Programa Oficial de Posgrado en Sistemas Multimedia

Departamento de Teoría de la Señal, Telemática y Comunicaciones

Universidad de Granada

TESIS DOCTORAL

Detección de intrusiones multi-capa basada en anomalías en entornos MANET

Realizada por:

D. Leovigildo Sánchez Casado

Dirigida por:

Prof. Dr. D. Pedro García Teodoro

Dr. D. Gabriel Maciá Fernández

Editor: Editorial de la Universidad de Granada
Autor: Leovigildo Sánchez Casado
D.L.: GR 2079-2014
ISBN: 978-84-9083-253-0

Official Postgraduate Program in Multimedia Systems

Department of Signal Theory, Telematics and Communications

University of Granada

Ph.D. THESIS DISSERTATION

Anomaly-based Multi-layer Intrusion Detection for MANET Environments

Author:

Mr. Leovigildo Sánchez Casado

Advisors:

Prof. Dr. Pedro García Teodoro

Dr. Gabriel Maciá Fernández

El doctorando D. Leovigildo Sánchez Casado y los directores de la tesis Dr. D. Pedro García Teodoro y Dr. D. Gabriel Maciá Fernández, catedrático y profesor titular de universidad respectivamente, y adscritos al Departamento de Teoría de la Señal, Telemática y Comunicaciones de la Universidad de Granada,

GARANTIZAMOS AL FIRMAR ESTA TESIS DOCTORAL

que el trabajo ha sido realizado por el doctorando bajo la tutela de los directores de la tesis y, hasta donde nuestro conocimiento alcanza, en la realización del trabajo se han respetado los derechos de otros autores a ser citados, cuando se han utilizado sus resultados o publicaciones.

Granada, a 22 de julio de 2014

Directores de la Tesis

Prof. Dr. D. Pedro García Teodoro

Dr. D. Gabriel Maciá Fernández

Doctorando

D. Leovigildo Sánchez Casado

Agradecimientos

R2D2, ¿te lo dijo la computadora central de la ciudad? ¡R2D2, sabes bien que no debes confiar en una computadora extraña!

C3PO

Mucha gente ha colaborado para que pudiese superar este ciclo de mi vida que comenzó hace 4 largos años. A todos ellos, gracias por permitirme conseguir lo que ansiaba.

En primer lugar me gustaría expresar de forma especial mi agradecimiento a mis tutores, Pedro y Gabri, por su ética de trabajo, ayuda constante, atención y dedicación durante estos años. Pero sobre todo, tengo que agradecerles que me hayan dado la oportunidad de formarme no solo como investigador, sino también como persona.

Gracias también al resto del grupo NESG (Jesús, Pepe, Rafa y Roberto) por su apoyo, sus ideas siempre útiles y los buenos momentos que hemos pasado. Gracias también, por supuesto, por los *NESGayunos* (¡Roberto paga!).

Gracias al resto de mis compañeros del Departamento de Teoría de la Señal, Telemática y Comunicaciones: Juanjo, Pablo A., Jorge, Juanma, Isaac, Sonia, Luz, Pablo P., etc. Gracias por todo lo que me habéis aportado. Tengo que agradecer también a todos los compañeros de fatigas que me han acompañado durante estos años. Gracias a Joaquín, Carlos, Santi, Alberto, Jose María, Povedano, Didi, Sofi, Sandra, Natalia y todos los demás por haber sido capaces de soportarme todo este tiempo ◡̈. Also thanks to my friends in Osnabrück for helping me during my stay there. Thanks to Nils, Alexander, Jan, Matthias and Wolfgang for letting me go there to increase my knowledge and for showing me how wonderful the people are there. Special thanks to Sabine for her invaluable support. I have no words to express my gratitude.

No puedo olvidarme de mis amigos, que durante estos años han hecho de mí una persona más madura y comprensiva, y me han ayudado a pulir algunos de mis defectos. Gracias a todos por los grandes momentos vividos junto a vosotros. Gracias a mis *erasmuses*, que me sacan sonrisas casi sin intentarlo.

Se lo agradezco sobre todo a mi familia, en especial a mis padres y mi hermana, que siempre han estado ahí para apoyarme y ayudarme en los momentos más difíciles. Gracias por vuestros ánimos y vuestra comprensión.

Por último, darle las gracias a la persona más importante en mi vida… Nuria, sin la que todo esto hubiera sido imposible. Gracias por tu cariño, amor, paciencia, apoyo, risas, compañía, … Gracias por hacer de mí una mejor persona. ¡Gracias!

A mis padres y mi hermana.

A Nuria, mi otra mitad.

Resumen

Es innegable la importancia hoy en día de las redes de comunicación en multitud de las actividades diarias de la sociedad actual. Entre las distintas tecnologías existentes, las redes inalámbricas han evolucionado considerablemente en los últimos años, hasta dar lugar a la aparición de novedosos sistemas y servicios.

En particular, es manifiesto el creciente interés de un nuevo paradigma de comunicación denominado *redes ad hoc*. Una red ad hoc es un tipo de red inalámbrica sin administración centralizada compuesta por un conjunto de nodos, geográficamente distribuidos en un área dada, formando topologías dinámicas y comunicándose mediante una estrategia que permite la adopción de rutas origen-destino multi-salto. Esta gran versatilidad, incrementada cuando los nodos que conforman la red tienen capacidad de movilidad, constituyendo las denominadas redes MANET (*Mobile Ad hoc NETwork*), hace de este tipo de entornos un candidato óptimo en multitud de áreas tales como aplicaciones de carácter medioambiental o militar, gestión de situaciones de crisis, terremotos o atentados terroristas, etc. Sin embargo, este paradigma de comunicación posee un claro inconveniente, y es la multitud de vulnerabilidades y amenazas a la seguridad a las que está sujeto, inherentes a la propia filosofía MANET.

En este contexto, el objetivo general del presente trabajo de tesis es *mejorar la seguridad de las redes MANET*. Este objetivo general se concreta en dos objetivos más específicos: (*i*) estudio y detección de ataques a la seguridad en redes MANET y (*ii*) desarrollo de procedimientos que permitan la integración de soluciones de seguridad.

Tras la realización de un estudio detallado de los ataques de seguridad existentes en entornos MANET y la propuesta de una nueva taxonomía para la categorización de los mismos, se ha identificado que dos de las principales amenazas de nuestros días son los ataques de *dropping* y los ataques *sinkhole*.

La detección de ataques es el proceso por el cual se determina la presencia de eventos o actividades maliciosas en la red. Para llevar a cabo dicha detección suele realizarse el despliegue de sistemas de detección de intrusiones o IDS (*Intrusion Detection System*), los cuales, en función de diversos parámetros obtenidos de la monitorización de la actividad habida en el entorno, son capaces de advertir y concluir la ocurrencia de comportamientos indeseados. En esta línea, se ha realizado una extensa revisión bibliográfica con el fin de conocer las principales técnicas defensivas propuestas en la bibliografía ante dichos ataques. Así mismo, y en relación con el objetivo central de esta tesis, se han diseñado nuevos mecanismos eficaces y eficientes para la *detección de ataques en redes MANET*. En particular, se ha propuesto un esquema para la detección de ataques de *dropping* basado en una heurística sencilla desarrollada a partir de una aproximación analítica del proceso de retransmisión

en entornos MANET, lo que permite distinguir entre comportamientos maliciosos y distintas causas legítimas para el descarte de paquetes. Por otro lado, se ha propuesto un esquema colaborativo que recopila información de la vecindad del nodo para la detección de ataques *sinkhole*, basado en una aproximación en dos pasos y en la existencia de los denominados "bordes de contaminación".

Una vez abordado el problema de la detección de ataques en redes MANET, es conveniente la subsiguiente *integración de soluciones de seguridad*. Para ello, primero se ha diseñado y desarrollado un mecanismo para la notificación y alerta de eventos de seguridad, cuyo fin principal es servir como una interfaz efectiva para la interoperación entre los distintos módulos defensivos. De este modo, ante la notificación de un cierto ataque detectado, será posible la adopción posterior de las medidas de respuesta oportunas. En esta misma línea de reunir distintos desarrollos de seguridad, aunque con propósitos más generales, se ha implementado un entorno de seguridad integral cuyo diseño flexible es apropiado para el despliegue de multitud de ataques, permitiendo implementar, integrar y comparar de forma precisa y bajo condiciones controladas de simulación nuevas técnicas de defensa. De esta manera, se pretende crear un marco de referencia que resulte una herramienta útil a la comunidad investigadora centrada en el campo de la seguridad en redes.

Abstract

Nowadays, communication networks are present in many of the daily normal activities of a great part of the world. Among the several existing technologies, wireless networks have considerably evolved in recent years, leading to the appearance of novel systems and services.

In particular, it is evident the increasing interest of the new communication paradigm that emerges with the so-called *ad hoc networks*. An ad hoc network is a particular type of wireless network composed of autonomous devices, geographically distributed in a given area and without a fixed infrastructure or centralized administration. Nodes that are within the communication range communicate directly, while those which are out of the range make use of other nodes to relay their messages to reach their destination (multi-hop strategy). The versatility of such a kind of environments, increased when the devices are mobile, setting up the so-called MANETs (*Mobile Ad hoc NETwork*s), make them a particularly useful candidate in certain areas, such as environmental and military applications, disaster and crisis management (earthquakes, terrorist attacks), etc. Despite their benefits, this kind of networks have a major limitation: the huge number of security threats inherently associated with these environments due to the intrinsic characteristics of MANETs.

In this general context, the central purpose of this thesis is *to strengthen the security in MANETs*. This general objective is divided into two more specific targets: (*i*) study and detection of security attacks in MANETs, and (*ii*) development of procedures to allow the integration of defensive solutions.

After a thorough study of the existing attacks in MANET environments and the proposal of a novel taxonomy for them, we have identified that two of the main current security threats are dropping and sinkhole attacks.

In order to detect the existence of attacks in a network, it is common to deploy IDSs (*Intrusion Detection Systems*) which, by means of monitoring several parameters related to the activity of the environment, are able to determine the occurrence of malicious behaviors against the system. In this line, a thorough study on the state of the art of research in the field of security defenses against these attacks has been performed. Besides, and related to the main objective of this thesis, we have introduced new efficient an effective *mechanisms to detect attacks in MANETs*. In particular, we have proposed a simple heuristic-based detection scheme aimed at distinguishing malicious dropping behaviors from other different circumstances which can lead to legitimate packet discards. After that, we have also proposed a collaborative approach that collects information from the node's vicinity to detect sinkhole attacks in MANETs, based on the existence of "contamination borders", and a two-phase process.

Once the detection of MANET attacks is addressed, it is convenient to *integrate security solutions*. For that, we have first designed and developed a mechanism for the notification and alert of security events, whose main goal is to be used as an effective interoperation procedure between the different defensive modules. This way, the notification of some given detected attack will allow to subsequently deploy some potential response solutions to solve it. Also in the line of putting together security developments, although with more general purposes, we have implemented an integral security framework. This constitutes a valuable tool to deploy attacks and implement, integrate and evaluate new defense schemes in a controlled simulation oriented testbed environment for the research community.

Contenido

Lista de Figuras

Lista de Tablas

Lista de Abreviaturas y Acrónimos

AA	*Action Agent*
ABM	*Anti-Blackhole Mechanism*
AP	*Access Point*
AR	*Attraction Ratio*
ARAN	*Authenticated Routing for Ad hoc Networks*
ARPANET	*Advanced Research Projects Agency NETwork*
AODV	*Ad hoc On-demand Distance Vector*
BBN	*BackBone Node*
BEB	*Binary Exponencial Backoff*
BEEP	*Blocks Extensible Exchange Protocol*
CBDS	*Cooperative Bait Detection Scheme*
CBR	*Constant Bit Rate*
CBRP	*Cluster-Based Routing Protocol*
CH	*Cluster Head*
CHT	*Call Holding Time*
CONFIDANT	*Cooperation Of Nodes: Fairness In Dynamic Ad hoc NeTworks*
CRADS	*Cross-layer Routing Attack Detection System*
CREP	*Confirmation REPly*

CREQ	*Confirmation REQuest*
CSMA	*Carrier Sense Multiple Access*
CSMA/CA	*CSMA with Collision Avoidance*
CSV	*Comma Separated Value*
CTS	*Clear To Send*
CW	*Contention Window*
DARPA	*Defense Advanced Research Projects Agency*
DCF	*Distributed Coordination Function*
DDoS	*Distributed Denial of Service*
DIFS	*Distributed Inter-Frame Space*
DoS	*Denial of Service*
DPRAODV	*Detection, Prevention and Reactive AODV*
DR	*Dropping Ratio*
DSDV	*Destination Sequence Distance Vector*
DSR	*Dynamic Source Routing*
DSSS	*Direct Sequence Spread Spectrum*
DTN	*Delay Tolerant Network*
DYMO	*DYnamic MANET On-demand*
E2ED	*End-to-End Delay*
EFSA	*Extended Finite State Automaton*
FANET	*Flying Ad hoc NETwork*
FDA	*Fisher Discriminant Analysis*
FHSS	*Frequency Hopping Spread Spectrum*
FN	*False Negatives*
FP	*False Positives*
FPR	*False Positives Rate*

FRp	*Further Reply*
FRq	*Further Request*
FTP	*File Transfer Protocol*
GloMo	*Global Mobile*
GloMoSim	*Global Mobile system Simulator*
HMM	*Hidden Markov Model*
IA	*Immune Agent*
IAT	*Inter-Arrival Time*
ICMP	*Internet Control Message Protocol*
ICT	*Information & Communications Technologies*
IEEE	*Institute of Electrical and Electronics Engineers*
IETF	*Internet Engineering Task Force*
IDAD	*Intrusion Detection based on Anomaly Detection*
IDIP	*Intrusion Detection and Isolation Protocol*
IDMEF	*Intrusion Detection Message Exchange Format*
IDS	*Intrusion Detection System*
IDWG	*Intrusion Detection Working Group*
IDXP	*Intrusion Detection eXchange Protocol*
IODEF	*Incident Object Description and Exchange Format*
IoT	*Internet of Things*
IP	*Internet Protocol*
IR	*InfraRed*
IRMEF	*Intrusion Response Message Exchange Format*
MAC	*Medium Access Control*
MANET	*Mobile Ad hoc NETwork*
MAODV	*Multicast AODV*

MIB *Management Information Base*

MIME *Multipurpose Internet Mail Extensions*

MOA *MOnitoring Agent*

MPR *MultiPoint Relay*

NAV *Net Allocation Vector*

NED *NEtwork Description*

NESG *Network Engineering & Security Group*

NETA *NETwork Attacks*

NTDR *Near Term Digital Radio*

NS-2 *Network Simulator 2*

NS-3 *Network Simulator 3*

ODMRP *On-Demand Multicast Routing Protocol*

OMH *One More Hop*

OMNeT++ *Objective Modular Network Test-bed in C++*

OLSR *Optimized Link State Routing*

OPNET *OPtimized Network Engineering Tools*

OTcl *Object-oriented Tool command language*

PCF *Puntual Coordination Function*

PDR *Packet Delivery Ratio*

PKI *Public Key Infrastructure*

PRNET *Packet Radio NETwork*

QoS *Quality of Service*

RERR *Route ERRor*

RFC *Request for Comments*

RIP *Restricted IP*

RIPPER *Repeated Incremental Pruning to Produce Error Reduction*

ROA	*ROuting Agent*
ROC	*Receiver Operating Characteristic*
RPGM	*Reference Point Group Mobility*
RREP	*Route REPly*
RREQ	*Route REQuest*
RTS	*Request To Send*
RWP	*Random Way Point*
SAODV	*Secure AODV*
SAR	*Security-Aware ad hoc Routing*
SEAD	*Secure Efficient Ad hoc Distance*
SIFS	*Short Inter-Frame Space*
SNMP	*Simple Network Management Protocol*
SORI	*Secure and Objective Reputation-based Incentive*
SRC	*Short Retry Count*
SRL	*Short Retry Limit*
SRP	*Secure Routing Protocol*
SRREP	*Secure RREP*
SRREQ	*Secure RREQ*
SVM	*Support Vector Machine*
SURAN	*SUrvivable RAdio Network*
TC	*Topology Control*
TCP	*Transmission Control Protocol*
TFT	*Tit-For-Tat*
TIC	*Tecnologías de la Información y las Comunicaciones*
TN	*True Negatives*
TP	*True Positives*

TPR *True Positives Rate*

TSR *Trust-based Secure Routing*

TTL *Time To Live*

UDP *User Datagram Protocol*

VANET *Vehicular Ad hoc NETwork*

VBR *Variable Bit Rate*

WLAN *Wireless Local Area Network*

WRP *Wireless Routing Protocol*

WSN *Wireless Sensor Network*

X-ARF *eXtended Abuse Reporting Format*

XML *eXtensible Markup Language*

XMPP *eXtensible Messaging and Presence Protocol*

YAML *YAML Ain't Markup Language*

ZRP *Zone Routing Protocol*

Parte I

Fundamentos de Seguridad en Redes MANET

Capítulo 1

Introducción

1.1. Redes MANET: alcance y retos

La época actual es conocida como la era de las comunicaciones y una de las razones que ha motivado este apelativo es la posibilidad de comunicarse de forma sencilla e instantánea con personas de todo el mundo. Esto es factible gracias a la impresionante evolución de Internet, que apareció en el año 1969 de la mano de ARPANET (*Advanced Research Projects Agency NETwork*) con únicamente dos ordenadores interconectados y en la que se comunican ahora miles de millones de equipos.

