocolo propuesto, la información recibida de esta forma es descartada, debido a que en el mensaje de actualización se incluye, por cada parámetro notificado, el identificador  $v$  del nodo del que inicialmente procedía la información. En la figura 5.4 se representa la estructura de una tabla de parámetros. En negrita se indica los posible elementos seleccionados como distancia efectiva.

Por último, cada entrada de la tabla de parámetros almacena información sobre un tipo de parámetro concreto. Sin embargo, como se explica en la sección 4.4, el uso de una ontología permite establecer relaciones de generalización y especificación entre diferentes conceptos. Por lo tanto, cuando una entrada de la tabla de parámetros hace referencia a un tipo concreto de la ontología, esa entrada puede representar tanto a un parámetro del tipo exacto

como a cualquier parámetro de un tipo más específico que el representado por el correspondiente a la entrada de la tabla.

### 5.2.3 Mensajes de actualización

Los nodos diseminan la información sobre los tipos de parámetros mediante el envío de mensajes de actualización a sus vecinos. El contenido de estos mensajes depende de la respuesta al evento que se haya producido en la red MANET. Así, los mensajes de actualización propagados por un nodo pueden transportar tanto el contenido completo de su tabla de parámetros como únicamente los cambios que se han producido desde la última actualización. Es por este motivo, la propagación de cambios diferenciales entre sucesivos mensajes de actualización, por el cual se hace uso de la *difusión confiable* explicada en la sección 4.5.2. La utilización de actualizaciones incrementales presupone que cualquier información propagada anteriormente sobre las tabla de parámetro a los vecinos fue recibida correctamente por estos.

Cada mensaje de actualización contiene los cambios que deben aplicar los vecinos que lo reciben a sus propias tablas de parámetros. Estos cambios son especificados mediante dos listas con las acciones a realizar: *adiciones* y *eliminaciones*. Cada vez que un nodo recibe un mensaje de actualización procesa estas listas y lleva a cabo las acciones de actualización de su tabla correspondientes. Un mensaje de actualización  $M_A$  se define con los siguientes campos

$$M_A = (L_A, L_E, n_e, [datos]); \quad n_e \in N$$

donde  $n_e$  es el identificador del nodo que envía el mensaje y  $N$  el conjunto de identificadores de los nodos de la red. El campo de *datos* permite la inclusión de información por parte de aquellas capas superiores que hagan uso del protocolo de diseminación de parámetros. Por su parte, la lista de adiciones  $L_A$  se define como

$$L_A = \{(p_1, e_1), (p_2, e_2), \dots, (p_k, e_k)\} \mid p \in O, e \in E\}$$

es decir, es una lista de entradas formadas por el tipo del parámetro según la ontología  $O$  y un elemento perteneciente al conjunto  $E$ , definido anteriormente, que contiene la *distancia estimada* para ese tipo y el identificador del nodo que ha proporcionado dicha información en una actualización anterior.

Por otro lado, la lista de eliminaciones  $L_E$  se define como

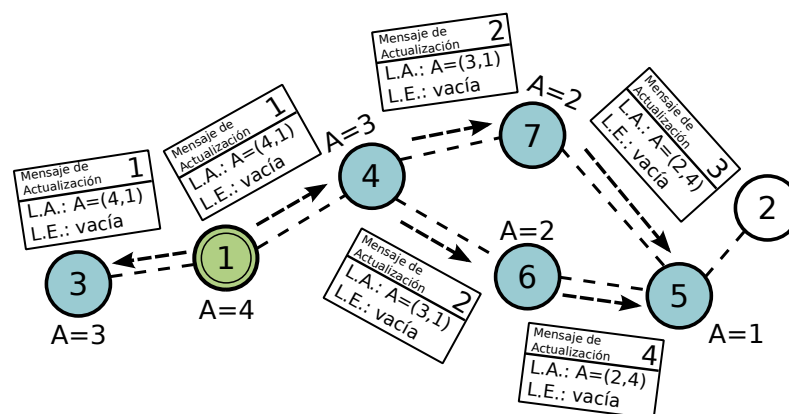
$$L_E = \{p_1, p_2, \dots, p_k\} \mid p \in O\}$$

no siendo, en este caso, necesario especificar más que los tipos de los parámetros a eliminar del nodo receptor del mensaje. Esto es debido a que los vecinos toman, como origen de la información a eliminar, el identificador  $n_e$  del nodo que está propagando el mensaje de actualización, que puede ser obtenido a

partir de la información proporcionada por la capa de gestión de red cuando se recibe el mensaje de actualización.

Cuando un nodo recibe un mensaje de actualización procesa en primer lugar las eliminaciones, que indican qué entradas debe eliminar de su tabla de parámetros. Por ejemplo, si un nodo ha eliminado de su tabla de parámetros un conjunto de entradas, debido a la desaparición de un vecino, debe indicar al resto de sus vecinos que la información que había propagado anteriormente ya no es válida. En tal caso, la lista de eliminaciones contiene las entradas que deben ser eliminadas.

Tras procesar las eliminaciones, el nodo aplica la lista de adiciones que contiene los nuevos elementos que deben ser añadidos a su tabla de parámetros. Por ejemplo, durante la diseminación de nuevos parámetros a través de la red, se utiliza esta lista para propagar las nuevas entradas a los vecinos. El procesamiento de los mensajes de actualización por parte de un nodo puede dar como resultado nuevos mensajes a propagar por cada nodo a sus respectivos vecinos.



**Figura 5.5:** Envío de mensajes de actualización para la diseminación de parámetros

En la figura 5.5 se muestra un ejemplo del envío de mensajes de actualización. El proceso de diseminación se inicia en el nodo 1, donde existe un servicio que define un parámetro de tipo  $A$ . Para el ejemplo se ha establecido una distancia máxima de diseminación de  $D_d = 4$ . En la figura, el número contenido en la parte superior derecha del mensaje de diseminación representa el orden de envío de los mensajes.

A partir del nodo 1 comienzan a propagarse mensajes de actualización con los cambios a realizar en las tablas de parámetros de sus vecinos. Así, dicho nodo envía primeramente un mensaje a sus vecinos, los nodos 3 y 4, que contiene, en la lista de adiciones  $L_A$ , una entrada para el parámetro de tipo  $A$  con el valor de la *distancia estimada* y el identificador del nodo que proporcionó esa información previamente. En este caso, como la información es proporcionada

por el mismo nodo, el parámetro es local y el identificador es el del propio nodo que está iniciando la diseminación de sus parámetros.

Cuando los vecinos reciben la información proceden a procesar el mensaje y a actualizar sus tablas. Como resultado, la *distancia estimada* para el parámetro *A* es disminuida en una unidad y se produce un nuevo mensaje para los vecinos de estos nodos. En la figura 5.5 se representan únicamente los mensajes que se propagan hacia delante pero, debido a la naturaleza compartida de la red ad hoc, el mensaje enviado por un nodo es recibido por todos sus vecinos en dicho momento. Así, por ejemplo, el mensaje de actualización enviado desde el nodo 4 llega también al nodo 1, sin embargo, su contenido no produce ninguna actualización porque la información provenía originalmente del propio nodo 1.

El proceso continúa nodo a nodo hasta que el valor de la *distancia estimada* para el parámetro llega a la unidad. Cuando esto sucede, por ejemplo, en el nodo 5, se detiene la propagación de más mensajes a los vecinos del nodo.

#### 5.2.4 Actualización de la tabla de parámetros

La movilidad de los nodos que constituyen la red ad hoc tiene como resultado que la información contenida en la tabla de parámetros quede obsoleta y necesite ser actualizada. El protocolo para la diseminación de la información relativa a los parámetros es *proactivo*. Esto significa que el contenido de las tablas de parámetros es mantenido frente a los cambios que se producen en la topología de la red y, por lo tanto, los mensajes de actualización se producen como respuesta a los eventos de movilidad detectados. Es el nivel de *gestión de red* y, en concreto, la capa de *detección de vecinos* explicada en la sección 4.5.1, el que lleva a cabo la notificación de estos eventos a las capas superiores.

El envío de mensajes de actualización se produce, inicialmente, como respuesta a los eventos de aparición y desaparición de vecinos. Sin embargo, la recepción de un mensaje de actualización y la posterior actualización de la tabla de parámetros puede resultar en la generación de nuevos mensajes, como se explica posteriormente en esta misma sección. La diseminación de parámetros responde a los siguientes eventos:

- **Detección de nuevo vecino:** la detección de un nuevo vecino por un nodo resulta en que dicho nodo lleve a cabo una difusión del estado actual de su tabla de parámetros. Debido a que el medio inalámbrico es compartido, este mensaje puede ser escuchado por todos los vecinos actuales del nodo, incluyendo tanto al vecino nuevo como a los antiguos. El algoritmo de diseminación lleva a cabo las comprobaciones necesarias para determinar si la tabla de vecinos de los nodos receptores debe actualizarse tras la recepción del mensaje. Aquellos nodos que tengan su tabla de parámetros vacía no llevarán a cabo ninguna propagación como resultado de la detección de nuevos vecinos.

- **Desaparición de un vecino:** como resultado de este suceso, el nodo actual elimina de su tabla toda la información que fue proporcionada por el vecino desaparecido. Para ello utiliza los identificadores de los vecinos contenidos en las entradas de las tablas de parámetros y que determinan qué vecino proporcionó cada información de *distancia estimada*.

Además, la información de la tabla de parámetros de un nodo también puede verse actualizada como resultado de la modificación de los servicios proporcionados por el propio nodo. Así, se distingue la ocurrencia de los siguientes sucesos:

- **Registro de nuevo servicio:** los nodos pueden añadir nuevos servicios a proveer en cualquier momento durante su participación en la red MANET. Cuando se añade un servicio nuevo en un nodo se inicia el registro de sus parámetros de entrada/salida correspondientes de forma local. Si los parámetros añadidos no habían sido ya registrados como provistos por otro servicio en el propio nodo se produce un mensaje de actualización que es propagado a los vecinos. Este mensaje resulta en un proceso de actualización de las tablas de parámetros a través de la red utilizando la propagación y actualización salto a salto explicada anteriormente.
- **Eliminación de un servicio:** si un nodo deja de proveer un servicio es necesario notificar a sus vecinos de que los parámetros proporcionados por dicho servicio ya no están disponibles. Se envía para ello un mensaje de actualización que contiene las eliminaciones correspondientes. El proceso de eliminación de la información continúa a través de la red hasta que se alcanza la distancia máxima de diseminación, y se han eliminado, por lo tanto, todas las entradas en la tablas de parámetros de los nodos afectados.

El algoritmo que representa los procesos de actualización de la tabla de parámetros se muestra en la figura 5.6 y su funcionamiento se describe a continuación.

Por cada parámetro contenido en el mensaje de actualización se comprueba si existe una *eliminación*. Si este es el caso, la información de *distancia estimada*, que fue proporcionada anteriormente por el nodo que está propagando el mensaje recibido, es eliminada de la tabla de parámetros. Si se ha producido una eliminación y la *distancia estimada* eliminada tenía un valor mayor que la unidad es necesario propagar dicha eliminación a los vecinos. El motivo para esto es que anteriormente se habrá producido una propagación de dicha información a los vecinos y, por lo tanto, es necesario comunicar que dicha información ya no es válida y debe ser eliminada.

```

Entrada:
     $m$  : mensaje recibido
     $v_i$  : vecino que ha enviado el mensaje
     $P_i$  : tabla de parámetros local
     $n_i$  : nodo actual

 $m_{nuevo} \leftarrow Vacio$ 
for each Parametro  $p$  in  $m$  do
    |  $L_{distancias} \leftarrow ObtenerListaDistancias(P_i, p)$ 
    | if ContieneEliminación( $m, p$ ) then
    | | ( $d_e, v$ )  $\leftarrow$  EliminarEntrada( $L_{distancias}, v_i$ )
    | | if  $d_e > 1$  then
    | | | InsertarEliminación( $m_{nuevo}, p$ )
    | | | ( $d_e', v$ )  $\leftarrow$  ObtenerDistancia( $p$ )
    | | | if  $d_e' > 1$  then
    | | | | InsertarAdición( $m_{nuevo}, p, d_e'$ )
    | | | end
    | | end
    | end
    | if ContieneAdición( $m, p$ ) then
    | | ( $d_e, v$ )  $\leftarrow$  ObtenerAdición( $m, p$ ) if  $v \neq n_i$  then
    | | |  $d_e \leftarrow$  ObtenerDistancia( $L_{distancias}, p$ )
    | | |  $d_e' \leftarrow$  InsertarDistancia( $L_{distancias}, d_e, v$ )
    | | | if  $d_e' > d_e$  and  $d_e' > 1$  then
    | | | | InsertarAdición( $m_{nuevo}, p, d_e'$ )
    | | | end
    | | end
    | end
end
if not EsVacio( $m_{nuevo}$ ) then
    | propagar( $m_{nuevo}$ )
end

```

Figura 5.6: Algoritmo para la diseminación de la información sobre parámetros

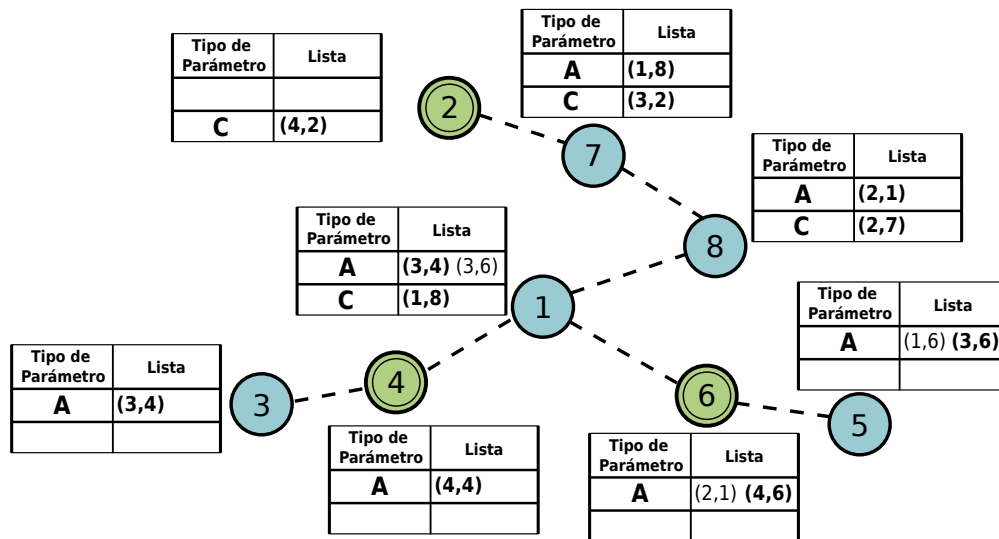
Por otro lado, si como resultado de la eliminación de una entrada de la lista de elementos para un parámetro se ha producido una modificación en el *indicador de distancia estimada*, siendo el nuevo valor de la *distancia estimada* mayor que la unidad, dicho nuevo valor debe ser propagado a los vecinos del nodo. Esto ocurre, por ejemplo, cuando un nodo ha recibido de sus vecinos más de un valor para la *distancia estimada* a un tipo de parámetro y, tras la eliminación del valor que se estaba tomando como actual, se produce la propagación de otro de los valores alternativos.

A continuación, para cada parámetro se procesan las posibles *adiciones* contenidas en el mensaje de actualización recibido. Solamente se procesan aquellas para las que la información añadida no hubiera sido anteriormente proporcionada por el nodo que procesa el mensaje. Para ello, todas las *adiciones* contienen el identificador del nodo que anteriormente había propagado dicha información. Si este no es el caso, y la información se añade a la tabla de parámetros, se comprueba si el *indicador de distancia estimada* ha cambiado como resultado de la actualización. Si no se ha producido ninguna modificación no es necesario seguir propagando dicha información a los vecinos del nodo actual. Además, solamente aquellas *adiciones* cuya *distancia estimada* es mayor que la unidad son propagadas en el mensaje de actualización a los vecinos.

Todas las *adiciones* y las *eliminaciones* resultantes del procesado de un mensaje de actualización son incluidas en un nuevo mensaje que será enviado a los vecinos actuales del nodo. La recepción de este mensaje de actualización por parte de los vecinos del nodo inicia de nuevo el algoritmo explicado anteriormente. El proceso continúa así a través de la red MANET hasta que se estabiliza y no se producen nuevas actualizaciones.

La figura 5.7 muestra el estado de una red de nodos en la que se ha producido la diseminación de la información sobre los tipos parámetros de los servicios ofertados. El valor destacado en negrita en cada lista de distancias estimadas representa al valor utilizado como *indicador efectivo*. Estos valores son los diseminados a los vecinos del nodo cuando se produce un mensaje de actualización. Tanto el nodo 1 como el nodo 6 proporcionan un servicio con el parámetro de tipo A, mientras que el nodo 2 contiene un parámetro de tipo C. El ejemplo se ha llevado a cabo con una distancia máxima de diseminación  $D_d = 4$ . Cada nodo mantiene en su tabla de parámetros la lista de entradas  $(d, v)$  que contienen el valor de la *distancia estimada* y el identificador del vecino que ha proporcionado la información.

En el ejemplo de la figura se supone que el nodo 4 ha llevado a cabo la diseminación de su información al nodo 1 antes que el nodo 6. Como puede observarse en la figura, la información sobre el mismo tipo de parámetros puede solaparse en nodos intermedios, como es el caso del nodo 1. Como en la información sobre la *distancia estimada* al parámetro A en dicho nodo, recibida desde los dos nodos vecinos, es  $d_e = 3$ , se usa como distancia efectiva el primer



**Figura 5.7:** Estado de las tablas tras la diseminación de la información sobre los parámetros de los servicios

valor que se ha recibido, en este caso el proveniente del nodo 4. Por otro lado, también el nodo 6 habrá recibido la información propagada desde el nodo 1, sin embargo, en este caso el valor para la distancia efectiva es  $d_e = 4$ , ya que existe un parámetro local del mismo tipo. Por otro lado, la información propagada por el nodo 2 se propagada de la forma usual hasta alcanzar la distancia máxima de diseminación.

### 5.2.5 Uso de la ontología de conceptos

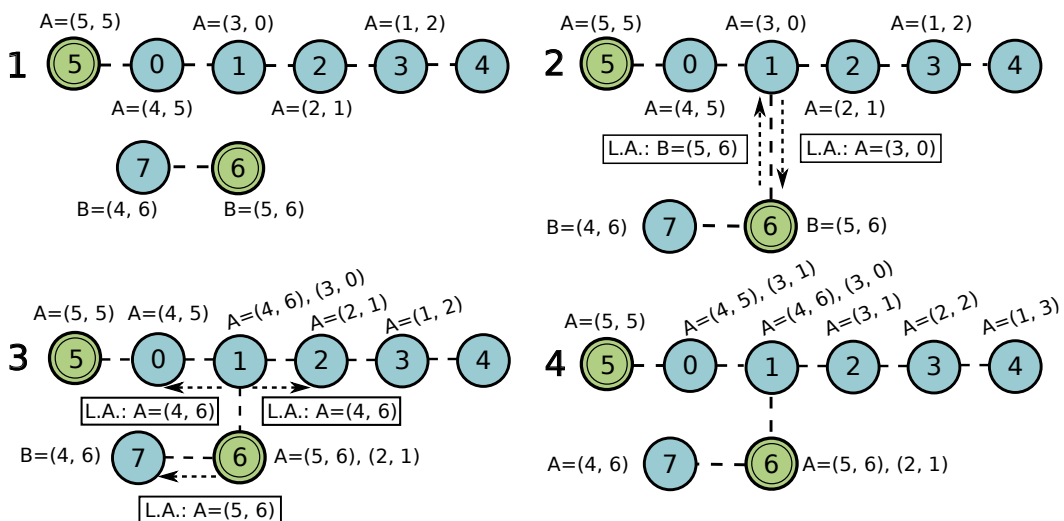
Hasta ahora se han presentando las características generales del algoritmo que gestiona la actualización de la tabla de parámetros de cada nodo. Sin embargo, no se ha hecho en ningún momento referencia directa a cómo se lleva a cabo la utilización de la ontología sobre los tipos de parámetros. Como se explica en la sección 4.4.2, dos parámetros no solamente pueden tener el mismo tipo sino que además pueden estar vinculados mediante una relación de generalización y especificación determinada por la ontología. Por lo tanto, el algoritmo presentado debe ser modificado con el objeto de utilizar esta información sobre las relaciones entre los tipos de los parámetros.

La tabla de parámetros mantiene una entrada por cada tipo de parámetro conocido, sin embargo, dicha entrada se refiere realmente a un *grupo de tipos de parámetros* en vez de a un único tipo concreto. Cada grupo representa a todos aquellos parámetros cuyos tipos están relacionados mediante *igualdad* o *subsunción*, de acuerdo a la información contenida en la ontología de conceptos



utilizada. Cada grupo está representado por el tipo de parámetro más general, de todos aquellos que están contenidos en el grupo, llamado *tipo representativo* del grupo. El tipo representativo cambia cada vez que se añaden parámetros de un tipo más general al grupo, durante la actualización de las entradas de la tabla. Por el contrario, la eliminación de parámetros de un grupo no modifica su tipo representativo que es mantenido hasta que la entrada de la tabla de parámetros es eliminada de manera definitiva. La eliminación de una entrada de la tabla ocurre cuando la entrada queda vacía debido a la eliminación de todos los elementos de la lista de distancias asociada.

Por otro lado, el cambio en el tipo representativo de un grupo de parámetros tiene siempre como resultado la propagación de la información de *distancia estimada* asociada a dicho grupo a los vecinos del nodo actual. El cambio en el tipo representativo de un grupo puede producirse como resultado de la recepción de un mensaje de actualización desde un nodo vecino o debido al registro de nuevos servicios locales en el nodo.



**Figura 5.8:** Diseminación de parámetros utilizando las relaciones contenidas en una ontología de conceptos

La figura 5.8 muestra qué ocurre cuando se lleva a cabo la diseminación de la información utilizando una ontología de conceptos que establece relaciones jerárquicas entre los tipos de los parámetros. La ontología utilizada en el ejemplo define dos tipos de parámetros  $A$  y  $B$  siendo el tipo del parámetro  $A$  un concepto más general que el de  $B$ . En el estado 1, el nodo 5 ha propagado información sobre los parámetros del servicio, o servicios, que proporciona a través de la red ad hoc utilizándose para ello una distancia máxima de diseminación de  $D_d = 5$ . A continuación, en el estado 2, aparece el nodo 6 en las cercanías del

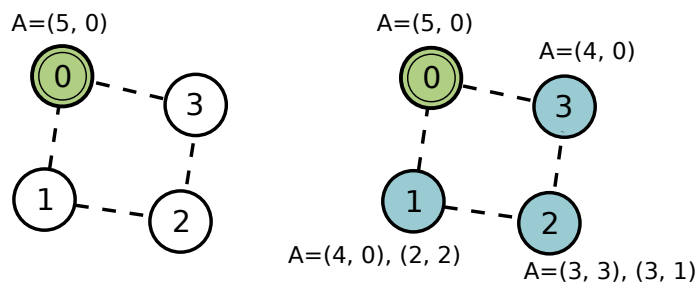
nodo 1, lo que produce que ambos nodos intercambien el contenido completo de sus tablas de parámetros mediante el uso de mensajes de actualización.

En el estado 3, y como resultado de la recepción del mensaje propagado por el nodo 1, el nodo 6 actualiza el tipo representativo para el tipo de parámetro que tiene almacenado en su tabla. Esto es debido a que dicho nodo mantenía en su tabla una entrada para el parámetro  $B$  y, a continuación, ha recibido una entrada con información para el parámetro  $A$  cuyo tipo está relacionado con el anterior y es, además, más general.

El cambio en el tipo representativo provoca la propagación de un mensaje de actualización a los vecinos desde nodo 6. Esto tiene como resultado que el nodo 7 actualice también la entrada correspondiente para  $B$  en su tabla de parámetros, así como el tipo representativo del grupo al tipo  $A$ , que es más general. Además, el nodo 1 propagará la información de *distancia estimada* a sus vecinos porque la nueva distancia recibida desde el nodo 6 es mayor que la actual. Finalmente, como se muestra en el estado 4, la recepción del mensaje produce que los nodos 2, 3 y 4 actualicen las entradas correspondientes del parámetro  $A$ , utilizando para ello los nuevos valores de *distancia estimada*.

### 5.2.6 Casos especiales de diseminación

Debido a la movilidad de los nodos es posible que aparezcan bucles durante el tiempo de vida de la red. Sin embargo, el algoritmo de diseminación finaliza en el caso de la existencia de bucles porque eventualmente dejan de producirse cambios en las tablas de parámetros de los nodos causadas por la recepción de mensajes, debido a que los valores para la *distancia estimada* son decrementados en cada salto realizado.



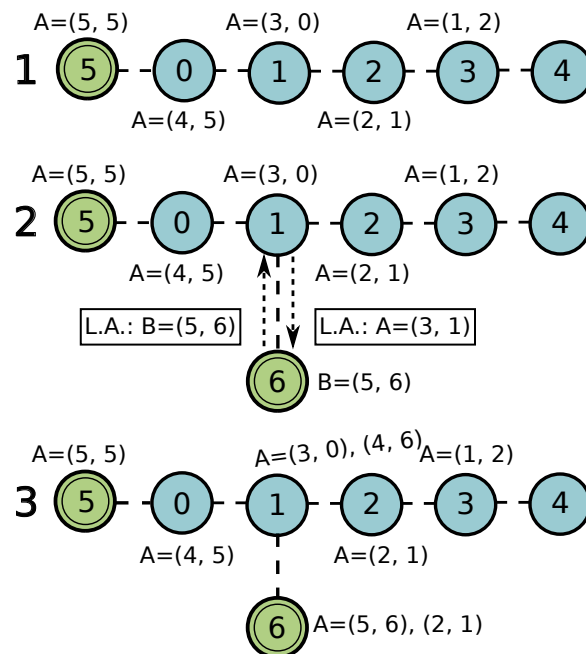
**Figura 5.9:** Diseminación de la información de parámetros en una configuración en bucle de los nodos de la red ad hoc

En la figura 5.9 se muestra el comportamiento de un bucle durante la diseminación de parámetros. Aunque el *indicador efectivo de la distancia estimada* que resulta tras estabilizarse el proceso de diseminación siempre es el mismo,

para una topología de nodos concreta y una distribución de parámetros determinada, las entradas en la lista pueden variar dependiendo del orden de los mensajes. En dicha figura se representa el resultado obtenido si el mensaje de actualización originado en el nodo 3 llega al nodo 2 antes que el mensaje proveniente del nodo 1. La ruptura de algún enlace del bucle ocasionaría la diseminación de las nuevas distancias estimadas utilizando los procesos explicados en las secciones anteriores.

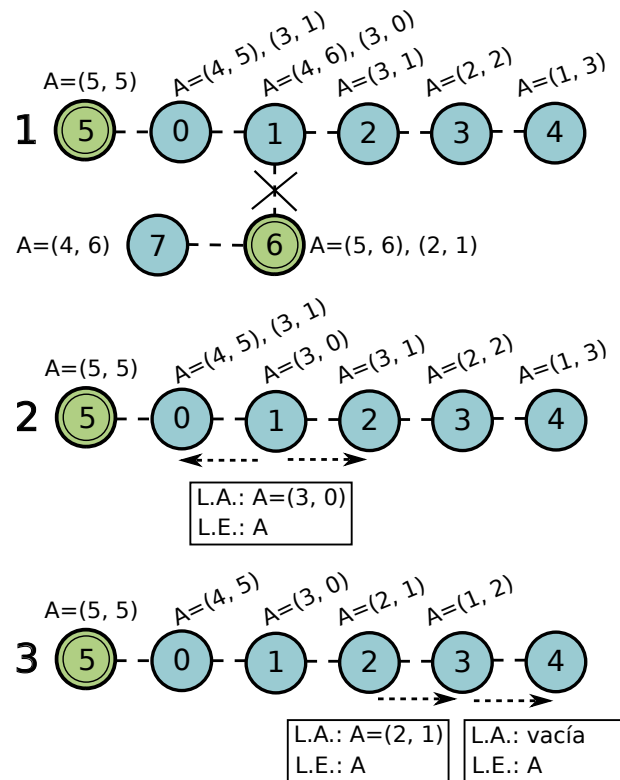
Por otro lado, la figura 5.10 muestra qué ocurre en una situación en la que anteriormente se había propagado a través de la red información con el mismo tipo, o un tipo más general, para un parámetro que está siendo diseminado desde otro nodo. Cuando el nodo 6 aparece en la red, sucede que se muestra en el estado 2, se produce un intercambio de mensajes entre los nodos 1 y 6 que produce la actualización de sus tablas.

A su vez, la modificación de la distancia efectiva en el nodo 1 para el parámetro diseminado resulta en la propagación de mensajes a sus vecinos correspondientes, los nodos 0 y 2. Este proceso se produce entre los estados 2 y 3 de la figura y no es mostrado en la misma. La diseminación de esta información es detenida en los nodos 0 y 2, debido a que la actualización de la tabla de parámetros en dichos nodos, como resultado de la recepción del mensaje propagado desde el nodo 1, no produce un cambio en los *indicador efectivo de la distancia estimada* para el parámetro actualizado.



**Figura 5.10:** Diseminación de parámetros en una red con información previa existente

Por último, la figura 5.11 representa qué sucede cuando la red se rompe como resultado de la desconexión del enlace entre un par de nodos. La *distancia estimada* del nodo 1 cambia después de eliminar la información procedente del nodo 6, tras detectar la desaparición de su vecino. La eliminación se propaga a través de la red de mediante el uso de mensajes de actualización. Dicha eliminación contiene únicamente el tipo del parámetro cuya entrada debe ser suprimida ya que el origen puede conocerse a partir del identificador del nodo que propagada la eliminación. El ejemplo muestra que la eliminación va acompañada de una adición con el nuevo valor para la *distancia estimada* del parámetro de tipo A. Esto se debe a la existencia de un valor alternativo para la distancia causado por la existencia de otro parámetro con el mismo tipo en el nodo 5. Como resultado de la propagación del mensaje de actualización, se produce una actualización en los valores para la distancia estimada de los nodos 2, 3 y 4 de la red de forma progresiva. El nodo 4 únicamente recibe una mensaje de eliminación del valor actual ya que con la distancia máxima de diseminación utilizada  $D_d = 5$  no se puede alcanzar dicho nodo desde el nodo 5.



**Figura 5.11:** Rotura en la red ad hoc produciendo la actualización de la información anteriormente diseminada

## 5.3 Búsqueda de servicios

La finalidad de esta capa de la arquitectura es llevar a cabo la búsqueda de los servicios en la red ad hoc, utilizando para ello la información sobre parámetros diseminada por el proceso explicado en la sección 5.2. Los nodos que quieren llevar a cabo la búsqueda de servicios deben iniciar ésta indicando cuáles son las características de los servicios a localizar. Hay que tener en cuenta que los nodos de la red no tienen conocimiento de cuáles son los servicios ofertados por otros nodos; por el contrario, sí que tienen conocimiento de cuáles son las características de los servicios que necesitan. En el caso concreto del protocolo de descubrimiento de servicios propuesto, las características de los servicios buscados se especifican mediante el tipo de los parámetros de entrada y salida requeridos. Si un nodo necesita localizar un determinado servicio debe describir el conjunto de los tipos de sus parámetros de entrada/salida, haciendo uso de la ontología de conceptos correspondiente.

Las búsquedas se propagan a través de la red ad hoc, de nodo a nodo, hasta que se alcanza una distancia de propagación máxima, que viene determinada por la información propagada anteriormente por la capa de diseminación. Los mensajes de búsqueda difundidos se mantienen en los nodos que los han recibido hasta que son cancelados específicamente mediante el envío de un mensaje especial por parte del nodo que inició la búsqueda correspondiente. El protocolo de propagación de los mensajes de búsqueda necesita, al igual que la diseminación de parámetros, la *detección de vecinos* que es parte de la arquitectura presentada en la sección 4.3. Por último, el descubrimiento de servicios está relacionado directamente con la gestión y mantenimiento de las rutas de comunicación entre los nodos de la red, aspecto que es explicado con mayor detalle en la sección 5.4.

### 5.3.1 Mensajes de búsqueda

El proceso de localización de servicios en la red ad hoc se inicia mediante el envío, por parte del nodo interesado, de un mensaje de búsqueda a sus vecinos. El formato de un mensaje de búsqueda  $M_B$  es el siguiente

$$M_B = (ID_B, \{S_1, S_2, \dots, S_k\}, [datos])$$

donde  $ID_B$  es un identificador único de la búsqueda que permite detener su propagación cuando es recibida más de una vez por un mismo nodo. El identificador único de cada búsqueda es generado a partir del identificador del nodo que la inicia y un contador local a dicho nodo. Un mismo mensaje de búsqueda puede contener descripciones de múltiples servicios  $S_i$  a localizar, y por cada una de ellas es posible especificar el tipo de los parámetros de entrada y salida que lo describen.

La descripción de uno de los servicios contenido dentro de los mensajes de búsqueda está definida como

$$S_i = \{P_E, P_S, TTL_i\}$$

donde  $P_E$  y  $P_S$  es el conjunto de los parámetros de entrada y salida, respectivamente, que describen el servicio a localizar. Además, cada descripción de servicio tiene asociado un tiempo de vida (*Time to Live - TTL*) que determina, durante la propagación del mensaje y junto con la información diseminada sobre los parámetros, si el mensaje debe seguir siendo propagado otro salto a través de la red. Los parámetros de entrada y salida del servicio a su vez como

$$P_E = \{p_1, p_2, \dots, p_j\} \mid p \in O, 0 \leq j$$

$$P_S = \{p_1, p_2, \dots, p_k\} \mid p \in O, 0 \leq k$$

donde  $O$  es la ontología de conceptos utilizada para describir los tipos de los parámetros.

### 5.3.2 Propagación de búsquedas

Cada vez que un mensaje de búsqueda es recibido por un nodo se lleva a cabo una comprobación entre los tipos de los parámetros contenidos en el mensaje y las entradas contenidas en la tabla de parámetros. En la sección 4.4.2 se explican las relaciones que pueden existir entre distintos tipos de parámetros de acuerdo a una ontología de conceptos. Basándose en estas relaciones, el protocolo de descubrimiento de servicios permite especificar dos tipos de búsquedas:

- **Exacta:** localiza aquellos parámetros que son exactamente del tipo indicado. Los parámetros cuyo tipo es más general o más específico que el tipo buscado son ignorados. Por ejemplo, si la ontología define que  $A$  subsume a  $B$  y se inicia una búsqueda de parámetros de tipo  $A$ , únicamente se encontrarán servicios cuyo tipo de parámetro tenga exactamente el tipo buscado, es decir  $A$ .
- **Genérica:** localiza aquellos parámetros que tengan el mismo o un tipo más específico que el indicado. Para el caso del ejemplo anterior se localizan tanto servicios que posean parámetros de tipo  $A$  como cualquier otro tipo más específico, como por ejemplo del tipo  $B$ .