Entre las muchas posibilidades que ofrecen las TIC (*Tecnologías de la Información y las Comunicaciones*), las redes inalámbricas han evolucionado considerablemente en los últimos años, dando lugar a la aparición de diferentes tecnologías, arquitecturas y aplicaciones. En particular, este trabajo de tesis se centra en el estudio de las redes ad hoc [1]. Así, una red ad hoc es un tipo de red inalámbrica sin administración centralizada que se compone de un conjunto autónomo de dispositivos auto-configurables (generalmente denominados nodos), de despliegue sencillo y económico, geográficamente distribuidos en un área dada. Una de las primeras redes ad hoc, propuesta allá por la década de los 70 del S. XX, fue PRNET (*Packet Radio NETwork*) [2], promovida por la agencia DARPA (*Defense Advanced Research Projects Agency*) con el fin de verificar la viabilidad del uso de redes de paquetes inalámbricas en entornos militares. Una evolución de PRNET fue SURAN (*SUrvivable RAdio Network*), otro proyecto militar propuesto unos años más tarde y también patrocinado por la agencia DARPA, cuyo objetivo era proporcionar soporte a protocolos de comunicación

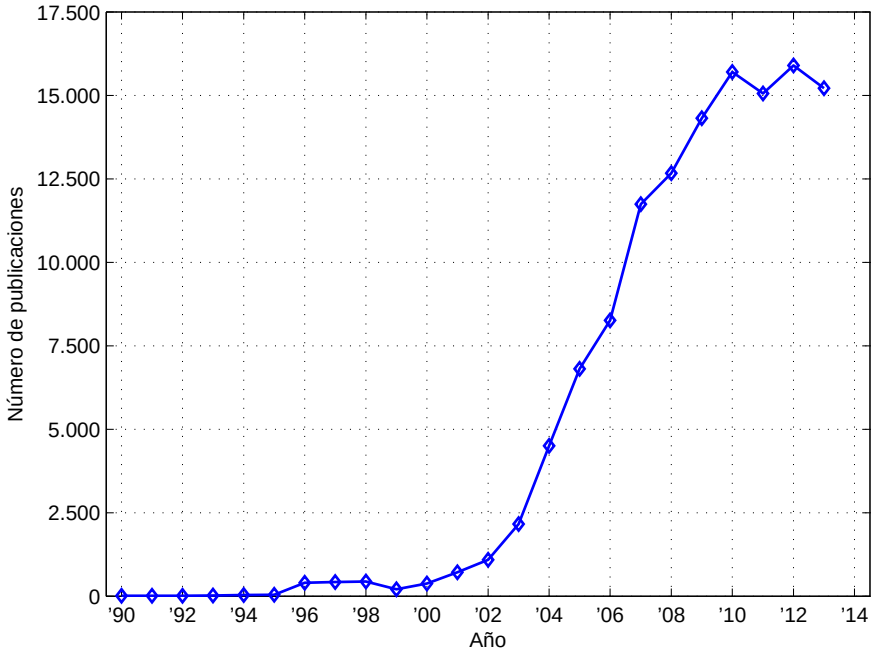


Figura 1.1: Evolución en el número de contribuciones directa o indirectamente relacionadas con redes ad hoc (Fuente: Scopus).

más sofisticados y desarrollar algoritmos más escalables y robustos. Debido a las numerosas investigaciones posteriores llevadas a cabo en este tipo de entornos, como GloMo (*Global Mobile*) o NTDR (*Near Term Digital Radio*), y a la amplia difusión de Internet y de los dispositivos portátiles inalámbricos, la IETF (*Internet Engineering Task Force*) (http://www.ietf.org) estableció en 1997 un grupo de trabajo específico denominado MANET [3], cuya principal tarea es la estandarización de protocolos de encaminamiento para redes ad hoc. Así, desde su aparición, es manifiesto el creciente interés en este tipo de redes, principalmente por parte de la comunidad investigadora. Una consecuencia de dicho interés puede observarse en la Figura 1.1, donde se muestra el crecimiento exponencial en el número de publicaciones especializadas relacionadas con este campo.

La comunicación en redes ad hoc tiene características específicas, como el uso de transmisiones inalámbricas, lo que proporciona una gran accesibilidad, o la ausencia de una infraestructura de transporte fija (encaminadores o puntos de acceso). Los nodos de la red ad hoc forman topologías dinámicas, comunicándose mediante una estrategia que permite la adopción de rutas origen-destino multi-salto; es decir, los nodos se comunican de forma directa con aquellos que se encuentran dentro de su rango de cobertura, y se apoyan en dichos nodos intermedios para que retransmitan sus mensajes, a modo de encaminadores, hasta el destino.

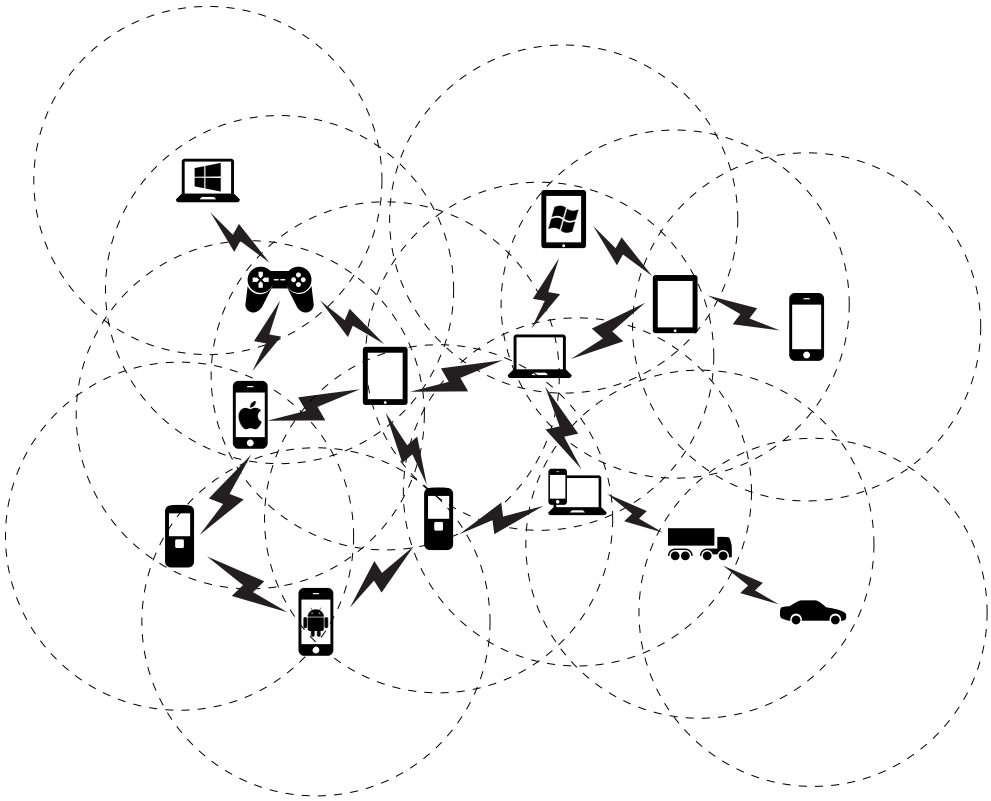


Figura 1.2: Ejemplo de escenario MANET, con diversos dispositivos móviles interconectados.

Las citadas peculiaridades dotan a estas redes de una gran versatilidad, haciendo de este tipo de entornos candidatos óptimos y altamente atractivos en multitud de áreas, tales como aplicaciones de carácter medioambiental o militar, gestión de situaciones de crisis, terremotos o atentados terroristas, etc. [4]. Esta versatilidad se ve incrementada cuando, además, los nodos que conforman la red tienen capacidad de movilidad, lo que constituye las denominadas MANET (*Mobile Ad hoc NETwork*) [5] [6], nombre del que derivan y con el que están relacionados otros también conocidos como VANET (*Vehicular Ad hoc NETwork*) y FANET (*Flying Ad hoc NETwork*) [7]. Así, es notable la relevancia que están cobrando este tipo de redes hoy día, pues su uso se está extendiendo, previéndose aún una mayor expansión en el futuro. Como se justificará más adelante, este trabajo de tesis se centra en el estudio de las redes MANET.

Un ejemplo de escenario típico de una red MANET es el observado en la Figura 1.2, en el que se distinguen diversos dispositivos o nodos inalámbricos, intercomunicándose entre sí siguiendo la ya citada estrategia multi-salto. Como se ha indicado, son numerosas las aplicaciones de este tipo de redes en entornos de lo más diverso. Una aplicación que surge de manera natural es la potencial gestión de situaciones de

emergencia o catástrofes. En estos escenarios, el servicio proporcionado por la red de telefonía móvil convencional suele verse interrumpido, lo que puede dificultar la labor de los distintos equipos de rescate. Gracias al inmediato despliegue de una red MANET sería posible proporcionar conectividad entre los distintos agentes. Así, por ejemplo, un grupo de bomberos podría solicitar la asistencia de un equipo médico que se encuentre en otra zona del área afectada.

Otro de sus principales ámbitos de aplicación es la intercomunicación en operaciones militares, en la que un escuadrón de soldados debe comunicarse entre sí, e incluso posiblemente con otros dispositivos ubicados en vehículos militares. En dichas situaciones, la comunicación con una unidad central puede no estar garantizada (condiciones ambientales, distancia, ataques enemigos, etc.), siendo entonces más adecuado el empleo de paradigmas de comunicación sin infraestructura y descentralizados.

A pesar de las numerosas ventajas de este tipo de entornos, son varias también sus limitaciones. Por una parte, la ausencia de infraestructura fija generalmente implica la necesidad de procedimientos o mecanismos más complejos para gestionar las comunicaciones de forma eficiente. La responsabilidad de la ejecución de estos procedimientos recae sobre los propios nodos, debiéndose dar respuesta a retos tales como el acceso compartido al medio inalámbrico, el descubrimiento de la topología de la red o la construcción de las rutas origen-destino para la entrega de los mensajes. Dichos procedimientos deben tener en cuenta la existencia de ciertas eventualidades, como la aparición de colisiones u otros efectos que alteren la propagación de la señal o la aparición de cambios rápidos e impredecibles en la topología, causados por la movilidad de los nodos. Por otro lado, aparecen restricciones inherentes a este tipo de dispositivos, generalmente relacionadas con los limitados recursos disponibles en los nodos, como pueden ser el tiempo de vida de la batería, la capacidad de almacenamiento o la potencia de cómputo.

Otro claro inconveniente de la utilización de las redes MANET, no menos importante que lo anterior, es la multitud de vulnerabilidades y amenazas a la seguridad inherentes a este tipo de redes y sistemas [8]. Especialmente motivados por la propia filosofía MANET y su naturaleza abierta (inalámbrica), además de por la potencial inexistencia de una infraestructura de control y gestión centralizados, son varios los tipos de amenazas existentes que adquieren cada vez mayor relevancia en estos entornos y que, en consecuencia, deben ser abordados convenientemente [9]. Por mencionar algunos, sírvanse citar, entre otros, los ataques de *dropping*, en los que un nodo malicioso elimina paquetes en su ruta hacia un destino dado; los ataques de *jamming*, donde un nodo genera interferencias y evita el acceso con éxito de otros al canal de comunicaciones; los ataques de suplantación de identidad (*spoofing*, *sybil*), consistentes en la falsificación de la identidad de un nodo; o los ataques de *route poisoning*, donde se falsifican las tablas de encaminamiento de los nodos a fin de

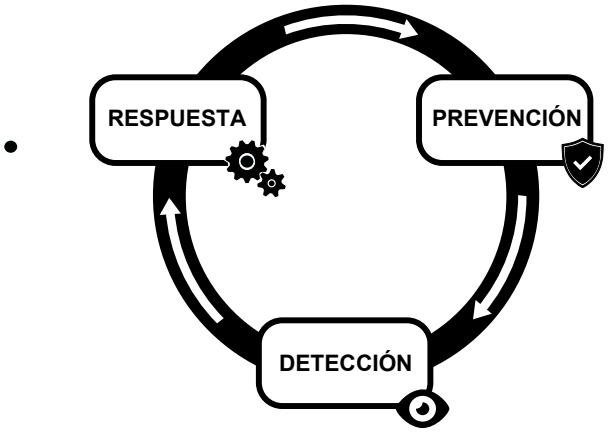


Figura 1.3: Líneas de defensa tradicionales ante incidentes de seguridad.

atraer tráfico hacia una cierta zona con fines maliciosos (eliminación, falsificación, acceso no permitido).

El problema de la seguridad en redes es un tema de especial relevancia en la era de las comunicaciones y recurrente en la investigación, siendo frecuente encontrar noticias relacionadas con el mismo. Un dato interesante acerca de la importancia de este problema lo proporciona un estudio publicado en 2011 por *Norton Symantec* [10]:

Con 431 millones de víctimas globales en el pasado año, y con unas pérdidas mundiales de 388.000 millones de dólares, el cibercrimen supera ampliamente la suma de dinero que mueven los mercados negros de marihuana, cocaína y heroína juntos, los cuales suponen unos 288.000 millones de dólares.

Aunque la principal motivación para desarrollar y llevar a cabo ataques contra la seguridad en redes MANET no sea la económica a fecha de hoy, los datos anteriores destacan el tremendo impacto que estas amenazas y vulnerabilidades pueden tener sobre las redes de comunicaciones actuales. En todo caso, considerando la creciente expansión en el uso de las tecnologías inalámbricas por parte de la sociedad, y en particular, el prometedor devenir del concepto de *Internet de las Cosas* o IoT (*Internet of Things*), donde el papel de las redes MANET podría ser relevante, no es en absoluto descabellado considerar el potencial auge de motivaciones financieras en un futuro cercano.