El proceso de descubrimiento de servicios basado en parámetros se lleva a cabo mediante la propagación de mensajes de búsqueda a través de la red ad hoc. Los nodos inician las búsquedas mediante el envío de un mensaje con el contenido explicado en la sección 5.3.1. Cada vez que un mensaje de búsqueda

es recibido por un nodo es comparado con el tipo de los parámetros de los servicios locales, es decir, aquellos servicios que son provistos por dicho nodo. Si se produce un emparejamiento entre una de las descripciones de servicio contenidas en el mensaje de búsqueda y alguno de los servicios ofertados por el nodo se considera que se ha encontrado un servicio compatible.

Cuando se localiza un servicio, el nodo que lo provee contesta con un mensaje de respuesta dirigido al nodo que ha iniciado la búsqueda. La propagación a través de la red de los mensajes de búsqueda y el envío de los mensajes de respuesta tienen como resultado la creación de las rutas de comunicación, tal y como se explica en la sección 5.4. Obviamente, es posible que se encuentren servicios compatibles en más de un nodo y que, por lo tanto, se produzcan respuestas desde varios de ellos con servicios compatibles, hacia el emisor de la búsqueda. Un mensaje de respuesta  $M_R$  contiene la descripción completa de uno o varios servicios localizados en el nodo que ha aceptado el mensaje de búsqueda, y se define como

$$M_R = (\{S_1, S_2, \dots, S_k\})$$

y cada servicio  $S_i$  se encuentra definido a su vez como

$$S_i = \{ID_S, P_E, P_S, n\} \mid n \in N$$

donde  $ID_S$  es el identificador del servicio en la red, compuesto por el identificador propio del servicio y el del nodo que lo provee, y  $P_E$  y  $P_S$  contienen la descripción de todas sus entradas y salidas y  $n$  el identificador del nodo que provee el servicio.

Las entradas y salidas del servicio compatible localizado son enviadas al nodo que inició la búsqueda con la finalidad de que así pueda conocerse el tipo exacto de los parámetros que posee dicho servicio. Hay que tener en cuenta que las búsquedas pueden contener parámetros de un tipo más general que el provisto realmente por un servicio localizado, y por lo tanto, el tipo exacto localizado solamente es conocido cuando dicha información es devuelta al nodo emisor desde el proveedor del servicio. Por último, como la descripción del servicio retornada en el mensaje de respuesta contiene la dirección única  $n$  del nodo que lo provee, se permite que un mismo servicio se encuentre replicado en distintos nodos.

Durante la propagación de los mensajes de búsqueda a través de la red se aplica un procedimiento de *poda* que tiene como objeto reducir el número de mensajes enviados. Los mensajes de búsqueda son propagados, desde el nodo que inicia la búsqueda, mediante operaciones de difusión a los vecinos. Se produce así una propagación mediante *inundación (flooding)* que avanza salto a salto a través de los nodos de la red ad hoc. La *poda* de los mensajes de búsqueda tiene como finalidad evitar la propagación de mensajes innecesarios por

aquellas zonas donde se tiene la seguridad de que no van a existir servicios compatibles. Al aplicar el proceso de *poda* los nodos únicamente propagan las búsquedas a sus vecinos si el TTL asociado a un parámetro  $p$  buscado, es mayor o igual que la *distancia en saltos* a dicho parámetro, según la información contenida en la tabla de parámetros del nodo que actualmente está procesando el mensaje. El TTL de un parámetro es decrementado en cada salto que lleva a cabo el mensaje de búsqueda.

Por lo tanto, para que la búsqueda de un parámetro sea propagada a los vecinos de un nodo se tiene que cumplir, en el caso de la búsqueda exacta, que el TTL para un parámetro  $p$ , representado como  $TTL(p)$  cumpla la siguiente condición

$$TTL(p) \geq d_s(p') \text{ cuando } p \equiv p'$$

y para el caso de una búsqueda de servicios genérica

$$TTL(p) \geq d_s(p') \text{ cuando } p \supseteq p'$$

donde  $p'$  se obtiene, en ambos casos, de la información contenida en la tabla de parámetros del nodo actual. A partir de la entrada de la tabla de parámetros para el parámetro  $p'$  puede obtenerse la *distancia en saltos estimada*  $d_s(p')$ , hasta el origen más cercano para dicho parámetro  $p'$ , mediante

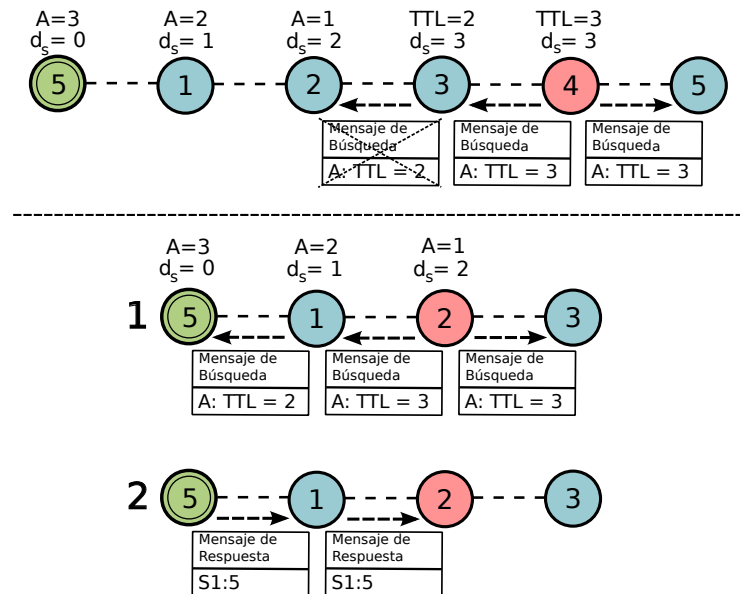
$$d_s(p') = D_d - d_e(p')$$

donde  $D_d$  es la distancia máxima de diseminación utilizada en dicha red ad hoc y  $d_e(p')$  la *distancia estimada* para el parámetro  $p'$  de acuerdo a la información contenida en la tabla de parámetros del nodo. El valor de la *distancia estimada* para un parámetro para el que no exista entrada en la tabla de parámetros, es decir, para el que el nodo actual no dispone de ninguna información, se toma como cero en la ecuación anterior.

Si las condiciones anteriores para el TTL de un parámetro no se cumplen significa que el mensaje de búsqueda se encuentra, durante su propagación a través de la red ad hoc en un nodo que, aunque situado dentro de la zona de diseminación de parámetros de un nodo proveedor del parámetro buscado, no puede alcanzar al proveedor del servicio correspondiente con el TTL actual.

Se muestra un ejemplo de esta situación en la figura 5.12. En la parte superior de esta figura, el mensaje de búsqueda no es propagado por el nodo 3 porque el TTL para el parámetro buscado es menor que la distancia calculada para alcanzarlo, según la información contenida en la tabla de parámetros. Es decir, el parámetro no puede ser encontrado porque el mensaje de búsqueda tiene un  $TTL = 2$ , y debido a que no existe una entrada en la tabla de parámetros del nodo 3 su  $d_e = D_d = 3$  y, por lo tanto, la distancia en saltos al parámetro es  $d_e = 3$ , no cumpliéndose que  $TTL = 2 \geq d_e = 3$ .





**Figura 5.12:** Propagación de los mensajes de búsqueda y posterior aplicación del proceso de *poda* usando la información diseminada

Sin embargo, en la segunda parte de la figura 5.12 el mensaje de búsqueda es propagado hasta el nodo que proporciona un servicio con un parámetro compatible. Al aceptar el mensaje de búsqueda, dicho nodo envía un mensaje de respuesta, mostrado en el estado 2, que contiene la descripción completa del servicio compatible localizado. Ese mensaje de respuesta es enviado hacia el nodo que ha iniciado la búsqueda gracias a las rutas de comunicación creadas durante la propagación del mensaje anterior, según explica en la sección 5.4.

### 5.3.3 Mantenimiento de búsquedas activas

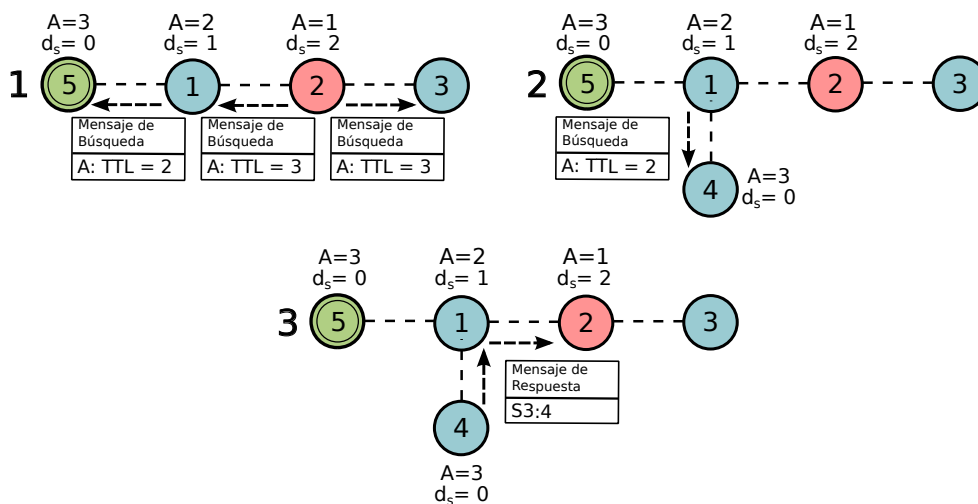
En el protocolo, las búsquedas realizadas por los nodos de la red son mantenidas hasta que son explícitamente canceladas. Cada vez que un nodo recibe un mensaje de búsqueda lo almacena en una lista de *búsquedas activas*. Cuando se producen cambios en la topología de la red, debido a la aparición o desaparición de nodos, se llevan a cabo acciones con las búsquedas activas almacenadas para adecuarse al nuevo estado de la red.

Se identifican los siguientes sucesos que suponen la gestión de sus búsquedas activas por parte de un nodo de la red:

- **Aparición de un vecino:** cuando un nodo detecta un nuevo vecino es posible que dicho nodo provea acceso a nuevos parámetros, ya sean locales al propio nodo o situados en otra subred a la que éste proporciona acceso. Si en la primera red se habían iniciado búsquedas de parámetros éstas

deberán propagarse a través del nuevo nodo, y de la nueva subred, para localizar los parámetros que puedan existir. Por esta razón, los nodos que detectan un nuevo vecino, y han recibido búsquedas que se encuentran activas, llevan a cabo la *repropagación* de dichas búsquedas con el objeto de que el nuevo nodo las reciba. De esta forma la propagación de las búsquedas continúa a través de los nuevos nodos que hayan sido detectados, aplicando de nuevo el proceso de  *poda*  explicado anteriormente.

- **Desaparición de un vecino:** la desaparición de un vecino tiene como resultado que los mensajes de búsqueda que habían sido propagados desde dicho vecino deban ser eliminados del nodo actual. Además, la desaparición del vecino supone también la eliminación de una ruta de comunicación con el nodo que originó la búsqueda. Para ello, el nodo que ha detectado la desaparición propaga a sus vecinos un mensaje de eliminación de la ruta inválida que será difundido a través de la red, aplicando el proceso explicado en la sección 5.4.



**Figura 5.13:** Propagación de una búsqueda activa tras la aparición de un nuevo vecino

En la figura 5.13 se muestra cómo la aparición de un nuevo vecino produce la repropagación de una búsqueda activa desde el nodo 1. Esta búsqueda ha sido realizada en el estado 1 de la figura. Cuando se produce la aparición del nodo 4 la búsqueda es procesada de la forma usual por el nuevo vecino, que responde con un mensaje de respuesta hacia el nodo 2 utilizando para ello las rutas de comunicación creadas durante la repropagación del mensaje. El intercambio de mensajes es mostrado en los estados 2 y 3 de la figura.

Por otro lado, y como resultado del registro de nuevos servicios en el nodo actual, pueden aparecer nuevos parámetros locales. Si el nodo tiene alguna búsqueda activa que es compatible con los parámetros recientemente añadidos se lleva a cabo la notificación del nodo que inició dicha búsqueda. Para la notificación se utiliza, como en cualquier otro caso, un mensaje de respuesta que se envía utilizando las rutas de comunicación existentes.

### 5.3.4 Cancelación de búsquedas

Las búsquedas de mensajes recibidas se mantienen como búsquedas activas hasta que son explícitamente canceladas por el nodo que las inició. Las búsquedas son propagadas automáticamente a través de los nuevos nodos que aparecen, como se explica en la sección 5.3.2. Como se ha comentado anteriormente, el objetivo de mantener las búsquedas es que, un nodo, no solamente localice los servicios que se encuentran disponibles en la red en el momento en el que una búsqueda se propaga, sino también aquellos otros servicios que pueden aparecer posteriormente. Así, a partir del momento en el que un nodo inicia una búsqueda, éste puede ser notificado con cualquier servicio compatible, tanto en el estado actual como posteriormente. Si el nodo quiere detener el proceso de búsqueda debe indicarlo mediante el envío de un mensaje especial que llevará a cabo la cancelación de la búsqueda activa.

Los mensajes de cancelación indican qué descripciones contenidas en una búsqueda deben ser eliminadas. Esto permite que un mensaje de búsqueda enviado anteriormente pueda ser cancelado en su *totalidad* o de forma *parcial*. Hay que tener en cuenta que una búsqueda puede agrupar varias descripciones de servicios que son propagadas de forma conjunta a través de la red, tal y como se explica en la sección 5.3.1. Por lo tanto, es posible que el nodo que ha iniciado una búsqueda quiera cancelar únicamente parte de la misma y no la búsqueda en su totalidad. Se permite, por lo tanto, que únicamente unas descripciones concretas de servicios puedan ser eliminadas de una búsqueda, mientras que el resto pueden continuar activas en la red.

El contenido de un mensaje de cancelación de búsqueda  $M_C$  se define como

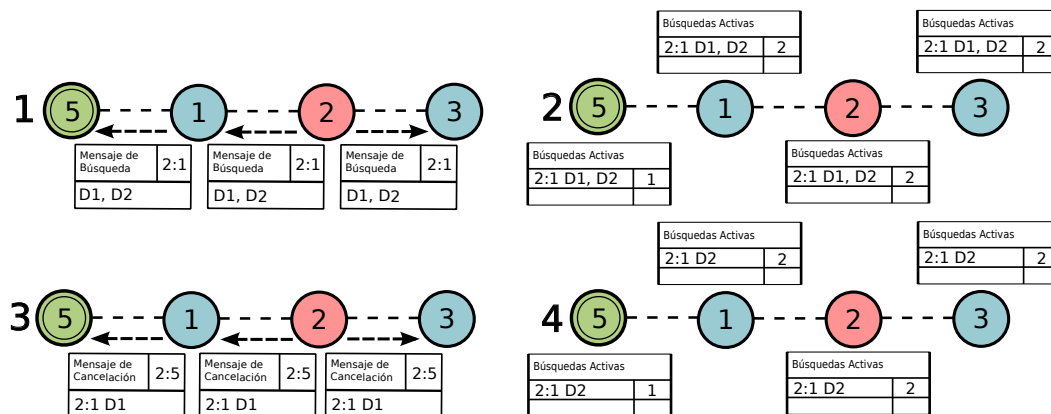
$$M_C = (ID_B, \{S_1, S_2, \dots, S_k\})$$

donde  $ID_B$  es el identificador de la búsqueda sobre la que se aplica la cancelación y  $S_i$  las descripciones de las búsquedas de servicios que deben ser eliminadas de la búsqueda activa correspondiente a dicho identificador.

Los mensajes de cancelación son enviados por el nodo que inició la búsqueda a eliminar. Los vecinos que reciben el mensaje de cancelación comprueban si tienen una búsqueda activa que se corresponda con el identificador  $ID_B$  contenido en el mensaje recibido. En el caso de que el mensaje suponga una

cancelación completa cada nodo receptor procederá a eliminar la búsqueda correspondiente de su lista de activas. La cancelación completa ocurre cuando el mensaje contiene todas las descripciones de servicio incluidas en una búsqueda activa determinada y mantenida por un nodo. Por el contrario, si el mensaje cancela una búsqueda de forma parcial, debido a que no incluye todas las descripciones, se lleva a cabo únicamente la eliminación de las descripciones indicadas y la búsqueda se sigue manteniendo como activa por el nodo. Esto significa que las descripciones de servicio que se han mantenido pueden seguir siendo emparejadas con servicios existentes o que aparezcan en la red.

En cualquiera de los dos casos anteriores, el nodo procede a propagar el mensaje de cancelación de búsqueda a sus propios vecinos. La propagación continúa hasta que el mensaje de cancelación llega a un nodo sobre el que no produce ninguna modificación en sus búsquedas activas. Esto significa que el nodo no había recibido anteriormente el mensaje de búsqueda que se quiere cancelar y, por lo tanto, no es necesario continuar con la propagación de la cancelación. Obviamente, los mensajes de cancelación ya procesados por un nodo son ignorados al ser recibidos una segunda vez, lo que supone que la propagación de los mensajes de cancelación a través de la red se detendrá en algún momento.



**Figura 5.14:** Cancelación parcial de una búsqueda activa propagada

Por ejemplo, en la figura 5.14 se muestra el proceso de cancelación parcial de una búsqueda activa por parte un nodo. Inicialmente, el nodo 2 lleva a cabo una búsqueda de servicios mediante el envío de un mensaje que contiene dos descripciones:  $D_1$  y  $D_2$ , cada una de ellas referente a un conjunto de parámetros a satisfacer por un servicio. La situación de las búsquedas activas mantenidas por cada nodo de la red, tras procesar la búsqueda, está representado en el estado 2 de la figura.

Un tiempo después, el nodo 2 quiere cancelar dicha búsqueda porque ya no sigue interesado en que el descubrimiento de servicios compatibles con la

descripción  $D_1$ . El nodo 2 envía, por lo tanto, un mensaje de cancelación que contiene el identificador de la búsqueda afectada y aquellas descripciones que deben ser eliminadas de la misma. En la figura se muestra que los mensajes de cancelación y búsqueda contienen un identificador único formado por el número de nodo emisor y un contador local.

El mensaje de cancelación es procesado por los nodos vecinos, y propagado por dichos nodos a sus respectivos vecinos, hasta que, tras ser procesado, no se produce ninguna modificación en las búsquedas activas. El resultado de este proceso se representa en el estado 4 de la figura, donde se observa que la búsqueda sigue activa en los nodos de la red, pero únicamente para la búsqueda de servicios que se correspondan con la descripción  $D_2$ . Si el mensaje de cancelación hubiera contenido la eliminación de las dos descripciones, las búsquedas activas correspondientes hubieran sido suprimidas de los nodos afectados. Por supuesto, el nodo 2 puede enviar, en cualquier otro momento posterior, otro mensaje de cancelación para eliminar la descripción restante y así suprimir completamente la búsqueda.

Es posible, por otro lado, que un nodo que ha iniciado una búsqueda desaparezca con respecto a sus vecinos, ya sea debido a su movimiento relativo o a otra causa, como por ejemplo, un posible fallo en el mismo o en la comunicación. Si esto ocurre, los nodos vecinos son notificados de esta desaparición por la capa de *detección de vecinos*, explicada en la sección 4.5.1. Tras la notificación, si estos nodos habían recibido búsquedas provenientes del vecino desaparecido, proceden a llevar a cabo la cancelación completa de las búsquedas inválidas y a la notificación a sus vecinos mediante el envío de los mensajes de cancelación correspondientes.

Por último, indicar que para evitar la cancelación errónea de búsquedas que todavía pueden ser correctas debido a la existencia de caminos alternativos, una búsqueda únicamente es cancelada si el mensaje de cancelación aplicado procede del mismo vecino que había propagado la búsqueda afectada anteriormente. Así, en la figura 5.14 se muestra, para los estados 2 y 4, como para cada búsqueda activa almacenada se guarda también el identificador del nodo vecino que había propagado dicha búsqueda al nodo.

## 5.4 Encaminamiento de mensajes

En una red ad hoc no existe un conocimiento previo de las rutas de comunicación existentes entre los diferentes nodos que la constituyen. Es necesario un mecanismo que permita a los nodos descubrir rutas para comunicarse con otros nodos de su interés. En el caso del protocolo de descubrimiento de servicios propuesto se supone que esta comunicación entre los nodos se establece con el fin de acceder a los distintos servicios disponibles. Por lo tanto, el descubrimiento de las rutas de comunicación está orientado a permitir el envío de

mensajes entre el nodo que inicia la búsqueda de servicio y aquellos nodos de la red que proporcionan los servicios compatibles.

### 5.4.1 Tabla de encaminamiento

Cada nodo de la red mantiene una *tabla de encaminamiento* con información sobre las rutas de comunicación, con otros nodos de la red, que han sido descubiertas. Esta tabla, que se actualiza como resultado de los procesos de envío de mensajes y la ocurrencia de distintos eventos, contiene una entrada por cada ruta de comunicación descubierta. Cada entrada de una tabla de encaminamiento  $E_i$  se define como una función

$$E_i : D \rightarrow V; D, V \subseteq N$$

que relaciona el conjunto de destinos alcanzables  $D$  con el conjunto de vecinos  $V$  a través de los cuales se puede realizar el siguiente salto de un mensaje para llegar a un destino concreto.  $V$  es un subconjunto de  $N$ , que representa a su vez los identificadores de todos los nodos que constituyen la red. La tabla de encaminamiento de un nodo mantiene entradas  $E_e$  de la forma

$$E_e = \{(ID_R, d, v) \mid d, v \in N\}$$

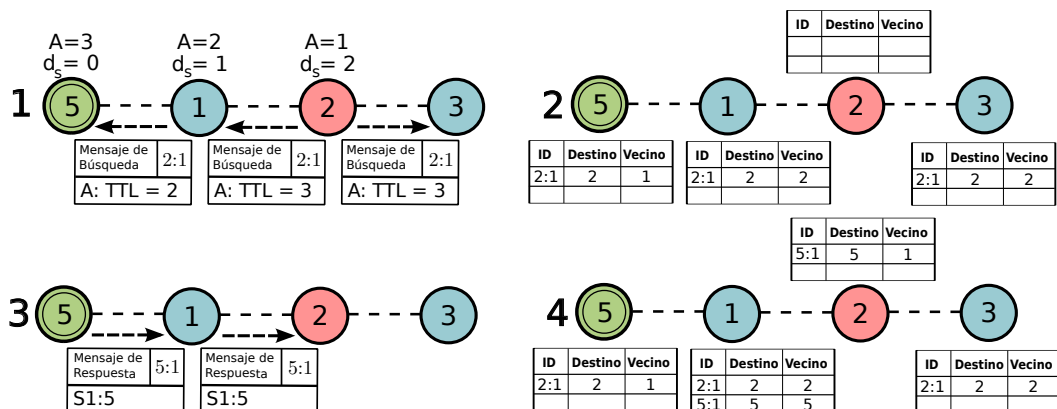
donde  $ID_R$  es el identificador único de una ruta,  $d$  es el nodo destino de la misma y  $v$  el siguiente salto para alcanzar dicho destino.

Se considera que dos rutas de una tabla son distintas si poseen un identificador de ruta distinto. El identificador de cada ruta se obtiene a partir del identificador del mensaje que, durante su propagación, ha creado la entrada de la tabla. De esta forma, una ruta a través de la red ad hoc está identificada de forma unívoca por el mensaje que la ha creado. Cada entrada de la tabla contiene, además de su identificador único, la dirección única del nodo destino de la ruta, la dirección del vecino que constituye el siguiente salto de la red y el número de saltos necesarios para alcanzarlo. La tabla de encaminamiento puede contener más de una ruta para alcanzar un mismo destino debido a la posibilidad de obtener respuestas del mismo nodo por distintos caminos.

Durante el envío de los mensajes de comunicación se ha seleccionado la ruta con un número menor de saltos para alcanzar el nodo destino, con objeto de reducir el tiempo de envío de los mensajes. Sin embargo, podrían aplicarse otras políticas para la selección de rutas que no han sido contempladas en esta investigación.

La figura 5.15 muestra el contenido de las tablas de encaminamiento, de los nodos de una red ad hoc en un momento determinado, creadas como resultado de la propagación de los mensajes de búsqueda y de respuesta. En el estado 1 se produce una búsqueda del parámetro de tipo  $A$  desde el nodo 2. La propagación del mensaje de búsqueda hasta el nodo 5 crea las rutas que permiten la

comunicación hacia el nodo 2 desde todos aquellos nodos que han recibido el mensaje, como se representa en el estado 2. Al aceptar el mensaje de búsqueda, el nodo 5 responde con un mensaje *unicast* que usa la ruta anteriormente creada, y que produce a su vez la actualización de las rutas de encaminamiento de los nodos que lo procesan. En el estado 4 se observa cómo quedan las tablas de encaminamiento en los nodos 1 y 2, que permiten el envío de mensajes desde dichos nodos hacia el nodo 5. Por el contrario, desde el nodo 3 no se puede establecer ninguna comunicación con el nodo 5 ya que no ha recibido mensaje de respuesta desde dicho nodo.



**Figura 5.15:** Estado de las tablas de encaminamiento tras la propagación de los mensajes de búsqueda y respuesta

### 5.4.2 Mensajes de comunicación

El protocolo propuesto permite, una vez que se han construido las rutas de comunicación correspondientes, el envío de dos tipos de mensajes: *unicast* y *multicast*. El primero posibilita el envío de un mensaje a un único nodo dentro de la red, mientras que el segundo se utiliza para que el mismo mensaje sea encaminado hacia varios nodos destinos. Para el envío de mensajes a un nodo es necesario conocer el identificador del destino. Así, el identificador del nodo destino debe haber sido descubierto anteriormente por el proceso de descubrimiento de servicios. Los mensajes *unicast* se han implementado como mensajes *multicast* en los que existe un único destino para el mensaje. Se aprovecha así el mecanismo construido para el envío de mensajes *multicast* a varios nodos.

Una búsqueda de parámetros originada en un nodo puede localizar varios destinos con los parámetros buscados. Por otro lado, un mismo nodo con parámetros puede ser encontrado por varios mensajes de búsqueda originados en distintos nodos. El envío de mensajes *multicast* permite, en ambos casos, comunicar un nodo con un grupo. En el primer caso, posibilita que el nodo que ha

iniciado la búsqueda se comunique con todos los nodos que proveen los parámetros buscados, mientras que en el segundo, permite que un nodo que ofrece uno o varios parámetros se pueda comunicar con los distintos nodos desde los que ha recibido una búsqueda.

El formato de un mensaje *multicast*  $M_M$  es el siguiente

$$M_M = (n, d_1, d_2, \dots, d_i, [datos]) \mid n, d \in N$$

donde  $n$  es el identificador del nodo que envía el mensaje y contiene además una lista con los identificadores  $d_i$  de los destinos del mensaje, así como un campo para incluir datos enviados por las capas superiores que usen el mecanismo de encaminamiento de mensajes.

Para poder realizar el envío de mensajes *multicast* deben existir tablas de encaminamiento en los nodos intermedios que permitan el envío de los mensajes siguiendo la ruta correcta. Cuando un nodo ha recibido un mensaje de búsqueda se habrá creado la ruta correspondiente hasta el nodo emisor, permitiendo el envío de mensajes destinados a dicho nodo. Así, el nodo emisor de una búsqueda no puede enviar mensajes de comunicación a los nodos en los que se encuentran los parámetros hasta que haya recibido de ellos un mensaje de respuesta a la búsqueda. Sólo a partir de dicho momento podrá existir una ruta hasta el nodo que envió la respuesta.

Los mensajes de comunicación se transmiten solamente a través de los vecinos indicados en la tabla de encaminamiento de cada nodo. Debido a que, al realizar el envío de un mensaje, éste es recibido por todos los vecinos del nodo, tanto por los que tienen una ruta hacia el destino como los que no, el mensaje *multicast* contiene en cada salto la lista de los nodos a través de los que tiene que ser enviado. Los vecinos son obtenidos de la tabla de encaminamiento del nodo que hace el envío. Por otro lado, durante la recepción de un mensaje, aquellos vecinos que no estén incluidos en la lista de vecinos a través de los que debe ser enviado el mensaje, procederán a descartarlo.

La figura 5.16 recoge el algoritmo que gestiona el envío de los mensajes *multicast*. El nodo que quiere enviar un mensaje a un conjunto de destinatarios crea un mensaje *multicast* que contiene la lista de nodos destinatarios del mensaje y su propio identificador como origen del mensaje. Además, como vecino a través del que se debe enviar el mensaje se encuentra inicialmente el propio nodo emisor. Así, el mensaje es procesado como si hubiera sido recibido desde el propio nodo emisor, simplificando el algoritmo de gestión de mensajes.

Cuando un nodo recibe un mensaje *multicast* primeramente comprueba si su identificador está incluido en la lista de vecinos que deben procesar el mensaje, en caso contrario descarta el mensaje. Si el mensaje no ha sido descartado se comprueba si el nodo actual está en la lista de destinatarios del mensaje *multicast*. Si esto es así, el mensaje es aceptado por el nodo y el identificador de dicho nodo es eliminado de la lista de destinatarios. A continuación, se procede



```

Entrada:
     $m$  : mensaje recibido
     $E_i$  : tabla de encaminamiento local
     $n_i$  : nodo actual

if  $n_i \in \text{Vecinos}(m)$  then
    | if  $n_i \in \text{Destinos}(m)$  then
    | | Aceptar( $m$ )
    | | EliminarDestino( $n_i, m$ )
    | end
    | for each Destino  $d$  in  $m$  do
    | | if  $d \notin E_i$  then
    | | | EliminarDestino( $d, m$ )
    | | end
    | end
    | if Destinos( $m$ ) > 0 then
    | |  $m_{\text{nuevo}} \leftarrow \text{Vacío}$ 
    | | for each Destino  $d$  in  $m$  do
    | | |  $v \leftarrow \text{ObtenerVecino}(d, E_i)$ 
    | | |  $m_{\text{nuevo}} \leftarrow v$ 
    | | |  $m_{\text{nuevo}} \leftarrow \text{Datos}(m)$ 
    | | | end
    | | | propagar( $m_{\text{nuevo}}$ )
    | | end
    | end
end

```

**Figura 5.16:** Algoritmo para el procesamiento de los mensajes *multicast*

a eliminar de la lista de destinos del mensaje aquellos que son inválidos. Un destino de un mensaje *multicast* es inválido si no existe desde el nodo actual una ruta, de acuerdo a su tabla de encaminamiento, hasta dicho destino. Si, tras realizar los pasos anteriores, el mensaje no contiene más destinos en su lista será descartado. Si por el contrario existen más destinos por alcanzar se obtiene, consultando la tabla de encaminamiento del nodo actual, los vecinos a través de los que debe enviarse el mensaje para alcanzarlos. Se crea así una nueva lista de vecinos para el mensaje *multicast*. El mensaje recibido se copia en uno nuevo substituyendo la lista de vecinos por los nuevos e incluyendo los destinatarios restantes. Se procede al envío del nuevo mensaje y el proceso continúa de esta manera hasta que se alcanzan todos los destinatarios o se determina que el mensaje debe ser descartado por no existir más rutas válidas.

### 5.4.3 Gestión de las rutas de comunicación

El proceso de creación de las rutas de comunicación no se inicia hasta que los nodos comienzan la búsqueda de servicios. Cuando un nodo requiere localizar un servicio envía a todos sus vecinos un mensaje de búsqueda que será propagado de la forma explicada en la sección 5.3.2. Durante la transmisión del mensaje de búsqueda a través de los nodos, se crea una ruta inversa que permite llegar, desde cualquier nodo receptor del mensaje, hasta aquel que inició la búsqueda. La ruta inversa tiene forma de árbol en cuya raíz se encuentra el nodo que inició la búsqueda. Los nodos que han recibido el mensaje contendrán en sus tablas información que permitirá el envío de mensajes hacia dicho nodo. Sin embargo, el nodo origen de la búsqueda no podrá comunicarse con ningún otro nodo de la red hasta que se creen las rutas inversas producidas como resultado de la localización de servicios y el envío de los mensajes de respuesta.

La desaparición de un vecino tiene como resultado que los destinos accedidos a través de dicho nodo ya no puedan ser alcanzados. Cuando un nodo detecta que un vecino suyo ha desaparecido elimina de su tabla de encaminamiento aquellas rutas que usaban dicho vecino como siguiente salto para alcanzar algún nodo remoto. Si se ha producido la eliminación de alguna ruta, el nodo envía a sus vecinos un mensaje de eliminación de ruta conteniendo una lista con los identificadores de todas las rutas que ya no son válidas.

El contenido del mensaje de eliminación  $M_E$  se define como

$$M_E = (n, ID_{R_1}, ID_{R_2}, \dots, ID_{R_i}) \quad n \in N, \quad 1 \leq i$$

que contiene el conjunto de los identificadores de rutas a eliminar por los vecinos del nodo emisor. Los nodos que reciben dicho mensaje proceden a comprobar sus tablas de encaminamiento y a eliminar las rutas correspondientes, si éstas existen. El proceso continúa hasta que el mensaje de eliminación de ruta es recibido por un nodo sobre el que no produce ninguna modificación en su tabla de encaminamiento.

En la figura 5.17 se muestra este proceso cuando el nodo 1 detecta la desaparición del nodo 2. En el estado 1 se muestra la situación inicial de las tablas de encaminamiento de cada nodo de la red. Se produce, posteriormente, la desconexión entre los nodos 1 y 2, evento que es detectado por el mecanismo para la gestión de vecinos explicado en la sección 4.5.1. El nodo 1 responde a la desaparición de su vecino, el nodo 2, con la eliminación de su tabla de aquella información proveniente de dicho nodo, y la propagación de un mensaje a sus vecinos. En este caso, dicho mensaje será procesado únicamente por el nodo 5, que procede a eliminar las entradas correspondientes de su tabla de encaminamiento, tal y como se muestra en el estado 3 de la figura. Por otro lado, el nodo 2, elimina también de su tabla las rutas que utilizaban el nodo 1 desaparecido.

En este caso, el mensaje de eliminación de rutas enviado por el nodo 2, no produce ningún cambio en el nodo 3, ya que dicho nodo no contiene ninguna ruta hacia el nodo 5.

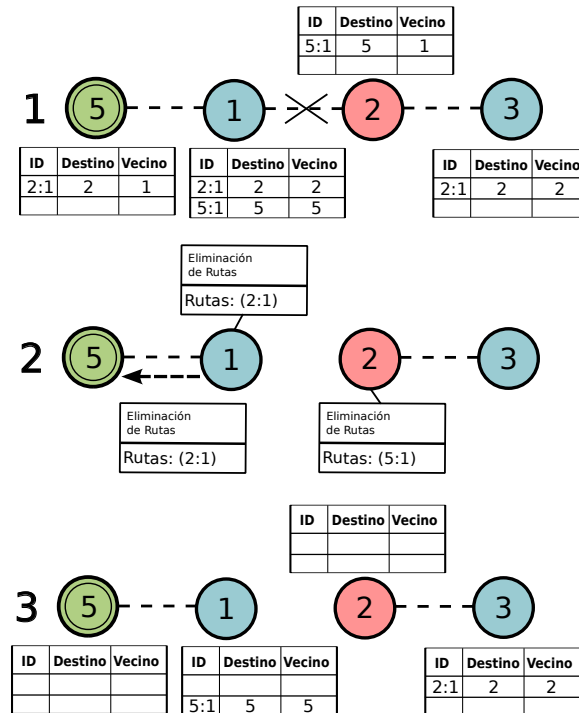


Figura 5.17: Eliminación de rutas debido a la desaparición del vecino de un nodo

## 5.5 Evaluación

El protocolo para el descubrimiento de servicios propuesto en este capítulo ha sido completamente implementado y evaluado. Se ha utilizado el simulador de red *ns-2* extendido con AgentJ para permitir la ejecución de código Java en cada nodo de la red. Se puede consultar más información acerca de las implicaciones de la experimentación con redes ad hoc mediante el uso de simuladores en la sección 2.2. Por otro lado, la implementación se ha llevado a cabo utilizando Java SE 1.6 y las librerías de comunicación estándar proporcionadas por dicho lenguaje. En concreto, se han utilizado *sockets* UDP como mecanismo fundamental para el envío de mensajes de difusión durante la comunicación inalámbrica en modo ad hoc entre los nodos.

Los escenarios de evaluación se han construido con los parámetros recogidos en la tabla 5.1, teniendo en cuenta también los parámetros del simulador

**Tabla 5.1:** Configuración de la evaluación del protocolo de descubrimiento

Parámetro simulación	Valor
Nodos	100
Área de simulación	700 m x 700 m
Modelo de movilidad	<i>Random waypoint</i>
Velocidad nodos	Distribución uniforme de [0, 5] m/s
Tiempo de pausa	50, 100 s
Velocidad de transmisión	11 Mb/s
Periodo de baliza	2000 s
Distancia máx. de diseminación $D_d$	10 saltos
TTL de búsqueda	10 saltos
Frecuencia de búsqueda	0,2 búsquedas/s
Búsqueda simultáneas	5 búsquedas

ns-2 recogidos en la tabla 2.2, la cual define distintos aspectos de la transmisión inalámbrica y de bajo nivel de la red MANET.

El modelo de movilidad *Random waypoint* seleccionado se basa en el movimiento de los nodos de la red con velocidades y direcciones aleatorias durante el tiempo de simulación. Los nodos de la red alternan sus movimientos con *tiempos de pausa*, en los que permanecen quietos en su posición. Así, cuando para un nodo expira su tiempo de pausa, dicho nodo elige un destino aleatorio dentro del área de simulación y una velocidad dentro del rango de velocidades disponibles y comienza a moverse. Cuando el nodo alcanza su destino vuelve a aplicar un tiempo de pausa aleatorio y se repite el proceso. En el trabajo realizado por Camp, Boleng y Davies (2002) se exponen las características de éste y otros modelo de movilidad.

De acuerdo a Kurkowski, Navidi y Camp (2007a), el tamaño del área de simulación, la distancia máxima de transmisión y el número de nodos seleccionados para las simulaciones permiten obtener un escenario que posee las siguientes características:

- *Partición media de la red*  $\leq 5\%$ . Esta característica hace referencia al porcentaje de nodos de la red que de media se encuentran desconectados en un instante determinado. La selección de un valor bajo para la partición de la red garantiza que el protocolo es probado en un escenario de experimentación en el que la topología de la red posibilita llegar a localizar casi la totalidad de los servicios disponibles.

- *Camino medio más corto = 4,15 saltos*. Representa la distancia media más corta que puede existir entre dos nodos de la red. El protocolo de descubrimiento se prueba así con una configuración de red que permite que realmente se haga uso de las características de encaminamiento del protocolo, mediante la propagación de los mensajes de las capas de diseminación y búsqueda.

Como se observa en la tabla 5.1, tanto la distancia máxima de diseminación  $D_d$  como el TTL para los mensajes de búsqueda han sido configurados con un valor de 10 saltos. Teniendo en cuenta que ambos valores, en la configuración usada, son mucho mayores que el camino medio más corto para la red ad hoc, se consigue que tanto los mensajes de diseminación como de búsqueda cubran por completo la red. Se permite así probar el protocolo garantizando la posibilidad de localizar casi la totalidad de los servicios existentes en la red y conocer, por lo tanto, si la solución se comporta de la manera adecuada.

Por otro lado, para la realización de las pruebas se ha utilizado una ontología en la que no se han incluido conceptos pertenecientes a ningún dominio específico. Únicamente han sido representadas relaciones de generalización/especialización entre los diferentes conceptos. Los tipos de los parámetros de entrada y salida de los servicios son asignados de forma aleatoria a una categoría, de tal forma que están distribuidos al azar, y se crea un número variable de relaciones jerárquicas entre los parámetros de los servicios. Se consigue así probar el protocolo propuesto en un caso general sin necesidad de crear un conjunto de servicios para un dominio concreto.

Por último, las simulaciones han sido repetidas 10 veces para cada una de las configuraciones en cada uno de los experimentos realizados. Los resultados obtenidos en cada configuración han sido promediados para obtener el resultado representado en las gráficas.

### 5.5.1 Reducción de mensajes de diseminación

El objetivo de este experimento ha sido evaluar la reducción en el número de mensajes de diseminación enviados como consecuencia de la existencia en la red de parámetros de entrada/salida que están relacionados según la ontología de conceptos utilizada. De acuerdo al mecanismo de diseminación propuesto en este capítulo, el uso de relaciones entre conceptos, ya sea mediante igualdad o subsunción, produce un agrupamiento de la información diseminada. Este agrupamiento debe tener un impacto directo en la disminución del número de mensajes de diseminación que intercambian los nodos en la red MANET.

En el primer caso evaluado, los parámetros de los diferentes servicios no se encuentran relacionados entre sí usando ninguna ontología, por lo tanto, el protocolo de diseminación de la información sobre los parámetros no va a

producir ningún agrupamiento. Posteriormente, estos resultados han sido comparados con otro experimento en el que se ha distribuido el mismo número de servicios pero, en este caso, han sido construidos mediante la selección de parámetros relacionados de acuerdo a una ontología generada aleatoriamente. Se pretende así comparar, por un lado, una configuración de un protocolo de búsqueda de servicios tradicional, que no tiene en cuenta las relaciones entre parámetros durante la diseminación, con las ventajas que la introducción del uso de las relaciones semánticas entre parámetros puede proporcionar.

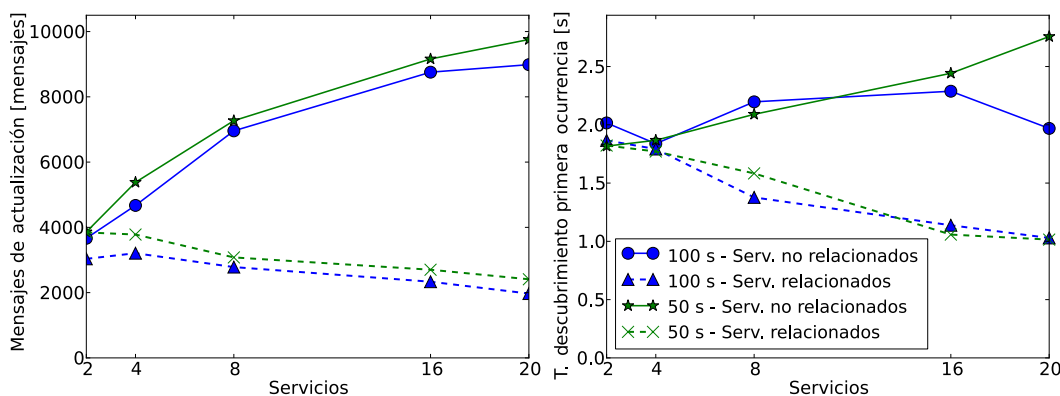
Por supuesto, la comparación podría haberse llevado a cabo con un protocolo concreto de los estudiados en la sección 3.1, sin embargo, el mayor problema a la hora de comparar soluciones para redes ad hoc reside en que los parámetros usados en la ejecución de los experimentos no suelen estar correctamente especificados, haciendo muy difícil, si no imposible, la comparación con los resultados publicados. Hay que tener en cuenta, como se explica en la sección 2.2, que una variación en los mismos produce cambios muy grandes en los resultados, por lo que una comparativa correcta únicamente puede llevarse a cabo si las soluciones son probadas exactamente con la misma configuración.

Por otro lado, es complicado obtener una especificación completa de las particularidades de un protocolo de descubrimiento, por lo que, si no se tiene acceso al código, resulta difícil su implementación correcta, tal y como fue diseñado por sus autores. Así, aunque la evaluación llevada a cabo en este capítulo no es cuantitativa, sí que resulta cualitativamente correcta al permitir comparar dos enfoques distintos para la resolución de un mismo problema.

Para realizar el experimento se ha llevado a cabo, primeramente, una simulación en la que un número variable de servicios (de 2 a 20) ha sido distribuido de manera aleatoria sobre los diferentes nodos de la red ad hoc, teniendo, cada uno de los servicios provistos, un total de 6 parámetros de entrada/salida.

Durante la simulación se han realizado búsquedas de servicios con parámetros seleccionados aleatoriamente entre todos los disponibles. Las búsquedas iniciadas se han mantenido durante un tiempo de 10 segundos hasta su cancelación, mediante el envío de un mensaje de finalización de la búsqueda. Las búsquedas se han realizado en bloques de 5 nodos seleccionados aleatoriamente, que han iniciado estas búsquedas de forma simultánea. Los bloques de búsquedas se han producido cada 5 segundos durante un total de 100 segundos de simulación. Las medidas para cada búsqueda se han promediado para obtener los resultados finales presentados en las gráficas.

Los resultados para los experimentos anteriores se muestran en la figura 5.18 y han sido repetidos para dos tiempos de pausa en la movilidad de los nodos (50 y 100 segundos). En dicha gráfica se observa que el número de mensajes enviados es bastante menor cuando se usan las relaciones proporcionadas por la ontología de parámetros y se procede a la agrupación de los mensajes.



**Figura 5.18:** Efecto en el número de mensajes de actualización y en el tiempo medio para el descubrimiento de la primera ocurrencia de un servicio compatible aplicando una ontología de conceptos

La figura demuestra, por lo tanto, que la categorización de los tipos de los parámetros no solamente tiene como ventaja el permitir una mayor expresividad durante la búsqueda de servicios, ya que posibilita una mayor riqueza en las búsquedas, sino también porque reduce, mediante el agrupamiento de parámetros por tipo, el número de mensajes de diseminación necesarios. Cuando no existe relación entre los parámetros el número de mensajes necesarios aumenta de forma directa con el incremento de los servicios, así como con el número de parámetros que deben ser diseminados a través de la red ad hoc.

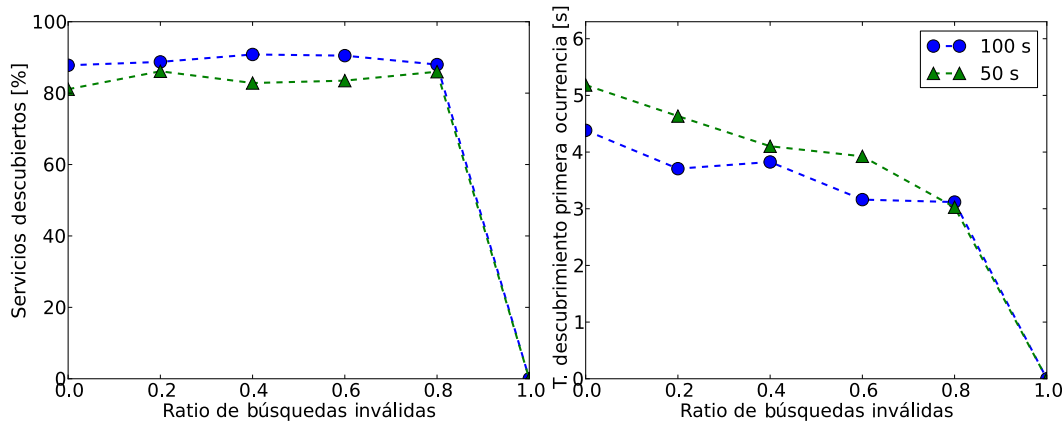
Por otro lado, en la misma figura se muestra el tiempo medio necesario para encontrar una ocurrencia de un servicio compatible con una búsqueda según se incrementa el número de servicios en la red. Como es lógico, este tiempo disminuye en el caso de que el número de servicios compatibles con una determinada búsqueda aumente, debido al incremento en la cantidad de parámetros relacionados jerárquicamente en cada configuración del experimento.

### 5.5.2 Poda de mensajes de búsqueda

El objetivo de este experimento es mostrar qué efecto tiene el proceso de *poda* aplicado a los mensajes de búsqueda de acuerdo a las reglas explicadas en la sección 5.3.2. El experimento ha sido llevado a cabo mediante la selección aleatoria de 30 nodos en la red ad hoc que han sido configurados con servicios que disponen de 6 parámetros de entrada/salida cada uno.

La *poda* de los mensajes ha sido simulada mediante la realización de búsquedas que tienen como objetivo la localización de servicios cuyos parámetros de entrada/salida no existen en la red. El porcentaje de estas búsquedas inválidas ha sido incrementado para comprobar cómo, la aplicación de las reglas

para la poda de mensajes, afecta a varios aspectos del protocolo. Nuevamente, se ha realizado el experimento para dos tiempos de pausa diferentes de 50 y 100 segundos.



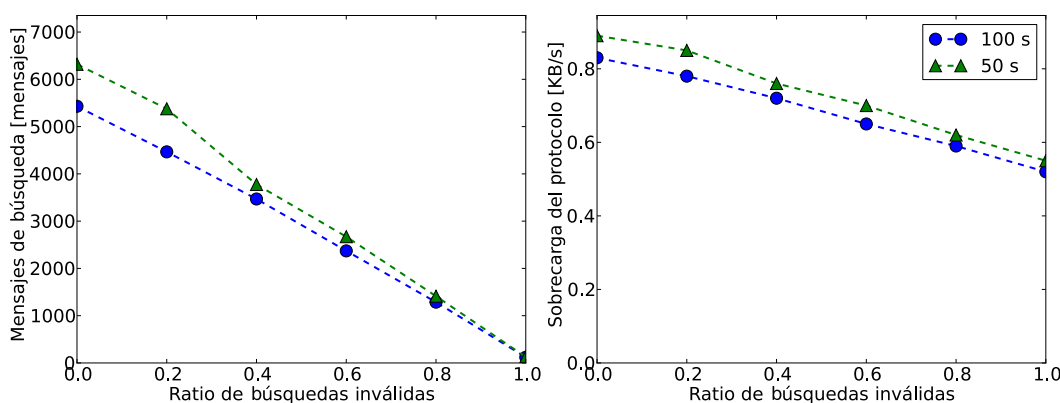
**Figura 5.19:** Efecto de la poda de mensajes de búsqueda en el porcentaje medio de servicios descubiertos y en el tiempo medio de descubrimiento de servicios

La figura 5.19 muestra el porcentaje medio de servicios descubiertos durante la realización de cada experimento. En la figura, las búsquedas inválidas se expresan como un *ratio* entre 0,0 y 1,0. Como puede observarse, el porcentaje de descubrimiento de servicios no es perfecto debido a la conectividad de la red ad hoc. De acuerdo a las características del escenario usado durante la simulación, existe un 5% de nodos que pueden encontrarse desconectados en cada momento, por lo tanto, existe la posibilidad de que tanto el nodo que inicia la búsqueda como el nodo que proporciona el servicio compatible, no dispongan de rutas para comunicarse entre ellos. Además, es también posible que el camino entre dos nodos sea mayor que el TTL utilizado en el experimento. Esto significa que algunos mensajes de búsqueda no serán capaces de llegar a todos los nodos con servicios compatibles, reduciendo el número de servicios encontrados. Por otro lado, cuando el *ratio* de servicios inválidos buscados llega a un total de 1,0, el porcentaje de servicios descubiertos decae hasta cero porque no existe ningún servicio compatible con las búsquedas realizadas.

En la figura 5.19 se muestra también el tiempo medio para el descubrimiento de servicios usando el protocolo propuesto, teniendo en cuenta tanto el tiempo para localizar el servicio como para recibir el mensaje de respuesta desde el nodo que provee el servicio descubierto. Como puede observarse en la figura, dicho tiempo se reduce debido a la disminución en el número de mensajes de búsqueda, como consecuencia de la aplicación del proceso de *poda*. A pesar de que el número de servicios existentes en la red es el mismo en cada caso, los mensajes de búsqueda son descartados con mayor frecuencia al incrementar el *ratio* de búsquedas inválidas. Por lo tanto, existe una menor congestión en el



medio inalámbrico de la red ad hoc, que se traduce en menores probabilidades para la colisión entre mensajes. Por su parte, la disminución en el número de colisiones de mensajes tiene como consecuencia un menor número de mensajes perdidos y, por lo tanto, no es necesario llevar a cabo nuevos reenvíos de los mensajes mediante el mecanismo de *difusión confiable* explicado en la sección 4.5.2.



**Figura 5.20:** Efecto de la poda de mensajes de búsqueda en el número de mensajes enviados y en la sobrecarga introducida por el protocolo de descubrimiento

Por otro lado, la gráfica representada en la figura 5.20 demuestra que la reducción en el tiempo de búsqueda se debe a la disminución en el número de mensajes explicado anteriormente. En dicha figura se recoge el número de mensajes enviados para el mismo experimento y, como puede verse, este número decrece gradualmente cuando se buscan parámetros inexistentes, debido a la aplicación del procedimiento de *poda*. Como se ha explicado en la sección 5.3.1, la utilización del proceso de *poda* de mensajes de búsqueda descarta determinados mensajes porque los nodos intermedios tienen constancia de que dichos mensajes no podrán alcanzar nunca servicios compatibles.

La figura 5.20 muestra también la sobrecarga introducida por los mensajes de diseminación y búsqueda durante el proceso. La sobrecarga, medida en  $KB/s$  transmitidos de media por cada nodo, se obtiene calculando el tamaño total de todos los mensajes relacionados con los procesos de diseminación y búsqueda enviados durante toda la simulación. La introducción del proceso de *poda* reduce la sobrecarga debida al tráfico de mensajes mediante la eliminación de las búsquedas innecesarias.

Por último, como puede observarse también en la figura, el uso del protocolo de descubrimiento propuesto tiene un coste en la información transmitida por cada nodo. La utilización de un mecanismo de difusión de la información *proactivo* supone que los nodos propagan, como respuesta a los cambios producidos en la red ad hoc, cierta información aunque no se esté llevando a cabo

una búsqueda. Esto puede observarse cuando el *ratio* de búsquedas inválidas es igual a  $1,0$ , que es casi equivalente a la situación en la que no se transmiten búsquedas en la red ad hoc, y que muestra que alrededor de un  $75\%$  de la sobrecarga es debido únicamente al proceso de diseminación de parámetros. Es decir, aproximadamente  $0,6\text{ KB/s}$  son transmitidos por cada nodo de la red en respuesta a los cambios producidos en la topología de la red ad hoc.

### 5.5.3 Otros experimentos

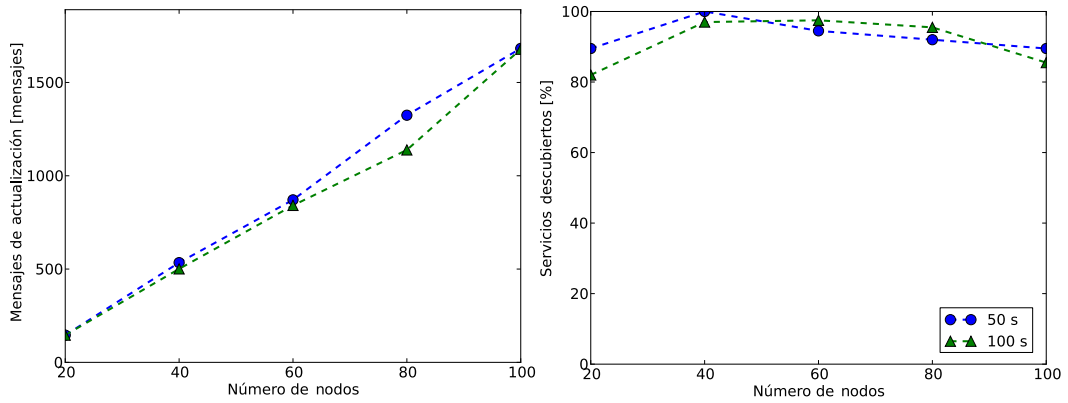
En esta sección se incluyen otros experimentos de ámbito más general realizados sobre el protocolo de búsqueda propuesto en el capítulo. En este caso se ha utilizado una red de 50 nodos como configuración básica de los escenarios para la realización de cada uno de los experimentos. El resto de parámetros se ha mantenido como se encuentran especificados en la tabla 5.1.

Primeramente, se ha evaluado cómo afecta la variación del número de nodos de la red ad hoc en el área de simulación definida en la tabla 5.1. Como puede observarse en la figura 5.21, el número de mensajes enviados aumenta de forma directa con el número de nodos. Cada nuevo nodo introducido en la red participa en el proceso de diseminación incrementando el valor absoluto de mensajes de actualización enviados.

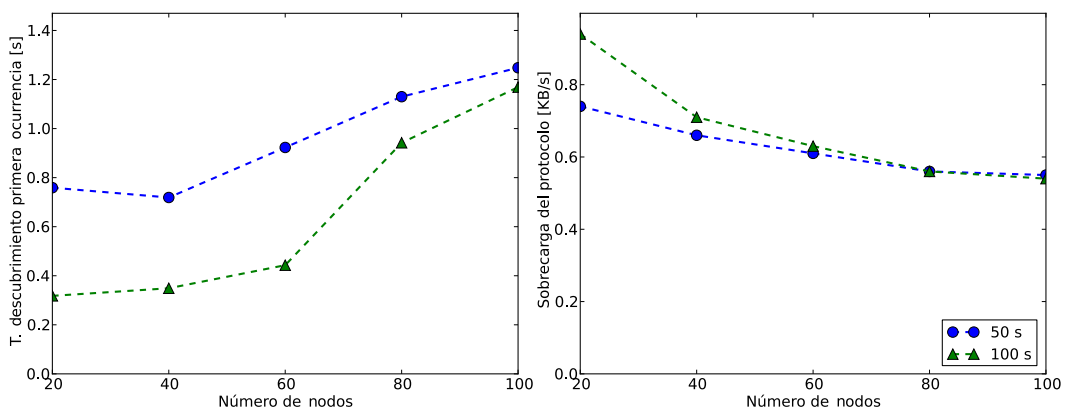
Por otro lado, como el área de simulación permanece constante en cada una de las configuraciones del experimento, al aumentar el número de dispositivos se produce un incremento de la densidad de nodos en cada simulación. Los nodos tienden a estar más juntos y es más probable que existan caminos que permitan encontrar los diferentes servicios distribuidos por la red ad hoc. Por este motivo, puede observarse, en la misma figura, que el porcentaje de servicios encontrados se incrementa inicialmente al aumentar el número de nodos. Posteriormente, este valor decrece ligeramente con el aumento en la densidad, debido a que se incrementa la posibilidad de desaparición de caminos al aumentar la movilidad relativa entre los nodos y las colisiones entre mensajes.

En la figura 5.22 se muestra que el tiempo de descubrimiento se incrementa al aumentar la densidad de los nodos. Esto se debe al aumento en la probabilidad de colisión entre mensajes y, por lo tanto, a la necesidad de realizar un mayor número de reenvíos de los mensajes no recibidos correctamente.

Además, en la parte derecha de la misma figura se observa también que la sobrecarga introducida por el protocolo de descubrimiento disminuye, para los dos tiempos de pausa del experimento, cuando se incrementa el número de nodos. Esta disminución es debida a que los nodos se encuentran en cada configuración más cercanos unos a otros, existiendo menos caminos distintos entre ellos y produciéndose, por lo tanto, un mayor agrupamiento de mensajes durante la diseminación de parámetros.

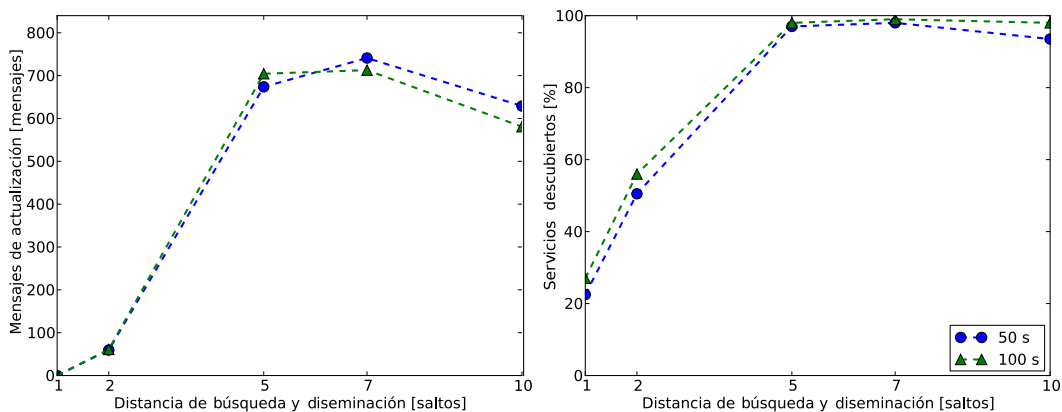


**Figura 5.21:** Efecto de la variación del número de nodos en la cantidad de mensajes de actualización y en el porcentaje de servicios encontrados



**Figura 5.22:** Efecto de la variación del número de nodos en el tiempo de descubrimiento y en la sobrecarga introducida por el protocolo de descubrimiento

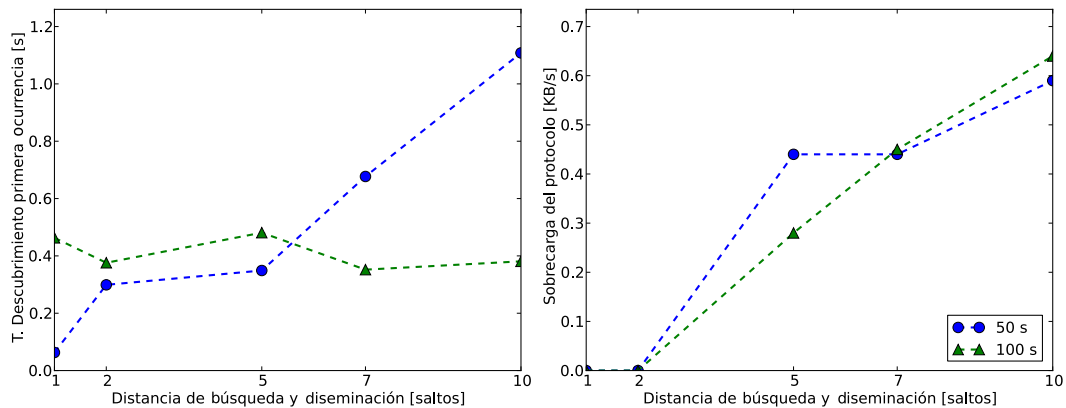
A continuación se ha experimentado con la variación en el número de saltos durante la diseminación y la realización de búsquedas. En la figura 5.23 puede observarse que el número de mensajes de actualización enviados aumenta rápidamente con la distancia de diseminación. Debido a las características de la red, cuando la distancia de diseminación pasa de 5 a 7 saltos no se produce un crecimiento de los mensajes e, incluso, se observa una pequeña reducción. Esto es debido a que a partir de dicha distancia de diseminación se cubre completamente toda la red del experimento y los mensajes son agrupados con mayor frecuencia, reduciendo así la cantidad de mensajes enviados. Además, y como era de esperar, el porcentaje de servicios localizados se incrementa de forma directa con la distancia de diseminación y búsqueda hasta encontrar casi la totalidad de los servicios disponibles.



**Figura 5.23:** Efecto de la variación de la distancia de diseminación y búsqueda en el número de mensajes enviados y en el porcentaje de servicios encontrados

Por otro lado, en la figura 5.24 se recoge el efecto que el experimento anterior tiene en el tiempo de descubrimiento de los servicios. Como puede apreciarse en dicha figura, el tiempo para el descubrimiento aumenta de forma más rápida en el caso de que la movilidad de los nodos sea mayor (50 s), debido a que al aumentar la distancia de diseminación es necesario propagar un mayor número de mensajes de actualización y búsqueda a través de la red.

La propagación de un mayor número de mensajes tiene, como se muestra en la figura 5.24, un impacto directo en la sobrecarga introducida por el protocolo. Como la mayor parte de la sobrecarga es debida a la diseminación de la información sobre parámetros, el aumento del radio de diseminación produce un mayor envío de mensajes y, por lo tanto, una mayor cantidad de información a propagar entre los nodos. La selección de un valor adecuado para la distancia de diseminación es un aspecto que depende del tamaño esperado de la red ad hoc. Así, valores pequeños, relativos al tamaño medio de la red en saltos, producirán que los mensajes solo cubran una parte pequeña de la misma, redu-

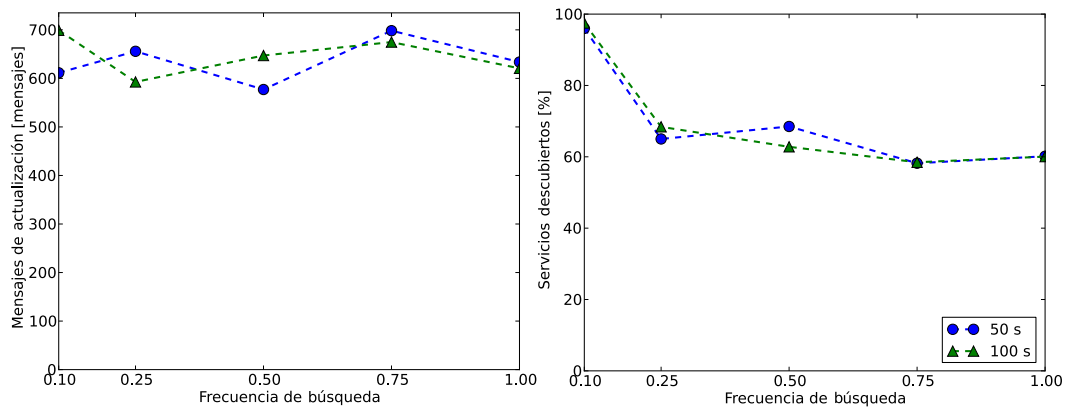


**Figura 5.24:** Efecto de la variación de la distancia de diseminación y búsqueda en el tiempo de descubrimiento y en la sobrecarga introducida por el protocolo de descubrimiento

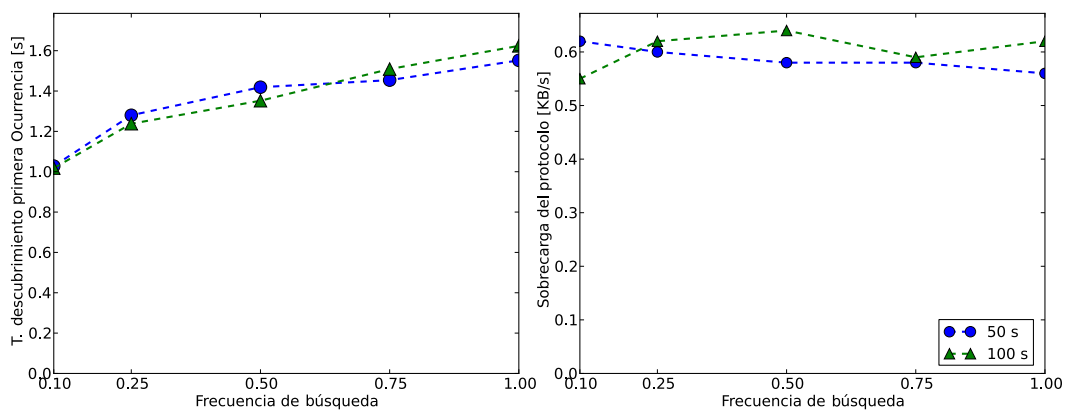
ciéndose el número de servicios encontrados. Por el contrario, un número grande en el radio de diseminación y búsqueda puede hacer que no se apliquen los procesos de *poda* y se envíen, como resultado, un número de mensajes mayor que los realmente necesarios para maximizar el descubrimiento de servicios.

Se ha realizado un último experimento que ha consistido en la modificación de la frecuencia de las búsquedas realizadas. En la figura 5.25 puede observarse que, como era de esperar, la variación de la frecuencia de búsqueda no afecta al número de mensajes de actualización, ya que únicamente modifica el número de mensajes de búsqueda enviados por los nodos. Sin embargo, como puede observarse en la segunda parte de la figura, el aumento de la frecuencia de búsqueda reduce el número de parámetros encontrados, debido a que el incremento en el número de mensajes, que deben ser enviados y procesados por los nodos, resulta en una mayor congestión en la red.

Por último, en la figura 5.26 puede observarse que el tiempo de descubrimiento de los servicios crece poco a poco debido al incremento de los mensajes de búsqueda. Esto es debido a que realmente no existe una modificación en la distribución de la información diseminada en cada caso. La variación observada en la figura se debe a la mayor cantidad de mensajes enviados en la red debido al incremento de las búsquedas. En último lugar, tampoco se observa en dicha gráfica una modificación apreciable en la sobrecarga del protocolo ya que la información transmitida por los nodos se debe en su mayor parte a los mensajes de diseminación de parámetros.



**Figura 5.25:** Efecto de la variación de la frecuencia de búsqueda en el número de mensajes enviados y en el porcentaje de servicios encontrados



**Figura 5.26:** Efecto de la variación de la frecuencia de búsqueda en el tiempo de descubrimiento y en la sobrecarga introducida por el protocolo

## 5.6 Conclusiones

En este capítulo se ha propuesto un protocolo para el descubrimiento de servicios en redes móviles ad hoc en el cual la búsqueda es llevada a cabo mediante el uso de los tipos de los parámetros de entrada/salida de los servicios. El protocolo se basa en la diseminación de la información sobre parámetros a través de la red MANET, utilizando para ello una ontología de conceptos que categoriza los tipos de los diferentes parámetros. El uso de la ontología tiene dos objetivos: primero, mejorar el proceso de descubrimiento introduciendo la posibilidad de llevar a cabo búsqueda exactas y genéricas de parámetros; segundo, reducir el número de mensajes enviados durante los procesos de diseminación y posterior búsqueda. La reducción en el número de mensajes se consigue gracias a la agrupación de los parámetros, utilizando para ello las relaciones que proporciona la ontología de conceptos. Los tipos de los parámetros son agrupados, durante la diseminación, de acuerdo a las relaciones de generalización y especificación definidas en la ontología.

Por otro lado, la utilización de la información diseminada permite aplicar un proceso de *poda* a los mensajes de búsqueda propagados en la red ad hoc. Este proceso, que utiliza la información sobre el tipo de los parámetros diseminada anteriormente, consiste en determinar si los mensajes de búsqueda deben seguir siendo propagados a través de la red en función del contenido en las tablas mantenidas por los diferentes nodos.