Para abordar el problema de la seguridad en redes, tradicionalmente se han propuesto tres líneas de defensa claramente diferenciadas: prevención, detección y respuesta (véase Figura 1.3). Si bien son conocidas las medidas *preventivas* de adopción aconsejada, su despliegue no garantiza en modo alguno la no aparición de actuaciones maliciosas. En consecuencia, ante la potencial superación de las medidas preventivas, se precisa complementar estas con otras orientadas a la *detección* de eventos indeseados. En *respuesta* a estos eventos maliciosos debieran ser adoptadas

las medidas oportunas para su resolución y, en suma, la recuperación del sistema. Adicionalmente, es recomendable la realimentación de todo el proceso a fin de permitir la adaptación dinámica del entorno [11][1]. Sin embargo, aunque en teoría estos módulos deberían interaccionar entre sí, en la práctica se suelen desarrollar como propuestas independientes, lo que da lugar a una problemática adicional relacionada con la inexistencia de mecanismos que permitan la interoperabilidad entre las distintas líneas defensivas.

En este contexto general, el objetivo de esta tesis está alineado con la resolución del problema de la seguridad en redes MANET, principalmente a partir del desarrollo de nuevas aproximaciones para la *detección* de ataques de alto impacto en este tipo de entornos.

1.2. Objetivos y metodología

Derivado de la importancia creciente del tema de la seguridad en redes en general, y en MANET en particular, el objetivo general del presente trabajo de tesis es *mejorar la seguridad de las redes MANET*. Este objetivo general se concreta en dos objetivos más específicos: (*i*) el estudio y detección de ataques a la seguridad en redes MANET y (*ii*) el desarrollo de procedimientos que permitan la integración de los mecanismos de detección con otras líneas defensivas. A continuación se expone la metodología que se ha seguido para la consecución de los citados objetivos.

El primer paso consiste en realizar un **estudio de los ataques de seguridad existentes en entornos MANET** en la actualidad, identificando aquellos con mayor relevancia desde el punto de vista del impacto sobre la red. En particular, tras la realización del estudio se ha identificado que dos de las principales amenazas de nuestros días son los ataques de *dropping* y los de *poisoning* [12]. Por este motivo, el desarrollo de esquemas de detección eficientes y precisos frente a dichos ataques es actualmente un problema muy popular en el campo de las redes de telecomunicación, y concretamente entre la comunidad investigadora.

La detección de ataques es el proceso automático por el cual se puede advertir la presencia de eventos indeseados en la red, permitiendo la categorización de los nodos responsables de dichos eventos en distintas clases (*maliciosos*, *sospechosos*, *legítimos*, etc.). Para llevar a cabo dicha detección suele realizarse el despliegue de sistemas de detección de intrusiones o IDS (*Intrusion Detection System*), los cuales, en función de diversos parámetros obtenidos de la monitorización de la actividad habida en el entorno, como pueden ser el número de paquetes retransmitidos o descartados, información de las rutas origen-destino aprendidas, características de

[1] La recuperación suele ser especificada de forma separada como una línea de defensa adicional.

los flujos, etc., son capaces de advertir y determinar la ocurrencia de comportamientos maliciosos contra la seguridad del sistema. Cabe destacar que, debido a la propia naturaleza de las redes MANET, la mayor parte de las técnicas y procedimientos desarrollados para redes cableadas o para redes WLAN (*Wireless Local Area Network*) no son apropiados en este nuevo tipo de entornos [13]. En esta línea, y en relación con el objetivo central de esta tesis, se han diseñado nuevos mecanismos eficaces y eficientes para la **detección de ataques en redes MANET**. Para evaluar los esquemas propuestos se ha desarrollado e implementado un entorno de experimentación. Puesto que los ataques evolucionan muy rápidamente, el desarrollo y despliegue de nuevas técnicas defensivas se está convirtiendo en una ardua tarea. De este modo, las herramientas de simulación ofrecen una solución intermedia entre coste y complejidad, por lo que hemos empleado este tipo de herramientas para analizar los esquemas de detección desarrollados. Esto permitirá comparar los resultados obtenidos con los proporcionados por otros esquemas similares en la literatura, extrayéndose así conclusiones válidas relativas a las capacidades reales de nuestras propuestas.

Una vez abordado el problema de la detección de ataques en redes MANET es necesaria la subsiguiente adopción de medidas de respuesta concretas frente a estos eventos intrusivos, habilitándose algún procedimiento que permita la notificación al resto de la red acerca de la existencia del evento intrusivo previamente detectado. Sin embargo, a pesar de que sobre el papel los módulos de defensa deben interoperar a fin de conseguir una seguridad integral, por lo general en la literatura estos se plantean y adoptan como soluciones independientes, obviando la necesidad de disponer de procedimientos efectivos de intercomunicación entre ellos. Esto es especialmente cierto y crítico para entornos de red ad hoc [14], en los que se evidencia una alta carencia de propuestas específicas orientadas a la interoperación de estos módulos. Así, el último paso de la labor desarrollada en este trabajo consiste en abundar en la **integración de soluciones de seguridad**. Para ello, primero se ha diseñado y desarrollado un mecanismo para la *notificación y alerta de eventos de seguridad*, cuyo fin principal es servir como una interfaz efectiva para la interoperación entre los módulos de detección y respuesta. Un diseño inteligente del mismo permitirá su empleo para un conjunto más amplio de actuaciones, como por ejemplo la intercomunicación de módulos de detección distribuidos y colaborativos. En esta misma línea de reunir distintos desarrollos de seguridad, aunque con propósitos más generales, se ha implementado un *marco de seguridad integral*. Un diseño flexible del mismo será apropiado para el despliegue de multitud de ataques, permitiendo implementar, integrar y comparar de forma precisa y bajo condiciones controladas de simulación, nuevas técnicas de prevención, detección y/o respuesta. De esta manera, se pretende crear un marco de referencia para analizar el impacto de diferentes ataques, así como de diversos esquemas defensivos, en múltiples tecnologías, protocolos y escenarios, proporcionando una herramienta útil a la comunidad investigadora centrada en el campo de la seguridad en redes.

En base a lo anteriormente expuesto, las tareas específicas a a realizar para la consecución de los citados objetivos son las siguientes:

I. *Estudio de ataques en redes MANET*

- Elaborar un estudio de los distintos ataques reportados en la actualidad en redes MANET y proponer una nueva taxonomía que permita una mejor clasificación de los mismos.
- Seleccionar los ataques más relevantes para focalizar el resto del trabajo en los mismos: ataques de *dropping* y *poisoning*.
- Realizar un estudio del estado del arte de la investigación en el campo de la defensa frente a dichos ataques, estructurando de una forma organizada e intuitiva los trabajos estudiados.

II. *Desarrollo de mecanismos de detección de ataques en redes MANET*

- Diseñar esquemas de detección frente a los ataques previamente indicados, amenazas especialmente dañinas en redes MANET.
- Evaluar el correcto funcionamiento de los esquemas de detección propuestos por medio de una experimentación basada en simulación.

III. *Procedimientos de interoperabilidad entre las líneas defensivas*

- Estudiar el estado del arte de la investigación referente a la interoperación entre las distintas líneas de defensa.
- Diseñar una nueva solución de comunicación que solvente las limitaciones actuales en este campo para redes MANET.
- Implementar y evaluar la solución propuesta.

IV. *Desarrollo de un marco de seguridad integral*

- Definir y diseñar una arquitectura flexible, apropiada para la implementación y evaluación de diversos ataques a la seguridad, así como de soluciones defensivas específicas.
- Llevar a cabo el desarrollo e implementación efectivos del marco de seguridad propuesto, proponiendo algunos ataques como prueba de concepto.
- Poner de manifiesto las capacidades del *framework* mediante la evaluación del funcionamiento de algunos de los ataques implementados.

1.3. Contribuciones principales

Las contribuciones fundamentales de este trabajo de tesis en relación a las tareas antes descritas pueden resumirse en los siguientes puntos:

1. Se presenta una revisión de los distintos ataques existentes en la actualidad frente a redes MANET.
2. Se diseña una nueva taxonomía para su clasificación, con el objetivo de permitir desarrollar mecanismos de defensa más eficientes y efectivos.
3. Se presenta una detallada revisión bibliográfica de los trabajos más relevantes de los últimos años en el campo de la detección de ataques en redes MANET, centrándonos en dos de las amenazas más dañinas en dichos entornos, los ataques de *dropping* y de *poisoning*. Como un caso particular y especialmente relevante del último tipo, aquí estudiaremos los ataques *sinkhole*.
4. Se contribuye con dos propuestas específicas para la detección de los mencionados ataques. Una de ellas está basada en una heurística sencilla que permite distinguir entre comportamientos de *dropping* maliciosos y distintas causas legítimas para el descarte de paquetes.
5. La segunda aproximación permite detectar ataques *sinkhole* por medio de un esquema colaborativo que recopila información de la vecindad de los nodos.
6. Se diseña e implementa también un nuevo protocolo de comunicación flexible, que permite la interoperación entre las distintas líneas defensivas tradicionales. Dicho protocolo puede ser empleado además para el desarrollo de sistemas de detección y/o reacción distribuidos y colaborativos.
7. Se desarrolla un nuevo *framework* de seguridad integral mediante simulación, para la evaluación de ataques y soluciones defensivas, basado en una arquitectura flexible y versátil.

1.3.1. Publicaciones

Las publicaciones derivadas de este período de tesis doctoral se resumen en 4 publicaciones enviadas a revistas internacionales (1 de ellas aceptada), 1 libro, 2 capítulos de libro, 3 congresos internacionales y 5 congresos nacionales. Aquellas publicaciones más directamente relacionadas con el trabajo aquí expuesto se indican a continuación:

Revistas internacionales

1. *Enviada* → **L. Sánchez-Casado**, G. Maciá-Fernández, P. García-Teodoro y N. Aschenbruck. "Identification of Contamination Zones for Sinkhole Detection in MANETs". *Journal of Network and Computer Applications (Elsevier)*, 20 páginas, 2014.

2. *Enviada* → **L. Sánchez-Casado**, G. Maciá-Fernández, y P. García-Teodoro. "A Model of Data Forwarding in MANETs for Lightweight Detection of Malicious Packet Dropping". *The Scientific World Journal (Hindawi)*, 25 páginas, 2014.

Capítulos de libro

3. P. García-Teodoro, **L. Sánchez-Casado** y G. Maciá-Fernández. "Taxonomy and Holistic Detection of Security Attacks in MANETs". *Security for Multihop Wireless Networks*, S. Khan y J. Lloret (Eds.), CRC Press, pp. 1-12, 2014.

4. **L. Sánchez-Casado**, R. Magán-Carrión, P. García-Teodoro y J. E. Díaz-Verdejo. "Defenses Against Packet Dropping Attacks in Wireless Multihop Ad Hoc Networks". *Security for Multihop Wireless Networks*, S. Khan y J. Lloret (Eds.), CRC Press, pp. 377-400, 2014.

Congresos internacionales

5. **L. Sánchez-Casado**, G. Maciá-Fernández, P. García-Teodoro y N. Aschenbruck. "A Novel Collaborative Approach for Sinkhole Detection in MANETs". *Workshop on Security in Ad Hoc Networks* (SecAN), pp. 42-55, 2014.

6. **L. Sánchez-Casado**, R. A. Rodríguez-Gómez, R. Magán-Carrión y G. Maciá-Fernández. "NETA: Evaluating the effects of NETwork Attacks. MANETs as a case study". *Advances in Security of Information and Communication Networks*, (SecNet), pp. 1-10, 2013.

7. **L. Sánchez-Casado**, G. Maciá-Fernández y P. García-Teodoro. "An Efficient Cross-Layer Approach for Malicious Packet Dropping Detection in MANETs". *11th IEEE International Conference on Trust, Security and Privacy in Computing and Communications* (Trustcom), pp. 231-238, 2012.

Congresos nacionales

8. **L. Sánchez-Casado**, R. Magán-Carrión, P. Garrido-Sánchez y P. García-Teodoro. "Protocolo para la Notificación y Alerta de Eventos de Seguridad en Redes

Ad hoc". Aceptado en *Reunión Española sobre Criptología y Seguridad de la Información* (RECSI), 6 páginas, 2014.

9. **L. Sánchez-Casado**, G. Maciá-Fernández y P. García-Teodoro. "Indicadores de Ataques Sinkhole en MANETs". *XI Jornadas de Ingeniería Telemática* (JITEL), pp. 475-480, 2013.

10. **L. Sánchez-Casado**, R. A. Rodríguez-Gómez, R. Magán-Carrión y G. Maciá-Fernández. "NETA: un Framework para Simular y Evaluar Ataques en Redes Heterogéneas. MANETs como Caso de Estudio". *XI Jornadas de Ingeniería Telemática* (JITEL), pp. 487-492, 2013.

11. **L. Sánchez-Casado**, G. Maciá-Fernández y P. García-Teodoro. "Multi-Layer Information for Detecting Malicious Packet Dropping Behaviors in MANETs". *Reunión Española sobre Criptología y Seguridad de la Información* (RECSI), pp. 57-62, 2012.

12. **L. Sánchez-Casado**, G. Maciá-Fernández y P. García-Teodoro. "Caracterización de Servicios en Redes Ad Hoc Inalámbricas mediante Métricas Cross-Layer". *X Jornadas de Ingeniería Telemática* (JITEL), pp. 381-384, 2011.

Existen otras publicaciones que, aunque relacionadas con la labor realizada, tienen un menor impacto en ella. Estas se indican a continuación.

Revistas internacionales

a) *Aceptada* → R. A. Rodríguez-Gómez, G. Maciá-Fernández, **L. Sánchez-Casado** y P. García-Teodoro. 2014. "Analysis and Modeling of Resources Shared in the BitTorrent Network". *Transactions on Emerging Telecommunications Technologies (Wiley)*, 2014.

b) *Enviada* → S. Salah, G. Maciá-Fernández, J. E. Díaz-Verdejo y **L. Sánchez-Casado**. "A Model for Incident Tickets Correlation in Network Management". *Journal of Network and System Management*, *major revision*, 2014.

Libros

c) G. Maciá Fernández, R. Magán Carrión, R. A. Rodríguez Gómez, **L. Sánchez Casado**: "Sistemas y Servicios Telemáticos". 2013. Ed. Avicam. ISBN: 978-84-941781-6-0.

Así mismo, en el marco de esta tesis doctoral, se han realizado los siguientes Proyectos Fin de Carrera y Trabajos Fin de Grado:

1. Pablo Garrido Sánchez, "Detección de ataques en OMNeT++: Dropping en redes MANET". *Trabajo Fin de Grado*, P. García-Teodoro y **L. Sánchez-Casado** (Directores), Grado en Ingeniería de Tecnologías de la Comunicación, Universidad de Granada, Julio 2014.

2. Jesús Ponce Medina, "Implementación de ataques de *jamming* en el entorno de simulación OMNeT++". *Proyecto Fin de Carrera*, G. Maciá-Fernández y **L. Sánchez-Casado** (Directores), Ingeniería de Telecomunicación, Universidad de Granada, Julio 2013.

Por su parte, el trabajo de investigación llevado a cabo en esta tesis doctoral está enmarcado dentro del siguiente proyecto de investigación:

- "SuMA: Supervivencia de redes MANET ante incidentes de seguridad" (Ref.: TEC2011-22579), Plan Nacional de I+D+i (2008–2011), Subprograma de Proyectos de Investigación Fundamental, Ministerio de Ciencia e Innovación, Gobierno de España.

El periodo de investigación predoctoral ha estado financiado por:

- "Programa de Formación de Profesorado Universitario" (AP2009-2926), Ministerio de Educación, Cultura y Deporte, Gobierno de España.

1.4. Estructura del documento

Este documento, de acuerdo a lo ya expuesto para los objetivos con anterioridad descritos, ha sido dividido en tres partes fundamentales, todas en la línea de proporcionar seguridad integral en redes MANET. Es importante destacar que se ha hecho un esfuerzo de redacción para que los capítulos sean, en la medida de lo posible, autocontenidos, de modo que un lector interesado solo en uno de ellos pueda comprenderlo en sus líneas básicas sin necesidad de leer el resto de capítulos.

1.4.1. Primera parte: Fundamentos de seguridad en redes MANET

Superado este primer capítulo, el Capítulo 2 presenta las principales amenazas a la seguridad existentes actualmente en las redes MANET, proponiéndose una nueva taxonomía (publicación 3) con dos objetivos principales: organizar y clasificar dichas amenazas a la seguridad en redes MANET de una forma más práctica, y proporcionar unas pautas generales para el potencial desarrollo de esquemas de detección más

flexibles y efectivos. Así mismo, en este capítulo se profundiza en el funcionamiento e implementación de dos de las potenciales amenazas más severas en este tipo de entornos, los ataques de *dropping* y de *poisoning*, detallándose los fundamentos básicos necesarios para su correcta comprensión. En este trabajo se considera un caso particular y especialmente relevante del último tipo, el ataque *sinkhole*.

1.4.2. Segunda parte: Detección de ataques en redes MANET

Adquiridos los conceptos principales con anterioridad, la segunda (y más importante) parte, que se corresponde con la contribución principal de este trabajo de tesis, se centra en la detección de las dos amenazas previamente identificadas, los ataques de *dropping* y los ataques *sinkhole*, como un caso particular de los ataques de *poisoning*.

Así, en el Capítulo 3 se expone una extensa revisión bibliográfica en el campo de la seguridad en redes MANET, haciendo especial hincapié en los esquemas de detección (publicación 4).

Incardinado en ello, en el Capítulo 4 se describe el sistema de detección propuesto para combatir los ataques de *dropping*. Dicho esquema se basa una heurística sencilla desarrollada a partir de una aproximación analítica del proceso de retransmisión en entornos MANET (véanse publicaciones 2, 7, 11 y 12).

Seguidamente, en el Capítulo 5 se detalla el sistema de detección de ataques *sinkhole* desarrollado. Dicho esquema se basa en una aproximación en dos pasos, en la primera de las cuales se realiza una pre-detección local. Solo si dicha pre-detección arroja un resultado positivo de comportamiento malicioso se computa una heurística (sencilla) de forma colaborativa con los nodos vecinos (publicaciones 1, 5 y 9).

1.4.3. Tercera parte: Integración de soluciones de seguridad

En esta última parte se aborda la interoperabilidad e integración de los esquemas de detección previamente desarrollados con el resto de líneas de defensa tradicionales. Por ello, en el Capítulo 6 se presenta el desarrollo de un nuevo protocolo ligero de notificación usado como interfaz de comunicación entre los módulos de detección y respuesta (publicación 8). El protocolo ha sido explícitamente diseñado para la notificación y alerta de eventos de seguridad en redes MANET, pudiendo ser utilizado también para la distribución de información de seguridad entre distintas entidades en procesos de detección y/o respuesta colaborativos. Por tanto, la aplicación de dicho protocolo permitirá desarrollar esquemas de seguridad más globales y, en consecuencia, más eficaces y robustos.

Para concluir esta tercera y última parte, y con un objetivo más ambicioso, en el Capítulo 7 se presenta el desarrollo e implementación de un marco de seguridad integral específico que permite la simulación de ataques en redes de comunicación, así como de soluciones defensivas, con el objetivo de proporcionar una herramienta útil a la comunidad investigadora (publicaciones 6 y 10). Su arquitectura flexible lo hace altamente extensible y versátil para la evaluación y comparación de nuevas técnicas de detección y/o respuesta bajo condiciones controladas.

Finalmente, en el Capítulo 8 se exponen las principales conclusiones que se derivan del trabajo completo, resaltando algunas líneas de trabajo futuro que han de abordarse para continuar avanzando en la línea de investigación iniciada con la presente tesis doctoral.

Apéndices

Adicionalmente a las tres partes mencionadas, y con objeto de cumplir con la normativa vigente relativa a las tesis con mención internacional, se incluyen dos apéndices en inglés al final de este documento. El primero de ellos, Apéndice A, presenta un resumen amplio del documento completo que expone de forma sintética los trabajos realizados, motivados por el objetivo global del presente documento, la mejora de la seguridad de las redes MANET.

En el Apéndice B se indican las conclusiones y líneas de trabajo futuro ya presentadas en el Capítulo 8.

Capítulo 2

Ataques a la seguridad en redes MANET

Durante los últimos años los ataques a la seguridad han constituido, y constituyen cada vez más, uno de los problemas más relevantes en el campo de las redes de comunicación y, debido a su creciente proliferación, en las redes MANET en particular.

En este capítulo se presenta una revisión de los principales ataques a la seguridad reportados para redes MANET hasta la fecha. Debido a la ausencia de una clasificación que agrupe los distintos ataques desde una perspectiva práctica, se propone aquí una nueva taxonomía que los englobe. Así mismo, se hará especial énfasis en la descripción de los ataques de *dropping* y de *sinkhole*, por ser estos los ataques sobre los que se centrarán los sistemas de detección propuestos y que serán objeto de estudio en los próximos capítulos de esta tesis. Para ello, es necesario proporcionar algunas nociones básicas acerca del funcionamiento del protocolo de encaminamiento con el que se trabajará con núcleo del estudio. En este trabajo de tesis nos centraremos en el uso de AODV (*Ad hoc On-demand Distance Vector*) [15], aunque como se explicará con posterioridad, las soluciones propuestas pueden hacerse extensivas a otros protocolos.

El resto del capítulo se estructura de la siguiente forma. En la Sección 2.1 se motiva la necesidad de conocer los ataques a la seguridad en redes MANET existentes en la actualidad, así como la de introducir cierto orden en dicho campo. Los principales ataques reportados en redes MANET se presentan en la Sección 2.2, introduciendo la nueva taxonomía propuesta en la Sección 2.3. Seguidamente, en la Sección 2.4 se introducen los fundamentos de comunicación en entornos MANET, focalizando el interés en los protocolos de capa de enlace y de encaminamiento empleados a

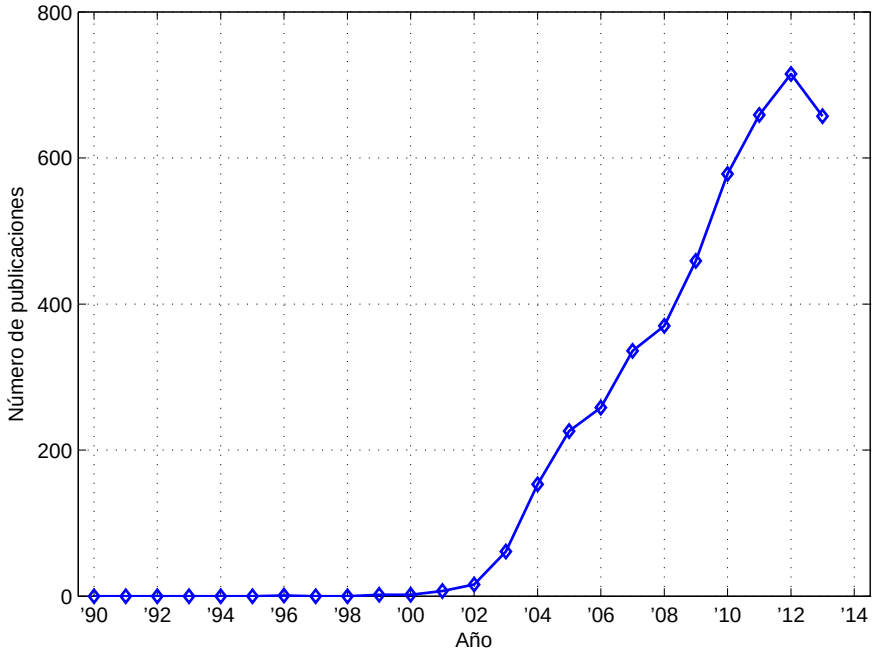


Figura 2.1: Número de publicaciones en el campo de la seguridad en redes MANET en los últimos años (Fuente: Scopus).

lo largo de este trabajo (IEEE 802.11 y AODV, respectivamente). En la Sección 2.5 se presentan en mayor detalle los ataques de *dropping* y *sinkhole*, sobre los que se centrarán los sistemas de detección propuestos en los siguientes capítulos. Por último, las conclusiones principales del presente capítulo se exponen en la Sección 2.6.

2.1. Motivación

La defensa frente a vulnerabilidades y amenazas de seguridad se ha convertido en los últimos años en un tema de especial relevancia en el campo de las redes de telecomunicación, debido principalmente a la amplia difusión de la que gozan actualmente las TIC. La vulnerabilidad de las redes de comunicación frente a ataques se pone especialmente de manifiesto en el caso de las redes MANET [5] [6], donde la movilidad de los nodos introduce un factor de dificultad adicional a la hora de desarrollar e implementar mecanismos de defensa eficientes. Como puede verse en la Figura 2.1, el elevado número de publicaciones en el campo de la seguridad en MANET, y su continuo crecimiento en los últimos años, evidencian la importancia de este tema entre la comunidad investigadora.

Una limitación importante de toda esta literatura es que, a pesar de la gran variedad de ataques y vulnerabilidades reportados, el número de contribuciones

existentes orientadas a introducir un cierto orden en el campo de la seguridad en redes MANET es reducido, siendo la mayoría de ellas clasificaciones teóricas que no sirven de especial utilidad para el diseño e implementación de sistemas de detección prácticos.

Así, con el objetivo de desarrollar e implementar sistemas defensivos eficientes, es necesario identificar y, en la medida de lo posible clasificar, los distintos ataques existentes en redes MANET reportados en la actualidad. Además, es imprescindible tener un conocimiento detallado de la posible implementación de los ataques, evaluando su comportamiento y los efectos sobre la red. De este modo se podrán diseñar esquemas de detección más precisos. Por todo ello, en este capítulo se presenta una revisión de los principales ataques reportados en redes MANET y de las clasificaciones existentes hasta la fecha, proponiéndose una taxonomía alternativa. Además, se profundiza en dos de las amenazas más dañinas para este tipo de entornos, los ataques de *dropping* y *sinkhole*, sobre los que se centrarán los sistemas de detección propuestos en los siguientes capítulos.

2.2. Principales ataques a la seguridad en redes MANET

Existen multitud de trabajos centrados en estudiar la seguridad en las redes MANET. Las peculiares características de estas redes, como la utilización de un medio no guiado para la transmisión (comunicación inalámbrica), la gestión descentralizada de las mismas o la utilización de terminales en movilidad y con recursos que pueden ser escasos (*p.ej.*, batería), dan como resultado que este tipo de entornos sean especialmente sensibles y propensos a la aparición de multitud de amenazas y ataques de seguridad. En la Tabla 2.1 se recopilan los ataques más comúnmente reportados en la literatura especializada [16] [17] [18] [19] [20], algunos de los cuales son específicos de entornos MANET y otros, aunque más genéricos, también afectan a dichos entornos. La tabla muestra los ataques, así como una pequeña descripción de su comportamiento y el efecto que producen.

Tabla 2.1: Principales ataques en redes MANET reportados en la literatura.

Ataque	Descripción
Blackhole	Descarte total del tráfico recibido, generalmente de los paquetes de datos. Suele ir precedido por actuaciones cuyo objetivo es conseguir que el resto de nodos retransmitan los paquetes hacia el nodo malicioso.

Continúa en la página siguiente

Tabla 2.1 – *Continúa de la página anterior*

Ataque	Descripción
Collision	Generación de interferencias selectivas para perturbar el correcto funcionamiento de los mecanismos MAC (*Medium Access Control*), lo que implica un número de errores de canal elevado y disminuye la probabilidad de captura del canal por parte de las transmisiones legítimas.
Delay	Introducción de un retardo temporal en la retransmisión de los paquetes.
*DoS (*Denial of Service*)*	Agotamiento de los recursos de la red, degradando el funcionamiento de la misma.
Eavesdropping	Escucha de las comunicaciones privadas, *i.e.*, interceptación de los datos. Ataque contra la confidencialidad.
Exhaustion	Repetidas colisiones y/o intentos de retransmisión continuos con el objetivo de ocupar el canal.
Fabrication	Creación de paquetes, generalmente destinados a engañar a los mecanismos de autenticación.
Flooding	Consumo significativo de los recursos de la red, *p.ej.*, mediante la inyección de multitud de paquetes inútiles. Es una variante del ataque DoS.
Grayhole	Ataque *blackhole* en el que el nodo realiza un descarte selectivo de los paquetes, *p.ej.*, con una cierta probabilidad, un paquete cada cierto tiempo, solo paquetes correspondientes a determinados flujos, etc.
HELLO flooding	Envío masivo de mensajes HELLO a los vecinos, inundándolos de información.
Impersonation	Adopción fraudulenta de la identidad legítima de otro nodo o aplicación, lo que resulta en distorsiones en la red.
Jamming	Generación de interferencias en la señal, lo que provoca interrupciones o alteraciones en la comunicación. Las interferencias pueden ser aleatorias, periódicas, etc.
Jellyfish	Introducción de retardos temporales en las retransmisiones TCP (*Transmission Control Protocol*), degradando el rendimiento extremo-a-extremo.
Link spoofing	Publicación de enlaces falsos con nodos que no son vecinos, alterando las operaciones de encaminamiento.
Link withholding	Se ignoran los anuncios de enlaces hacia rutas, provocando el aislamiento de los nodos.

Continúa en la página siguiente

Tabla 2.1 – *Continúa de la página anterior*

Ataque	Descripción
Link-broken error	Envío de paquetes de error falsos, provocando pérdidas de conectividad.
Man-in-the-middle	Actuación entre el emisor y el receptor, suplantando la identidad de uno de ellos o de ambos.
Modification	Modificación de los paquetes, alterando la integridad de los mensajes intercambiados.
Replication	Almacenamiento y posterior reenvío fraudulento de los mensajes previamente intervenidos en una comunicación legítima.
Routing cache poisoning	Falseo de la información de las tablas de rutas, modificando el correcto encaminamiento.
Routing table overflow	Anuncio de un número excesivo de rutas hacia nodos no existentes, evitando que los nodos vecinos puedan aprender nuevas rutas legítimas.
Rushing	Retransmisión inmediata de paquetes de rutas, provocando el aprendizaje de rutas incorrectas.
Selfish	Incumplimiento de ciertas reglas de los protocolos para ahorrar recursos (*p.ej.*, batería), haciendo decrecer el rendimiento de la red.
Sinkhole	Envío de información de encaminamiento publicando una ruta óptima falsa hacia el destino, lo que hace que el resto de nodos retransmitan los paquetes hacia el nodo malicioso, el cual podrá actuar sobre el tráfico recibido.
Sleep deprivation	Introducción de repetidas colisiones que inducen al nodo a intentar múltiples retransmisiones, causando el agotamiento de sus recursos.
Sybil	Adopción de múltiples identidades, *p.ej.*, convirtiéndose en parte "legítima" de la red.
SYN flooding	Creación de multitud de conexiones TCP sin completar, provocando el agotamiento de los recursos del nodo objetivo.
Tampering	Manipulación física de un nodo que afecta alguna funcionalidad, comprometiéndolo.
Wormhole	Dos nodos en confabulación almacenan los paquetes en una localización dada y los replican en otra distinta, utilizando para ello un enlace privado (generalmente de alta velocidad).