En esta propuesta de protocolo de descubrimiento, las búsquedas de servicios son mantenidas en la red por los nodos que las han recibido hasta que son finalizadas explícitamente por los clientes que las habían iniciado. La cancelación de las búsquedas se lleva a cabo mediante el envío de un mensaje del tipo correspondiente. Además, las búsquedas también son canceladas como respuesta a las desconexiones entre los nodos que puedan surgir debido a la movilidad de la red ad hoc. Se permiten dos tipos de búsquedas: *exactas*, cuyo cometido es buscar servicios con parámetros cuyo tipo sea idéntico al especificado en el mensaje de búsqueda; y búsquedas *genéricas*, que posibilitan la localización de parámetros cuyo tipo sea, además de exacto, una especialización del indicado, de acuerdo a la ontología utilizada.

El protocolo de descubrimiento ha sido implementado completamente en Java y probado utilizando el simulador de red ns-2. Para su evaluación, se ha construido un escenario de prueba, con unas configuraciones que han permitido probar las características del protocolo de una forma correcta. Los resultados muestran que la solución propuesta permite la reducción de los mensajes de diseminación y búsqueda gracias al uso de la información contenida en la ontología de conceptos y utilizada durante los procesos anteriores.

Como trabajo futuro para la mejora de la solución se contempla la introducción de un mecanismo que permita la difusión inicial de la ontología de

conceptos utilizada por los nodos. En la solución actual la ontología debe ser conocida *a priori* por los diferentes nodos que participan en la red ad hoc, lo que puede complicar el proceso de configuración de los nodos. Por otro lado, también puede ser interesante la inclusión de mecanismos para la selección de servicios en los nodos cliente que puedan ayudar a las aplicaciones y usuarios a determinar qué servicio, de entre los recibidos, es más adecuado a sus necesidades. Por último, se plantea como investigación futura la posibilidad de generalizar la solución realizada para que sea utilizada no solamente para la diseminación y descubrimiento de servicios basado en los tipos de los parámetros de entrada/salida, sino también para cualquier recurso cuya información pueda ser categorizada mediante una ontología de conceptos.



## Composición basada en un grafo distribuido de servicios

El objetivo de la composición de servicios es permitir que los usuarios puedan realizar tareas mediante el uso combinado de los servicios que están disponibles en su contexto actual. En este capítulo se propone una solución para la composición automática de servicios en redes móviles ad hoc que se basa en la utilización de un grafo distribuido de servicios (GDS). Este grafo es creado y mantenido por los diferentes nodos de la red por medio del descubrimiento de relaciones de compatibilidad entre las entradas y salidas de los servicios cercanos. El uso del grafo de servicios permite, ante un problema de composición, reducir el tiempo necesario para encontrar las soluciones disponibles.

La composición de servicios presentada en este capítulo utiliza el protocolo de descubrimiento de servicios basado en parámetros propuesto en el capítulo 5. La diseminación de la información de parámetros y las funcionalidades de búsqueda de servicios permiten determinar qué servicios disponibles en la red pueden conectarse entre sí y dar lugar a la creación del grafo de servicios. Las aportaciones de este capítulo son:

- Definición del proceso de detección de colisiones entre parámetros de entrada y salida que permite llevar a cabo la creación del GDS en la red MANET.
- Mecanismo de creación y mantenimiento del GDS frente a los cambios que se producen durante la movilidad de los nodos en la red ad hoc.

- Definición del proceso de búsqueda de composiciones adecuado a las características del GDS y a la naturaleza de las redes MANET. Se evita así la necesidad de construir un *flujo de trabajo* que describa de forma abstracta el servicio a buscar, mejorando así, la automatización del proceso.
- Evaluación de la solución propuesta mediante la utilización de un simulador de red para conocer sus ventajas y limitaciones.

La sección 6.1 ofrece una visión general de la solución propuesta para la composición de servicios. En la sección 6.2 se presentan las características del GDS y de los procesos de detección de colisiones y mantenimiento que hacen posible su construcción y gestión frente a la movilidad de la red ad hoc. Por otro lado, la sección 6.4 explica en qué consiste el proceso de búsqueda en el grafo, mientras que en la sección 6.5 se presentan los resultados de la evaluación realizada. Por último, la sección 6.6 contiene las conclusiones finales del capítulo.

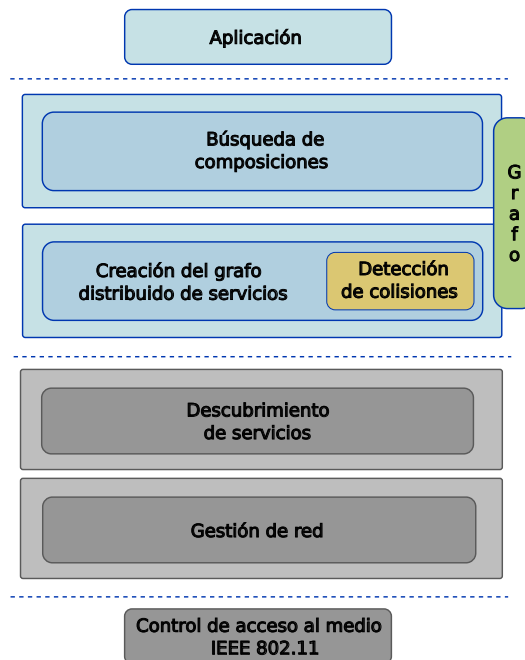
## 6.1 Visión general

La solución propuesta para la composición de servicios en redes móviles ad hoc se encuentra dividida en tres funcionalidades: la detección de colisiones entre los parámetros de entrada/salida de los servicios existentes en la red que, hace uso de la búsqueda de servicios explicada en el capítulo anterior; la construcción y mantenimiento del grafo distribuido de servicios y, por último, la búsqueda de soluciones en el grafo construido.

La diseminación y búsqueda de los servicios compatibles aplica el protocolo presentado en el capítulo 5 y es utilizado por el proceso de detección de colisiones entre parámetros. Por su parte, la detección de colisiones consiste en un proceso por el cual cada nodo de la red utiliza la información contenida en su tabla de parámetros para determinar si existen posibles conexiones entre parámetros de entrada/salida.

El proceso de creación del grafo es el mecanismo que construye y mantiene las estructuras de información en la red que posteriormente permiten llevar a cabo la búsqueda de composiciones de acuerdo a los requerimientos especificados. Por último, la búsqueda de composiciones es un proceso de recorrido del GDS que comienza en el nodo que inicia la búsqueda y, mediante una búsqueda hacia delante, explora los caminos existentes para encontrar una o más soluciones al problema de composición planteado en la red. La figura 6.1 muestra estos elementos dentro de la arquitectura global presentada en el capítulo 4.

La propuesta para la composición de servicios en redes MANET realizada en este capítulo puede ser clasificada de acuerdo a la categorización presentada en la sección 3.2.1. En cada apartado se incluye una referencia a la sección de la revisión bibliográfica en la que se comenta dicho aspecto.



**Figura 6.1:** Arquitectura interna para la composición de servicios basada en un grafo distribuido de servicios

- **Arquitectura:** la solución propuesta es de naturaleza *distribuida* y todos los nodos participan de la misma manera en el proceso de detección de colisiones y en la construcción del GDS. Se pretende así evitar la necesidad de utilizar un gestor centralizado del proceso de composición (sección 3.2.2).
- **Mecanismo de composición:** la propuesta se basa en la construcción del grafo distribuido de servicios. La estructura construida es utilizada durante el proceso de composición con el objeto de reducir el tiempo de búsqueda necesario para encontrar una solución (sección 3.2.3).
- **Especificación de la composición:** en la propuesta realizada en este capítulo, la solicitud de composición se especifica mediante los tipos de los parámetros de entrada y salida del servicio compuesto. El objetivo es evitar la definición de un *flujo de trabajo* completo que deba ser instanciado durante el proceso de composición (sección 3.2.4).
- **Descripción de servicios:** al igual que en el caso del protocolo de descubrimiento de servicios presentado en el capítulo 5, los servicios se describen mediante el tipo de sus parámetros de entrada y salida. Estos parámetros se encuentran categorizados en base a una ontología que permite

utilizar sus relaciones de generalización y especificación entre los conceptos para establecer relaciones de compatibilidad entre los servicios.

- **Participación del usuario:** debido a que el proceso de composición es automático, una vez que se han especificado las entradas y salidas del servicio a obtener, la participación del usuario puede reducirse a la selección entre diferentes soluciones que puedan encontrarse a un problema de composición (sección 3.2.4 y sección 3.2.6).
- **Modo de operación:** la construcción del GDS es un proceso *proactivo* que responde a la movilidad de la red de nodos y que se realiza de manera independiente a las búsquedas de composiciones que puedan ser realizadas (sección 3.2.7).
- **Respuesta a la movilidad:** el mantenimiento de la estructura que representa el GDS requiere responder a los cambios que se producen en la topología de la red. Por lo tanto, los eventos de aparición y desaparición de nodos de la red deben ser capturados para actuar en consecuencia y actualizar el grafo de servicios (sección 3.2.9).
- **Ejecución:** los resultados de la composición pueden ejecutarse mediante la propagación de un mensaje a través de los nodos que proveen los servicios seleccionados. Se evita así la utilización de un enfoque centralizado para la ejecución que requiera el envío de mensajes desde y hacia el nodo coordinador (sección 3.2.8).

## 6.2 Grafo distribuido de servicios

El grafo distribuido de servicios (GDS) se define como un grafo formado por los servicios, los parámetros de dichos servicios y las conexiones entre ellos que se encuentran distribuidas a través de la topología de la red. Los servicios pueden estar proporcionados, y por lo tanto localizados, en diferentes nodos de la red MANET. Cada servicio proporcionado en la red se encuentra descrito por su identificador único y el del nodo que lo provee, de tal forma que sea posible distinguir entre diferentes replicas de un mismo servicio localizadas en distintos nodos de la red.

El GDS se define formalmente de la siguiente manera

$$GDS = \{S, C\}$$

donde  $S$  es el conjunto de vértices del grafo formado por los servicios disponibles en la red, mientras que  $C$  es el conjunto de conexiones entre los parámetros del grafo.

Cada nodo de la red contiene únicamente una parte del grafo de servicios, donde cada uno de esos grafos parciales representa la relación entre los servicios locales contenidos en el propio nodo y los servicios remotos, provistos por otros nodos de la red. Con objeto de permitir la navegación a través de la red, el grafo mantiene referencias a las direcciones (identificador del servicio y dirección del nodo que lo provee) de los servicios conectados. La existencia de una referencia entre dos nodos remotos supone la existencia de una ruta de comunicación entre ellos. Estas rutas de comunicación son creadas durante las fases de creación y mantenimiento del grafo.

En esta propuesta, el GDS es un grafo dirigido y las aristas permiten únicamente la navegación de un servicio hacia sus servicios *sucesores*. Dados dos servicios  $S_i$  y  $S_j$ , se dice que  $S_j$  es *sucesor* de  $S_i$  si algún parámetro de salida de  $S_i$  es compatible con algún parámetro de entrada de  $S_j$  y, por lo tanto puede establecerse una conexión dirigida desde el primer servicio hasta el segundo. En este caso se dice también que  $S_i$  es *antecesor* de  $S_j$ , sin embargo, en la solución presentada en este capítulo cada servicio mantiene solamente información sobre sus sucesores y, por lo tanto, solo es posible la exploración hacia delante en el GDS. En la sección 7.2.2 de las líneas futuras de investigación se expone la posibilidad de extender la propuesta actual utilizando otro tipo de búsquedas en el grafo con objeto de mejorar el proceso de composición.

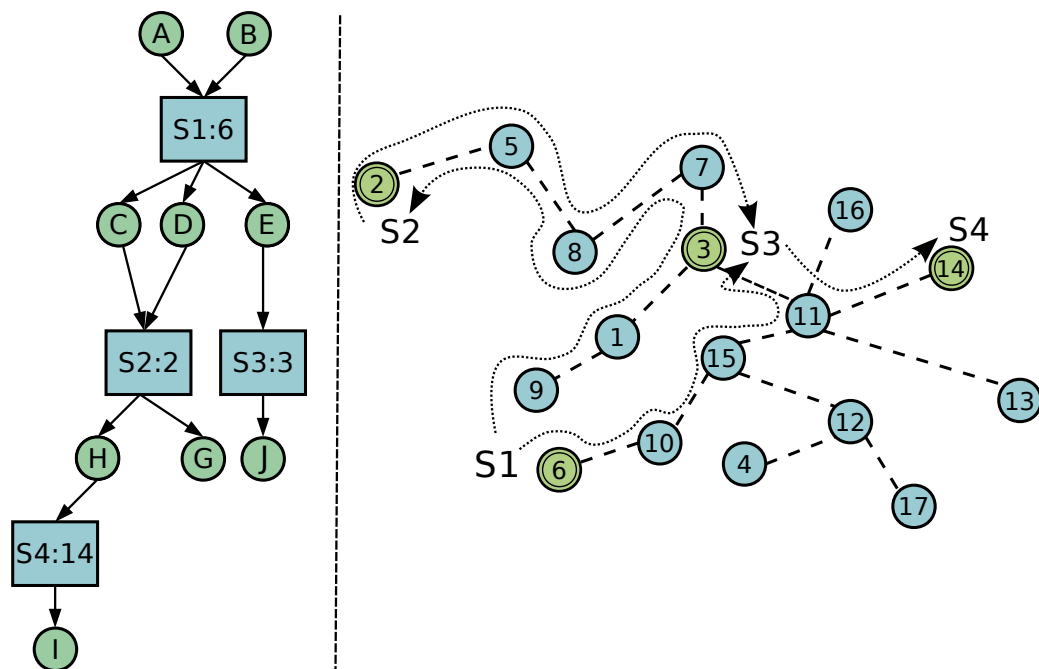


Figura 6.2: Grafo distribuido de servicios conectando servicios a través de la red

La figura 6.2 muestra un ejemplo de un grafo distribuido de servicios. La parte izquierda de la figura representa los parámetros de entrada/salida y los servicios a los que dichos parámetros pertenecen. La dirección de las flechas en el grafo indica si los parámetros son de entrada o de salida. Además, en el lado derecho de la figura se muestra la distribución del grafo a través de la red ad hoc y como éste conecta los diferentes servicios a través de ella. Así, puede verse que en el nodo 6 existe un servicio cuyos parámetros de salida están conectados a través de ciertos caminos en la red con otros servicios sucesores y situados en los nodos 2 y 3. Es decir, sobre la red de nodos existe un grafo que determina la dependencia entre los parámetros de entrada y salida pertenecientes a los servicios provistos en los nodos.

### 6.3 Colisiones de parámetros

El proceso de creación del GDS comienza con cada nodo de la red diseminando información sobre los parámetros de los servicios que provee, utilizando para ello el mecanismo de diseminación presentado en el capítulo 5. La diseminación es realizada de forma continua durante el tiempo de vida de la red.

Los nodos que se encuentran situados en los caminos intermedios que conectan nodos que proveen servicios reciben información sobre los tipos de los parámetros de entrada y salida de dichos servicios cercanos. Cada vez que un nodo conoce información acerca de los parámetros diseminados lleva a cabo un proceso para comprobar si existen colisiones entre los nuevos parámetros añadidos a su tabla de parámetros y aquellos que ya se encontraban registrados anteriormente en ella.

Una *colisión* ocurre cuando, entre un parámetro de entrada y otro parámetro de salida, existe una compatibilidad de acuerdo a la ontología de conceptos utilizada. Como se explica en la sección 4.4.2, la relación entre los tipos de dos parámetros puede ser de *igualdad* o de *subsunción*. En el caso del protocolo de descubrimiento de servicios estas relaciones entre los tipos de parámetros son utilizadas para agrupar la información y reducir el número de mensajes en los procesos de diseminación y búsqueda de servicios, así como para mejorar el proceso de descubrimiento de servicios.

Para la construcción del GDS es necesario determinar la compatibilidad entre las entradas y salidas de los diferentes servicios distribuidos en la red ad hoc. Para ello, se aplican las relaciones entre parámetros explicadas a continuación, donde  $P_S$  representa el parámetro de salida de un servicio  $S_i$  y  $P_E$  el parámetro de entrada de otro servicio  $S_j$ .

- **Igualdad:** si  $P_E$  y  $P_S$  tienen el mismo tipo significa que la salida de un servicio puede utilizarse como entrada para el otro. Se puede así establecer

una conexión del servicio  $S_i$  al servicio  $S_j$  de tal forma que el segundo es *sucesor* del primero.

- **Subsuncción:** este caso sucede cuando el tipo del parámetro de entrada  $P_E$  es más general, un supertipo, que el tipo del parámetro de salida  $P_S$ . Esto significa que la salida del servicio  $S_i$  puede usarse como entrada para  $S_j$ . El caso inverso, cuando  $P_S$  es más general que  $P_E$  no ha sido tratado en esta tesis doctoral. Su solución requeriría la obtención de información extra para completar el tipo correspondiente a la entrada del sucesor.
- **No relacionados:** en el caso de que no exista ninguna relación directa entre los tipos de los parámetros de entrada y salida, de acuerdo a la ontología utilizada, no se podrá establecer ninguna conexión entre ellos.

Se establece que una colisión está definida como una tupla de dos elementos

$$C = (T(P_E), T(P_S))$$

donde  $T(P_E)$  es el tipo del parámetro de entrada y  $T(P_S)$  el tipo del parámetro de salida.

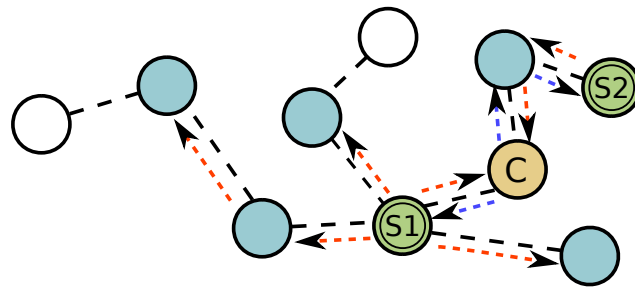
La detección de las colisiones entre parámetros y la gestión del grafo se lleva a cabo mediante la realización de las siguientes acciones: la detección de una o más colisiones a partir de la recepción de la información sobre parámetros de un nodo vecino; la conexión entre los servicios compatibles; la *inhibición* de colisiones idénticas en nodos cercanos; y por último, después de localizar los servicios participantes en la colisión, el envío de un mensaje a los servicios *antecesores* de cada colisión, por parte del nodo que ha detectado la colisión, para que estos lleven a cabo la actualización del GDS.

En la figura 6.3 se muestra como un nodo intermedio detecta una colisión entre los parámetros de entrada y salida de dos servicios provistos por diferentes nodos en la red ad hoc tras realizarse su disseminación, representada en rojo, y que, posteriormente realiza la búsqueda de los mismos, en azul.

### 6.3.1 Detección de colisiones

La detección de colisiones entre parámetros se realiza siempre que un nodo recibe un mensaje de actualización enviado por la capa de disseminación de parámetros. Se puede consultar la sección 5.2.3 junto con la sección 5.2.4 para más información sobre los mensajes de actualización y la tabla de parámetros.

Cuando un nodo recibe un mensaje de actualización se notifica a la capa de *gestión del grafo* y, a partir del contenido de la *tabla de parámetros* del nodo receptor y del mensaje recibido, se lleva a cabo el proceso explicado a continuación, representado en forma de algoritmo en la figura 6.4.



**Figura 6.3:** Detección de una colisión entre parámetros de entrada y salida compatibles por parte de un nodo intermedio

El proceso comienza por la detección de colisiones entre los tipos de parámetros que son añadidos a la *tabla de parámetros* del nodo, incluyendo aquellos cuya *distancia estimada* ha cambiado como resultado del procesamiento del mensaje de actualización recibido, así como el resto de tipos de parámetros que ya se encontraban en la tabla del nodo. Solamente aquellos tipos de parámetros cuyo nuevo valor de *distancia estimada* es mayor que el valor contenido anteriormente en la tabla de parámetros del nodo actual, lo que incluye también a las nuevas entradas, son tenidos en cuenta durante la comprobación de las colisiones producidas entre los diferentes parámetros.

Como se explica en la sección 5.2.1, la *distancia estimada* representa cuánto de cerca se encuentra un nodo, que proporciona un tipo de parámetro determinado, del nodo actual. Dicha distancia es un valor estimado de acuerdo al mecanismo seguido para la propagación de la información de los nodos en la red ad hoc. Se supone que, antes de llevar a cabo la detección de colisiones, se ha procesado el mensaje recibido aplicando para ello el algoritmo presentado en la sección 5.2.4 para la actualización de la tabla de parámetros.

Utilizando el contenido de la tabla de parámetros y del mensaje de actualización recibido se obtienen los siguientes grupos:

- $P_E$ : los *antiguos parámetros de entrada*, contenidos en la tabla antes de procesar el mensaje de actualización.
- $P'_E$ : los *nuevos parámetros de entrada* tras procesar el mensaje de actualización.
- $P_S$ : los *antiguos parámetros de salida* contenidos en la tabla antes de procesar el mensaje de actualización.
- $P'_S$ : los *nuevos parámetros de salida* tras procesar el mensaje de actualización.

A partir de los grupos anteriores se comprueba si existe una colisión entre  $P_E$  y  $P'_E$  por un lado, y entre  $P_S$  y  $P'_E$  por otro. El proceso para detectar una colisión



```

Entrada:  $m$  : mensaje recibido
            $P_i$  : tabla de parámetros local
            $L_c$  : lista de colisiones detectadas por el nodo
            $v_i$  : vecino que ha enviado el mensaje
            $n_i$  : nodo actual

Salida :  $C_{detectadas}$  : colisiones detectadas

// Procesar mensaje actualización
( $P_{cambiados}, P_{nuevos}, P_{eliminados}$ )  $\leftarrow$  ProcesarMensaje( $m, v_i, P_i, n_i$ )

// Procesar eliminaciones de parámetros
foreach Colisión  $c$  in  $L_c$  do
|    $p_e \leftarrow$  ObtenerEntrada( $c$ )
|    $p_s \leftarrow$  ObtenerSalida( $c$ )
|   if  $p_e \in P_{eliminados}$  or  $p_s \in P_{eliminados}$  then
|   |   Eliminar( $c, L_c$ )
|   |   EnviarMensajeCancelación( $p_e, p_s$ )
|   end
end

// Obtener parámetros válidos
 $P_v \leftarrow$  Vacio
foreach Parámetro  $p$  in  $P_{nuevos}$  do
|   if  $p \in P_{cambiados}$  then
|   |    $d_{previa} \leftarrow$  DistanciaPrevia( $p$ )
|   |    $d_{actual} \leftarrow$  DistanciaActual( $p$ )
|   |   if  $d_{actual} > d_{previa}$  then Insertar( $p, P_v$ )
|   end
end

// Detectar nuevas colisiones entre parámetros
 $P_E, P_S, P'_E, P'_S \leftarrow$  ObtenerParámetros( $m, P_i$ )

 $C_{detectadas} \leftarrow$  Vacio
 $C_{detectadas} \leftarrow$  CalcularColisiones( $P_S, P_E$ )
 $C_{detectadas} \leftarrow$  CalcularColisiones( $P'_S, P'_E$ )

// Eliminar colisiones ya detectadas
foreach Colisión  $c$  in  $C_{detectadas}$  do
|   if  $c \in L_c$  then Eliminar( $c, C_{detectadas}$ )
end

```

**Figura 6.4:** Algoritmo para la detección de colisiones tras la recepción de un mensaje de actualización

```

Entrada:  $P_E$  : parámetros de entrada
            $P_S$  : parámetros de salida
            $O$  : ontología de conceptos
Salida :  $C_{detectadas}$  : colisiones detectadas

 $C_{detectadas} \leftarrow Vacio$ 
foreach Parámetro  $e$  in  $P_E$  do
  | foreach Parámetro  $s$  in  $P_S$  do
  | | if relaciónJerarquía( $e, s, O$ ) then
  | | |  $c \leftarrow (e, s)$ 
  | | | Insertar( $c, C_{detectadas}$ )
  | | end
  | end
end

```

**Figura 6.5:** Algoritmo para el cálculo de colisiones entre parámetros de entrada y salida

entre parámetros consiste en comprobar cada parámetro de un grupo con cada uno de los parámetros del otro y determinar si existe alguna de las relaciones de compatibilidad entre parámetros explicadas en la sección anterior.

Como resultado de la comprobación de relaciones se obtiene la lista de las nuevas colisiones detectadas, aplicando para ello el algoritmo contenido en la figura 6.5. Las nuevas colisiones son comprobadas con aquellas que el nodo ya había detectado anteriormente, y que habían sido almacenadas en una lista, con objeto de evitar múltiples detecciones de colisiones entre parámetros idénticas. Así, si una colisión ya fue descubierta por un nodo es descartada, y, por lo tanto, se notifican únicamente a las capas superiores las nuevas colisiones detectadas por el nodo. Estas serán las colisiones utilizadas para llevar a cabo la conexión entre los servicios compatibles durante la construcción del grafo.

En la figura 6.4, antes de llevar a cabo el proceso de detección de nuevas colisiones explicado anteriormente, se comprueba si los parámetros que deben ser eliminados de la tabla de parámetros del nodo actual afectan a aquellas colisiones que habían sido detectadas anteriormente por el nodo. Hay que tener en cuenta que los mensajes de actualización contienen no solo nuevos parámetros a añadir a la tabla, sino también aquellos otros que deben ser eliminados.

Así, por cada colisión detectada en el nodo se comprueba si los parámetros contenidos en la lista de *eliminaciones*, del mensaje de actualización recibido, afectan a alguna colisión detectada anteriormente. En el caso de que el parámetro de entrada, o el parámetro de salida, de una colisión detectada deban

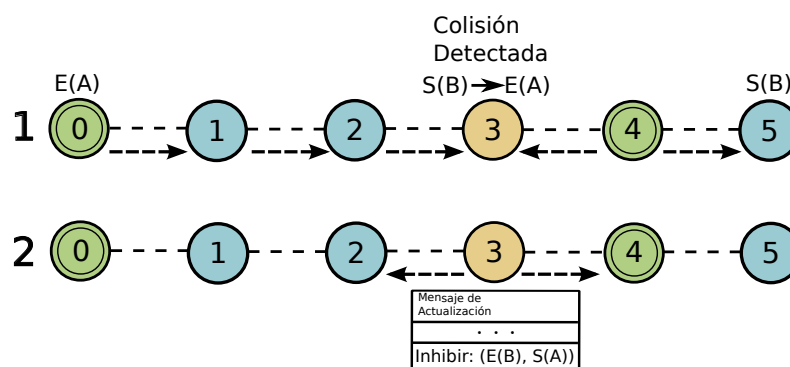
ser eliminados de la tabla del nodo actual se produce también, como resultado, la supresión de la colisión correspondiente de la lista de colisiones detectadas.

### 6.3.2 Inhibición de colisiones

Debido a la naturaleza asíncrona de la red, es posible que varios nodos lleven a cabo la detección de colisiones idénticas. En algunos casos esto será deseable, ya que permite conocer durante la creación del GDS la existencia de rutas alternativas para conectar servicios. Sin embargo, en otros casos puede producir una multiplicación de mensajes innecesarios que no aportan ninguna ruta alternativa y que dan como resultado una mayor congestión en la red. Estas colisiones secundarias deben ser inhibidas con objeto de reducir así el número de mensajes enviados.

Hay que tener en cuenta que, tras producirse la detección de la colisión en un nodo, no se detiene la diseminación de la información de parámetros correspondientes a través de la red. Así, nodos más alejados pueden también recibir los mismos cambios en sus tablas de parámetros y, producir como resultado, la detección de las mismas colisiones en diferentes nodos.

Por ejemplo, en la figura 6.6 se muestra una situación en la se ha producido la diseminación de un parámetro desde el nodo 0. Posteriormente, se produce la diseminación de un parámetro, compatible con el anterior, desde el nodo 4, y que resultará en colisiones en los nodos intermedios. Los mensajes de diseminación se propagan a través de la red hasta que llegan al nodo 3, que es el primer nodo, en el caso del ejemplo, en el que se produce la detección de la colisión. Tras la colisión en este nodo, cuando el mensaje de actualización de las tablas de parámetros se propague por el resto de la red pueden detectarse las mismas colisiones una y otra vez, por ejemplo, en los nodos 2 y 4.



**Figura 6.6:** Inhibición de colisiones duplicadas en los nodos intermedios

Para evitar el problema comentado anteriormente, cuando se detecta una colisión en un nodo se incluye, mediante *piggyback* en el mensaje de actuali-

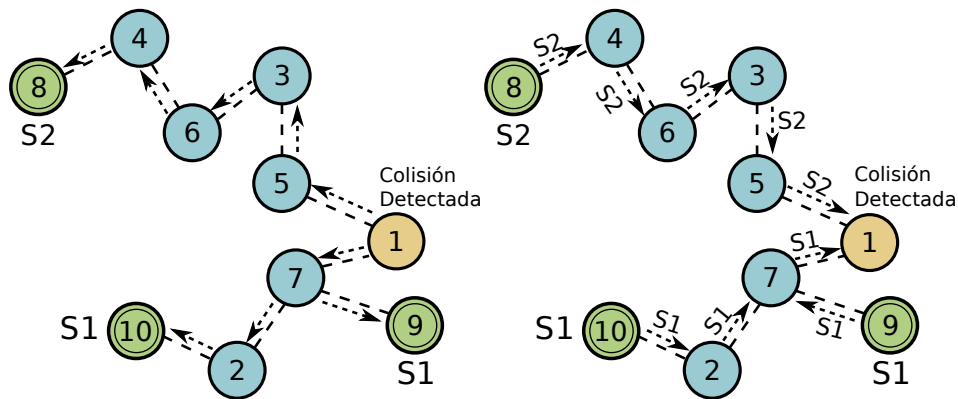
zación enviado por la capa de diseminación de parámetros, las colisiones detectadas en dicho nodo. Esto incluye tanto las colisiones válidas como aquellas invalidadas, tras aplicar el algoritmo de la figura 6.5. Todos los nodos que reciban este mensaje de actualización inhibirán aquellas colisiones idénticas a las contenidas en el mensaje recibido. La aplicación de la inhibición de colisiones puede observarse en la figura 6.6, donde el mensaje incluye la colisión a inhibir por los vecinos. Hay que tener en cuenta que el proceso de inhibición no afecta a la creación de las rutas de comunicación entre los nodos, ya que los mensajes se siguen recibiendo de la forma usual.

Por último, cuando un nuevo nodo aparece, éste recibe la información procedente de las tablas de parámetros de sus vecinos. Si debido a una actualización procedente de un mismo vecino se produce una colisión, ésta no es tenida en cuenta si los parámetros de entrada/salida que han colisionado proceden del mismo vecino. Se evita así que la aparición de nuevos vecinos produzca la duplicación de colisiones detectadas anteriormente por el propio vecino o, incluso, en otros cercanos. Sin embargo, en el caso de que la actualización proceda del propio nodo, debido, por ejemplo, a que se han añadido nuevos servicios, sí que se llevará a cabo el emparejamiento de las nuevas entradas y nuevas salidas para detectar posibles colisiones locales.

### 6.3.3 Conexión de servicios compatibles

Cuando un nodo detecta un conjunto de colisiones entre distintos parámetros de entrada/salida debe llevar a cabo la búsqueda de aquellos nodos que proporcionan dichos parámetros. Para ello, el nodo que ha detectado la colisión envía un mensaje de búsqueda, utilizando la capa de descubrimiento de servicios basada en parámetros explicada en el capítulo 5. Los parámetros contenidos en el mensaje de búsqueda son aquellos que forman parte de las colisiones recientemente detectadas, incluyendo tanto los parámetros de entrada como de salida. Este mensaje de búsqueda contiene además, en la zona de datos, un código especial con el fin de indicar que el mensaje de búsqueda ha sido generado por la capa de gestión del grafo y no por otra aplicación.

Aquellos nodos que proporcionan los parámetros buscados, es decir, que proveen servicios que tienen como entrada o salida, según corresponda, los parámetros requeridos, y que se encuentren a una distancia determinada por el TTL del mensaje, aceptan esta búsqueda. Posteriormente, comprueban sus servicios locales para seleccionar aquellos que sean compatibles con la búsqueda realizada, y a continuación envían un *mensaje de respuesta de colisión* que es incluido como *datos* en el mensaje de respuesta  $M_R$  enviado por la capa de descubrimiento de servicios, tal y como se explica en el capítulo 5. El mensaje de respuesta contiene la lista de los servicios compatibles descubiertos junto con su identificador completo: servicio y nodo.



**Figura 6.7:** Búsqueda de los servicios compatibles como resultado de la detección de una colisión y posterior recepción de las respuestas

Los mensajes de respuesta, que provienen de los nodos con servicios compatibles, son recibidos por el nodo que detectó la colisión e inició la búsqueda. Los nodos gestionan las respuestas correspondientes a cada colisión detectada de manera independiente. Cada vez que se recibe una respuesta compatible con una colisión, ya sea correspondiente a los parámetros de entrada, a los parámetros de salida o a ambos, se procede a notificar a los otros servicios relacionados con ella, produciéndose únicamente la notificación de las nuevas conexiones entre servicios.

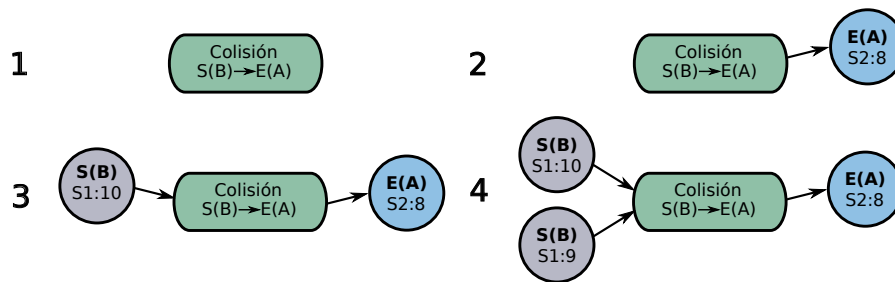
La figura 6.7 muestra cómo, desde un nodo que ha detectado una colisión, el nodo 1, se lleva a cabo la propagación de los mensajes de búsqueda. Estos mensajes de búsqueda son respondidos por los nodos que propagaron los parámetros, los nodos 8, 9 y 10, dando lugar a la creación de las rutas de comunicación correspondientes. Cada vez que un nodo, que ha detectado una colisión, recibe un mensaje de respuesta con los servicios correspondientes a la misma, actualiza su tabla de colisiones. Esta tabla relaciona las colisiones entre parámetros de entrada y salida con las diferentes respuestas recibidas a las búsquedas realizadas. El objetivo de este intercambio de mensajes es crear las conexiones entre servicios compatibles y sucesores en la red MANET.