2.2.1. Clasificación de los ataques en redes MANET

En la lista de ataques presentada coexisten distintos tipos, sub-tipos y variantes, que constituyen únicamente diferencias sutiles entre los ataques. Por ejemplo, los ataques *sybil*, *man-in-the-middle* y de *suplantación* son similares en muchos aspectos, así como también lo son entre sí los distintos tipos de ataques de *inundación* (*i.e.*, por SYN, por HELLO) y los de *denegación de servicio*. ¿Y qué decir de los ataques de *modificación* o *fabricación* frente a los ataques de *link spoofing*?

Esta diversidad de ataques, en cierto modo artificial según nuestro punto de vista, puede justificarse por diversas causas. La diferencia entre distintos ataques es, en ocasiones, un mero asunto de grado o impacto, como ocurre con los ataques de *blackhole* y *grayhole*. En otros casos, la diferencia es debida a leves connotaciones, como en los ataques de *route cache poisoning* frente a los de *link spoofing* o de *link-broken error*. En resumen, varios de estos ataques son, en definitiva, el mismo ataque.

A veces, la distinción entre ataques es difícilmente comprensible. Por ejemplo, aunque el ataque de *blackhole* y *selfish* exhiben en ciertos casos un comportamiento similar (*i.e.*, descarte y no retransmisión de los paquetes), se consideran ataques distintos simplemente por su motivación: "malicia" en el primer caso frente a "egoísmo" en el segundo. Sin embargo, el perjuicio final ocasionado a la red difícilmente difiere en cada caso.

El propósito del ataque es confundido en algunas situaciones con el procedimiento que da lugar a su ejecución. Así, en este trabajo de tesis no se considera al ataque de *tampering* como un ataque per se, puesto que únicamente representa el acceso físico ilegítimo al nodo, a pesar de que esto pueda comprometerlo en algún sentido. Algo similar ocurre con el ataque *wormhole*. En este caso, la existencia de un enlace de alta velocidad entre dos nodos es explotada para redirigir el tráfico a través de él. Por tanto, se puede concluir que el ataque *wormhole* no es realmente un ataque, puesto que el enlace entre los nodos realmente existe. Distinto es el hecho de que se puedan realizar acciones subsecuentes sobre el tráfico al que se tiene acceso (*p.ej.*, descarte, modificación, etc.).

Otra diversificación en el estudio de los ataques en entornos MANET se refiere a la existencia de numerosos trabajos que analizan ataques frente a protocolos específicos, como AODV, OLSR (*Optimized Link State Routing*) o DSR (*Dynamic Source Routing*) [21] [22] [23]. En estas situaciones se suele obviar el análisis de las características generales del ataque, centrándose en la investigación de las causas y efectos específicos que se producen sobre la red.

Considerando lo previamente mencionado, se pone de manifiesto la necesidad de establecer una clasificación de los ataques en redes MANET que permita mejorar y clarificar este campo de estudio. Esta clasificación permitiría, por un lado, una mejor

comprensión de las vulnerabilidades presentes en dichos entornos. Por otro lado, establecer diferentes clases podría influir en la definición y despliegue de esquemas de defensa más efectivos y flexibles, que es el principal objetivo del presente trabajo.

Desde esta perspectiva, algunas de las clasificaciones que ya han sido propuestas con anterioridad en la literatura se explican brevemente a continuación:

- *Clasificación en ataques activos/pasivos*. Los ataques en redes MANET en ocasiones se clasifican como activos o pasivos [24]. Los primeros implican la modificación del flujo de datos o la creación de flujos falsos, con la intención de alterar los recursos del sistema y afectar a su funcionamiento. Por otro lado, en los segundos no se altera en modo alguno la comunicación, limitándose a la escucha o monitorización de la información del sistema, sin afectar a los recursos del mismo. Puesto que la amplia mayoría de los ataques existentes en la actualidad son considerados como activos, excepto el ataque de *eavesdropping*, esta clasificación no es especialmente útil.

- *Clasificación en ataques internos/externos*. Los ataques también pueden clasificarse en internos y externos [25]. Los primeros son realizados por nodos ya pertenecientes a la red como una parte autorizada de la misma y, por tanto, protegidos por los mecanismos de seguridad del sistema. Por el contrario, los ataques externos son llevados a cabo por nodos remotos que no son considerados todavía como pertenecientes a la red. Un nodo "dentro" del sistema debería realizar ataques que envíen información autenticada, mientras que un atacante externo podría llevar a cabo, *p.ej.*, ataques de *jamming*. Sin embargo, puesto que esta clasificación es más teórica que real (un atacante interno también puede realizar ataques de *jamming*, mientras que un atacante externo podría realizar una suplantación), se puede concluir también que la clasificación de ataques en internos frente a externos no proporciona información de discriminación relevante.

- *Clasificación por capa objetivo*. Otro criterio utilizado comúnmente para organizar los ataques en MANET es la capa a la que corresponde el protocolo vulnerable que explotan (capa objetivo) [18]. A pesar de que esta clasificación está ampliamente aceptada, no permite su empleo para desplegar sistemas de detección prácticos. Por ejemplo, un ataque de inundación por SYN (contra TCP) es, independientemente de la implementación específica, completamente similar a un ataque de inundación por paquetes HELLO (contra OLSR).

  Por tanto, el nivel de granularidad introducido por una clasificación de los ataques en función de la capa objetivo no tiene mucho sentido a la hora de crear una organización razonable que tenga como fin la implementación práctica de sistemas de defensa.

- *Clasificaciones alternativas*. A causa de la baja utilidad práctica de las clasificaciones previas, han aparecido nuevas propuestas en la literatura especializada. Así, Jawandhiya *et al.* [26] clasifican los ataques de acuerdo con el servicio de seguridad objetivo del ataque (*i.e.*, confidencialidad, integridad, disponibilidad, control de acceso), buscando clarificar las consecuencias del ataque en la red. Por su parte, Cardenas *et al.* [27] y Jain *et al.* [28] clasifican los ataques DoS en redes MANET en base a otro criterio, incluyendo el objetivo del ataque. En cambio, Liao *et al.* [29] proponen una clasificación basada en teoría de juegos, mientras que Yu *et al.* [30] realizan una categorización en función de mecanismos de confianza.

En este contexto se propone una nueva taxonomía para ataques en redes MANET con dos características principales [9]. Primero, y siguiendo la definición de "taxonomía", se consideran diversos criterios sucesivos para realizar la clasificación de los ataques, desde la disposición de una *raíz* común hasta derivar las distintas *especies* existentes. Por otro lado, la definición de *especies* permite derivar conjuntos de características que, más allá de las peculiaridades de cada ataque, puedan simplificar el diseño y desarrollo de entornos de defensa mas genéricos y, en consecuencia, versátiles.

2.3. Propuesta de taxonomía de ataques

La Figura 2.2 muestra la taxonomía de ataques MANET en este trabajo propuesta como alternativa hasta las ahora existentes. Desde una *raíz* común, denominada *ataques a la seguridad en redes MANET*, se obtienen agrupaciones de los distintos ataques reportados, hasta derivar la variante (o *especie*) específica. Los distintos criterios empleados son: (*i*) *acción* del atacante, (*ii*) *efecto* del ataque, (*iii*) *procedimiento* del ataque, y (*iv*) *función/servicio* atacado.

Primeramente clasificamos los ataques como *activos* o *pasivos*, en función de la **acción** llevada a cabo por el atacante para ejecutar el ataque. Si el ataque afecta al funcionamiento normal del sistema de alguna forma se considerará activo, en caso contrario se considerará pasivo.

Un segundo nivel de clasificación considera el **efecto** del ataque en el sistema. Los ataques pasivos implican *escuchas de información*, que afectan directamente a la confidencialidad. Por otro lado, los ataques activos pueden producir principalmente tres tipos de efectos, que hacen referencia bien a las entidades que toman parte en el proceso de comunicación, bien a la información transmitida entre ellas o bien a la calidad del servicio proporcionado. Dichos efectos son los siguientes:

Figura 2.2: Taxonomía propuesta para los ataques a la seguridad en redes MANET.

- Suplantación (*impersonation*), que puede afectar de algún modo a la autenticidad de las entidades participantes.
- Falsificación (*spoofing*), que supone un riesgo para la integridad de la información.
- Interrupción (*interruption*), es decir, una degradación (total o parcial) de la calidad del servicio, afectando a la disponibilidad de este.

Un tercer nivel para diferenciar los ataques en MANET se refiere al **procedimiento** llevado a cabo para ejecutar el ataque. Así, los ataques *pasivos* son posibles sin más que monitorizar el canal. Sin embargo, los ataques *activos* pueden ser ejecutados de diversas formas dependiendo del efecto considerado:

- *Ataques de suplantación*. Pueden llevarse a cabo falsificando la identidad de alguna de las entidades. Se distinguen dos procedimientos:
  - Replicación (*replication*): se utiliza una identidad ya existente en el sistema.
  - Invención (*invention*): se crea una nueva identidad específica para el atacante.
- *Ataques de falsificación*. Se ejecutan mediante la introducción de información "errónea" en los nodos y, consecuentemente, dando lugar a toma de decisiones "equivocadas". Se pueden distinguir dos situaciones:
  - Envenenamiento (*poisoning*): directamente se introduce información falsa en las comunicaciones, por ejemplo, en las tablas de encaminamiento.
  - Retraso (*delay*): a través de la captura de paquetes y su retransmisión, introduciendo un retardo temporal artificial (*delay*), el cual puede ser positivo o negativo.
- *Ataques de interrupción*. Son tres los principales procedimientos que pueden dar lugar a una interrupción total o parcial del servicio:
  - Retraso (*delay*): la introducción de retardos temporales en las comunicaciones también puede dar lugar a alteraciones en los tiempos de respuesta del servicio, degradando la calidad del mismo.
  - Descarte (*dropping*): mediante el descarte de paquetes, lo que además implica que se malgasten los recursos en transmitir información que no alcanzará el destino final.
  - Inundación (*flooding*): generando de forma artificial flujos de algún tipo de tráfico, que bloquean e impiden el acceso normal a los recursos del sistema.

En resumen, se propone que cada uno de los ataques en MANET reportados puede clasificarse en una de las 7 categorías obtenidas de la clasificación multi-dimensional *acción* → *efecto* → *procedimiento*, tal y como se muestra en la Figura 2.2. Así, la existencia de dichas clases permite ir más allá de posibles particularidades debidas a la consideración de servicios y/o protocolos específicos.

No obstante ello, en este punto se pueden obtener las *especies* de los ataques en nuestra taxonomía aplicando un cuarto criterio de clasificación: la **función objetivo** atacada, *i.e.*, el servicio y/o protocolo que es objetivo del ataque. Pueden existir multitud de funciones (y protocolos asociados) implementadas en la red y, por tanto, existen también numerosos sub-tipos o variantes de los ataques. Así, *p.ej.*, la implementación de *escuchas* será distinta si se quiere obtener información de la capa física o de la capa MAC, puesto que para llevar a cabo el ataque en el segundo caso se requiere un conocimiento específico que permita interpretar las tramas MAC y la temporización asociada. Análogamente, un ataque de *suplantación* puede llevarse a cabo a distintos niveles, incluyendo la identificación MAC, el puerto TCP o la dirección IP (*Internet Protocol*), entre otros. Por tanto, es necesario conocer la sintaxis específica de la función/servicio correspondiente para poder diseñar defensas concretas.

Debe tenerse en cuenta que ciertos ataques pueden ser considerados como *complejos*, puesto que involucran la ejecución de más de un tipo (clase) de ataque. Así, la mayoría de los autores definen el ataque *blackhole* como la combinación de dos procesos: un primer proceso en el que se intenta "envenenar" las tablas de rutas de los nodos, de modo que el nodo malicioso se convierta en parte de la ruta; y un segundo paso en el que el nodo malicioso comienza a descartar los paquetes que le llegan (de forma total, selectivamente, etc.).

Por otro lado, nótese que el ataque de *tampering* reportado en la Tabla 2.1 no se ajusta en principio a la clasificación propuesta en la Figura 2.2, puesto que no consideramos este tipo como un ataque como tal, sino como la acción de manipular físicamente el nodo. Así, la pregunta interesante desde una perspectiva centrada en la defensa frente a ataques sería: ¿Cuál es el objetivo de la manipulación del nodo? ¿Es tener acceso a las credenciales privadas? ¿Modificar características de la transmisión radio? ¿Alterar su identidad? ¿O incluso un objetivo completamente diferente? Nuestra tesis es que, en función del objetivo para el que se realice el *tampering* del nodo, se podrá clasificar el ataque según la taxonomía propuesta.

También es de destacar el hecho de que los nodos maliciosos pueden actuar de forma individual o en *confabulación*. Sin embargo, este último hecho no es tenido en cuenta en nuestra clasificación, puesto que el principal objetivo de la confabulación es dificultar la defensa frente al ataque ocultando o dispersando la acción entre distintos ataques. Un paradigma de ataque en *confabulación* sería, *p.ej.*, un ataque

de DoS distribuido o DDoS (*Distributed Denial of Service*) frente a un ataque DoS tradicional.

2.4. Fundamentos de comunicación en redes MANET

Una vez detallados los distintos ataques reportados en la actualidad y la nueva taxonomía propuesta para los mismos, se puede concluir que, a pesar de que puedan ser incluidos en distintas clases, los ataques son al final implementados contra servicios o protocolos concretos (especies). De este modo, es necesario introducir algunos conceptos fundamentales, importantes para entender las bases de la comunicación en entornos MANET, y sobre los que se implementan dichas especies. Además, generalmente las redes MANET se caracterizan por implementarse sobre una capa de enlace inalámbrica que permita la operación ad hoc, así como sobre protocolos de encaminamiento que permitan la comunicación igual a igual entre los nodos de la red, permitiendo que estos actúen a su vez como *routers*. Por tanto, es importante definir algunos de los fundamentos básicos de este paradigma de comunicación, centrándonos, como se ha indicado, en las capas de enlace y de red.

A pesar de las múltiples tecnologías inalámbricas existentes hoy día, como las propuestas en los estándares IEEE 802.15.1 (Bluetooth®) [31], 802.15.4 (sobre la que se basa Zigbee®) [32], 802.16 (WiMAX®) [33], 802.22 (redes cognitivas) [34], una de las tecnologías más ampliamente utilizadas a la hora de implementar redes MANET realmente operativas es la definida por el estándar IEEE 802.11 [35], más comúnmente conocida como *Wi-Fi*. Así, a pesar de no haber sido diseñada originalmente como una tecnología específica para este tipo de entornos, tanto la posibilidad de operar en modo ad hoc como su amplia difusión hacen de esta tecnología un candidato perfectamente útil para ser empleado en escenarios MANET.

Además, de entre los distintos protocolos de encaminamiento disponibles para redes MANET, AODV [15] quizá sea uno de los más conocidos y ampliamente utilizados. Sin embargo, este no fue diseñado originalmente teniendo en cuenta aspectos de seguridad y, en consecuencia, no especifica medidas como cifrado, autenticación o protección de la integridad de los datos. Por tanto, aún continúa llevándose a cabo una extensiva investigación en este campo con el fin de resolver todas las potenciales vulnerabilidades presentes en AODV.