El mecanismo de notificación de las respuestas recibidas, así como de la creación de las conexiones en el GDS, es el siguiente:

- Cuando se recibe un servicio con un parámetro de entrada, es decir, un servicio *sucesor* correspondiente a una respuesta a una determinada colisión, se notifica a los nodos con parámetros de salida relacionados, esto es, a sus *antecesores*, a través de los datos contenidos en la tabla para la entrada de la colisión correspondiente.

- Cada vez que se añade un servicio con un parámetro de salida, es decir, un nuevo *antecesor* para una colisión, se notifica, al nodo que provee dicho parámetro de salida, la existencia de los servicios con parámetros de entrada, esto es, los *sucesores* compatibles que se hayan recibido anteriormente para la colisión dada.

En la figura 6.8 se muestra un ejemplo del proceso de notificación de servicios que continúa la situación mostrada en la figura 6.7. Cierta nodo de la red ha detectado una colisión entre dos parámetros de entrada/salida:  $E(A)$  que representa una entrada de tipo  $A$  y  $S(B)$  un parámetro de salida de tipo  $B$ . Además, se supone que el tipo  $A$  es más general que el  $B$ , de acuerdo a la ontología utilizada. Tras detectar la colisión, se han enviado los mensajes de búsqueda correspondientes para localizar los servicios que proveen los parámetros involucrados en la misma.



**Figura 6.8:** Gestión de conexiones entre servicios compatibles tras la detección de una colisión entre parámetros

Como resultado de la búsqueda, se recibe un mensaje de respuesta desde el nodo 8 con el servicio compatible  $S2$  que contiene como entrada el parámetro de tipo  $A$ . En esta ocasión, no es necesario notificar a ningún nodo ya que no existe ninguna conexión completa realizada a través de la colisión detectada. A continuación, se recibe otro mensaje de respuesta a la búsqueda pero esta vez desde el nodo 10 que contiene el parámetro de tipo  $B$  como salida en el servicio  $S1$ . Como se acaba de recibir un parámetro de salida y, anteriormente, se habían recibido parámetros de entrada, se produce una notificación al nodo 10 con los servicios con entradas compatibles, es decir, el servicio  $S2$  del nodo 8, para que lleve a cabo la conexión del GDS, creándose así una relación de sucesión desde el servicio  $S1$  en el nodo 10 hasta el  $S2$  localizado en el nodo 8.

Un tiempo después, se recibe otro mensaje de respuesta desde el nodo 9 con el servicio  $S1$  que proporciona otro parámetro de salida compatible con la colisión. Se envía, por lo tanto, un mensaje a este nuevo nodo, indicando el servicio con una entrada compatible ya conectado, de tal forma que en el GDS se crea otra conexión desde el servicio  $S1$  del nodo 9 hasta el servicio  $S2$

en el 8. El proceso explicado se repite mientras se sigan recibiendo respuestas para las búsquedas correspondientes a la colisión de parámetros. Así, en caso de aparecer un nuevo servicio con un parámetro de entrada compatible, se llevaría a cabo una notificación de dicho nuevo servicio a los servicios S1 localizados en los nodos 9 y 10, creándose las rutas del grafo correspondientes.

En la figura 6.9 se representa el proceso de creación del GDS explicado anteriormente y su distribución en la red MANET. El nodo 1, que ha detectado la colisión y ha recibido las respuestas con servicios tras la realización de la búsqueda, lleva a cabo el reenvío de los mensajes recibidos a los servicios antecesores. En este caso, envía el mensaje de respuesta que se ha recibido desde el nodo 8 hacia los nodos 10 y 9. Para ello utiliza las rutas de comunicación que se han establecido durante los procesos de búsqueda y recepción de las respuestas. Como resultado de todos estos procesos, los nodos 9 y 10 actualizan su parte del GDS con una conexión desde sus servicios S1 hacia el servicio remoto S2 localizado en el nodo 8, permitiendo el envío de mensajes hacia dicho sucesor.

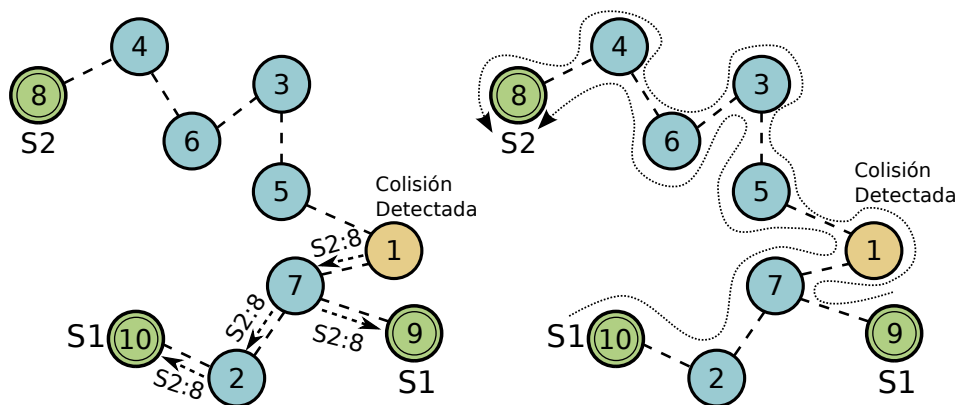


Figura 6.9: Creación del grafo distribuido de servicios (GDS)

### 6.3.4 Encaminamiento de mensajes en el grafo

La existencia de una arista en el GDS, entre un servicio y su sucesor, supone que existe la posibilidad de llevar a cabo una comunicación desde el primero hacia el segundo. Sin embargo, estas rutas de comunicación no son siempre directas y se pueden encontrar divididas en dos partes: la primera, entre el servicio antecesor y el nodo que detectó la colisión y, la segunda, entre este último y el servicio sucesor. Por lo tanto, los mensajes entre un nodo y su sucesor no podrán ser siempre enviados de forma directa utilizando los mecanismos explicados en la sección 5.4.

Se distinguen así, en el encaminamiento de mensajes a través del grafo de servicios, dos tipos de rutas de comunicación entre los nodos de una red, o de forma más concreta, entre los servicios que estos proveen:

- **Ruta directa:** estas rutas son gestionadas por la capa de *descubrimiento de servicios* y permiten la comunicación mediante el envío de un mensaje *multicast* directo entre los nodos que proveen los servicios.
- **Ruta indirecta:** son rutas gestionadas por la capa de *creación del grafo distribuido de servicios* y, por lo tanto, únicamente accesibles a través de ésta. Permiten la comunicación entre los nodos que ofrecen los servicios de una forma indirecta mediante un reenvío intermedio.

La capa de *creación del grafo distribuido de servicios* proporciona los mecanismos necesarios para llevar a cabo esta comunicación, de forma indirecta, mediante el *reenvío* de mensajes por parte de los nodos intermedios que han detectado la colisión entre los dos servicios implicados. Así, cuando un servicio quiere realizar el envío de un mensaje a su sucesor, este envío se lleva a cabo aplicando dos pasos:

1. El nodo emisor consulta en su grafo parcial de servicios cuál es el identificador del nodo intermedio que permite la conexión con el servicio sucesor al cuál quiere enviar un mensaje. El nodo envía un mensaje de tipo  $M_{RE}$  como contenido en un mensaje *multicast* enviado al nodo intermedio correspondiente, utilizando para ello los mecanismos explicados en la sección 5.4. El mensaje  $M_{RE}$  contiene el servicio origen del mensaje y el conjunto de servicios destinatarios del mismo

$$M_{RE} = (S_{origen}, \{S_{destino}\}, [datos])$$

2. Cuando el nodo intermedio recibe el mensaje de reenvío obtiene la lista de servicios destinatarios contenidos en el mismo y procede a reenviar su contenido a dichos nodos mediante el uso de un nuevo mensaje *multicast*, completando la información de destinatarios correspondiente.

Es posible que, durante el tiempo de vida de la red, puedan existir simultáneamente varias rutas para comunicar dos servicios entre sí, pudiendo ser tanto rutas directas como indirectas. Cuando un nodo quiere enviar un mensaje a otro, a través de la capa de gestión del grafo, se elige primeramente la ruta directa más corta, en número de saltos, entre las disponibles. En el caso de que no haya rutas directas para comunicar con un servicio determinado se selecciona, de nuevo, la ruta más corta pero, en este caso, entre las rutas indirectas conocidas por el nodo origen y que comunican con el servicio requerido.



### 6.3.5 Desconexión de servicios

La movilidad de los nodos de la red puede resultar en la desaparición de rutas de comunicación entre los servicios compatibles conectados y, por otro lado, los nodos pueden dejar de ofrecer, en cualquier momento, los servicios que proveen. Cualquiera de las situaciones anteriores tiene como resultado que el GDS deba ser modificado para reflejar las desconexiones de servicios que se han producido.

La desaparición de una ruta de comunicación con un nodo remoto que ofrece un servicio es detectada gracias a las notificaciones que produce la capa de descubrimiento de servicios. Cuando un nodo detecta la desaparición de una ruta obtiene cuáles son los servicios que, hasta ese momento, eran accedidos a través de dicha ruta desaparecida. Si no existen otras rutas alternativas, que conecten el nodo actual con los servicios remotos afectados, significa que dichos servicios han dejado de ser accesibles.

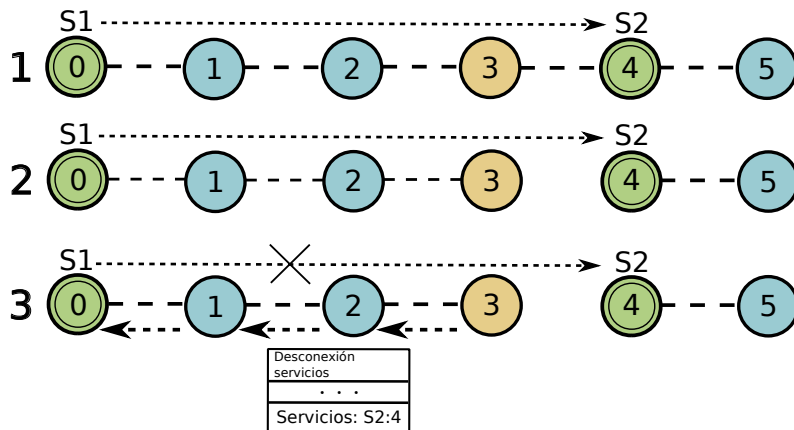
Cuando un servicio remoto ya no es accesible, debe ser eliminado del grafo parcial de servicios mantenido en el nodo que conecta con él. En el caso de que el nodo hubiera detectado colisiones entre parámetros de servicios se comprueba si las rutas desaparecidas afectan a las colisiones detectadas. Si para alguna conexión entre servicios, mantenida por el nodo actual, se detecta que ya no es posible acceder al servicio remoto se procede a eliminar del grafo distribuido de servicios dicha conexión. Así, la desconexión de un servicio es notificada a los servicios remotos que se encuentran conectados con el servicio inaccesible. Se envía para ello un mensaje de desconexión de servicio  $M_{DS}$  que contiene los servicios a eliminar por el nodo remoto. Este mensaje es enviado utilizando las rutas de comunicación mantenidas por esta capa, así como por la de descubrimiento de servicios.

En la figura 6.10 se muestra una desconexión de la red detectada por el nodo 3 y que resulta en el envío de un mensaje al nodo 1 para que éste actualice su grafo. La figura representa el caso en el que el nodo desaparecido es vecino directo del nodo que detectó la colisión. Sin embargo, tal y como se explica en la sección 5.4, las rutas desaparecidas son también notificadas, mediante el reenvío del mensaje correspondiente, hasta cualquier nodo afectado.

Por otro lado, cuando un servicio local es eliminado en un nodo, es necesario notificar a aquellos otros servicios que se encuentran conectados con él. Para ello, el nodo que ha eliminado el servicio envía un mensaje de servicio eliminado  $M_{SE}$  a aquellos nodos intermedios que habían detectado colisiones en las que participa dicho servicio. Este mensaje se encuentra definido de la siguiente forma

$$M_{SE} = ((ID_S, n)_1, (ID_S, n)_2, \dots, (ID_S, n)_i) \quad 1 \leq i$$

conteniendo la lista de los identificadores únicos de los servicios eliminados en el nodo  $(ID_S, n)$ . Por otro lado, la lista de nodos de colisión destinatarios



**Figura 6.10:** Notificación de desconexión del grafo enviada por el nodo que ha detectado la colisión

de cada mensaje de eliminación es obtenida de la información contenida en el grafo parcial del nodo que envía el mensaje.

El mensaje es enviado mediante *multicast* utilizando la funcionalidad proporcionada por la capa de descubrimiento de servicios. Cuando el nodo intermedio recibe el mensaje de eliminación de servicio, comprueba cada una de las conexiones de servicios que mantiene para buscar aquellos otros que se encuentran conectados con el servicio eliminado. A continuación, envía un mensaje de desconexión  $M_{SD}$  a los servicios remotos conectados, indicando así que los nodos proveedores de los mismos deben actualizar su grafo parcial de servicios y eliminar el servicio remoto desaparecido.

La figura 6.11 muestra un ejemplo de este proceso en el cual se ha eliminado el servicio  $S2$  en el nodo 4 y se procede a notificar dicha eliminación al servicio  $S1$  localizado en el nodo 0 mediante el envío de un mensaje. Dicho mensaje se dirige inicialmente al nodo 3, que había detectado la colisión que conecta los servicios entre sí, y, posteriormente, es reenviado por dicho nodo intermedio hasta el nodo 0. Este último nodo actualiza su GDS en consecuencia para reflejar los cambios acontecidos.

Por último, el borrado de una colisión anteriormente detectada en un nodo, como resultado de la recepción de un mensaje de actualización que contiene la eliminación de los parámetros implicados en la misma, supone la cancelación de las búsquedas activas que se iniciaron cuando se detectó la colisión, tal y como se explica en la sección 6.3.3. Para ello, el nodo que ha eliminado una colisión envía un mensaje de cancelación de búsqueda que es propagado a través de la red y que anula la búsqueda de aquellos tipos de parámetros involucrados en la colisión eliminada. El envío y propagación de estos mensajes de cancelación se realiza de acuerdo a lo explicado en la sección 5.3.4.

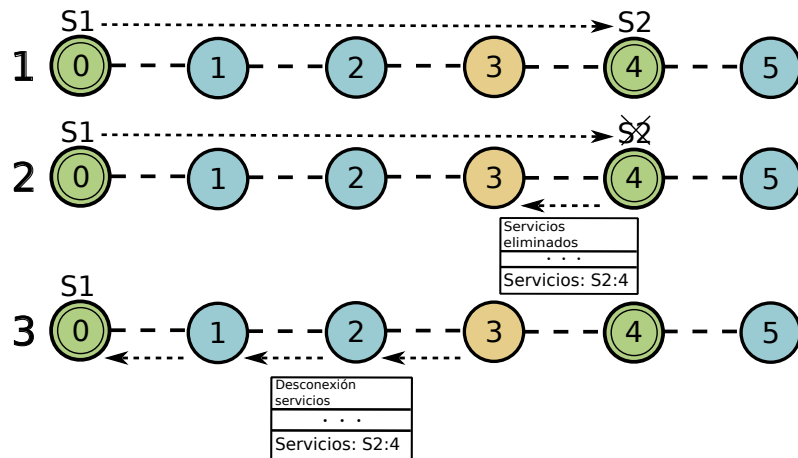


Figura 6.11: Eliminación de un servicio en un nodo y notificación a los servicios conectados

## 6.4 Búsqueda de composiciones

La búsqueda de composiciones puede ser iniciada por cualquier nodo durante el tiempo de vida de la red y consiste en la búsqueda de un camino en el GDS que satisfaga la especificación para el servicio compuesto requerido. Como se ha explicado en la sección 4.2, la búsqueda de composiciones planteada en esta investigación tiene únicamente en cuenta las relaciones de compatibilidad entre los tipos de los parámetros de entrada/salida de los servicios disponibles. Es el nodo que inicia la composición, que recibe las soluciones encontradas, el que debe determinar mediante otras características de las composiciones recibidas si éstas son adecuadas para sus fines.

El proceso de búsqueda de composiciones se divide en tres fases: creación de los *servicios de especificación* del servicio compuesto a partir de la descripción proporcionada; registro de estos servicios como servicios locales en el nodo que inicia la búsqueda; y, por último, búsqueda en el GDS para encontrar las composiciones disponibles utilizando las conexiones que se han establecido entre los distintos servicios registrados en la red MANET.

### 6.4.1 Servicios de especificación

Para que la búsqueda de un servicio compuesto pueda llevarse a cabo es necesario especificar cuáles son las características del servicio buscado. En la solución propuesta, esta especificación se realiza mediante la definición del tipo de los parámetros de entrada y salida de la composición resultante. Antes de llevar a cabo una búsqueda es necesario indicar cuáles son los parámetros de

entrada/salida que el servicio compuesto requerido debe satisfacer, tal y como se explica en la sección 4.1.

Los nodos que necesitan llevar a cabo la búsqueda de un servicio compuesto inician la búsqueda mediante el registro de los parámetros de entrada y de salida que definen al servicio requerido. Esto se lleva a cabo mediante la creación de dos servicios especiales que tienen como finalidad describir el inicio y el final de la composición buscada. Así, a partir de cada descripción de búsqueda de un servicio compuesto se crean los servicios de especificación de la composición:  $S_{INICIO}$  y  $S_{FIN}$ , que se corresponden con el servicio inicial y final de la misma, respectivamente.

Se supone que la especificación de una composición de servicio  $C_E$  se encuentra definida como

$$C_E = \{P_E, P_S\}$$

$$P_E = \{p_1, p_2, \dots, p_j\} \mid p \in O$$

$$P_S = \{p_1, p_2, \dots, p_k\} \mid p \in O$$

donde  $P_E$  y  $P_S$  son los tipos de los parámetros de entrada/salida, respectivamente, de acuerdo a la ontología  $O$ . A partir de la especificación anterior del servicio requerido se, definen los servicios de especificación de una composición buscada como

- El servicio  $S_{INICIO}$ , que representa el comienzo de una composición, se construye como un servicio que no tiene entradas y que proporciona como parámetros de salida el conjunto de las entradas contenidas en la especificación de servicio  $C_E$ .

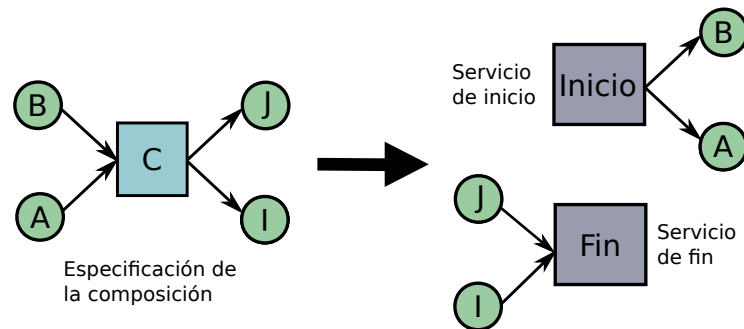
$$S_{INICIO} = \{\{\emptyset\}, P_E, C_{ID}\}$$

- El servicio  $S_{FIN}$ , que representa el servicio final de una composición, se construye como un servicio que posee, como parámetros de entrada, el conjunto de parámetros de salida de la especificación del servicio compuesto y no proporciona ningún parámetro de salida.

$$S_{FIN} = \{P_S, \{\emptyset\}, C_{ID}\}$$

Además, ambos servicios de especificación se construyen con un identificador único  $C_{ID}$  obtenido a partir del identificador de la solicitud de composición de servicio realizada. Este identificador permite, durante el proceso de búsqueda de composiciones, determinar qué parejas de servicios  $S_{INICIO}$  y  $S_{FIN}$  se corresponden entre sí y, por lo tanto, con una misma búsqueda. Se permite, de esta manera, que se pueden llevar a cabo en la red varias búsquedas de composiciones de servicios de forma simultánea.

En la figura 6.12 se muestra un ejemplo de la creación de los servicios de especificación a partir de una descripción de composición. Un nodo que quiere buscar aquellos servicios compuestos que proporcionen los parámetros con tipos  $P_S = \{I, J\}$  como sus salidas y que debe poder ser iniciado con un subconjunto de parámetros de entrada  $P_E = \{A, B\}$ , proporciona como resultado los servicios de especificación indicados.



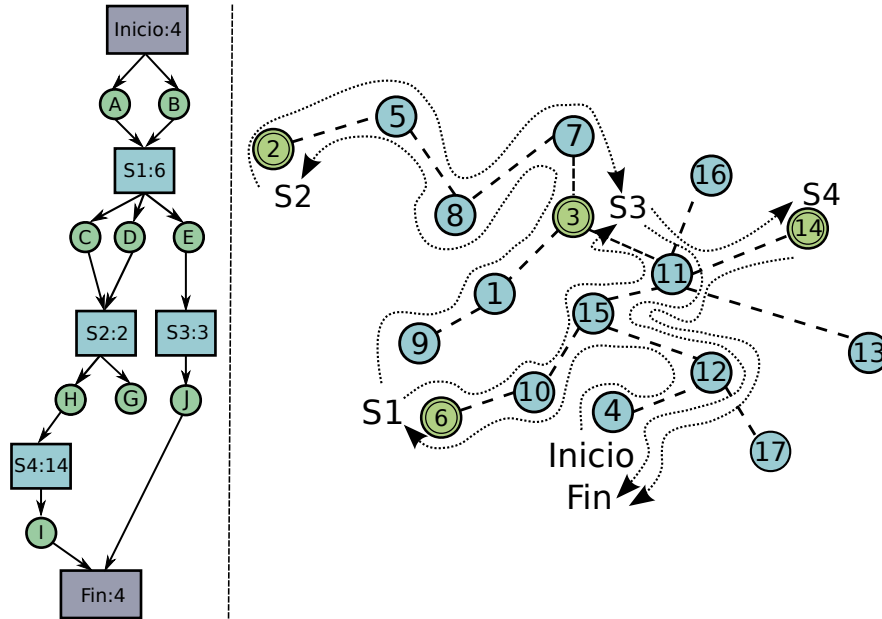
**Figura 6.12:** Servicios de especificación creados a partir de la definición del servicio compuesto

### 6.4.2 Registro de servicios de especificación

Después de crear los servicios de especificación para una búsqueda de composición, se lleva a cabo el registro de los estos como servicios locales del nodo que inicia la búsqueda. El registro de los nuevos servicios inicia los procesos de diseminación explicados en el capítulo 5. La diseminación de los parámetros de los servicios registrados puede producir colisiones con otros servicios provistos por diferentes nodos de la red ad hoc. En el caso de que existan servicios cuyas entradas/salidas sean compatibles con los servicios  $S_{INICIO}$  y  $S_{FIN}$  registrados, dichos servicios pasarán a formar parte del GDS y se podrá iniciar la búsqueda de composiciones en el grafo.

El proceso para el descubrimiento y establecimiento de las conexiones con los servicios recién registrados no es instantáneo, ya que es necesario llevar a cabo la comunicación de una serie de mensajes para descubrir las conexiones, actualizar las rutas y crear las aristas del grafo que permiten navegar hasta los servicios remotos. En el caso de la utilización de la *búsqueda hacia delante* en el GDS, que es la aplicada en esta investigación, el proceso de búsqueda no puede comenzar hasta que el servicio  $S_{INICIO}$  está conectado con algún servicio sucesor en la red. Por esta razón, la actualización del grafo de servicios proporciona notificaciones a la capa de búsqueda que indican cuando un servicio ha sido conectado con algún otro servicio sucesor en la red.

La figura 6.13 muestra cómo los servicios de especificación se unen al grafo de servicios para crear conexiones que puedan ser navegadas desde el nodo que inicia la búsqueda del servicio compuesto hacia los sucesores, y completar la ruta hasta el nodo inicial que había comenzado la búsqueda de la composición.



**Figura 6.13:** Conexión de los servicios de especificación de la composición en el grafo distribuido de servicios (GDS)

### 6.4.3 Búsqueda en el grafo distribuido de servicios

El proceso de búsqueda supone que los servicios  $S_{INICIO}$  y  $S_{FIN}$  han sido creados y registrados en el nodo que inicia la composición. Si existe una composición, y caminos en la red para encontrarla, los servicios se habrán conectado con otros servicios compatibles cercanos creando un subgrafo desde el nodo que inicia la búsqueda. Este grafo puede ser recorrido para obtener una o más soluciones al problema de composición planteado.

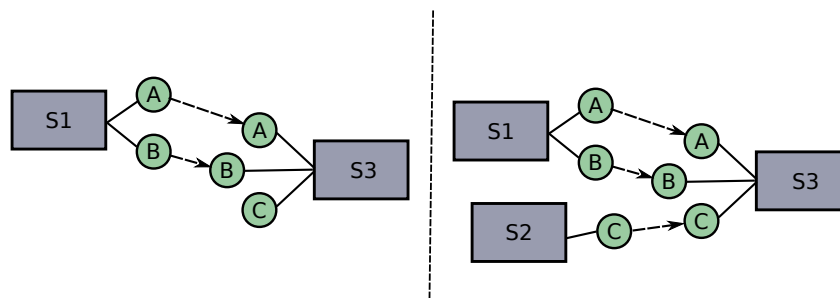
El proceso de composición comienza desde el servicio  $S_{INICIO}$  e intenta localizar un camino a través del grafo conectado que permita retornar al nodo original que inicia la búsqueda a través del servicio  $S_{FIN}$  correspondiente. Cuando comienza el proceso, desde el servicio  $S_{INICIO}$  se envía un mensaje de búsqueda de composición  $M_{BC}$  que se define como:

$$M_{BC} = (ID_C, C_I)$$

donde  $ID_C$  es el identificador de la búsqueda y  $C_I$  un conjunto, inicialmente vacío, de los servicios que se van añadiendo a la composición.

El mensaje de búsqueda es enviado desde el servicio inicial hasta sus sucesores y, para ello, se supone que se ha recibido una notificación de la capa de creación del grafo indicando que el servicio  $S_{INICIO}$  se ha conectado a uno o más servicios sucesores, gracias a la compatibilidad entre el tipo de sus parámetros. Los mensajes son dirigidos mediante una combinación de la funcionalidad de encaminamiento proporcionada por la capa de gestión del grafo, explicada en la sección 6.3.4, y el mecanismo de la capa de descubrimiento de servicios, presentado en la sección 5.4.

Cuando un nodo recibe un mensaje de búsqueda de composición comprueba si el servicio receptor del mensaje se encuentra cubierto completamente. Un servicio se establece como *completamente cubierto* cuando ha recibido mensajes de composición, correspondientes a la misma búsqueda, de diferentes servicios antecesores y, además, la unión de las composiciones recibidas en dichos mensajes satisface todas las entradas del servicio. Es decir, si las entradas de un servicio se encuentran completamente cubiertas significa que sus parámetros de entrada están conectados con servicios antecesores mediante alguna arista del GDS y que, por lo tanto, existe un camino que permite que los mensajes de búsqueda puedan llegar desde el servicio antecesor hasta él.



**Figura 6.14:** Entradas de un servicio cubiertas por los parámetros de salida de sus antecesores

Por ejemplo, en la figura 6.14, el servicio  $S3$  solamente podría ser utilizado dentro de una composición si puede recibir todos los parámetros necesarios,  $A$ ,  $B$  y  $C$  en este caso, desde sus antecesores. Durante una búsqueda de composición el servicio recibirá mensajes desde algunos de sus antecesores que estén conectados a sus entradas a través de parámetros compatibles. En el lado izquierdo de la figura, el servicio  $S3$  se encuentra conectado a un servicio antecesor que le proporciona salidas compatibles con los parámetros  $A$ ,  $B$ , sin embargo, el servicio  $S3$  no podrá estar completamente cubierto hasta que exista otro servicio que le pueda proporcionar el parámetro restante, situación que se muestra en el lado derecho.

Cada nodo mantiene una tabla interna con los mensajes de composición que recibe durante su tiempo de vida. El tiempo que los mensajes son mantenidos en la tabla antes de ser eliminados viene determinado por el proceso de expiración de las composiciones explicado en la sección 6.4.4. Cuando un servicio recibe un mensaje de composición que resulta en que sus entradas se encuentren completamente cubiertas, se produce la propagación de la composición a los sucesores de dicho servicio. Para ello, se crea un nuevo mensaje que contiene la unión de todas las listas de composición recibidas y, que además, han conseguido cubrir completamente el servicio. A la unión de las composiciones se añade también el propio servicio actual. El mensaje resultante es enviado a los sucesores del servicio para continuar la búsqueda. Además, cada vez que la capa de creación del grafo notifica que un servicio tiene un nuevo sucesor, se comprueba si el servicio actual ya se encuentra completamente cubierto, en cuyo caso se reenvía el último mensaje de composición al dicho nuevo sucesor.

Así, en la figura 6.14 mostrada anteriormente, cuando se propagan los mensajes relativos a una búsqueda de composición concreta, para que el servicio  $S_3$  sea añadido al resultado y, por lo tanto, propagado a sus sucesores, debe recibir el mismo mensaje de composición  $M_{BC}$  a través de todas sus entradas, es decir, tanto desde el servicio  $S_1$  como  $S_2$ . Obviamente, el servicio  $S_1$  y el servicio  $S_2$  deben estar, a su vez, completamente cubiertos, y haber recibido el mensaje correspondiente a la búsqueda a través de todas sus entradas. Únicamente el servicio  $S_{INICIO}$  de una composición se encuentra siempre completamente cubierto debido a que no tiene ninguna entrada definida.

El proceso de composición finaliza cuando el mensaje de composición vuelve al nodo que inició la búsqueda, a través del servicio  $S_{FIN}$  correspondiente y, además, dicho servicio de finalización se encuentra completamente cubierto. La comprobación del identificador del servicio de finalización, que permite determinar si el servicio se corresponde con la búsqueda que se está llevando a cabo, se realiza antes de enviar el mensaje a dicho sucesor. Se evita así el envío de mensajes innecesarios a servicios  $S_{FIN}$  que no se correspondan con la búsqueda que se está llevando a cabo.

Cuando el servicio de finalización  $S_{FIN}$  de una búsqueda determinada se encuentra completamente cubierto, las composiciones obtenidas, como resultado de la unión de los diferentes mensajes de búsqueda recibidos por dicho servicio, son combinadas para obtener la composición resultante. La unión de todos los mensajes recibidos es una lista de de servicios que debe ser convertida en un grafo de dependencia de servicios que proporcionará la composición o composiciones resultantes. Este proceso se lleva de forma local en el nodo que ha iniciado la búsqueda de la composición.

Por ejemplo, en el caso de la situación mostrada en la figura 6.13, el nodo 4, que había iniciado un proceso de búsqueda de composición, recibe el grafo de composición recogido en la parte izquierda de la figura. Dicho grafo con-



tiene la información necesaria para, a partir del nodo inicial 4 llevar a cabo la ejecución del servicio paso a paso, desde las entradas a las salidas. Los nodos de la red poseen, debido a la construcción del GDS, la información necesaria en su tabla de encaminamiento para alcanzar desde un servicio a aquellos otros que son sus sucesores, mediante la aplicación del número de saltos intermedios necesarios. Un ejemplo de estas rutas de comunicación se muestra en la parte derecha de la misma figura. Así, por ejemplo, desde el nodo 6 existe una ruta que comunica el servicio  $S_1$  con los servicios  $S_2$  y  $S_3$ , situados en los nodos 2 y 3, respectivamente.

Por último, los mensajes de composición son enviados sin la utilización de la *difusión confiable* explicada en la sección 4.5.2, al contrario que lo que ocurren con los mensajes de creación del grafo en los que sí se aplica el uso de esta técnica. Esto significa que estos mensajes son más proclives a las colisiones, lo que reduce el porcentaje de las composiciones encontradas debido a la pérdida de mensajes. Sin embargo, el envío de estos mensajes a los vecinos mediante difusión directa reduce el tiempo necesario para la búsqueda de las composiciones, ya que no es necesario esperar a las respuestas de confirmación de los vecinos durante la difusión. En la sección 6.5, que contiene la evaluación realizada de esta solución, se comentan más detalles acerca de esta cuestión.

#### 6.4.4 Expiración de búsquedas

Para evitar que las búsquedas de composiciones se propaguen de forma indefinida a través de la red se incluyen dos mecanismos que permiten controlar su tiempo de vida y producir su expiración.

El primero de los mecanismos tiene que ver con el uso de un temporizador que determina el tiempo durante el que se lleva a cabo la búsqueda de soluciones para una petición de composición dada. El tiempo máximo de validez de una búsqueda de composición es incluido en el mensaje que se envía desde el nodo inicial  $S_{INICIO}$ . Cada vez que un nodo recibe un mensaje de una búsqueda incluye dicho mensaje en una lista de mensajes recibidos junto con el tiempo de validez correspondiente. El tiempo de validez se almacena solamente una vez y se corresponde con aquel que estaba contenido en el primer mensaje recibido para una búsqueda determinada.

Cada vez que un nodo propaga un mensaje de búsqueda a los sucesores de un servicio, el tiempo de validez enviado a dichos sucesores es decrementado según el tiempo transcurrido desde que se recibió el primer mensaje para dicha búsqueda. Si el tiempo transcurrido hace que el tiempo de validez para una búsqueda sea menor o igual que cero, se procede a eliminar de la lista de mensajes dicha búsqueda de composición y no se lleva a cabo la propagación de ningún mensaje más, relativo a esa búsqueda, a los sucesores.

Con el procedimiento anterior se consigue que, cuando el tiempo de validez para una búsqueda expira, todos los nodos de la red que habían recibido mensajes correspondientes a dicha búsqueda eliminen toda la información relacionada con ella de manera simultánea. Si el nodo inicial no ha recibido una respuesta en el tiempo de validez indicado puede considerar que no existen soluciones a su solicitud de composición y repetir la búsqueda de nuevo o llevar a cabo otras acciones para solucionar el problema.

Por otro lado, los mensajes de búsqueda de composiciones llevan asociado un contador de tiempo de vida TTL que se decrementa con cada salto que el mensaje realiza a través de la red. Cuando el contador de un mensaje de búsqueda propagado a través del grafo en la red ad hoc llega a cero, dicho mensaje es descartado y la búsqueda finaliza. Se consigue así limitar la longitud de las composiciones en número de saltos realizados a través de la red ad hoc.

Además del contador anterior, es posible incluir otro  $TTL_s$ , más que se decrementa cada vez que el mensaje de búsqueda es propagado por un servicio hacia sus sucesores y que posibilita controlar la longitud en servicios de la composición resultante.

En cualquiera de los dos casos, la detección de la expiración de la búsqueda, debido al descarte del mensaje de composición al agotarse alguno de los contadores, será realizada por el nodo que ha iniciado la búsqueda cuando expire el tiempo de validez asociado y no haya recibido ninguna respuesta.

### 6.4.5 Ejecución de composiciones

Cuando un nodo obtiene una solución, a una petición de composición realizada, puede llevar a cabo la ejecución de la misma. El proceso de ejecución consiste en recorrer, mediante el envío de mensajes a través del grafo, los servicios que forman parte del sub-grafo de solución obtenido. Este mecanismo de ejecución es similar al proceso de búsqueda hacia adelante en el GDS, sin embargo, ahora el mensaje contiene, desde su inicio, la composición a ejecutar y el valor de los parámetros iniciales para su invocación.