Es por estas razones por las que el presente trabajo de tesis se centrará en el desarrollo de esquemas de defensa específicos para este protocolo de encaminamiento en particular. Sin embargo, es de destacar que, tal y como se detallará en los capítulos correspondientes, los esquemas de detección propuestos son fácilmente extensibles a otros protocolos similares existentes en la actualidad para redes MANET, como DSR (*Dynamic Source Routing*) o DYMO (*DYnamic MANET On-demand*).

Antes de comenzar el estudio de los fundamentos de comunicación en redes MANET, particularizados para las tecnologías y protocolos previamente indicados, nótese que estos se estudiarán única y exclusivamente hasta el nivel de detalle necesario para comprender las actuaciones posteriores que serán presentadas en este trabajo de tesis.

2.4.1. Fundamentos de IEEE 802.11

La familia de estándares 802.11 [35] define y gobierna las redes de área local inalámbricas o redes WLAN, que operan en el espectro de los 2,4 y 5 GHz. La primera implementación del estándar original data de 1997 y especificaba la operación a 1 y 2 Mbps usando tres tecnologías diferentes: FHSS (*Frequency Hopping Spread Spectrum*), DSSS (*Direct Sequence Spread Spectrum*) e IR (*InfraRed*), asegurando la interoperabilidad entre equipos de comunicación dentro de cada una de estas tecnologías inalámbricas. Dicha versión original ha tenido numerosas revisiones con el fin de mejorarla. Así, el estándar 802.11 define dos posibles arquitecturas:

1. *Redes con punto de acceso, o modo infraestructura*: en estas las estaciones se comunican mediante un dispositivo de interconexión llamado punto de acceso, o AP (*Access Point*). La ventaja de esta arquitectura es la escalabilidad.

2. *Redes sin punto de acceso, o modo ad hoc*: en estas las estaciones se comunican directamente entre sí, sin necesidad de emplear una infraestructura fija intermediaria.

Este trabajo de tesis se centra en el estudio de la segunda arquitectura. Para permitir la interconexión de más de dos dispositivos es necesario administrar el medio inalámbrico por el que circula la información. Para realizar esta tarea se define la subcapa MAC.

Capa MAC IEEE 802.11

La subcapa MAC de IEEE 802.11 se encarga de proporcionar a las capas superiores una interfaz independiente de la tecnología empleada en la capa de enlace de datos y física, utilizando CSMA (*Carrier Sense Multiple Access*) como método de detección de portadora. Así, antes de intentar el envío de una trama se sondea el medio para determinar si este se encuentra libre de comunicaciones, en cuyo caso comienza el proceso de transmisión. Por el contrario, si el medio está ocupado, se darán distintos casos en función de la implementación de CSMA: bien se espera hasta que quede libre el medio y se pueda transmitir (modo persistente), bien se espera un tiempo

aleatorio y se reintenta la transmisión (modo no persistente), o bien se transmite con probabilidad p cuando el medio quede libre (modo p-persistente). Proceder a la escucha del medio y, por lo tanto, detectar las colisiones producidas puede resultar complicado en medios inalámbricos. Por ello se utiliza el protocolo CSMA/CA (*CSMA with Collision Avoidance*), en el cual, en lugar de transmitir la trama en cuanto se detecta el medio libre, se espera un tiempo aleatorio adicional y, solamente si tras ese intervalo el medio sigue estando libre, se procede a la transmisión. Con ello se reduce la probabilidad de colisiones en el canal, como se verá más adelante cuando se discuta el problema de la estación oculta.

Para controlar el acceso al medio inalámbrico compartido, la capa MAC en IEEE 802.11 utiliza dos funciones: una *función de coordinación distribuida* o DCF (*Distributed Coordination Function*), implementada en todos los nodos de la red, y una *función de coordinación puntual* o PCF (*Puntual Coordination Function*), implementada únicamente en los puntos de acceso, AP.

Para una gestión eficiente del acceso, únicamente se permiten transmisiones en unos determinados períodos fijos de tiempo, denominados *slots* o ranuras, por lo que las colisiones se producen solo cuando distintos nodos solicitan transmitir en el mismo *slot*.

Cuando un nodo desea enviar datos, y antes de poder comenzar con la transmisión de la trama, debe escuchar el medio para comprobar que este se encuentra libre durante un determinado intervalo de tiempo, denominado DIFS (*Distributed Inter-Frame Space*). Transcurrido este, si el medio aún permanece libre, podrá proceder al envío (véase Figura 2.3). Si la trama es correctamente transmitida, el nodo receptor confirmará su correcta recepción mediante una trama ACK, enviada tras esperar un intervalo SIFS (*Short Inter-Frame Space*).

Por el contrario, si durante el intervalo DIFS se detecta que existe una transmisión en el canal, el nodo aplaza el envío de la trama deseada. Para evitar que todos los nodos que han detectado el canal ocupado aplacen sus transmisiones el mismo tiempo e intenten posteriormente transmitir a la vez, provocando gran cantidad de colisiones, se emplea el retroceso exponencial binario o BEB (*Binary Exponencial*

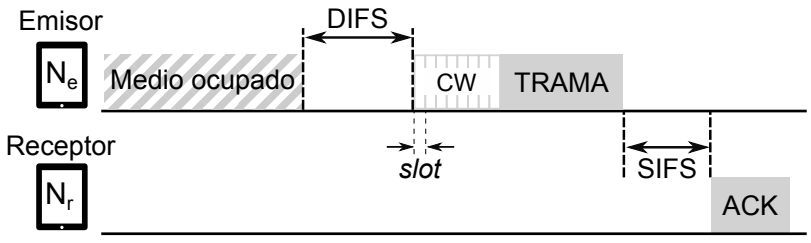


Figura 2.3: Espaciado inter-tramas en IEEE 802.11.

Backoff). Este mecanismo fuerza que los nodos que han detectado el canal ocupado esperen un tiempo extra aleatorio, cuya longitud viene determinada como función de un número dado de *slots* temporales. Dicho número de ranuras toma un valor entero entre 0 y una ventana de contención, o CW (*Contention Window*). El valor de dicha ventana viene definido por las características físicas del medio y varía entre unos valores mínimo y máximo, CW_{min} y CW_{max}. Así, cada vez que un nodo escucha el medio y lo encuentra ocupado, duplica el valor de la ventana de contención, hasta llegar a un valor máximo definido por $CW_{max} + 1$. De igual modo, ante una transmisión exitosa, la ventana vuelve a su valor mínimo CW_{min}. De este modo, el tiempo de espera del nodo tras detectar la ocupación del canal es:

$$\text{Tiempo de } backoff = random(0, CW) \times \text{tiempo de ranura} \tag{2.1}$$

Además, durante el proceso de transmisión, el resto de nodos que escuchan esta transmisión ajustan su tiempo de reserva, NAV (*Net Allocation Vector*), al tiempo de duración de la trama detectada, siendo este el tiempo que deberán esperar antes de volver a comprobar el estado del medio. Se evitan así envíos simultáneos a la trama actual, lo que daría lugar a colisiones.

La utilización de la DCF no resuelve completamente, sin embargo, algunos de los problemas que se dan al acceder a medios compartidos, como es el *problema de la estación oculta*. Esta situación (véase Figura 2.4) se produce cuando un nodo N_a intenta transmitir hacia un nodo N_b creyendo que el canal está libre (1), pero en realidad está ocupado por otro nodo N_c al que no oye (2), al no estar dentro de su radio de acción. Se produciría así una colisión que impide que N_b reciba correctamente ninguno de los mensajes. El resultado es un incremento considerable

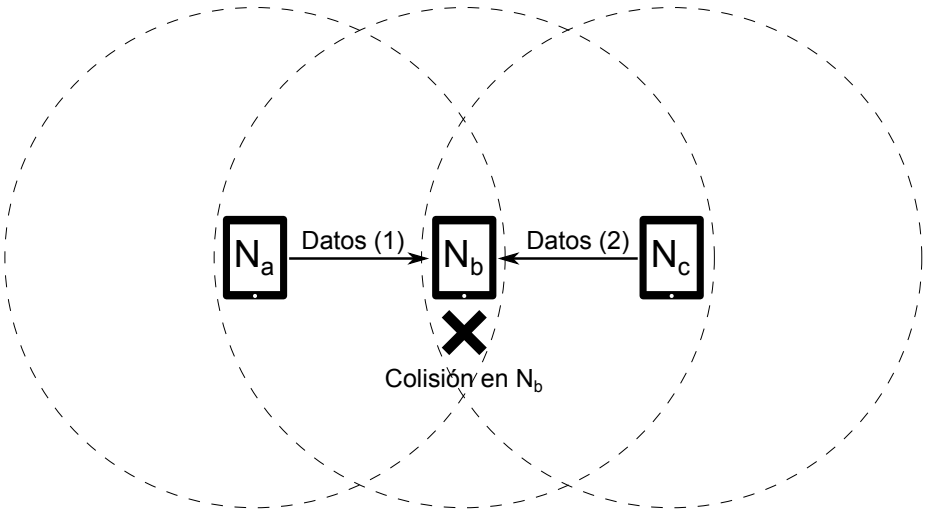


Figura 2.4: Problema de la estación oculta.

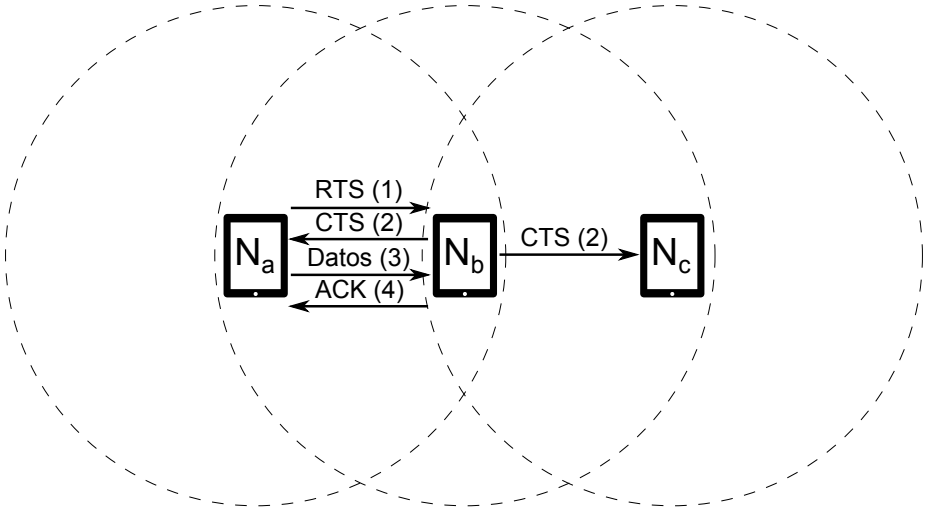


Figura 2.5: Mecanismo de sondeo de portadora virtual en IEEE 802.11.

en el número de colisiones, dando lugar a repetidas retransmisiones que redundan en un gasto innecesario de los recursos de la red (batería, ancho de banda, ...).

Con intención de aliviar este problema, en IEEE 802.11 también se puede utilizar un mecanismo de *sondeo de portadora virtual* (Figura 2.5). Mediante dicho procedimiento, el nodo que desea transmitir, N_a, genera una trama especial de solicitud de transmisión o RTS (*Request To Send*) (1), hacia el destino deseado, N_b. Dicho destino N_b, tras recibir la trama RTS, responderá con una trama de autorización o CTS (*Clear To Send*) (2) hacia el nodo origen, N_a. Así, bien la trama RTS, bien la trama CTS, será recibida por los nodos que se encuentren en el radio de cobertura respectivo de N_a y N_b, detectando dichos nodos vecinos (N_c en nuestro ejemplo) por tanto que el medio está ocupado con una transmisión. Finalmente, el nodo N_a podrá transmitir sin colisiones los datos (3) una vez recibida dicha autorización.

Además, como se ha mencionado anteriormente, la función DCF de IEEE 802.11 también incluye confirmaciones positivas, o ACK, que indican la correcta recepción de los datos enviados por parte del destino, así como la liberación del medio (4).

Es de destacar que existen diversas vulnerabilidades que permitirían explotar el funcionamiento del protocolo IEEE 802.11 por parte de nodos maliciosos con el fin de llevar a cabo diversos ataques, o simplemente de nodos que quisieran incumplir el protocolo para obtener algún tipo de beneficio. Por ejemplo, un nodo podría proceder al envío de tramas sin esperar el intervalo DIFS requerido, siendo siempre el primero en transmitir, acaparando permanentemente el canal y evitando el acceso a este por parte de otros nodos legítimos. Otra posibilidad es la implementación de ataques de *jamming*, en los que un nodo dado incumple los tiempos de espera y realiza transmisiones a sabiendas de que el medio se encuentra ocupado actualmente,

provocando gran cantidad de colisiones, que darán como resultado multitud de retransmisiones y, en consecuencia, un mayor agotamiento de los recursos de los nodos legítimos. Aunque estos ataques quedan fuera del ámbito de este trabajo de tesis, se ha considerado importante citarlos al menos de forma breve a fin de dejar constancia de los riesgos existentes en este tipo de entornos y del impacto y alcance de los mismos.

2.4.2. Fundamentos de encaminamiento en redes MANET

Una vez explicada la subcapa de acceso al medio utilizada, y teniendo en cuenta que las redes MANET están principalmente especificadas tanto en esta capa de enlace como en la capa de red, consideramos oportuno detallar el funcionamiento del protocolo de encaminamiento empleado. Así, uno de los principales métodos para clasificar los protocolos de encaminamiento específicos de redes ad hoc se basa en una categorización en función de cómo es adquirida y mantenida la información de *routing* por parte de los nodos [36] [37] [38]. Este método permite clasificar los protocolos en tres grupos principales: protocolos reactivos, proactivos e híbridos. Son como sigue:

- *Protocolos proactivos*: en este tipo de protocolos (denominados también *table-driven*), los nodos de la red intentan mantener información actualizada y consistente sobre el resto de los nodos (o al menos sobre parte de ellos), manteniendo una visión global de la topología completa de la red. Las rutas hacia el destino se determinan al inicio del despliegue de la red MANET y se almacenan en una o más tablas de rutas, que serán actualizadas a lo largo del tiempo mediante procesos periódicos de actualización de rutas. Además, la actualización de las rutas también puede venir determinada como respuesta a cambios en la topología de la red, permitiendo una rápida adaptación frente a cambios por parte de estos protocolos. De este modo, los nodos que desean transmitir pueden conocer la ruta hacia el destino de forma inmediata. Sin embargo, esta actualización periódica provoca la introducción de una gran sobrecarga en las comunicaciones (*overhead*) y un alto consumo energético, además de una menor eficiencia, al ser necesario el mantenimiento de todas las rutas independientemente de si por ellas está transitando tráfico o no. Algunos protocolos proactivos típicos son OLSR (*Optimized Link State Routing*) [39], DSDV (*Destination Sequence Distance Vector*) [40] y WRP (*Wireless Routing Protocol*) [41].
- *Protocolos reactivos*: estos se diseñaron con el objetivo de reducir el *overhead* introducido por los protocolos proactivos. De este modo, las rutas son obtenidas únicamente cuando un nodo desea originar una comunicación y no conoce la ruta deseada. Para ello se hace necesaria la existencia de procedimientos que permitan realizar el descubrimiento de las rutas bajo demanda, generalmente

mediante procesos de *inundación*. Estos procedimientos finalizan cuando se encuentra una ruta válida o cuando todas las posibles permutaciones han sido examinadas. Una vez que la ruta se ha establecido se hace necesario mantenerla, pues pueden producirse desconexiones causadas por la naturaleza cambiante de estas redes. Para ello se realiza el mantenimiento de la ruta mientras que esta permanezca activa, es decir, hasta que el destino sea inaccesible o hasta que la ruta ya no sea necesaria. Los protocolos reactivos introducen un menor *overhead*, haciendo un uso más eficiente del ancho de banda y siendo, en consecuencia, más escalables que los proactivos. Sin embargo, los nodos origen pueden sufrir mayores retardos causados por la búsqueda de las rutas. De entre los numerosos protocolos reactivos caben destacar AODV [15], DSR [42] y DYMO [43].