El proceso de ejecución se inicia en el  $S_{INICIO}$  y se propaga hacia sus sucesores, de acuerdo al grafo de composición utilizado. Cada vez que un servicio recibe el mensaje de ejecución utiliza los parámetros que contiene para llevar a cabo su ejecución dando como resultado un conjunto de parámetros de salida que serán utilizados como entrada en la siguiente etapa de la ejecución. Al igual que ocurre durante la búsqueda de la composición, un servicio solamente puede ser ejecutado si tiene todos sus parámetros de entrada satisfechos, es decir, si ha recibido mensajes desde todos sus antecesores. Estos mensajes deben contener valores para todos los parámetros de entrada del servicio. El proceso de ejecución termina, al igual que en el caso de la búsqueda, cuando se alcanza el servicio  $S_{FIN}$ .

Sin embargo, como ya se ha comentado anteriormente, esta investigación se centra en los aspectos relacionados con la búsqueda de las composiciones de servicios disponibles en una red MANET. Por lo tanto, los problemas asociados a la ejecución de las composiciones encontradas no han sido tenidos en cuenta. Puede consultarse la sección 3.2.8 para conocer qué soluciones existen en la actualidad para la ejecución de servicios compuestos en este tipo de redes.

## 6.5 Evaluación

La evaluación de la propuesta para la composición de servicios en redes ad hoc ha sido llevada a cabo mediante la utilización del simulador de red ns-2 extendido con AgentJ. Se puede consultar más información sobre el uso de simuladores de red en la sección 2.2. La experimentación ha sido realizada comparando la solución propuesta con un mecanismo de composición de servicios *reactivo* y basado en la exploración de la red mediante *inundación*.

En la búsqueda por inundación los nodos que necesitan componer un servicio inician el proceso mediante el envío de un mensaje al resto de nodos de la red. La distancia a la que un mensaje de búsqueda es propagado en este tipo de búsqueda se controla mediante el uso de un contador *TTL* que es decremen-tado en cada salto. El mensaje inicial contiene el servicio de inicio  $S_{INICIO}$  que ha sido obtenido de la descripción del servicio compuesto requerido. Cada vez que el mensaje de búsqueda es recibido por un nodo de la red, la composición contenida en el mensaje es comprobada contra todos los servicios locales provistos por dicho nodo. Si un servicio local es cubierto completamente por la composición recibida, de acuerdo a la definición dada en la sección 6.4.3, dicho servicio es añadido a la composición. El mensaje resultante, que contiene la nueva composición, es enviado nuevamente a toda la red.

El proceso anterior continúa hasta que el servicio de  $S_{FIN}$ , para la búsqueda correspondiente, es cubierto por la composición o, en su defecto, cuando se ha explorado toda la red y no han podido añadirse nuevos servicios al mensaje de composición. Para determinar que se ha explorado toda la red, los nodos almacenan temporalmente los mensajes recibidos y descartan aquellos cuyo identificador ha sido recibido anteriormente. Cada vez que se añade un servicio a la composición se cambia el identificador del mensaje y se propaga de nuevo mediante inundación.

Los experimentos se han llevado a cabo mediante la distribución de servicios de forma aleatoria en la red ad hoc. Estos servicios constituyen diferentes composiciones con una longitud variable en el número de servicios que la conforman. Cada configuración del escenario ha sido evaluada 10 veces y, como resultado final, se han obtenido los valores medios de cada configuración. En cada simulación se seleccionan de manera aleatoria los nodos que llevan a cabo

**Tabla 6.1:** Configuración de la evaluación del protocolo de composición

Parámetro simulación	Valor
Nodos	30
Área de simulación	300 x 300 m
Modelo de movilidad	<i>Random waypoint</i>
Velocidad nodos	Distribución uniforme de [0, 5] m/s
Tiempo de pausa	[0, 20, 50, 100] s
Velocidad de transmisión	54 Mb/s
Periodo de baliza	1000 s
Distancia máx. de diseminación $D_d$	5 saltos
TTL de búsqueda	5 saltos
Longitud las composiciones	[3, 5, 7] servicios
Frecuencia de búsqueda	1,0 búsquedas/s

las búsquedas de las diferentes composiciones disponibles, de tal forma que en la red se lleven búsquedas con una frecuencia fija.

Las características del escenario de experimentación han sido elegidas de tal forma que la relación entre el área de simulación y la distancia de transmisión, junto con la movilidad de los nodos participantes, es equivalente a un conjunto de personas que se mueven andando dentro de un área pequeña (10 m de rango de transmisión en un área de 30 m x 30 m). Además, en cada simulación un 30% de los nodos son seleccionados aleatoriamente para proporcionar los servicios que pueden ensamblarse para formar una composición.

Se utiliza el modelo de movilidad *Random waypoint* que se basa en el movimiento de los nodos de la red con velocidades y direcciones aleatorias durante el tiempo de simulación. Además, los nodos de la red alternan sus movimientos con *tiempos de pausa* en los que permanecen estáticos en su posición. En el trabajo realizado por Camp, Boleng y Davies (2002) se exponen las características de éste y otros modelos de movilidad.

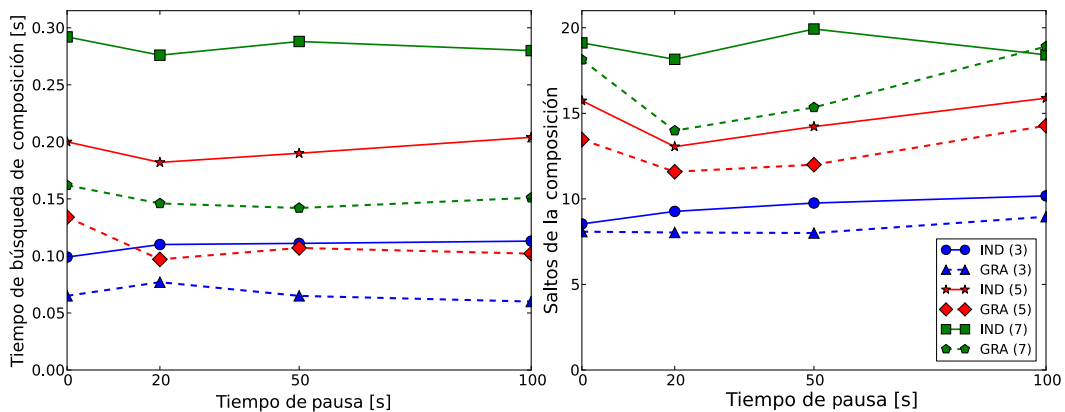
En los experimentos realizados se han utilizado, para la configuración del simulador, los parámetros recogidos en la tabla 2.2. Por otro lado, las propiedades específicas del escenario de evaluación están resumidas en la tabla 6.1. De acuerdo al trabajo de Kurkowski, Navidi y Camp (2007a), el tamaño del área de simulación, la distancia máxima de transmisión y el número de nodos utilizados permiten obtener un escenario con las siguientes características:

- *Partición media de la red  $\leq 8\%$* . Esta característica hace referencia al porcentaje de nodos de la red que de media se encuentran desconectados en un instante determinado. La selección de un valor bajo para la partición de la red garantiza que el protocolo es probado en un escenario de experimentación en el que la topología de la red, pudiendo ser posible el llegar a conectar la casi totalidad de los servicios disponibles.
- *Camino medio más corto = 3 saltos*. Representa la distancia media más corta que puede existir entre dos nodos de la red. El protocolo de descubrimiento se prueba así con una configuración de red que permite que realmente se haga uso de las características de encaminamiento que el protocolo proporciona.

En las gráficas correspondientes a cada uno de los experimentos realizados en esta sección, *GRA*, en líneas punteadas, identifica al protocolo para la composición de servicios propuesto, mientras que *IND*, en líneas continuas, es el mecanismo de búsqueda de servicios basado en inundación que se utiliza como referencia base para realizar las distintas comparaciones.

### 6.5.1 Tiempo de pausa

En este experimento se ha variado el tiempo de pausa de los nodos entre 0 y 100 segundos para determinar cómo se comportan diferentes aspectos del protocolo propuesto.



**Figura 6.15:** Efecto del intervalo de pausa en el tiempo medio de búsqueda de una composición y en el número medio de saltos efectuados para longitudes de composición de 3, 5 y 7 servicios

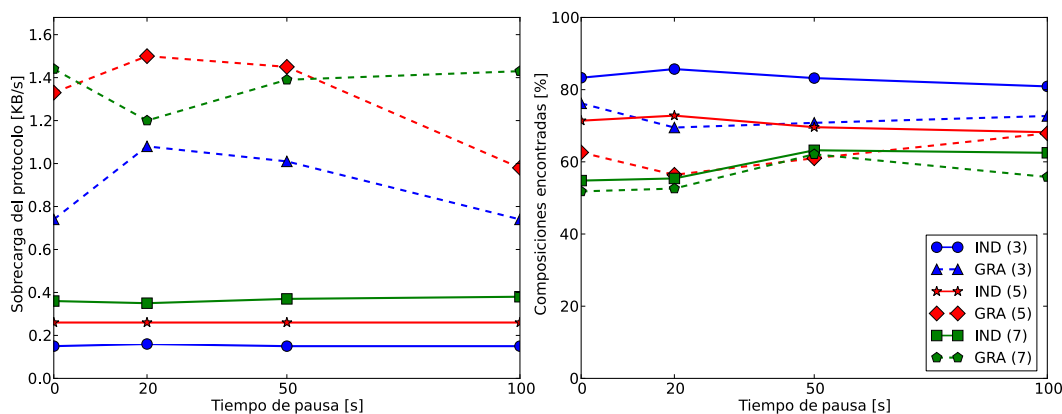
Como se muestra en la figura 6.15, el uso del grafo distribuido de servicios *GRA* reduce el tiempo de búsqueda necesario para encontrar una composición,

comparado con la búsqueda basada en inundación *IND*. La razón para esta reducción es que, en la solución propuesta, los mensajes de composición son enviados hacia los servicios sucesores utilizando las rutas más cortas entre las disponibles. Tal y como puede observarse en la misma figura, el número de saltos requeridos y, como consecuencia, el tiempo necesario para localizar una respuesta, son menores en aquellos casos en los que se usa el GDS.

Además, en las soluciones tradicionales, como la búsqueda basada en inundación, en las cuales no se realiza un proceso inicial para encontrar las conexiones entre los servicios disponibles, es necesario llevar a cabo una exploración de la red en el momento en el que se inicia cada búsqueda. Por lo tanto, se produce una mayor congestión de la red en dichos instantes que tiene como resultado un incremento en el tiempo de descubrimiento de cada composición.

Debido a los motivos explicados anteriormente, el tiempo requerido es menor para cada longitud de servicio determinada cuando se utiliza el grafo de servicios precalculado. Por otro lado, y como puede observarse, el tiempo de composición se incrementa, como era de esperar, cuando se aumenta la longitud de las soluciones disponibles en número de servicios.

El mantenimiento del grafo de servicios distribuido tiene un coste asociado debido a que la sobrecarga introducida por su gestión, como se muestra en la figura 6.16. En la solución propuesta, la creación del grafo y el mantenimiento del mismo son tareas *proactivas*, realizadas durante el tiempo de vida de la red y en respuesta a los cambios que suceden en su topología. Sin embargo, en las soluciones *reactivas*, como es el caso de la búsqueda por inundación comparada, los mensajes de búsqueda de composiciones son propagados solamente cuando hay una solicitud para componer un servicio.



**Figura 6.16:** Efecto del intervalo de pausa en la sobrecarga debida al protocolo y en el porcentaje de composiciones encontradas para longitudes de composición de 3, 5 y 7 servicios

Como puede observarse en la figura 6.16, en la solución reactiva la sobrecarga introducida por el proceso de composición es independiente del tiempo de pausa de los nodos. En este caso, la sobrecarga está producida por el incremento sucesivo en la longitud de las soluciones encontradas, que supone que la solución basada en inundación deba explorar la red un mayor número de veces para encontrar cada servicio sucesivo de la composición, debido a la inexistencia de un grafo de servicios precalculado. Por el contrario, la solución para la composición de servicios propuesta en esta tesis doctoral tiene una fuerte dependencia de la movilidad de los nodos y de los cambios en la topología.

En la figura 6.16 se observa también que el uso de un grafo precalculado para la dependencia entre servicios posee una desventaja en el hecho de que reduce el porcentaje de composiciones encontradas. La causa de esta reducción está explicada por los siguientes dos motivos.

Primero, la movilidad de la red tiene un efecto en la capa de creación del grafo de servicios, con el objeto de añadir o eliminar nuevos caminos. La congestión de la red incrementa la probabilidad de colisión de mensajes durante la propagación de los mensajes de búsqueda. Esto tiene como resultado que se pierda un mayor número de mensajes y se incremente el número de búsquedas de composición que no son correctamente finalizadas.

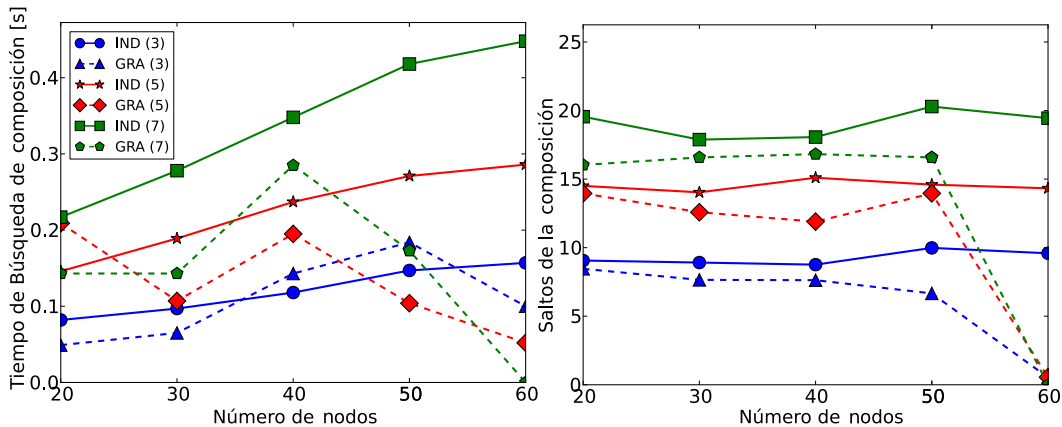
La otra razón para la reducción en el porcentaje de composiciones encontradas es que la aparición y desaparición de los vecinos no es detectada inmediatamente, debido al uso de *balizas periódicas*, tal y como se explica en la sección 4.5.1. Las nuevas rutas no son detectadas de forma instantánea y las rutas desaparecidas pueden ser todavía usadas cuando se lleva a cabo la propagación de los mensajes de búsqueda, durante un periodo de tiempo, hasta que son completamente eliminadas. Esto produce que el incremento en la movilidad de la red y, como consecuencia, en el número de rutas invalidadas, produzca un mayor cantidad de fallos en la búsqueda de las composiciones al resultar en un número más alto de colisiones de mensajes.

## 6.5.2 Número de nodos

El objetivo de este experimento ha sido determinar cómo responde el protocolo propuesto a la variación del número de nodos de la red. Para ello, en la misma configuración del escenario anterior, el número de nodos de la red ad hoc ha sido modificado entre 20 y 60 nodos.

La figura 6.17 muestra que la solución propuesta, basada en el grafo distribuido de servicios, sigue encontrando soluciones a las peticiones de composición en un tiempo menor que la basada en inundación. En el caso de la búsqueda por inundación se puede observar que el tiempo necesario para localizar una solución aumenta con el número de nodos. Esto es debido a que los mensajes deben difundirse a través de una red con un mayor número de nodos

para encontrar cada servicio que pertenece a una composición dada. Además, como puede observarse en la segunda gráfica de la figura, esto se debe también a que el número de saltos sigue siendo mayor, para cada longitud de servicios disponible, con respecto al protocolo propuesto.

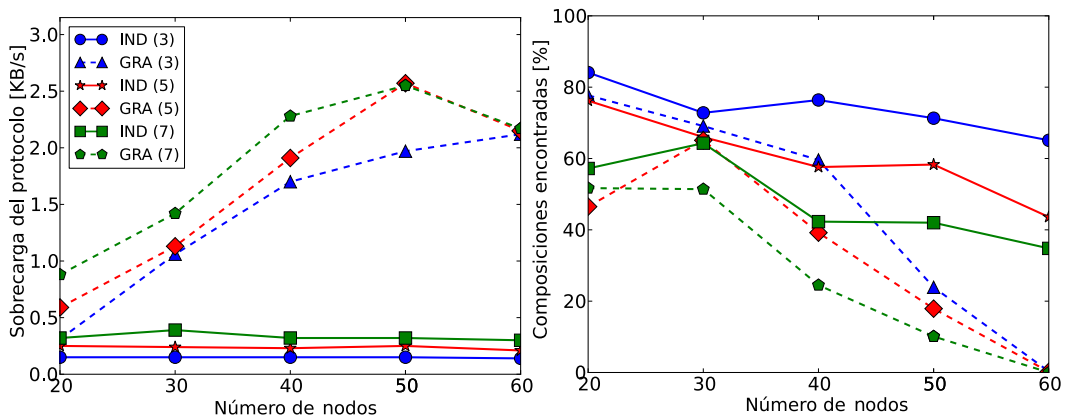


**Figura 6.17:** Efecto de la variación del número de nodos en el tiempo medio de búsqueda de una composición y en el número medio de saltos efectuados para longitudes de composición de 3, 5 y 7 servicios

Por otro lado, y al igual que en el caso del experimento anterior, el mantenimiento del grafo introduce una sobrecarga en los mensajes enviados a través de la red ad hoc. Como puede observarse en la primera gráfica de la figura 6.18 el aumento en el número de nodos en la red supone un gran incremento en la información transmitida para el protocolo propuesto *GRA*. Sin embargo, en el caso del protocolo basado en inundación *IND*, no existe un aumento apreciable de la sobrecarga, debido a que, como el protocolo es reactivo, los mensajes relacionados con la búsqueda de composiciones se propagan únicamente cuando se llevan a cabo las solicitudes.

En la figura 6.18 se muestra también el efecto que la variación en el número de nodos tiene en el porcentaje de composiciones encontradas. Como puede observarse, el protocolo propuesto no escala bien cuando el número de nodos aumenta en la red ad hoc. La relación entre las composiciones encontradas con respecto a las disponibles disminuye cuando el número de nodos de la red aumenta. Así, para red de 60 nodos no se llegan a encontrar soluciones debido a que los mensajes de búsqueda a través del grafo, que se propagan sin controlar si han sido correctamente transmitidos, colisionan más frecuentemente con otras transmisiones. Como puede observarse, la solución basada en inundación no sufre este problema, aunque para el mismo aumento de nodos, las composiciones encontradas sufren una disminución de hasta un 20% debido también al aumento de las colisiones entre las transmisiones de mensajes a través del medio inalámbrico.

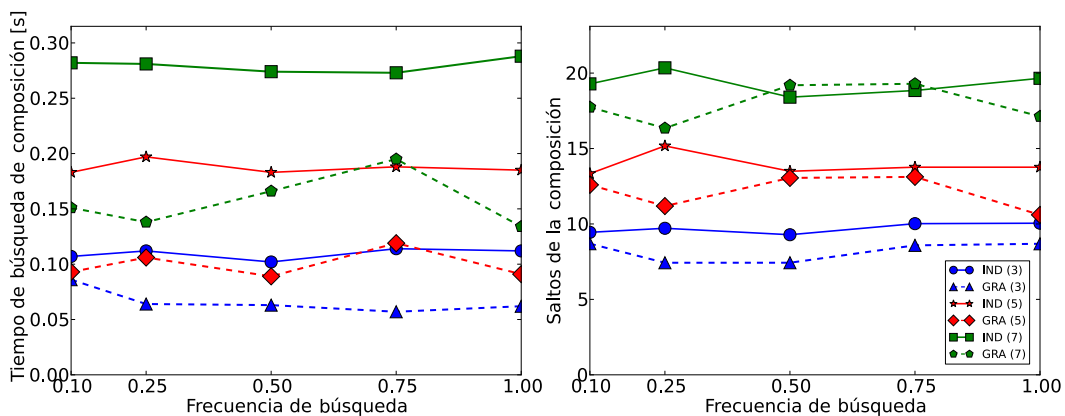




**Figura 6.18:** Efecto de la variación de la cantidad de nodos en la sobrecarga debida al protocolo y en el porcentaje de composiciones encontradas para longitudes de composición de 3, 5 y 7 servicios

### 6.5.3 Frecuencia de búsqueda

Se ha realizado un último experimento que ha consistido en determinar cómo afecta la variación en la frecuencia de búsqueda de composiciones. Como recoge la figura 6.19, la variación en la frecuencia de búsqueda no tiene un impacto apreciable en el tiempo de búsqueda para ninguna de las dos soluciones evaluadas.



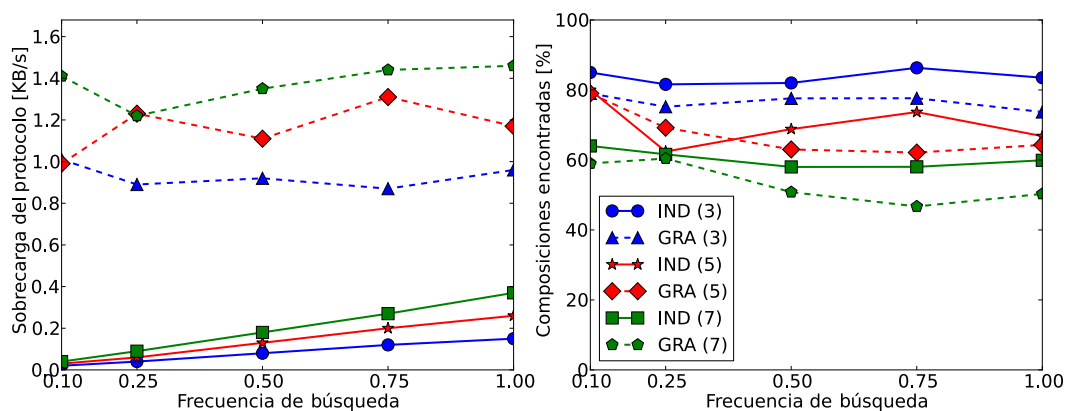
**Figura 6.19:** Efecto de la frecuencia de búsqueda en el tiempo medio de búsqueda de una composición y en el número medio de saltos efectuados para longitudes de composición de 3, 5 y 7 servicios

En el caso del protocolo basado en *inundación* solamente se produce una propagación de mensajes cuando se requiere realizar una búsqueda. Así, debido a que la cantidad de información transmitida por los mensajes de búsqueda, en

el caso de dicho protocolo, es pequeña, el aumento en la frecuencia de estos mensajes, no llega a producir la suficiente congestión en la red ad hoc para que se produzca un incremento en las colisiones entre mensajes.

En el caso del protocolo propuesto en esta tesis doctoral, el incremento en la frecuencia de búsquedas realizadas no supone tampoco una variación significativa en el tiempo de composición. Esto es debido a que, como se ha visto en los experimentos anteriores, la solución depende principalmente del número de nodos y de la movilidad de los mismos, parámetros que afectan al proceso de construcción y mantenimiento del grafo. Por otro lado, y como era de esperar, los saltos realizados por el mensaje de búsqueda al realizar una composición tampoco se ven afectados por la variación en la frecuencia de composiciones en la red ad hoc.

Los razonamientos anteriores pueden observarse de nuevo en la figura 6.20 que muestra que la sobrecarga introducida por el protocolo de composición propuesto no tiene un comportamiento definido a pesar del incremento en la frecuencia de las búsquedas realizadas. Por otro lado, tampoco tienen un efecto apreciable en el porcentaje de las composiciones encontradas, que no varía con una tendencia determinada con el incremento de las frecuencias de búsqueda utilizadas en cada simulación, aunque en algunos casos puede observarse un pequeño descenso en las composiciones encontradas.



**Figura 6.20:** Efecto de la frecuencia de búsqueda en la sobrecarga debida al protocolo y en el porcentaje de composiciones encontradas para longitudes de composición de 3, 5 y 7 servicios

## 6.6 Conclusiones

En este capítulo se ha presentado una solución para la composición dinámica de servicios en redes móviles ad hoc que se basa en la creación y mante-

nimiento de un grafo precalculado de dependencia entre los servicios. La creación del grafo está basada en el protocolo de diseminación de parámetros y búsqueda de servicios presentado en el capítulo 5. A partir de la diseminación de parámetros se lleva a cabo una detección de colisiones entre las entradas y salidas de los diferentes servicios disponibles que son compatibles entre sí y la posterior conexión de estos servicios formando el grafo distribuido de servicios.

Como se ha demostrado en la evaluación realizada, el uso del grafo distribuido de servicios se traduce en una reducción en el tiempo de búsqueda de composiciones. Sin embargo, la creación y el mantenimiento del grafo tiene un coste asociado como resultado del incremento en la comunicación requerida para su gestión y en la posible congestión de la red. Hay que tener en cuenta que el envío de los mensajes se realiza como respuesta a los cambios en la topología que suceden durante la movilidad de los nodos.

Sin embargo, la solución propuesta puede ser adecuada en aquellas situaciones en las que un nodo que busca una composición necesita obtener las soluciones disponibles de una manera más rápida, seleccionando posteriormente de entre éstas la más adecuada a sus necesidades. Además, la disminución en el tiempo obtenida para la búsqueda de las composiciones permite reducir también el tiempo durante el cual una aplicación, que necesita cierto servicio para llevar a cabo sus funciones, no se encuentra disponible para el usuario.

Por otro lado, como se ha constatado durante la realización de los experimentos, la solución propuesta no escala bien cuando se produce un aumento en el número de nodos de la red, debido a que también se incrementa la sobrecarga introducida por las tareas de creación y mantenimiento del grafo. Esto hace que la solución presentada sea adecuada solamente para redes que tengan un número medio-bajo de nodos (hasta unos 40) y con una velocidad de los nodos cercana al movimiento de personas andando (0 - 5 m/s). En estos casos, el protocolo propuesto proporciona ciertas ventajas al reducir el tiempo de búsqueda efectivo para las composiciones.

Como línea futura de investigación se plantea estudiar la posibilidad de utilizar búsquedas bidireccionales en el grafo cuando se localizan las composiciones disponibles. Hay que tener en cuenta que la solución propuesta puede permitir, con ligeras modificaciones, no solo la creación de conexiones hacia los sucesores de un servicio determinado, sino también desde estos a sus servicios antecesores, creando conexiones en ambos sentidos.

La existencia de conexiones en dos direcciones, entre los servicios compatibles de la red, podría permitir aplicar procesos de búsqueda *hacia atrás* o, incluso, *bidireccionales*. En este último caso, las búsquedas podrían comenzar de forma simultánea tanto desde el servicio  $S_{INICIO}$  como desde el  $S_{FIN}$  de una búsqueda, lo que podría reducir aún más el tiempo necesario para encontrar una composición, debido a la naturaleza paralela de las redes MANET.

Por otro lado, un aspecto que no se ha explorado en esta tesis doctoral es la selección entre las múltiples composiciones recibidas para una solicitud de composición. El algoritmo propuesto para la búsqueda de composiciones podría ser mejorado con la inclusión de otros mecanismos que, además de la compatibilidad funcional entre los parámetros de entrada y salida de los servicios, tuvieran en cuenta otros aspectos de los mismos. El objetivo es que los servicios fueran seleccionados durante el proceso de búsqueda en el grafo distribuido de servicios, atendiendo a las restricciones incluidas en la búsqueda. Estas restricciones permitirían limitar las posibles soluciones recibidas por el nodo solicitante, así como el número de mensajes enviados a través de la red al explorar soluciones no adecuadas a los requisitos especificados.

# Conclusiones

Este capítulo identifica las contribuciones principales realizadas en esta tesis así como las posibles líneas futuras de investigación que pueden continuar este trabajo. El capítulo se encuentra estructurado de la siguiente manera: la sección 7.1 proporciona un resumen de las contribuciones realizadas en el área del descubrimiento y la composición de servicios en redes MANET, mientras que en la sección 7.2 ofrece una visión de las posibles direcciones futuras de investigación.

## 7.1 Contribuciones

El objetivo principal de la investigación ha sido el desarrollo de un mecanismo automático para la composición de servicios en redes MANET que se basa en la construcción y mantenimiento de un grafo distribuido de servicios (GDS). Así, en el capítulo 6 se han presentado las características fundamentales de la solución propuesta. Además, durante la realización de la investigación se ha mostrado también que la composición de servicios está íntimamente relacionada con el proceso de descubrimiento de servicios. Esta relación ha producido como resultado otra aportación principal en la investigación, que ha consistido en la propuesta de un nuevo protocolo para el descubrimiento de servicios en redes móviles ad hoc, explicado en profundidad en el capítulo 5.

El estudio de las ventajas y limitaciones de las distintas soluciones existentes en la actualidad, tanto para el descubrimiento de servicios como para su composición, ha dado lugar a la revisión bibliográfica de la literatura científica correspondiente, y que ha sido presentada en el capítulo 3. La revisión del estado actual ha permitido construir una clasificación de las diferentes soluciones

existentes para abordar los problemas relativos al descubrimiento y a la composición de servicios de una forma general y, específicamente, para su aplicación a las características de las redes MANET. Esta clasificación ha permitido acotar claramente cuáles son las aportaciones de la investigación presentada.

A pesar de que el protocolo de descubrimiento de servicios propuesto puede ser utilizado de forma independiente a la solución para la composición de servicios, en el capítulo 4 se propone una arquitectura que integra las dos soluciones. Se consigue así comprobar la validez de la hipótesis  $H(A)$ , planteada en la sección 1.2, que hace referencia a la aplicación en redes móviles ad hoc de un mecanismo de composición de servicios basado en la construcción y el mantenimiento de un grafo de conexión entre servicios distribuido.

Por otro lado, el protocolo de descubrimiento presentado en el capítulo 5 valida la hipótesis  $H(B)$ , presentada a su vez en la sección 1.2, que propone la diseminación de la información sobre los parámetros de entrada y salida de los servicios aplicando una categorización de los mismos mediante una ontología de conceptos compartida.

Se incluye a continuación un resumen de las contribuciones principales de la tesis doctoral. Se puede consultar más información al respecto de cada una de las aportaciones, así como de las consecuencias que tienen las diferentes decisiones tomadas durante su diseño, en las secciones de evaluación y conclusiones de los capítulos correspondientes.

- **Capítulo 4:** donde se presenta la visión general de la problemática de la tesis y cuyas aportaciones se resumen a continuación:
  - La definición de una arquitectura integral para la composición de servicios en redes ad hoc de una forma colaborativa y dinámica. Esta arquitectura está estructurada en tres niveles: *gestión de red*, *descubrimiento de servicios* y *gestión del grafo*. Tanto el protocolo propuesto para el descubrimiento de servicios como el de composición son protocolos *proactivos* y se basan, por lo tanto, en la detección de los cambios en la topología de la red. La capa de *gestión de red* proporciona los mecanismos necesarios para la *detección de vecinos* y la *difusión confiable*, aspectos ambos necesarios para la solución propuesta.
  - Se introducen también las características de la ontología de conceptos que se usa para la anotación de los distintos parámetros de los servicios. La utilización de una ontología de conceptos para la descripción de los parámetros de los servicios beneficia los procesos de búsqueda de servicios en la red ad hoc, tal y como se deduce de los experimentos realizados en los capítulo 5 y capítulo 6.

- **Capítulo 5:** que contiene la propuesta de descubrimiento de servicios para redes MANET, con las siguientes aportaciones:
  - Definición del proceso de diseminación de la información sobre los parámetros de entrada y salida a través de la red ad hoc, utilizando para ello las relaciones entre los tipos de los parámetros con objeto de agrupar la información y reducir el número de mensajes.
  - Mecanismo de búsqueda de servicios, basado en el tipo de los parámetros de entrada y salida, que utiliza la información diseminada para aplicar un proceso de *poda*. Este proceso se aplica durante la propagación de los mensajes de búsqueda, reduciendo así el número de transmisiones.
  - Integración del proceso de diseminación y búsqueda con el mecanismo de encaminamiento y gestión de rutas entre los nodos. El descubrimiento de un servicio crea las rutas de comunicación entre *proveedor* y *cliente*, permitiendo el posterior envío de mensajes entre ellos. La integración de los diferentes mecanismos permite reducir el número de mensajes enviados en el proceso.
  - Evaluación de la solución propuesta mediante la utilización de un simulador de red y la construcción de un escenario de prueba con unas características adecuadas. Los resultados muestran los procesos de *agrupamiento* y *poda* de los mensajes durante la difusión y búsqueda, respectivamente. Se puede ver que las propuestas realizadas se traducen en una reducción de los mensajes de diseminación y búsqueda gracias a la aplicación de la información contenida en la ontología de conceptos.
- **Capítulo 6:** que trata sobre la propuesta realizada para la composición de servicios en redes MANET y cuyas aportaciones principales son:
  - Definición del proceso de detección de colisiones entre parámetros de entrada y salida que permite llevar a cabo la creación del grafo distribuido de servicios (GDS). Este proceso lleva a cabo la detección de dependencias entre los diferentes servicios distribuidos en la red. La colisión entre parámetros se realiza aplicando el proceso de diseminación definido en el protocolo de descubrimiento.
  - Mecanismo de creación y mantenimiento del GDS frente a los cambios que se producen durante la movilidad de los nodos en la red ad hoc. Cuando un nodo detecta una colisión, entre dos o más parámetros, inicia un proceso de búsqueda utilizando la funcionalidad de búsqueda de servicios proporcionada por el protocolo de descubrimiento. Se crean así las rutas de comunicación que permitirán

posteriormente utilizar el GDS durante la realización del proceso de búsqueda de composiciones.

- Definición del proceso de búsqueda de composiciones adecuado a las características del GDS y a la naturaleza de las redes móviles ad hoc. La estructura distribuida del grafo de servicios es utilizada para llevar a cabo una *búsqueda hacia delante* que permite localizar las composiciones. La especificación de las composiciones a buscar se realiza mediante la definición del servicio compuesto a localizar. Se evita así la necesidad de construir un *flujo de trabajo* que describa de forma abstracta el servicio requerido, mejorando de esta forma la automatización del proceso.
- Evaluación de la solución propuesta utilizando un simulador de red mediante su comparación con otra solución de composición de servicios de naturaleza *reactiva* y basada en la *inundación* de la red. Esta comparación permite observar cómo la introducción del GDS resulta en una mejora al reducir el tiempo de búsqueda de composiciones. Sin embargo, la misma evaluación demuestra también que el nuevo mecanismo de composición propuesto tiene sus limitaciones. La solución no escala bien cuando la movilidad de los nodos aumenta, o cuando existe un mayor número de los mismos en un área. Esto se debe a la sobrecarga de comunicación introducida por los mensajes relacionados con los procesos de construcción y mantenimiento del grafo distribuido de servicios. Todos estos aspectos son razonados con mayor profundidad en la sección 6.5 del capítulo correspondiente.

Además, las soluciones propuestas en esta investigación han sido implementadas de forma completa utilizando el lenguaje de programación Java, así como evaluadas utilizando dicha implementación de forma directa mediante el simulador de red ns-2.

La mayoría de las soluciones estudiadas en el capítulo 3, para el descubrimiento y la composición de servicios en redes ad hoc, se basan en la realización de pruebas mediante modelos o en la implementación parcial de las mismas, utilizando en muchas ocasiones lenguajes propios de la herramienta de simulación correspondiente. Sin embargo, en el caso de esta investigación, la utilización de una implementación completa y real demuestra que el protocolo es realizable en la práctica y que podría llegar a ser utilizado en dispositivos reales sin la necesidad de aplicar otras modificaciones.



## 7.2 Líneas abiertas de investigación

Se recogen en esta sección aquellas posibles líneas de trabajo futuras, tanto para aspectos relacionados con el descubrimiento de servicios como para su composición en redes móviles ad hoc, que no han sido explorados durante la realización de la investigación en esta tesis doctoral.

### 7.2.1 Descubrimiento de servicios

En el caso del protocolo propuesto para el descubrimiento de servicios en redes móviles ad hoc se proponen las siguientes líneas que pueden dar lugar a posibles investigaciones futuras:

- *Mecanismo para la difusión inicial de la ontología de conceptos utilizada por los nodos.* En la solución actual, la ontología utilizada para determinar la compatibilidad entre los diferentes tipos de datos debe ser conocida *a priori* por los diferentes nodos que participan en la red ad hoc. Esto puede complicar el proceso de configuración de los nodos y hacerlo menos flexible. Hay que tener en cuenta que los dispositivos pueden conectarse a la red ad hoc de una forma espontánea y, por lo tanto, es necesario reducir la participación del usuario en dicho proceso. Una posible solución a este problema es que, tras la aparición de un nuevo nodo en la red, sus vecinos le proporcionen la ontología utilizada. Sin embargo, en dicho caso sería necesario tener en cuenta aquellas situaciones en las que existen varias ontologías disponibles en la red y que, por lo tanto, es necesario aplicar una determinada selección de aquella que sea más adecuada al dominio de aplicación de cada búsqueda de servicios.
- *Selección de servicios.* En la propuesta de descubrimiento de servicios planteada no se han tenido en cuenta procesos de selección más allá de la búsqueda por tipo de parámetros. Podría ser interesante introducir otros mecanismos de selección llevados a cabo durante la búsqueda, por ejemplo, a través de aspectos *no funcionales* del servicio (precio, calidad, propietario, etc.). El protocolo de descubrimiento propuesto podría ser modificado para llevar a cabo un filtrado que permitiese aplicar las restricciones para la selección del servicio más adecuado. Estas restricciones podrían ser aplicadas tanto por parte del cliente, cuando recibe las respuestas, como por parte de otros nodos que participan en la búsqueda, durante el proceso de propagación de la misma.
- *Generalización del protocolo de descubrimiento propuesto.* El descubrimiento de servicios presentado podría llegar a ser generalizado para ser utilizado, no solamente para la disseminación y descubrimiento de servicios

basado en los tipos de los parámetros de entrada/salida, sino también para cualquier recurso cuya información pueda ser categorizada mediante una ontología de conceptos. Los procesos de agrupamiento de mensajes por parámetros podrían ser aplicados a los nuevos elementos para la descripción de recursos. Por otro lado, el proceso de *poda* de mensajes de búsqueda podría llegar a ser utilizado de una forma similar a la aquí expuesta, pero teniendo en cuenta las condiciones para su generalización.

### 7.2.2 Composición de servicios

Durante la realización de la propuesta para la composición de servicios se han identificado también posibles líneas de investigación futuras que pueden dar como resultado nuevas aportaciones al trabajo presentado:

- *Realización de diferentes tipos de búsquedas en el grafo distribuido de servicios.* El mecanismo de creación del grafo propuesto en esta investigación puede permitir la creación de enlaces entre servicios en ambas direcciones. En el trabajo presentado únicamente se han utilizado conexiones entre un servicio y sus *sucesores*, sin embargo, es posible también crear conexiones inversas entre un servicio y sus *antecesores*. Usando estas rutas inversas el protocolo de composición podría ser mejorado mediante la realización de búsquedas *hacia atrás* y *bidireccionales*. En el caso de una búsqueda hacia atrás, ésta empezaría desde el servicio  $S_{FIN}$  de una composición buscada e intentaría encontrar caminos inversos hasta el servicio de inicio  $S_{INICIO}$ . Sería necesario comparar si este tipo de búsqueda supone una mejora con respecto a la búsqueda hacia delante. Por último, la búsqueda hacia delante y hacia atrás podrían combinarse para dar lugar a una búsqueda bidireccional que podría posibilitar la búsqueda de servicios en la red de manera simultánea tanto desde el servicio de inicio como de fin. Esto podría suponer una mayor reducción del tiempo de búsqueda debido a la naturaleza paralela de la red ad hoc.
- *Selección entre múltiples composiciones disponibles.* Las composiciones de servicios encontradas con el mecanismo propuesto solamente tienen en cuenta aspectos funcionales, basados en la compatibilidad de servicios, para determinar si una composición satisface o no cierta solicitud del usuario. El algoritmo podría ser mejorado utilizando otros aspectos de la composición resultante, de tal forma que durante el proceso de búsqueda en el grafo se aplicasen restricciones que permitiesen limitar las posibles soluciones recibidas por el nodo solicitante. Estas restricciones deberían estar incluidas en la solicitud de búsqueda de composiciones y podrían ser aplicadas tanto por el nodo solicitante, al recibir las soluciones al

problema de composición, como por los nodos intermedios durante el proceso de búsqueda en el grafo.

Existen aspectos sobre la validez del servicio compuesto que solamente pueden comprobarse en su totalidad una vez que se dispone de la solución completa. Por ejemplo, comprobar que las precondiciones y efectos del servicio completo no den lugar a inconsistencias lógicas requiere comprobar el flujo completo de ejecución del servicio obtenido. Pueden llevarse a cabo comprobaciones parciales del grafo pero, debido nuevamente a la distribución de la información sobre los servicios a través de la red, se requeriría del intercambio de más información entre los nodos. Como solución al problema puede proponerse realizar un pre-cálculo para ciertos subgrafos de servicios dentro de la red de estos aspectos. Este pre-cálculo podría permitir determinar, sin la necesidad de recorrer todo el grafo distribuido, si la solución actual satisface las restricciones requeridas y detener la búsqueda en caso contrario. Sin embargo, este proceso requeriría el intercambio de mensajes entre servicios que no se encuentran directamente conectados y podría tener un impacto en el número de mensajes enviados, así como en la congestión de la red.

- *Ejecución de los servicios compuestos.* La investigación realizada en esta tesis doctoral no tiene en cuenta cómo se lleva a cabo la ejecución de una composición resultante. Tal y como se recoge en el estado actual de la composición de servicios en redes móviles ad hoc, en la sección 3.2.8, existen diferentes mecanismos para llevar a cabo la ejecución de un servicio compuesto. La solución más básica consiste en utilizar la información del grafo de composición obtenido para propagar mensajes desde el servicio inicial hacia sus sucesores, obteniendo en cada paso nuevas salidas que son enviadas a través de los caminos existentes en la red, hasta completar la ejecución del grafo completo. Sin embargo, este proceso no es tan sencillo, ya que se pueden producir fallos durante la ejecución del servicio, que podrían dejar el conjunto de la red en un estado inconsistente y, por lo tanto, hacer necesaria la aplicación de mecanismos de reparación.
- *Implementación y prueba con dispositivos reales.* Como se ha comentado en las secciones relativas a la evaluación de las diferentes propuestas realizadas, el proceso ha sido llevado a cabo mediante la utilización de un simulador de red. Sin embargo, toda la implementación se ha desarrollado en un lenguaje que puede ser transportado a dispositivos reales que cuenten con tecnología Java. En el momento actual, se ha optado por presentar los resultados obtenidos con la simulación que permiten una mayor facilidad a la hora de crear escenarios de prueba con características variables. Se deja como trabajo futuro la posibilidad de validar la arquitectura propuesta con dispositivos reales.



# Validación

El desarrollo de protocolos para las redes móviles ad hoc es un proceso complejo que debe ser validado para que los resultados obtenidos sean correctos. Esta complejidad es debida, por ejemplo, a que la movilidad de los nodos de la red resulta en la aparición de interrelaciones entre los nodos no tenidas en cuenta durante el diseño de la solución y que pueden dar lugar a comportamientos inesperados. Por otro lado, y aunque el diseño de un protocolo sea teóricamente correcto contemplando todas las posibilidades, es natural cometer fallos durante su implementación.

Los contratiempos anteriores son comunes a la resolución de cualquier problema de computación, sin embargo, su detección y resolución es más complicada en el caso de su aplicación en entornos distribuidos y asíncronos como las redes MANET. Con objeto de garantizar la validez de la solución presentada en esta tesis doctoral, se han llevado a cabo una serie de procesos que han permitido detectar errores, tanto durante las fases de diseño como las de implementación de los protocolos de descubrimiento y composición de servicios.

## A.1 Metodología de validación

La validación de la solución desarrollada durante la realización de la tesis doctoral, tanto para el descubrimiento de servicios explicado en el capítulo 5 como para la propuesta de composición del capítulo 6, se ha llevado a cabo mediante la realización de dos tipos de pruebas: unitarias y de comportamiento distribuido.

Las *pruebas unitarias* han tenido como finalidad comprobar la correcta funcionalidad de cada uno de los elementos que constituyen la implementación de

una forma autónoma. Por otro lado, las *pruebas de comportamiento distribuido* han sido realizadas para cada uno de los niveles de la arquitectura propuesta en el capítulo 4, y han permitido determinar que la implementación utilizada en la experimentación se comporta de acuerdo a su definición y diseño.

### A.1.1 Pruebas unitarias

Dado que la implementación de la solución propuesta ha sido llevada a cabo utilizando el lenguaje de programación Java, la forma natural de realizar las pruebas unitarias ha sido la utilización de JUnit (JUnit.org Community, 2011), un *framework* para la realización de este tipo de pruebas de una forma automática en dicho lenguaje. Las pruebas unitarias han permitido comprobar la funcionalidad de los distintos elementos (*clases*) que constituyen la implementación. En concreto, se ha tenido especial cuidado para comprobar la funcionalidad de aquellos aspectos fundamentales y cuya complejidad puede introducir mayor cantidad de problemas en durante el desarrollo.

Los aspectos de la funcionalidad implementada que han sido comprobados con mayor detalle durante la realización de las pruebas unitarias han sido los siguientes:

- **Serialización de los mensajes:** los mensajes enviados por las capas que constituyen la solución son convertidos a una secuencia de *bytes* para su envío a través de la red. Con la aplicación de las pruebas unitarias se ha comprobado que los datos enviados son convertidos y enviados correctamente a través de la red y, posteriormente, son recibidos y recuperados de forma correcta por el nodo receptor.
- **Tabla de parámetros:** se ha comprobado la funcionalidad de la tabla de parámetros para determinar si reacciona de forma correcta a las distintas situaciones que pueden suceder: actualización de su contenido tras recibir un mensaje, obtención de la distancia estimada, generación de los mensajes de actualización para los vecinos y gestión de los grupos de parámetros. Estas operaciones han sido definidas en la sección 5.2.
- **Tabla de encaminamiento:** en este caso se ha comprobado que la tabla de encaminamiento de mensajes funciona de forma correcta frente a las operaciones que pueden surgir durante el tiempo de vida de la red, y que han sido definidas en la sección 5.4. Las funcionalidades comprobadas incluyen la creación y eliminación de rutas mediante la recepción del mensaje correspondiente, así como la obtención de las distintas rutas disponibles para alcanzar un determinado nodo de la red.
- **Detección de colisiones:** la detección de colisiones entre los parámetros de la tabla de un nodo es un proceso complejo, como puede observarse

en la sección 6.3. Mediante la aplicación de pruebas unitarias a los elementos responsables de la realización de la detección, se ha comprobado que éstas responden de forma correcta a diferentes situaciones: cálculo de colisiones como resultado de la recepción de nuevos parámetros, actualización de conexiones entre servicios tras la recepción de mensajes de respuesta y eliminación de colisiones inválidas cuando se modifican los parámetros de la tabla de un nodo.

- **Gestión del grafo de servicios:** el grafo distribuido de servicios contiene información sobre las conexiones existentes entre los aquellos provistos por los nodos de la red, como puede verse en la sección 6.2 y, como se explica en la sección 6.4, permite además la búsqueda de composiciones. Con la aplicación de pruebas unitarias, se ha comprobado que los parámetros y los servicios crean las aristas de interconexión entre los nodos de una forma correcta y que, además, se permite obtener dicha información para navegar de forma adecuada.

### A.1.2 Pruebas de comportamiento

La realización únicamente de pruebas unitarias no garantiza que la implementación de la solución sea correcta, principalmente en un entorno distribuido como el de las redes MANET. En este tipo de entornos pueden producirse situaciones no previstas en el diseño que produce fallos en el funcionamiento.

Se han realizado, por lo tanto, pruebas cuya finalidad ha sido determinar el correcto funcionamiento de la arquitectura frente a situaciones conocidas de la red. Cada una de las capas de la arquitectura definida en la sección 4.3 ha sido comprobada de forma independiente, teniendo siempre en cuenta que las capas superiores hacen uso de la funcionalidad proporcionada por las inferiores y que, por lo tanto, los errores se propagan hacia arriba. Esto significa que ha sido necesario determinar primeramente la correcta funcionalidad de un nivel de la arquitectura antes de poder asegurar el funcionamiento de los superiores.

El proceso para comprobar el comportamiento de una capa de la arquitectura ha sido el siguiente:

1. *Identificación de una o varias situaciones de prueba.* Para ello se han tenido en cuenta los distintos comportamientos que puede tener la capa correspondiente frente a diferentes situaciones de la red. El objetivo de este proceso ha sido obtener una o más situaciones de configuración a partir de las cuales se producen situaciones especiales a comprobar.
2. *Creación de redes de prueba y su configuración.* Se han creado redes con un número reducido de nodos que representan cada una de las situaciones identificadas. El objetivo es tener un representación de las situaciones que

pueda ser comprobada manualmente si existen fallos, y poder resolverlos de una forma más sencilla.

3. *Definición de los resultados esperados.* Cada situación de prueba debe ir acompañada de unos resultados esperados que puedan ser comprobados fácilmente. Estos resultados pueden ser, por ejemplo, el contenido de las tablas de parámetros, las tablas de encaminamiento, el grafo contenido por cada nodo de la red o las composiciones encontradas.
4. *Ejecución de las simulaciones de prueba.* Las simulaciones de prueba son puestas en funcionamiento para comprobar que los resultados obtenidos se ajustan a los esperados.

Las pruebas de comportamiento han sido implementadas de tal forma que pueden ser llevadas a cabo de una manera completamente automática. Se han creado *scripts* de configuración de la red y de inicio del simulador que permiten realizar las pruebas cada vez que se realiza algún cambio en la implementación de los protocolos. Los *scripts* de prueba se encargan de configurar y poner en marcha el simulador para cada uno de los comportamientos establecidos.

Al finalizar la simulación de prueba, los nodos escriben en distintos ficheros los estados de configuración en los que se encuentran, que son comparados con los esperados para cada una de las pruebas. En el caso de que, los resultados obtenidos de una simulación de pruebas, no coincidan exactamente con los esperados, se llevan a cabo los procedimientos de notificación correspondientes que pueden facilitar la detección del problema.

Un aspecto considerado, durante la realización de las pruebas de validación del comportamiento, es que se ha visto que existen problemas que dependen de situaciones de sincronización. Estas situaciones se producen debido a pequeños cambios en los aspectos aleatorios generados por el simulador, por ejemplo, en el orden de procesamiento de los eventos debido a la variación en la secuencia de los mensajes enviados. Por lo tanto, con el fin de detectar estos problemas que se pueden producir de forma esporádica, se han creado *scripts* que realizan la misma prueba de comportamiento un cierto número de veces (50 en el caso de las pruebas realizadas).

La repetición de la misma prueba permite que se puedan reproducir esas situaciones excepcionales en el procesado de los mensajes. Una vez que se ha detectado una situación no contemplada en las pruebas, y que produce un resultado inesperado, se ha llevado a cabo la creación de una prueba específica para dicha situación y se han aplicado las variaciones en el diseño e implementación necesarias para resolverla.

La finalización de las pruebas de comportamiento sin que se detecten errores no garantiza la total ausencia de estos, ya sea por una implementación incorrecta o por un diseño incompleto. Sin embargo, sí que permite obtener



cierta seguridad sobre la comprobación de aquellos casos que han sido tenidos en cuenta durante el desarrollo de la propuesta. Además, ha permitido durante la realización de la investigación, detectar y refinar situaciones que de otra forma hubieran sido imposibles de tener en cuenta.

En la tabla A.1 y tabla A.2 se recogen los aspectos que han sido comprobados mediante este procedimiento para cada una de las diferentes capas de la arquitectura propuesta.

## A.2 Entorno de simulación y experimentación

Como se ha indicado en la sección 2.2, se ha seleccionado la herramienta ns-2 para llevar a cabo la simulación de las redes MANET, tanto en los procesos de validación como de evaluación de las soluciones propuestas. En esta herramienta la configuración de la red y el comportamiento de los nodos durante la simulación se especifican mediante la creación de *scripts* en el lenguaje Tcl.

Para facilitar el proceso de creación de configuraciones y lanzamiento automático del entorno de simulación se ha desarrollado un conjunto de herramientas utilizando el lenguaje de programación Python. La entrada de los scripts de simulación son ficheros XML que especifican aspectos como el número de nodos, tamaño del área de simulación, comportamiento de los participantes, etc.

Además, se han creado herramientas para su uso durante la experimentación que permiten generar, a partir de una configuración inicial, diferentes variaciones que modifican un único parámetro de la misma. Esta característica se ha utilizado durante la realización de las evaluaciones llevadas a cabo en el capítulo 5 y capítulo 6 para variar, por ejemplo, el número de nodos, la frecuencia de búsqueda de composiciones, el número de servicios, etc.

Por otro lado, se han creado también utilidades para la generación automática de gráficas a partir de los resultados obtenidos en cada experimentación realizada. La salida de un experimento son *ficheros de trazas* con todos los eventos que se han producido durante cada simulación. Cada línea del fichero de traza contiene el instante y el evento ocurrido: envío o recepción de un tipo de mensaje, inicio de una búsqueda, obtención del resultado, etc. Estos ficheros de trazas son procesados para obtener medidas concretas: número de mensajes enviados, porcentaje de composiciones encontradas, etc. Por último, a partir de estos resultados, se lleva a cabo un proceso de generación de gráficas de forma automática utilizando para ello la librería *matplotlib* (Hunter, 2012).

Tabla A.1: Pruebas de comportamiento. Parte I

Capa de la arquitectura	Pruebas aplicadas
Detección de vecinos	Aparición de un vecino Desaparición de un vecino Creación de un bucle
Diseminación de parámetros	Diseminación sencilla de parámetros Diseminación en un bucle de la red Aparición de un nuevo parámetro en un nodo Conexión de una nueva subred Eliminación de un parámetro existente en un nodo Rotura de la red tras una diseminación de parámetros Diseminación de parámetros con otros compatibles en la red Aplicación de los casos anteriores usando una ontología de conceptos
Encaminamiento	Búsqueda simple de parámetros Búsqueda descartada mediante el proceso de <i>poda</i> Conexión y desconexión de una subred Aparición y desaparición de un parámetro en un nodo Cancelación de una búsqueda Aplicación de los casos anteriores usando una ontología de conceptos

Tabla A.2: Pruebas de comportamiento. Parte II

Capa de la arquitectura	Prueba aplicadas
Creación del grafo	<p>Aparición y desaparición de servicios en los nodos</p> <p>Detección de colisiones entre servicios distribuidos</p> <p>Colisiones entre múltiples parámetros compatibles</p> <p>Detección local de colisiones</p> <p>Colisión detectada en el vecino de un nodo</p> <p>Creación del grafo distribuido de servicios en los casos anteriores</p> <p>Conexión entre múltiples servicios compatibles</p> <p>Rotura de la red y detección de las conexiones afectadas</p> <p>Aplicación de los casos anteriores usando una ontología de conceptos</p>
Búsqueda en el grafo	<p>Búsqueda de composición simple en el GDS</p> <p>Búsqueda con caminos alternativos</p> <p>Búsqueda con solución retardada por la aparición posterior de un servicio compatible</p> <p>Expiración de una búsqueda de composición</p>



## Lista de publicaciones

UNAI AGUILERA, DIEGO LÓPEZ-DE-IPÍÑA. Service Composition for Mobile Ad Hoc Networks Using Distributed Matching. 6th International Conference, UCAmI 2012, Vitoria-Gasteiz, España, 2012. Proceedings. LNCS 7656. Ubiquitous Computing and Ambient Intelligence. páginas 290-297. Springer-Verlag. ISBN: 978-3-642-35376-5

UNAI AGUILERA, DIEGO LÓPEZ-DE-IPÍÑA. A Parameter-Based service discovery protocol for mobile ad-hoc networks. 11th International Conference, ADHOC-NOW 2012, Belgrado, Serbia, 2012. Proceedings. LNCS 7363 Ad-hoc, Mobile, and Wireless Networks. páginas 274–287, Springer-Verlag. ISBN: 978-3-642-31637-1

UNAI AGUILERA, AITOR ALMEIDA, PABLO ORDUÑA, DIEGO LÓPEZ-DE-IPÍÑA, RAFAEL DE LAS HERAS. Continuous service execution in mobile prosumer environments. *Actas del IV International Symposium of Ubiquitous Computing and Ambient Intelligence, UCAmI 2010*, páginas 229-238, 2010, Valencia, España. ISBN: 978-84-92812-61-5

UNAI AGUILERA, JOSEBA ABAITUA, JOSUKA DÍAZ, DAVID BUJÁN, DIEGO LÓPEZ-DE-IPÍÑA. A semantic matching algorithm for discovery in UDDI. *ICSC '07 Proceedings of the International Conference on Semantic Computing*, páginas 751-758, 2007, Irvine, California, USA. ISBN:0-7695-2997-6

UNAI AGUILERA, JOSEBA ABAITUA, JOSUKA DÍAZ, DAVID BUJÁN, DIEGO LÓPEZ-DE-IPÍÑA. A Generic Matching Algorithm for Semantic Discovery. *Third International Conference on Semantics, Knowledge and Grid (SKG 2007)*, páginas 442-445, 2007, Xian, Shan Xi, China. ISBN: 0-7695-3007-9

### Otras publicaciones del autor

RAMÓN ALCARRIA, TOMÁS ROBLES, AUGUSTO MORALES, DIEGO LÓPEZ-DE-IPÍÑA, UNAI AGUILERA. Enabling Flexible and Continuous Capability Invocation in Mobile Prosumer Environments. *Sensors 2012 Journal* vol. 12, núm. 7, páginas 8930-8954, 2012.

MIKEL EMALDI, JON LÁZARO, UNAI AGUILERA, OSCAR PEÑA, DIEGO LÓPEZ DE IPÍÑA. Short Paper: Semantic Annotations for Sensor Open Data. *Proceedings of the 5th International Workshop on Semantic Sensor Networks* Boston, Massachusetts, USA, 2012.

RAMON ALCARRIA, UNAI AGUILERA, TOMÁS ROBLES, DIEGO LÓPEZ-DE-IPÍÑA, AUGUSTO MORALES. Ubiquitous Capability Access for Continuous Service Execution in Mobile Environments. *V International Symposium on Ubiquitous Computing and Ambient Intelligence (UCAmI'11)* Riviera Maya, México, 2011. ISBN: 978-84-694-9677-0

PABLO ORDUÑA, AITOR ALMEIDA, UNAI AGUILERA, XABIER LAISECA, DIEGO LÓPEZ-DE-IPÍÑA, AITOR GÓMEZ GOIRI. Identifying Security Issues in the Semantic Web: Injection attacks in the Semantic Query Languages. *Actas de las VI Jornadas Científico-Técnicas en Servicios Web y SOA (JSWEB 2010)* páginas 43 - 50, 2010, Valencia, España. ISBN: 978-84-92812-59-2

AITOR ALMEIDA, DIEGO LÓPEZ-DE-IPÍÑA, UNAI AGUILERA, IKER LARIZGOITIA, XABIER LAISECA, PABLO ORDUÑA, ANDER BARBIER. An Approach to Dynamic Knowledge Extension and Semantic Reasoning in Highly-Mutable Environments. *Proceedings of 3rd Symposium of Ubiquitous Computing and Ambient Intelligence 2008, Advances in Soft Computing*, vol. 51 2008, Salamanca, España. ISBN: 978-3-540-85866-9

DIEGO LÓPEZ-DE-IPÍÑA, AITOR ALMEIDA, UNAI AGUILERA, IKER LARIZGOITIA, XABIER LAISECA, PABLO ORDUÑA, ANDER BARBIER, JUAN IGNACIO VAZQUEZ. Dynamic Discovery and Semantic Reasoning for Next Generation Intelligent Environments. *The 4th IET international conference on Intelligent Environments IE08*, 2008, Washington, Seattle, USA. ISBN: 978-0-86341-894-5

AITOR ALMEIDA, DIEGO LÓPEZ-DE-IPÍÑA, UNAI AGUILERA, IKER LARIZGOITIA, XABIER LAISECA, PABLO ORDUÑA, ANDER BARBIER. Dynamic Ontology Enrichment and Reasoning in AmI Environments. *Proceedings of the 3rd Workshop on Artificial Intelligence Techniques for Ambient Intelligence (AITAmI2008)*, 2008, Patras, Grecia. ISBN: 978-960-6843-07-5





# Bibliografía

- ABOU EL SAOUD, M., KUNZ, T. y MAHMOUD, S. BENCEManet: an evaluation framework for service discovery protocols in MANET. En *Sensor and Ad Hoc Communications and Networks, 2006. SECON'06. 2006 3rd Annual IEEE Communications Society on*, vol. 3, páginas 860–865. IEEE, 2006. ISBN 1424406269. (ref. pág. 23)
- AGUILERA, U., ABAITUA, J., DIAZ, J., BUJAN, D. y LOPEZ DE IPINA, D. A semantic matching algorithm for discovery in UDDI. En *International Conference on Semantic Computing (ICSC 2007)*, vol. 1, páginas 751 – 758. IEEE Computer Society, Irvine, CA, USA, 2007. ISBN 0-7695-2997-6. (ref. pág. 34)
- AGUILERA, U., ALMEIDA, A., ORDUÑA, P., LÓPEZ-DE IPIÑA, D. y DE LAS HERAS, R. Continuous service execution in mobile prosumer environments. En *Proceedings of the 4th Symposium of Ubiquitous Computing and Ambient Intelligence*, páginas 229–238. 2010. (ref. pág. 59)
- AL MALLAH, R. y QUINTERO, A. A light-weight service discovery protocol for ad hoc networks. *Journal of Computer Science*, vol. 5(4), páginas 330–337, 2009. (ref. pág. 31)
- ANDEL, T. R. y YASINSAC, A. On the credibility of MANET simulations. *Computer*, vol. 39(7), páginas 48–54, 2006. (ref. pág. 18)
- ARIAS-TORRES, D. y GARCÍA-MACÍAS, J. Performance analysis of two approaches to service discovery in mobile ad hoc networks. En *Advanced Distributed Systems*, vol. 3563 de *Lecture Notes in Computer Science*, páginas 368–376. Springer Berlin / Heidelberg, 2005. ISBN 978-3-540-28063-7. (ref. pág. 33, 35, 36)
- ARTAIL, H., HAMZE, H., MERSHAD, K. y SAFA, H. A collaborative service discovery and service sharing framework for mobile ad hoc networks. *IFIP Lecture Notes in Computer Science (LNCS)*, vol. 4672(4672), páginas 151–160, 2011. (ref. pág. 31)

- ARTAIL, H., SAFA, H., HAMZE, H. y MERSHAD, K. A cluster based service discovery model for mobile ad hoc networks. En *Wireless and Mobile Computing, Networking and Communications, 2007. WiMOB 2007. Third IEEE International Conference on*, páginas 57–57. IEEE, 2007. ISBN 0769528899. (ref. pág. 31, 35, 36)
- BASU, P., KE, W. y LITTLE, T. D. C. Scalable service composition in mobile ad hoc networks using hierarchical task graphs. En *Proc. 1st Annual Mediterranean Ad Hoc Networking Workshop*. 2002. (ref. pág. 44, 46, 56)
- BERNERS-LEE, T., HENDLER, J. y LASSILA, O. The semantic web. *Scientific American*, páginas 29–37, 2001. (ref. pág. 2)
- BLOOM, B. H. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, vol. 13(7), páginas 422–426, 1970. (ref. pág. 30)
- BLUETOOTH SPECIAL INTEREST GROUP. Service discovery protocol (SDP). core system package [Host volume] part b. 2004. Disponible en <http://www.bluetooth.org> (último acceso, septiembre 2012). (ref. pág. 27)
- BRØNSTED, J., HANSEN, K. M. y INGSTRUP, M. Service composition issues in pervasive computing. *IEEE Pervasive Computing*, vol. 9(1), páginas 62–70, 2010. ISSN 1536-1268. (ref. pág. 39, 56)
- CAMP, T., BOLENG, J. y DAVIES, V. A survey of mobility models for ad hoc network research. *Wireless Communications and Mobile Computing*, vol. 2(5), página 483–502, 2002. ISSN 1530-8677. (ref. pág. 23, 116, 156)
- CAMPO, C., MUNOZ, M., PEREA, J. C., MANN, A. y GARCIA-RUBIO, C. PDP and GSDL: a new service discovery middleware to support spontaneous interactions in pervasive systems. En *Pervasive Computing and Communications Workshops, 2005. PerCom 2005 Workshops. Third IEEE International Conference on*, páginas 178–182. IEEE, 2005. ISBN 0769523005. (ref. pág. 32)
- CAREY, K., LEWIS, D., HIGEL, S. y WADE, V. Adaptive composite service plans for ubiquitous computing. En *2nd International Workshop on Managing Ubiquitous Communications and Services (MUCS2004)*, páginas 13–14. 2004. (ref. pág. 48, 49)
- CASATI, F., SAYAL, M., SHAN, M.-C., DITTRICH, K., GEPPERT, A. y NORRIE, M. Developing e-services for composing e-services. En *Advanced Information Systems Engineering*, vol. 2068 de *Lecture Notes in Computer Science*, páginas 171–186. Springer Berlin / Heidelberg, 2001. ISBN 978-3-540-42215-0. (ref. pág. 54)

- CASTELLS, P. Aplicaciones de técnicas de la web semántica. Workshop de investigación en entornos de interacción colectiva (COLINE'02), 2002. (ref. pág. 2)
- CAVIN, D., SASSON, Y. y SCHIPER, A. On the accuracy of MANET simulators. En *Proceedings of the second ACM international workshop on Principles of mobile computing*, páginas 38–43. ACM, 2002. ISBN 1581135114. (ref. pág. 18)
- CHAKRABORTY, D., JOSHI, A., FININ, T. y YESHA, Y. Service composition for mobile environments. *Mobile Network Applications*, vol. 10(4), páginas 435–451, 2005. ISSN 1383-469X. (ref. pág. 39)
- CHAKRABORTY, D., JOSHI, A., YESHA, Y. y FININ, T. Toward distributed service discovery in pervasive computing environments. *Mobile Computing, IEEE Transactions on*, vol. 5(2), páginas 97–112, 2006. (ref. pág. 5, 32, 33, 34, 47)
- CHAKRABORTY, D., YESHA, Y. y JOSHI, A. A distributed service composition protocol for pervasive environments. En *Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE*, vol. 4, páginas 2575–2580. IEEE, 2004. ISBN 0780383443. (ref. pág. 35, 42, 44, 48, 49, 50, 56)
- CHEN, W., HE, Z., REN, G. y SUN, W. Service recovery for composite service in MANETs. En *Wireless Communications, Networking and Mobile Computing, 2008. WiCOM'08. 4th International Conference on*, páginas 1–4. IEEE, 2008. ISBN 1424421071. (ref. pág. 50, 51)
- CHRISTENSEN, E., CURBERA, F., MEREDITH, G. y WEERAWARANA, S. Web services description language (WSDL) 1.1. W3C Note, 2001. Disponible en <http://www.w3.org/TR/wsdl> (último acceso, septiembre 2012). (ref. pág. 47)
- CLAUSEN, T., ADAMSON, B. y DEARLOVE, C. RFC 5148 - jitter considerations in mobile ad hoc networks (MANETs). IETF - Network Working Group, 2008. Disponible en <http://tools.ietf.org/html/rfc5148> (último acceso, septiembre 2012). (ref. pág. 78)
- CLAUSEN, T. y JACQUET, P. RFC 3626 - optimized link state routing protocol (OLSR). IETF - Network Working Group, 2003. Disponible en <http://www.ietf.org/rfc/rfc3626.txt> (último acceso, septiembre 2012). (ref. pág. 16)
- COLOBERTI, M., LOMBRISER, C., ROGGEN, D., TRÖSTER, G., GUARNERI, R. y RIBONI, D. Service discovery and composition in body area networks. En *Proceedings of the ICST 3rd international conference on Body area networks*, página 7. 2008. ISBN 9639799173. (ref. pág. 46)

- CONSTANTINESCU, I., FALTINGS, B. y BINDER, W. Type based service composition. En *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters, WWW Alt. '04*, páginas 268–269. ACM, 2004. ISBN 1-58113-912-8. (ref. pág. 56)
- CRESPO, A. y GARCIA-MOLINA, H. Routing indices for peer-to-peer systems. En *Distributed Computing Systems, 2002. Proceedings. 22nd International Conference on*, páginas 23–32. IEEE, 2002. ISBN 0769515851. (ref. pág. 32)
- CROW, B. P., WIDJAJA, I., KIM, L. G. y SAKAI, P. T. IEEE 802.11 wireless local area networks. *Communications Magazine, IEEE*, vol. 35(9), páginas 116–126, 1997. (ref. pág. 15)
- CUI, Y., LI, Z. y JIANG, S. Adaptive deployment of directory for service discovery in MANET. *Procedia Engineering*, vol. 15(0), páginas 2971–2975, 2011. (ref. pág. 32, 35)
- DABROWSKI, C., MILLS, K. y ELDER, J. Understanding consistency maintenance in service discovery architectures during communication failure. En *Proceedings of the 3rd international workshop on Software and performance, WOSP '02*, páginas 168–178. 2002. ISBN 1-58113-563-7. (ref. pág. 36)
- DEEPA, R. y SWAMYNATHAN, S. A service discovery model for mobile ad hoc networks. En *Recent Trends in Information, Telecommunication and Computing (ITC), 2010 International Conference on*, páginas 135–139. IEEE, 2010. ISBN 1424459567. (ref. pág. 31, 35)
- EDWARDS, W. K., NEWMAN, M. W., SECIVY, J. Z. y SMITH, T. F. Bringing network effects to pervasive spaces. *Pervasive Computing, IEEE*, vol. 4(3), páginas 15–17, 2005. (ref. pág. 42)
- ENGELSTAD, P. E. y ZHENG, Y. Evaluation of service discovery architectures for mobile ad hoc networks. En *Proceedings of the Second Annual Conference on Wireless On-demand Network Systems and Services, WONS '05*, página 2–15. IEEE Computer Society, 2005. ISBN 0-7695-2290-0. (ref. pág. 38)
- FIKES, R. E. y NILSSON, N. J. STRIPS: a new approach to the application of theorem proving to problem solving. *Artificial intelligence*, vol. 2(3-4), páginas 189–208, 1972. (ref. pág. 44)
- FORGY, C. L. Rete: A fast algorithm for the many pattern/many object pattern match problem. *Artificial Intelligence*, vol. 19(1), páginas 17 – 37, 1982. (ref. pág. 44)