- *Protocolos híbridos*: los protocolos híbridos combinan las características básicas de las dos categorías anteriores, es decir, tienen un comportamiento tanto reactivo como proactivo, con el objetivo de lograr los beneficios de ambas clases y superar sus limitaciones. Normalmente se basan en arquitecturas jerárquicas, de modo que se realiza un tipo de encaminamiento u otro en función del nivel jerárquico. Así, los nodos que se encuentran próximos entre sí (intra-zona) trabajan de forma conjunta para mantener, proactivamente, las rutas entre ellos; mientras que las rutas hacia nodos "lejanos" (inter-zona) se obtienen de forma reactiva bajo demanda, mediante procesos de descubrimiento. La mayoría de los protocolos híbridos particionan la red en distintos grupos, de modo que los nodos se pueden agrupar por zonas, *clusters*, árboles, etc. Algunos de los protocolos híbridos más conocidos son ZRP (*Zone Routing Protocol*) [44] y CBRP (*Cluster-Based Routing Protocol*) [45].

Es de destacar también la existencia de diversas clasificaciones alternativas, como las presentadas en [46], [47] o [48]. A pesar de ello, en este trabajo de tesis se ha considerado la aproximación previamente explicada, al ser una organización más ampliamente reconocida y aceptada actualmente por la comunidad investigadora.

El protocolo de encaminamiento AODV

Como se ha indicado previamente, AODV [15] es el protocolo de *routing* sobre en el que centraremos nuestra atención en este trabajo de tesis. AODV es un protocolo de encaminamiento para redes MANET de tipo reactivo, es decir, las rutas hacia un destino determinado se obtienen bajo demanda cuando son necesarias. Esta aproximación minimiza el número de paquetes de control precisos para establecer y mantener dichas rutas, mejorándose así la escalabilidad y el rendimiento del protocolo, al tiempo que se reduce el *overhead* introducido. Sin embargo, estas características implican, a su vez, mayores retardos a la hora de realizar la primera conexión entre un origen y un destino dados.

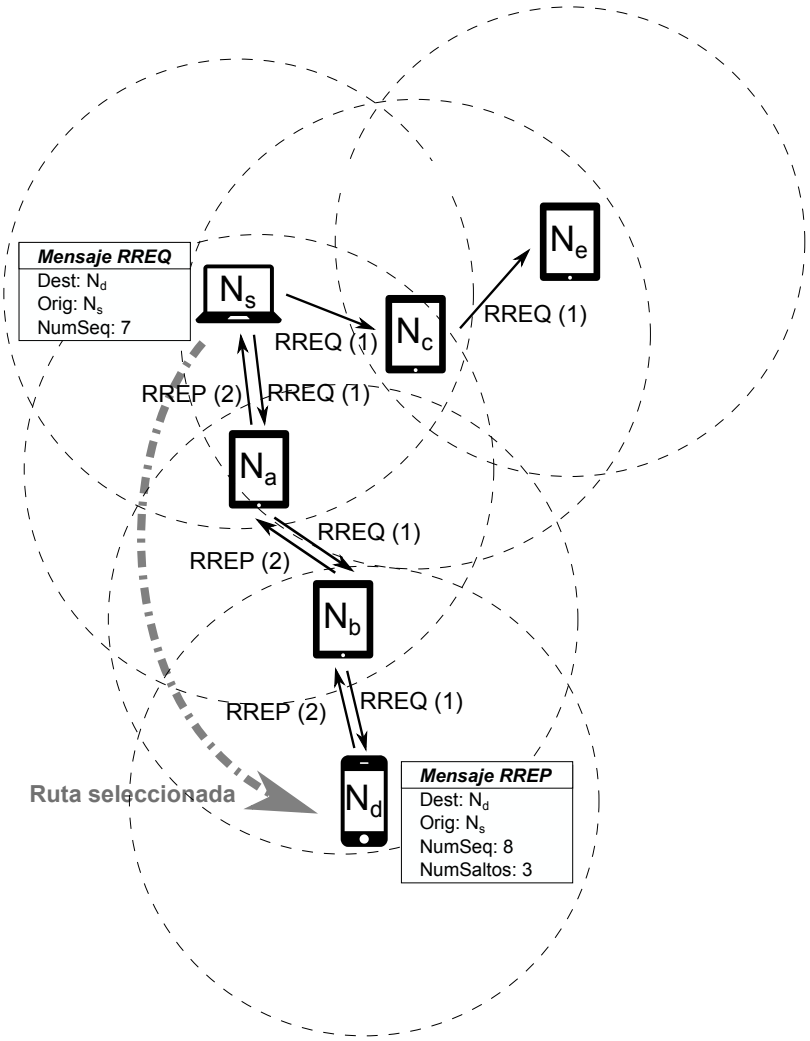


Figura 2.6: Ejemplo de proceso de descubrimiento de ruta en AODV.

Así, si un nodo N_s desea establecer comunicación con un nodo destino N_d y no posee una ruta válida hacia este, el nodo N_s inicia un *proceso de descubrimiento de ruta* (véase Figura 2.6) mediante un proceso de inundación o *flooding*, es decir, mediante la difusión en modo *broadcast* de un paquete de *solicitud de ruta* o RREQ (*Route REQuest*) (1). Al recibir este mensaje RREQ, los vecinos del nodo N_s (nodos intermedios) lo retransmiten a su vez a sus propios vecinos, repitiéndose el proceso hasta que el paquete RREQ alcanza el destino pretendido. Una vez que el nodo destino N_d recibe el primer RREQ, envía un *paquete de respuesta de ruta* o RREP (*Route REPly*) de vuelta al origen a través de la ruta inversa por la que recibió el RREQ (2), siendo los mensajes RREQ recibidos con posterioridad ignorados por N_d. Además de este proceso de descubrimiento, AODV permite que los nodos intermedios que conozcan una ruta válida hacia el destino solicitado por el paquete RREQ generen y envíen de

vuelta hacia el origen mensajes RREP como respuesta. De esta forma, tanto los nodos origen como los nodos intermedios son responsables de almacenar y gestionar, para cada flujo de comunicación, la información de encaminamiento relativa al siguiente salto en dicha ruta hacia el destino.

Para mantener la coherencia en las rutas y evitar bucles y otros problemas asociados a los protocolos clásicos basados en vector-distancia (como la "cuenta al infinito"), AODV utiliza *números de secuencia* ($NumSeq$), *i.e.*, unos identificadores o números monótonamente crecientes con un significado global en la red, que son especificados para cada ruta. Estos números de secuencia permiten a los nodos determinar cómo de "reciente" es la información que poseen. Cada vez que un nodo envía un paquete de control, incrementa su número de secuencia. Además, los nodos almacenan los números de secuencia de todos los otros nodos con los que mantienen comunicación, por lo que cada nodo incluirá el número de secuencia observado para un nodo destino determinado en su correspondiente entrada en la tabla de rutas. Este número de secuencia será actualizado cada vez que el nodo reciba información nueva (*i.e.*, no caducada) en mensajes de control relacionados con dicho destino. De este modo, un nodo solo actualizará su información de ruta si el número de secuencia del mensaje RREP recibido es mayor que el último número de secuencia almacenado, o igual a este pero con un menor número de saltos, lo cual indicará una ruta más reciente o mejor. Si se da el caso de que un nodo tiene que decidir entre dos posibles rutas hacia un destino, el nodo solicitante deberá seleccionar aquella con un mayor número de secuencia.

De este modo, las tablas de rutas de los nodos en AODV contienen la siguiente información: destino, siguiente salto ($NextHop$), distancia al destino medida en número de saltos ($HopCount$), estado (VAL -válida- o INV -caducada-) y número de secuencia ($NumSeq$), además de otros parámetros usados por AODV como el tiempo de vida de la ruta, una serie de banderas o *flags*, la interfaz de salida, etc. La Figura 2.7 muestra un ejemplo simplificado de una tabla de rutas de un nodo, mostrando los campos de interés que serán empleados a lo largo de este trabajo.

A partir de lo anterior, el hecho de que la selección entre dos posibles rutas deba resultar en aquella con mayor número de secuencia puede ser fácilmente explotado por potenciales nodos maliciosos para lograr introducirse en la ruta hacia el destino, como se verá más adelante.

Mantenimiento de rutas en AODV

Además de todo lo antes descrito, para que el protocolo funcione correctamente es necesario que realice un *proceso de mantenimiento de ruta*, mediante el cual cada nodo mantiene un registro de los nodos con los que es capaz de comunicarse directamente, los cuales son considerados sus vecinos. Dicho registro puede realizarse mediante la

Dest	NextHop	HopCount	Estado	NumSeq
2	12	2	VAL	1
3	3	1	VAL	3
6	10	2	INV	4
9	12	2	INV	4
10	10	1	VAL	3
12	12	1	VAL	5
15	12	3	VAL	6

Figura 2.7: Ejemplo simplificado de tabla de rutas en AODV para un nodo dado.

escucha de paquetes HELLO propios de AODV, que son difundidos periódicamente por los nodos con rutas activas a los vecinos (a un único salto), fijando el campo TTL (*Time To Live*) de la cabecera IP al valor 1 y utilizando la dirección IP de *broadcast* (255.255.255.255).

Sin embargo, para evitar un consumo innecesario de ancho de banda y de batería a causa del envío/recepción de estos mensajes HELLO, es común en redes MANET utilizar un procedimiento basado en capa de enlace para realizar la actualización de la lista de vecinos. Así, cuando un nodo determina la disponibilidad del medio y envía paquetes RTS para transmitir un paquete de datos, el procedimiento de mantenimiento comprueba si el mecanismo IEEE 802.11 RTS/CTS ha alcanzado el número máximo de retransmisiones, es decir, el número máximo de paquetes RTS enviados sin una respuesta CTS por parte del siguiente salto en la ruta. Por defecto, el máximo número de retransmisiones en IEEE 802.11 se produce cuando el parámetro SRC (*Short Retry Count*) alcanza el límite SRL (*Short Retry Limit*), cuyo valor es 7. En caso de superarse dicho valor, AODV considera que el enlace está roto e inicia un procedimiento de mantenimiento. Cuando dicho procedimiento comienza pueden aparecer dos posibles situaciones (Figura 2.8):

- *Escenario 1*: si el enlace roto se encuentra más cerca del nodo origen que del nodo destino, el nodo intermedio que ha detectado el fallo en el enlace marcará la ruta como inválida en su tabla e inmediatamente después enviará hacia el origen un mensaje de error, RERR (*Route ERRor*), alertando a sus precursores acerca del enlace fallido. En tal caso, los nodos precursores dejarán de enviar paquetes al nodo intermedio, y retransmitirán recursivamente el mensaje RERR.
- *Escenario 2*: en el caso de que el enlace fallido se encuentre más cerca del nodo destino, el nodo intermedio intentará realizar una *reparación local* de la ruta, enviando para ello un mensaje de solicitud, RREQ, tal y como lo haría el propio nodo origen. Si tras un cierto tiempo la ruta no ha podido ser reparada, el nodo

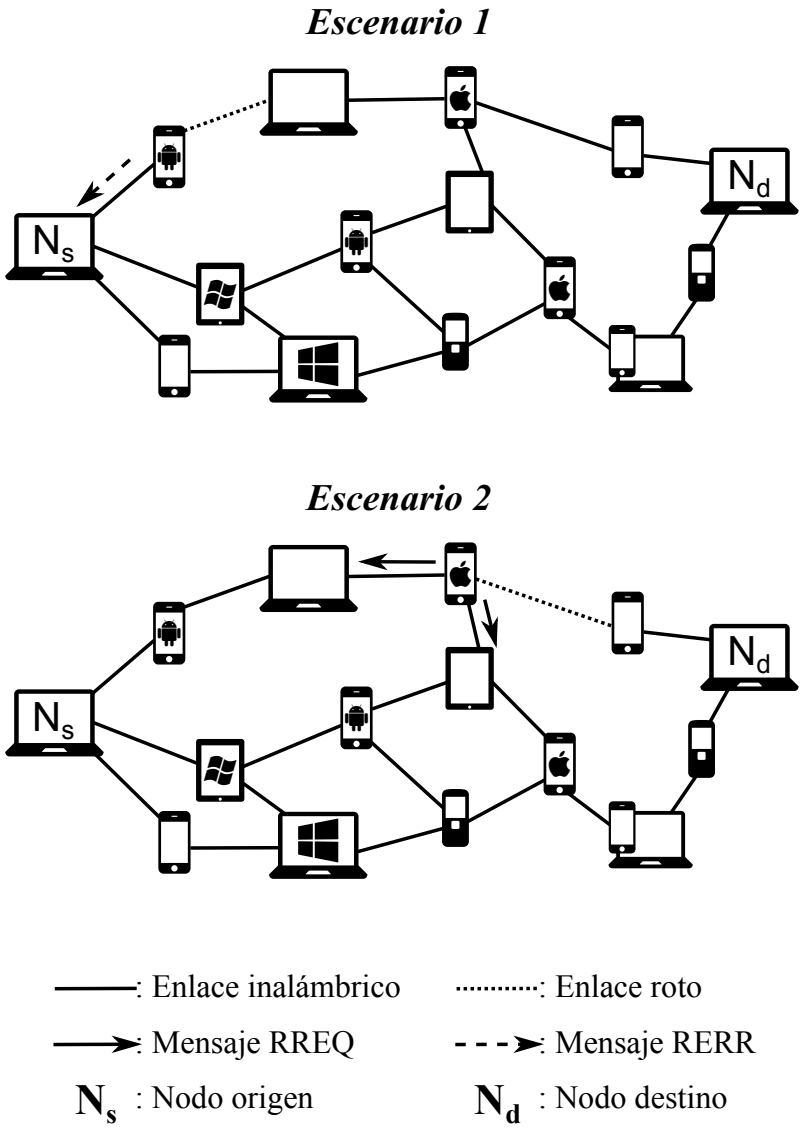


Figura 2.8: Posibles escenarios que resultan tras fallar el mecanismo RTS/CTS.

marcará la ruta como inválida y enviará un mensaje RERR a sus precursores, de modo análogo al caso del *escenario 1*.

Nótese que, durante un cierto tiempo, el nodo intermedio que ha detectado el enlace como roto continuará recibiendo mensajes que será incapaz de retransmitir, *i.e.*, el nodo se comportará de un modo *anómalo*, descartando paquetes. Este período de tiempo será considerablemente mayor en el caso de producirse la situación explicada en el *escenario 2*, puesto que el proceso de mantenimiento de ruta puede durar hasta docenas de segundos antes de que se genere un mensaje RERR.