- GAO, Z., WANG, L., YANG, M. y YANG, X. CNPGSDP: an efficient group-based service discovery protocol for MANETs. *Computer Networks*, vol. 50(16), páginas 3165–3182, 2006. (ref. pág. 33)
- GARCIA-MACIAS, J. y TORRES, D. Service discovery in mobile ad-hoc networks: better at the network layer? En *Parallel Processing, 2005. ICPP 2005 Workshops. International Conference Workshops on*, páginas 452 – 457. 2005. ISSN 1530-2016. (ref. pág. 36)
- GHAZOU LI, M. y BOUFAIDA, M. A generic P2P collaborative strategy for discovering and composing semantic web services. En *Proceedings of the 2009 Fourth International Conference on Internet and Web Applications and Services*, páginas 449–454. IEEE Computer Society, 2009. ISBN 978-0-7695-3613-2. (ref. pág. 43)
- GIRUKA, V. C. y SINGHAL, M. Hello protocols for ad-hoc networks: Overhead and accuracy tradeoffs. En *Proceedings of the Sixth IEEE International Symposium on World of Wireless Mobile and Multimedia Networks, WOWMOM '05*, páginas 354–361. IEEE Computer Society, 2005. ISBN 0-7695-2342-0-01. (ref. pág. 75)
- GRIMM, R. One. world: Experiences with a pervasive computing architecture. *Pervasive Computing, IEEE*, vol. 3(3), páginas 22–30, 2004. (ref. pág. 42)
- GROBA, C. Towards opportunistic service composition in dynamic ad hoc environments. En *Service-Oriented Computing - ICSOC 2011 Workshops*, vol. 7221 de *Lecture Notes in Computer Science*, páginas 189–194. Springer Berlin / Heidelberg, 2012. ISBN 978-3-642-31874-0. (ref. pág. 46)
- GRUNINGER, M., BODENREIDER, O., OLKEN, F., OBRST, L. y YIM, P. Ontology summit 2007–Ontology, taxonomy, folksonomy: Understanding the distinctions. *Applied Ontology*, vol. 3(3), páginas 191–200, 2008. ISSN 1875-8533. (ref. pág. 71)
- GU, X., NAHRSTEDT, K. y YU, B. SpiderNet: an integrated peer-to-peer service composition framework. En *High performance Distributed Computing, 2004. Proceedings. 13th IEEE International Symposium on*, páginas 110–119. 2004. ISBN 1082-8907. (ref. pág. 4, 43, 45, 49, 50)
- GU, Z., LI, J. y XU, B. Automatic service composition based on enhanced service dependency graph. En *Proceedings of the 2008 IEEE International Conference on Web Services*, páginas 246–253. IEEE Computer Society, 2008. (ref. pág. 4)

- GUTTMAN, E. y PERKINS, C. RFC 2608 - service location protocol, version 2. IETF - Network Working Group, 1999. Disponible en <http://tools.ietf.org/html/rfc2608> (último acceso, septiembre 2012). (ref. pág. 27)
- HAAS, Z. J., PEARLMAN, M. R. y SAMAR, P. The zone routing protocol (ZRP) for ad hoc networks. IETF Internet Draft, 2002. Disponible en <http://tools.ietf.org/id/draft-ietf-manet-zone-zrp-04.txt> (último acceso, septiembre 2012). (ref. pág. 16, 35)
- HAN, S., ZHANG, S., LI, G. y ZHANG, Y. An on-demand QoS service composition protocol for MANETs. *IEICE - Transactions on Information and Systems*, vol. E90-D(11), páginas 1877–1880, 2007. (ref. pág. 48, 49, 50)
- HAN, S. y ZHANG, Y. Design and implementation of service composition protocol based on DSR. En *Parallel and Distributed Computing Applications and Technologies, International Conference on*, vol. 0, páginas 323–328. IEEE Computer Society, 2010. ISBN 978-0-7695-4287-4. (ref. pág. 45, 48, 49, 50)
- HEISSENBÜTTEL, M., BRAUN, T., WÄLCHLI, M. y BERNOULLI, T. Evaluating the limitations of and alternatives in beaconing. *Ad Hoc Networks*, vol. 5(5), páginas 558–578, 2007. ISSN 1570-8705. (ref. pág. 75)
- HELAL, S., DESAI, N., VERMA, V. y LEE, C. Konark-a service discovery and delivery protocol for ad-hoc networks. En *Wireless Communications and Networking*, vol. 3, páginas 2107–2113. IEEE, 2003. ISBN 0780377001. (ref. pág. 35)
- HENI, M. y BOUALLEGUE, R. Adaptive service discovery and proactive routing protocol for mobile ad hoc network. En *Mediterranean Microwave Symposium (MMS), 2011 11th*, páginas 193–196. IEEE, 2011. ISBN 1457718146. (ref. pág. 36)
- HERBORN, S. y SENEVIRATNE, A. Service composition for mobile personal networks. En *Mobile and Ubiquitous Systems: Networking & Services, 2006 Third Annual International Conference on*, páginas 1–8. 2006. (ref. pág. 4, 45, 49, 50)
- HESSELMAN, C., TOKMAKOFF, A., PAWAR, P. y IACOB, S. Discovery and composition of services for context-aware systems. En *Smart Sensing and Context*, vol. 4272 de *Lecture Notes in Computer Science*, páginas 67–81. Springer Berlin Heidelberg, 2006. ISBN 978-3-540-47842-3. (ref. pág. 46)
- HU, S., MUTHUSAMY, V., LI, G. y JACOBSEN, H. Distributed automatic service composition in large-scale systems. En *Proceedings of the second international conference on Distributed event-based systems*, páginas 233–244. ACM, 2008. (ref. pág. 4, 45, 47, 49, 50, 60)

- HU, Z., TANG, X., WANG, X. y JI, Y. A distributed algorithm for DAG-Form service composition over MANET. En *Wireless Communications, Networking and Mobile Computing, 2007. WiCom 2007. International Conference on*, páginas 1664–1667. 2007. (ref. pág. 4, 45, 48, 49, 50, 56)
- HUANG, J., BAI, Y., ZHANG, Z., KONG, J. y QIAN, D. Service forest: Enabling dynamic service composition in mobile ad hoc networks. En *Proceedings of the The 2007 International Conference on Intelligent Pervasive Computing*, páginas 174–177. IEEE Computer Society, 2007. (ref. pág. 4, 45, 46, 56)
- HUNTER, J. matplotlib: python plotting. 2012. Disponible en <http://http://matplotlib.org/> (último acceso, septiembre 2012). (ref. pág. 177)
- HWANG, C., TALIPOV, E. y CHA, H. Distributed geographic service discovery for mobile sensor networks. *Computer Networks*, vol. 55(5), páginas 1069–1082, 2011. (ref. pág. 31)
- ISLAM, N. y SHAIKH, Z. A. Towards a robust and scalable semantic service discovery scheme for mobile ad hoc network. *Pak. J. Engg. & Appl. Sci. Vol.*, vol. 10, páginas 68–88, 2012. (ref. pág. 5, 34)
- JIANG, S., XUE, Y. y SCHMIDT, D. C. Minimum disruption service composition and recovery in mobile ad hoc networks. *Computer Networks*, vol. 53(10), páginas 1649–1665, 2009. (ref. pág. 50, 51)
- JOHNSON, D. B., MALTZ, D. A. y BROCH, J. DSR: the dynamic source routing protocol for multi-hop wireless ad hoc networks. *Ad hoc Networking*, vol. 5, páginas 139–172, 2001. (ref. pág. 16)
- JUNIT.ORG COMMUNITY. Resources for test driven development. 2011. Disponible en <http://junit.org> (último acceso, septiembre 2012). (ref. pág. 174)
- KALOFONOS, D. N. y WISNER, P. A framework for end-user programming of smart homes using mobile devices. En *Consumer Communications and Networking Conference, 2007. CCNC 2007. 4th IEEE*, páginas 716 –721. 2007. (ref. pág. 44, 47, 48)
- KANG, E., IM, Y. y KIM, U. Adaptive neighbor selection for service discovery in mobile ad hoc networks. En *Network and Parallel Computing*, vol. 5245 de *Lecture Notes in Computer Science*, páginas 13–23. Springer Berlin / Heidelberg, 2008. ISBN 978-3-540-88139-1. (ref. pág. 33)
- KARMOUCH, E. y NAYAK, A. Capability reconciliation for virtual device composition in mobile ad hoc networks. En *Wireless and Mobile Computing, Networking and Communications (WiMob), 2010 IEEE 6th International Conference on*, páginas 27–34. 2010. (ref. pág. 44, 46, 56)

- KLEIN, M., KONIG-RIES, B. y OBREITER, P. Service rings - a semantic overlay for service discovery in ad hoc networks. En *Database and Expert Systems Applications, 2003. Proceedings. 14th International Workshop on*, páginas 180–185. IEEE, 2003. ISBN 0769519938. (ref. pág. 5, 31, 34, 35, 36)
- KLEIN, M. y KÖNIG-RIES, B. Multi-layer clusters in ad-hoc networks — an approach to service discovery. En *Web Engineering and Peer-to-Peer Computing*, vol. 2376 de *Lecture Notes in Computer Science*, páginas 187–201. Springer Berlin / Heidelberg, 2002. ISBN 978-3-540-44177-9. (ref. pág. 31)
- KOZAT, U. C. y TASSIULAS, L. Network layer support for service discovery in mobile ad hoc networks. En *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, vol. 3, páginas 1965–1975. 2003. ISBN 0780377524. (ref. pág. 30)
- KURKOWSKI, S., NAVIDI, W. y CAMP, T. Constructing MANET simulation scenarios that meet standards. En *Mobile Adhoc and Sensor Systems, 2007. MASS 2007. IEEE International Conference on*, páginas 1–9. 2007a. (ref. pág. 22, 116, 156)
- KURKOWSKI, S., NAVIDI, W. y CAMP, T. Discovering variables that affect MANET protocol performance. En *Global Telecommunications Conference, 2007. GLOBECOM '07. IEEE*, páginas 1237–1242. 2007b. (ref. pág. 21)
- LEE, C., KO, S., LEE, S., LEE, W. y HELAL, S. Context-aware service composition for mobile network environments. En *Ubiquitous Intelligence and Computing*, vol. 4611 de *Lecture Notes in Computer Science*, páginas 941–952. Springer Berlin Heidelberg, 2007. ISBN 978-3-540-73548-9. (ref. pág. 46, 47)
- LENDERS, V., MAY, M. y PLATTNER, B. Service discovery in mobile ad hoc networks: A field theoretic approach. *Pervasive and Mobile Computing*, vol. 1(3), páginas 343–370, 2005. ISSN 1574-1192. (ref. pág. 33, 37)
- LEVESQUE, H., REITER, R., LESPERANCE, Y., LIN, F. y SCHERL, R. B. GOLOG: A logic programming language for dynamic domains. *Journal of Logic Programming*, vol. 31(1-3), páginas 59–83, 1997. (ref. pág. 44)
- LI, J., BLAKE, C., DE COUTO, D. S., LEE, H. I. y MORRIS, R. Capacity of ad hoc wireless networks. En *Proceedings of the 7th annual international conference on Mobile computing and networking, MobiCom '01*, páginas 61–69. ACM, 2001. ISBN 1-58113-422-3. (ref. pág. 76)
- LIAO, R. y GRIGORAS, D. An integrated discovery-routing service for mobile ad hoc networks. En *Communications and Mobile Computing, 2009. CMC'09. WRI International Conference on*, vol. 2, páginas 157–161. IEEE, 2009. ISBN 0769535011. (ref. pág. 35, 37)



- LUO, Y. S., YANG, K., TANG, Q., ZHANG, J. y XIONG, B. A multi-criteria network-aware service composition algorithm in wireless environments. *Computer Communications*, vol. 35(15), páginas 1882 – 1892, 2012. ISSN 0140-3664. (ref. pág. 46)
- MACKER, J. y TAYLOR, I. INDI: adapting the multicast DNS service discovery infrastructure in mobile wireless networks. En *Military Communications Conference, (MILCOM 2011)*, páginas 1616–1621. IEEE, 2011. ISBN 1467300799. (ref. pág. 35)
- MANDREOLI, F., PERDICHIZZI, A. M. y PENZO, W. A P2P-based architecture for SemanticWeb service automatic composition. En *Proceedings of the 18th International Conference on Database and Expert Systems Applications*, páginas 429–433. IEEE Computer Society, 2007. ISBN 0-7695-2932-1. (ref. pág. 43)
- MASUOKA, R., PARSIA, B. y LABROU, Y. Task computing – the semantic web meets pervasive computing. En *The Semantic Web - ISWC 2003*, vol. 2870 de *Lecture Notes in Computer Science*, páginas 866–881. Springer Berlin / Heidelberg, 2003. ISBN 978-3-540-20362-9. (ref. pág. 2, 42)
- MCILRAITH, S. A. y SON, T. C. Adapting golog for composition of semantic web services. En *8th International Conference on Principles of Knowledge Representation and Reasoning*. 2002. (ref. pág. 44, 46)
- MEIER, R., CAHILL, V., NEDOS, A. y CLARKE, S. Proximity-based service discovery in mobile ad hoc networks. En *Distributed Applications and Interoperable Systems*, vol. 3543 de *Lecture Notes in Computer Science*, páginas 1147–1149. Springer Berlin / Heidelberg, 2005. ISBN 978-3-540-26262-6. (ref. pág. 32)
- MOHAN, U., ALMEROOTH, K. y BELDING-ROYER, E. Scalable service discovery in mobile ad hoc networks. En *NETWORKING 2004. Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications*, vol. 3042 de *Lecture Notes in Computer Science*, páginas 137–149. Springer Berlin / Heidelberg, 2004. ISBN 978-3-540-21959-0. (ref. pág. 35, 38)
- MOKHTAR, S. B., LIU, J., GEORGANTAS, N. y ISSARNY, V. QoS-aware dynamic service composition in ambient intelligence environments. En *Proceedings of the 20th IEEE/ACM international Conference on Automated software engineering*, páginas 317–320. ACM, 2005. ISBN 1581139934. (ref. pág. 5, 42, 48)
- MOTEGI, S., YOSHIHARA, K. y HORIUCHI, H. Service discovery for wireless ad hoc networks. En *Wireless Personal Multimedia Communications, 2002. The 5th International Symposium on*, vol. 1, páginas 232–236 vol. 1. IEEE, 2002. ISBN 0780374428. (ref. pág. 33)

- MULLIGAN, G. The 6LoWPAN architecture. En *Proceedings of the 4th workshop on Embedded networked sensors*, páginas 78–82. ACM, 2007. ISBN 1595936947. (ref. pág. 16)
- MUNJAL, A., CAMB, T. y NAVIDI, W. C. Constructing rigorous MANET simulation scenarios with realistic mobility. En *Wireless Conference (EW), 2010 European*, páginas 817–824. 2010. (ref. pág. 22)
- MURTHY, C. S. R. y MANOJ, B. S. *Ad Hoc Wireless Networks: Architectures and Protocols*. Prentice Hall PTR, 2004. ISBN 9780131470231. (ref. pág. 12)
- NAU, D., ILGHAMI, O., KUTER, U., MURDOCK, J. W., WU, D. y FUSUN, Y. SHOP2: an HTN planning system. *Journal of Artificial Intelligence Research*, vol. 20, páginas 379–404, 2003. (ref. pág. 44)
- NEDOS, A., SINGH, K. y CLARKE, S. Service\*: Distributed service advertisement for multi-service, multi-hop MANET environments. En *7th IFIP International Conference on Mobile and Wireless Communication Networks (MWCN'05)*. 2005. (ref. pág. 35)
- NI, Q. Service composition in ontology enabled service oriented architecture for pervasive computing. En *Workshop on Ubiquitous Computing and e-Research*. 2005. (ref. pág. 44, 48)
- NIDD, M. Service discovery in DEAPspace. *Personal Communications, IEEE*, vol. 8(4), páginas 39–45, 2001. (ref. pág. 32)
- NOH, D. y SHIN, H. SPIZ: an effective service discovery protocol for mobile ad hoc networks. *EURASIP Journal on Wireless Communications and Networking*, vol. 2007(1), páginas 27–27, 2007. (ref. pág. 34, 35, 36)
- NS-3 DEVELOPMENT GROUP. The ns-3 network simulator. 2011. Disponible en <http://www.nsnam.org/> (último acceso, septiembre 2012). (ref. pág. 19)
- OASIS GROUP. Web services business process execution language (WSBPEL) TC. 2007. Disponible en [http://www.oasis-open.org/committees/tc\\_home.php?wg\\_abbrev=wsbpel](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel) (último acceso, septiembre 2012). (ref. pág. 47)
- OBAID, A., KHIR, A. y MILI, H. A routing based service discovery protocol for ad hoc networks. En *Networking and Services, 2007. ICNS. Third International Conference on*, páginas 108–108. IEEE, 2007. (ref. pág. 36)
- OPNET TECHNOLOGIES, INC. Wireless Network Simulator - OPNET. 2012. Disponible en <http://www.opnet.com/> (último acceso, septiembre 2012). (ref. pág. 19)

- OUTAY, F., KAISSER, F., VEQUE, V. y BOUALLEGUE, R. BF-SD-ZRP: a smart integrated scheme for service and route discovery in mobile ad hoc network. En *High Performance Computing and Communications (HPCC), 2010 12th IEEE International Conference on*, páginas 404–412. IEEE, 2010. ISBN 1424483352. (ref. pág. 33, 37)
- PALMIERI, F. Scalable service discovery in ubiquitous and pervasive computing architectures: A percolation-driven approach. *Future Generation Computer Systems*, vol. 1(0), páginas –, 2012. ISSN 0167-739X. (ref. pág. 33)
- PAOLUCCI, M., KAWAMURA, T., PAYNE, T. y SYCARA, K. Semantic matching of web services capabilities. En *The Semantic Web — ISWC 2002*, vol. 2342 de *Lecture Notes in Computer Science*, páginas 333–347. Springer Berlin / Heidelberg, 2002. ISBN 978-3-540-43760-4. (ref. pág. 5, 34, 42, 71)
- PEER, J. A PDDL based tool for automatic web service composition. En *Principles and Practice of Semantic Web Reasoning*, vol. 3208 de *Lecture Notes in Computer Science*, páginas 149–163. Springer Berlin / Heidelberg, 2004. ISBN 978-3-540-22961-2. (ref. pág. 56)
- PELTZ, C. Web services orchestration and choreography. *Computer*, vol. 36(10), páginas 46–52, 2003. (ref. pág. 49)
- PERKINS, C. E. y BHAGWAT, P. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. *ACM SIGCOMM Computer Communication Review*, vol. 24(4), páginas 234–244, 1994. (ref. pág. 16)
- PERKINS, C. E., ROYER, E. y DAS, S. RFC 3561 - ad hoc on-demand distance vector (AODV). IETF - Network Working Group, 2003. Disponible en <http://www.ietf.org/rfc/rfc3561.txt> (último acceso, septiembre 2012). (ref. pág. 16)
- PIRRÒ, G., TALIA, D. y TRUNFIO, P. A DHT-based semantic overlay network for service discovery. *Future Generation Computer Systems*, vol. 28(4), páginas 689–707, 2012. ISSN 0167-739X. (ref. pág. 31)
- PISTORE, M., BARBON, F., BERTOLI, P., SHAPARAU, D. y TRAVERSO, P. Planning and monitoring web service composition. En *Artificial Intelligence: Methodology, Systems, and Applications*, vol. 3192 de *Lecture Notes in Computer Science*, páginas 106–115. Springer Berlin Heidelberg, 2004. ISBN 978-3-540-22959-9. (ref. pág. 56, 59)
- PONNEKANTI, S., LEE, B., FOX, A., HANRAHAN, P. y WINOGRAD, T. ICrafter: A service framework for ubiquitous computing environments. En *Proceedings of the 3rd international conference on Ubiquitous Computing, UbiComp '01*, páginas 56–75. 2001. ISBN 3-540-42614-0. (ref. pág. 42, 48)

- POURREZA, H. y GRAHAM, P. On the fly service composition for local interaction environments. En *Proceedings of the 4th annual IEEE international conference on Pervasive Computing and Communications Workshops, PERCOMW '06*, página 393. IEEE Computer Society, 2006. ISBN 0-7695-2520-2. (ref. pág. 49)
- PROCHART, G., WEISS, R., SCHMID, R. y KAEFER, G. Fuzzy-based support for service composition in mobile ad hoc networks. En *Pervasive Services, IEEE International Conference on*, páginas 379–384. 2007. (ref. pág. 45, 48, 49, 50, 56)
- QASEM, A., HEFLIN, J. y MUÑOZ-AVILA, H. Efficient source discovery and service composition for ubiquitous computing environments. En *Workshop on Semantic Web Technology for Mobile and Ubiquitous Applications. ISWC*. 2004. (ref. pág. 44, 46)
- RANGANATHAN, A. y CAMPBELL, R. H. Autonomic pervasive computing based on planning. En *Autonomic Computing, 2004. Proceedings. International Conference on*, páginas 80–87. IEEE, 2004. ISBN 0769521142. (ref. pág. 48)
- RAO, J. y SU, X. A survey of automated web service composition methods. En *Semantic Web Services and Web Process Composition*, vol. 3387 de *Lecture Notes in Computer Science*, páginas 43–54. Springer Berlin / Heidelberg, 2005. ISBN 978-3-540-24328-1. (ref. pág. 45, 54)
- RATSIMOR, O., CHAKRABORTY, D., JOSHI, A. y FININ, T. Allia: Alliance-based service discovery for ad-hoc environments. En *Proceedings of the 2nd international workshop on Mobile commerce*, páginas 1–9. ACM, 2002. ISBN 1581136005. (ref. pág. 32, 33)
- RAYCHOUDHURY, V., CAO, J., WU, W., LAI, Y., CHEN, C. y MA, J. K-directory community: Reliable service discovery in MANET. *Pervasive and Mobile Computing*, vol. 7(1), páginas 140–158, 2011. (ref. pág. 32)
- ROMÁN, M., ZIEBART, B. y CAMPBELL, R. H. Dynamic application composition: Customizing the behavior of an active space. En *Pervasive Computing and Communications, (PerCom 2003). Proceedings of the First IEEE International Conference on*, páginas 169–176. IEEE, 2003. ISBN 0769518931. (ref. pág. 42)
- RUTA, M., ZACHEO, G., GRIECO, L. A., DI NOIA, T., BOGGIA, G., TINELLI, E., CAMARDA, P. y DI SCIASCIO, E. Semantic-based resource discovery, composition and substitution in IEEE 802.11 mobile ad hoc networks. *Wireless Networks*, vol. 16(5), páginas 1223–1251, 2010. (ref. pág. 5, 34, 42, 44, 47, 48, 49, 50, 51, 56)

- SADIQ, U., KUMAR, M., PASSARELLA, A. y CONTI, M. Modeling and simulation of service composition in opportunistic networks. En *Proceedings of the 14th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems*, MSWiM '11, página 159–168. ACM, 2011. (ref. pág. 56)
- SAID, R. y MAHÉO, Y. Toward a platform for service discovery and invocation in disconnected mobile ad hoc networks. En *Embedded and Ubiquitous Computing, 2008. EUC'08. IEEE/IFIP International Conference on*, páginas 238–244. IEEE, 2008. ISBN 0769534929. (ref. pág. 5, 34)
- SAILHAN, F. y ISSARNY, V. Scalable service discovery for MANET. En *Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications*, páginas 235–244. IEEE Computer Society, 2005. ISBN 0-7695-2299-8. (ref. pág. 30)
- SALUTATION CONSORTIUM INC. Salutation architecture specification v2.1. 1998. Disponible en <http://www.salutation.org> (último acceso, septiembre 2012). (ref. pág. 28)
- SCALABLE NETWORK TECHNOLOGIES, INC. QualNet. 2011. Disponible en <http://www.scalable-networks.com/content/products/qualnet> (último acceso, septiembre 2012). (ref. pág. 20)
- SCHIELE, G., BECKER, C. y ROTHERMEL, K. Energy-efficient cluster-based service discovery for ubiquitous computing. En *Proceedings of the 11th workshop on ACM SIGOPS European workshop*, página 14. ACM, 2004. (ref. pág. 31)
- SEADA, K. y HELMY, A. Rendezvous regions: A scalable architecture for service location and data-centric storage in large-scale wireless networks. En *Parallel and Distributed Processing Symposium, 2004. Proceedings. 18th International*, página 218. 2004. ISBN 0769521320. (ref. pág. 31, 35)
- SEN, R., ROMAN, G.-C. y GILL, C. CiAN: a workflow engine for MANETs. En *Coordination Models and Languages*, vol. 5052 de *Lecture Notes in Computer Science*, páginas 280–295. Springer Berlin / Heidelberg, 2008. ISBN 978-3-540-68264-6. (ref. pág. 46)
- SENO, S. A. H., BUDIARTO, R. y WAN, T. C. Survey and new approach in service discovery and advertisement for mobile ad hoc networks. *International Journal of Computer Science and Network Security (IJCSNS)*, vol. 7(2), páginas 275–284, 2007. (ref. pág. 31, 35, 36)
- SERHANI, M. A. y GADALLAH, Y. A service discovery protocol for emergency response operations using mobile ad hoc networks. En *Telecommunications (AICT), 2010 Sixth Advanced International Conference on*, páginas 280–285. IEEE, 2010. ISBN 1424467489. (ref. pág. 34)

- SHESHAGIRI, M., DESJARDINS, M. y FININ, T. A planner for composing services described in DAML-S. En *Proceedings of the AAMAS Workshop on Web Services and Agent-based Engineering*. 2003. (ref. pág. 44, 46, 49, 56)
- SIRIN, E., PARSIA, B., WU, D., HENDLER, J. y NAU, D. HTN planning for web service composition using SHOP2. *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 1(4), páginas 377–396, 2004. (ref. pág. 44, 46, 48)
- SIVAVAKEESAR, S., GONZALEZ, O. F. y PAVLOU, G. Service discovery strategies in ubiquitous communication environments. *Communications Magazine, IEEE*, vol. 44(9), páginas 106–113, 2006. (ref. pág. 31)
- SONG, Z., LABROU, Y. y MASUOKA, R. Dynamic service discovery and management in task computing. En *Mobile and Ubiquitous Systems: Networking and Services, 2004. MOBIQUITOUS 2004. The First Annual International Conference on*, páginas 310–318. IEEE, 2004. ISBN 0769522084. (ref. pág. 46, 47)
- SOUSA, J. P., POLADIAN, V., GARLAN, D., SCHMERL, B. y SHAW, M. Task-based adaptation for ubiquitous computing. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 36(3), páginas 328–340, 2006. (ref. pág. 42)
- SUN, W., ZHANG, Z., CHEN, W., PENG, B. y XU, Y. Decentralized execution of composite service in MANETs. En *Asia-Pacific Services Computing Conference, 2008. APSCC'08. IEEE*, páginas 355–360. IEEE, 2008. ISBN 0769534732. (ref. pág. 50)
- SUN MYCROSYSTEMS. Jini architecture specification v2.0. 2003. Disponible en [http://www.sun.com/software/jini/specs/jini2\\_0.pdf](http://www.sun.com/software/jini/specs/jini2_0.pdf) (último acceso, octubre 2007). (ref. pág. 28)
- TAYLOR, I. AgentJ - Java network simulations in NS-2. 2010. Disponible en <http://cs.itd.nrl.navy.mil/work/agentj/> (último acceso, septiembre 2012). (ref. pág. 20)
- TCHAKAROV, J. B. y VAIDYA, N. H. Efficient content location in wireless ad hoc networks. En *Mobile Data Management, 2004. Proceedings. 2004 IEEE International Conference on*, páginas 74–85. IEEE, 2004. ISBN 0769520707. (ref. pág. 33)
- THE VINT PROJECT. The network simulator - ns-2. 2011. Disponible en <http://www.isi.edu/nsnam/ns/> (último acceso, septiembre 2012). (ref. pág. 19)

- TYAN, J. y MAHMOUD, Q. H. A network layer based architecture for service discovery in mobile ad hoc networks. En *Electrical and Computer Engineering, 2004. Canadian Conference on*, vol. 3, páginas 1379–1384. IEEE, 2004. ISBN 0780382536. (ref. pág. 37)
- UCLA PARALLEL COMPUTING LABORATORY. GloMoSim - global mobile information systems simulation library. 1999. (ref. pág. 19)
- UPnP FORUM. UPnP device architecture 1.1. 2008. Disponible en <http://www.upnp.org/specs/arch/UPnP-arch-DeviceArchitecture-v1.1.pdf> (último acceso, septiembre 2012). (ref. pág. 26, 34)
- URBIETA, A., BARRUTIETA, G., PARRA, J. y URIBARREN, A. A survey of dynamic service composition approaches for ambient systems. En *Proceedings of the 2008 Ambi-Sys workshop on Software Organisation and Monitoring of Ambient Systems*, página 1. ICST Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, 2008. (ref. pág. 39)
- VALLÉE, M., RAMPARANY, F. y VERCOUTER, L. Flexible composition of smart device services. En *The 2005 International Conference on Pervasive Systems and Computing (PSC-05)*, páginas 27–30. 2005. (ref. pág. 46, 49)
- VARSHAVSKY, A., REID, B. y DE LARA, E. A cross-layer approach to service discovery and selection in MANETs. En *Mobile Adhoc and Sensor Systems Conference, 2005. IEEE International Conference on*, páginas 8 pp.–466. 2005. (ref. pág. 36)
- VERVERIDIS, C. N. y POLYZOS, G. C. Service discovery for mobile ad hoc networks: A survey of issues and techniques. *IEEE Communications Surveys & Tutorials*, vol. 10(3), páginas 30–45, 2008. (ref. pág. 28, 37)
- VERVERIDIS, C. N. y POLYZOS, G. C. A routing layer based approach for energy efficient service discovery in mobile ad hoc networks. *Wireless Communications and Mobile Computing*, vol. 9(5), páginas 655–672, 2009. (ref. pág. 33, 37)
- VUKOVIC, M. y ROBINSON, P. Adaptive, planning based, web service composition for context awareness. En *Proceedings of the Second International on Pervasive Computing*. 2004. (ref. pág. 44, 46, 47, 48, 49, 56)
- W3C MEMBER SUBMISSION. OWL-S: Semantic Markup for Web Services. 2004. Disponible en <http://www.w3.org/Submission/OWL-S/> (último acceso, septiembre 2012). (ref. pág. 47)

- W3C MEMBER SUBMISSION. Web Service Modeling Ontology (WSMO). 2005. Disponible en <http://www.w3.org/Submission/WSMO/> (último acceso, septiembre 2012). (ref. pág. 69)
- W3C OWL WORKING GROUP. OWL 2 web ontology language document overview. 2009. Disponible en <http://www.w3.org/TR/owl2-overview/> (último acceso, septiembre 2012). (ref. pág. 34, 70)
- WALZER, K., ANKE, J., LOSER, A. y WU, H. A concept for flexible event-driven invocation of distributed service compositions. En *Proceedings of the 27th International Conference on Distributed Computing Systems Workshops*, página 64. IEEE Computer Society, 2007. ISBN 0-7695-2838-4. (ref. pág. 50)
- WEISER, M. The computer for the 21st century. *Scientific American*, vol. 265(3), páginas 66–75, 1991. (ref. pág. 1)
- WENG, H., WEN, Y., JIN, B. y ZHANG, B. Scn4m-dl: An adaptive directory-less service discovery system for MANETs. En *Parallel and Distributed Systems (ICPADS), 2009 15th International Conference on*, páginas 640 –647. 2009. ISSN 1521-9097. (ref. pág. 33)
- WILLIAMS, B. y CAMP, T. Comparison of broadcasting techniques for mobile ad hoc networks. En *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing, MobiHoc '02*, páginas 194–205. ACM, 2002. ISBN 1-58113-501-7. (ref. pág. 76)
- WISTER, M. A. y TORRES, D. A. LIFT: an efficient cross-layer service discovery protocol in MANET. En *Advanced Information Networking and Applications Workshops, 2009. WAINA'09. International Conference on*, páginas 781–786. IEEE, 2009. ISBN 142443999X. (ref. pág. 31)
- ZENG, X., BAGRODIA, R. y GERLA, M. Glomosim: a library for parallel simulation of large-scale wireless networks. *SIGSIM Simulation Digest*, vol. 28(1), páginas 154–161, 1998. ISSN 0163-6103. (ref. pág. 19)
- ZHANG, J. F. y KOWALCZYK, R. Agent-based dis-graph planning algorithm for web service composition. En *Computational Intelligence for Modelling, Control and Automation, 2006 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, International Conference on*, página 258. 2006. (ref. pág. 44, 46)
- ZHANG, Z., SUN, W., CHEN, W., MAO, D. y XU, Y. An extended composite service choreography language for decentralized execution in MANETs. En *Communication Technology, 2008. ICCT 2008. 11th IEEE International Conference on*, páginas 592–595. IEEE, 2008. ISBN 1424422507. (ref. pág. 50)



- ZHOU, X., GE, Y., CHEN, X., JING, Y. y SUN, W. A distributed cache based reliable service execution and recovery approach in MANETs. En *Services Computing Conference (APSCC), 2011 IEEE Asia-Pacific*, páginas 298–305. IEEE, 2011. ISBN 1467302066. (ref. pág. 50, 51)
- ZHU, F., MUTKA, M. y NI, L. Splendor: A secure, private, and location-aware service discovery protocol supporting mobile services. En *Pervasive Computing and Communications. (PerCom 2003). Proceedings of the First IEEE International Conference on*, páginas 235–242. 2003. ISBN 0-7695-1893-1. (ref. pág. 30)
- ZHU, W., DU, Z. y ZHU, S. Dynamic service composition based on peer-to-peer network. En *Proceedings of the Second IEEE International Symposium on Service-Oriented System Engineering*, páginas 205–209. IEEE Computer Society, 2006. ISBN 0-7695-2726-4. (ref. pág. 43)
- ZIGBEE ALLIANCE. ZigBee specification. 2006. Disponible en <http://www.zigbee.org/Specifications.aspx> (último acceso, septiembre 2012). (ref. pág. 16)



Esta tesis fue terminada de escribir en Bilbao el 21 de noviembre de 2012