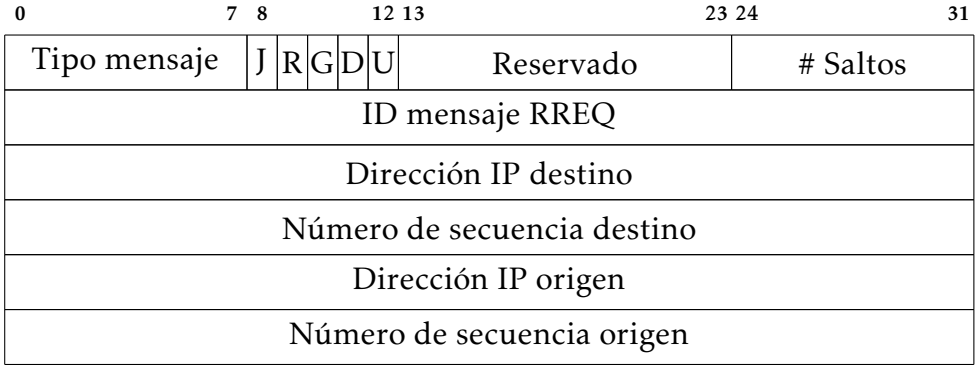


Figura 2.9: Formato de los mensajes de solicitud, RREQ.

Formato de los mensajes en AODV

A continuación se presenta el formato de los paquetes de control utilizados por AODV. De los distintos campos presentes en los mensajes se detallarán exclusivamente aquellos necesarios para la correcta comprensión de las actuaciones que serán presentadas posteriormente en este trabajo de tesis.

La Figura 2.9 presenta el formato de los paquetes de solicitud de ruta, RREQ. Los campos de interés son los siguientes:

- *Tipo mensaje*: para diferenciar entre los distintos mensajes. RREQ = tipo 1.
- *# Saltos*: número de saltos desde el nodo origen al nodo actual.
- *ID mensaje RREQ*: número monótonamente creciente que, junto con la dirección IP del nodo origen, identifica unívocamente el mensaje RREQ.
- *Dirección IP destino*: dirección IP del nodo para el que se desea conocer la ruta.
- *Número de secuencia destino*: el último número de secuencia recibido en el pasado por el origen, para cualquier ruta hacia el destino.
- *Dirección IP origen*: dirección IP del nodo que origina la solicitud de ruta.
- *Número de secuencia origen*: el número de secuencia actual que se usará en la entrada de la tabla de rutas hacia el origen del mensaje RREQ.

Por su parte, el formato de los paquetes de respuesta de ruta, RREP, se muestra en la Figura 2.10, siendo los principales campos:

- *Tipo mensaje*: RREP = tipo 2.

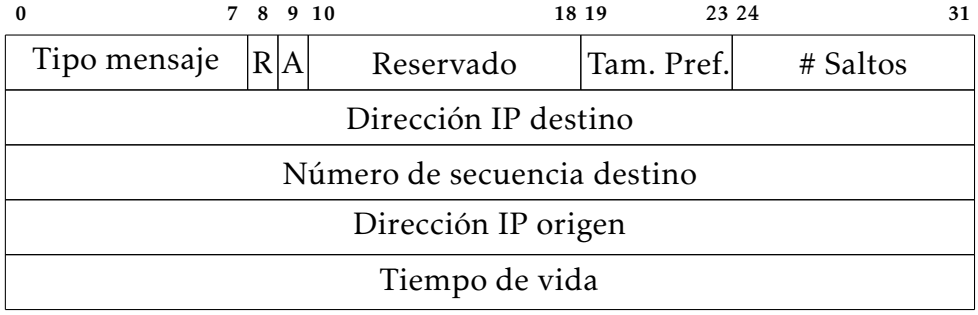


Figura 2.10: Formato de los mensajes de respuesta, RREP.

- *# Saltos*: número de saltos desde el nodo origen al nodo destino.
- *Dirección IP destino*: dirección IP del nodo para el que se proporciona la ruta.
- *Número de secuencia destino*: número de secuencia asociado a la ruta dada.
- *Dirección IP origen*: dirección IP del nodo que originó la solicitud y para el que se proporciona la ruta.
- *Tiempo de vida*: tiempo, en milisegundos, durante el cual el nodo que recibe el mensaje RREP considerará la ruta como válida.

Por último, el formato de los paquetes de error, RERR, es el indicado en la Figura 2.11. Los campos más relevantes de este tipo de mensaje son:

- *Tipo mensaje*: RERR = tipo 3.
- *# Destinos*: número de destinos inalcanzables incluidos en el mensaje.

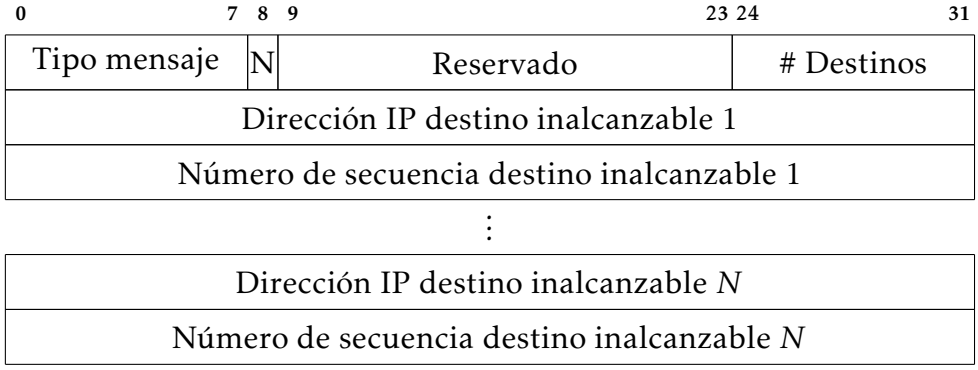


Figura 2.11: Formato de los mensajes de error, RERR.

- *Dirección IP destino inalcanzable 1...N*: dirección IP del nodo inalcanzable por rotura del enlace.
- *Número de secuencia destino inalcanzable 1...N*: número de secuencia en la entrada de la tabla de rutas para el destino listado.

Otros protocolos de encaminamiento en redes MANET

Aunque nuestro estudio se centrará en el protocolo AODV, los ataques objeto de estudio de este trabajo de tesis pueden ser implementados fácilmente en otros protocolos similares, como DYMO [43], DSR [42], ODMRP (*On-Demand Multicast Routing Protocol*) [49] o muchas de las extensiones de AODV, como MAODV (*Multicast AODV*) [50]. A continuación se presenta una breve descripción a modo orientativo de cada uno de ellos.

DSR es un protocolo reactivo basado en encaminamiento del origen (*source-initiated*), *i.e.*, cada paquete de datos contiene en su cabecera la lista completa y ordenada de nodos a atravesar en su ruta hasta el destino. DSR, al igual que AODV, también ejecuta un mecanismo de descubrimiento de rutas por inundación, e incluye unos *identificadores de solicitud* en cada mensaje RREQ. Además, los nodos deben mantener una *cache* para almacenar las rutas aprendidas recientemente. Sin embargo, DSR introduce un mayor consumo de ancho de banda y sufre problemas de escalabilidad debido al encaminamiento desde el origen, necesitando de una cantidad de recursos significativamente mayor que otros protocolos.

DYMO es el sucesor de AODV, operando de forma similar y manteniendo el modo de operación básico de este. Aunque DYMO trata de simplificar AODV en lugar de extenderlo o añadirle nuevas características, hereda la "acumulación de ruta" de DSR, es decir, permite que los nodos intermedios incluyan entradas subsiguientes en los mensajes de control, en las que proporcionan información de encaminamiento acerca de ellos mismos. Por otro lado, DYMO no tiene soporte para enlaces asimétricos, además de no ser tan conocido ni usado como AODV.

La extensión *multicast* de AODV, MAODV, construye un árbol bidireccional compartido y *hard-state*, es decir, que debe lidiar de forma explícita con redes particionadas y con roturas de enlaces, forzando la reparación del árbol. En él, el líder de cada grupo *multicast* envía mensajes HELLO de grupo de forma periódica para mantener la información actualizada. Sin embargo, puesto que MAODV envía los mensajes RREP de forma *unicast* hacia el origen, si algún nodo intermedio se mueve o falla, el mensaje se pierde y, consecuentemente, la ruta también se perderá. Ello provoca la introducción de una mayor sobrecarga.

Por su parte, ODMRP emplea una aproximación *soft-state* para mantener una topología en malla, y utiliza el concepto de "grupo de retransmisión" en el que solo

un subconjunto de los nodos retransmiten los paquetes *multicast*. Además envía los paquetes de respuesta en modo *broadcast*, permitiendo el aprendizaje de rutas redundantes, lo que resulta en una mayor robustez frente a movilidad y unas pérdidas mínimas de datos. Estas respuestas *broadcast* incrementan sin embargo la sobrecarga. Además, las rutas son actualizadas de forma periódica por el origen y, dependiendo de este intervalo de actualización, la sobrecarga de paquetes de control puede dar lugar a problemas de escalabilidad.

Aunque hay diferencias significativas entre AODV y el resto de protocolos, todos comparten algunos conceptos relevantes. Específicamente, todos ellos emplean algún tipo de identificador necesario para evitar bucles y determinar cómo de recientes son las rutas, cuya aplicación es muy similar a la de los números de secuencia de AODV. Como se verá en la siguiente sección, estos identificadores pueden ser explotados explícitamente por parte de los nodos maliciosos para llevar a cabo diversos ataques. Desde esta perspectiva, los esquemas de detección presentados en la parte subsiguiente y principal de este trabajo de tesis podrían ser fácilmente extendidos a otros protocolos sin más que realizar algunas ligeras modificaciones.

2.5. Ataques de *dropping* y *sinkhole* en redes MANET

Una vez detallados los fundamentos de operación de las redes MANET, a continuación se describen los ataques para los que se van a desarrollar los mecanismos de detección propuestos en los siguientes capítulos.

De entre las muchas amenazas existentes en las redes MANET [9] [51], dos de las clases de ataque que se consideran potencialmente más perjudiciales (sin menospreciar las demás) son las de *dropping* y de *poisoning* (véase la Figura 2.2). En la primera los nodos descartan paquetes de datos en lugar de retransmitirlos. La segunda clase incluye ataques consistentes en la modificación, creación o eliminación de paquetes de *routing*, con la intención de alterar el correcto funcionamiento de los protocolos de encaminamiento. La consecuencia directa de ambas clases de ataques es la interrupción y/o degradación de la operación normal de la red y sus servicios.

Este trabajo se centrará, por una parte, en la detección genérica de aquellos ataques pertenecientes a la clase *dropping*, incluyendo ataques *blackhole*, *grayhole*, *selfish*, etc. También de relevancia, en lo que respecta a la segunda clase de ataques, nos centraremos en el estudio específico del ataque *sinkhole*, posiblemente uno de los ataques de *route poisoning* más representativos. Seguidamente se describen brevemente, aunque en mayor detalle que hasta ahora, ambos tipos de ataque citados, especialmente en relación a los entornos de trabajo aquí considerados: MANET. Ello nos permitirá afrontar con mayor confianza las subsiguientes actuaciones realizadas en este trabajo.

2.5.1. El ataque de *dropping* en MANET

Como ya se ha comentado, los ataques de *dropping* son una de las amenazas más perjudiciales en las redes MANET. Así, los nodos que exhiben este comportamiento malicioso descartan los paquetes de datos o de control recibidos en vez de retransmitirlos, afectando al rendimiento normal de la red [12]. Son diferentes las variantes de ataques que se pueden clasificar como ataques de *dropping*, dependiendo de la estrategia particular adoptada. Los más populares son los ataques *blackhole* y *grayhole*. El primero se refiere a aquellos ataques en los que los nodos descartan completamente todos los paquetes recibidos. El segundo es causado por un descarte selectivo de paquetes, *p.ej.*, un paquete descartado de cada N recibidos, uno cada cierto tiempo, solo los paquetes correspondientes a un determinado flujo, etc.

Son varias las motivaciones que puede tener un nodo para tratar de evadir su responsabilidad en la retransmisión de los paquetes de la red. Por ejemplo, un nodo podría rechazar reenviar los paquetes con la intención de preservar o economizar sus recursos (energéticos, de procesamiento, etc.). Estos nodos suelen denominarse nodos *egoístas* (*selfish*). Por otro lado, un nodo llevando a cabo un ataque de *dropping* podría tener intenciones maliciosas, buscando interrumpir la disponibilidad de un servicio dado.

Así, aunque el daño específico causado depende del nivel de descarte implementado en cada caso (*p.ej.*, indiscriminado *vs.* selectivo, comportamiento malicioso *vs.* egoísta), el potencial impacto y relevancia de estos ataques en las comunicaciones es incuestionable [52].

Cabe destacar por otro lado que la implementación de ataques de *dropping* en MANET es extremadamente sencilla. Un nodo atacante no tiene más que modificar la función de retransmisión del protocolo IP para que realice el descarte de paquetes de la forma deseada (total, parcial, selectiva, etc.).

Para conseguir un mayor impacto en el rendimiento de la red, es usual que los nodos que llevan a cabo ataques de *dropping* intenten previamente introducirse en las rutas de comunicación, con la intención de atraer todo el tráfico hacia ellos. Dicho ataque, denominado comúnmente como ataque *sinkhole*, se detalla a continuación.

2.5.2. El ataque *sinkhole* en MANET

Como ya se ha indicado, además del *dropping*, los ataques de *poisoning* también se encuentran entre las amenazas más dañinas en entornos MANET. De entre ellos, nos centraremos en el ataque *sinkhole*, uno de los más representativos de esta clase. Los nodos que exhiben este comportamiento malicioso intentan falsear las rutas origen-destino para así atraer hacia ellos el tráfico circundante. Con este propósito,

modifican los paquetes de control del protocolo de encaminamiento, publicando información de *routing* falsa (número de saltos hasta el destino, números de secuencia, calidad del enlace, etc.) que los haga aparecer como la ruta más atractiva hacia ciertos destinos. De esta manera consiguen que otros nodos legítimos los elijan como siguiente salto en el proceso de retransmisión de la información.

Existen distintas motivaciones por las que un nodo puede llevar a cabo un ataque *sinkhole*. Por ejemplo, un propósito sería simplemente realizar escuchas de los datos recopilados para extraer información sensible de los mismos. Por otro lado, un nodo malicioso podría descartar o modificar ciertos paquetes para degradar el rendimiento y la eficiencia de la red. De hecho, un ataque *sinkhole* podría realizarse como paso previo para la realización de ataques más complejos, como por ejemplo ataques *wormhole* o *blackhole*. En el caso de que el nodo *sinkhole* no realice ninguna de las mencionadas acciones subsiguientes, como la modificación o el descarte, su interacción con la red será menor y, en consecuencia, su detección (y posible solución) resultará más compleja.

Una vez se conocen los conceptos básicos de AODV (detallados en la Sección 2.4.2), es sencillo comprender cómo un nodo malicioso puede aprovecharse del funcionamiento de este protocolo para realizar el ataque *sinkhole*. Para ello, el nodo podría crear o modificar un mensaje RREP para publicar una ruta hacia un destino dado indicando una métrica óptima, es decir, un mínimo número de saltos hacia el destino, así como un número de secuencia mayor que los previamente recibidos en los mensajes RREQ. Si el número de secuencia es suficientemente grande, se invalidarán las posibles rutas alternativas que publiquen otros nodos con rutas válidas hacia el destino. Como consecuencia de esto, el nodo malicioso (*sinkhole*) consigue que el nodo origen aprenda que la mejor ruta para alcanzar el citado destino es a través suyo, siendo así seleccionado como el siguiente salto en la ruta. Si el nodo *sinkhole* responde con mensajes RREP falsos a cualquier petición de rutas RREQ que reciba, terminará convirtiéndose en un sumidero de tráfico pues gran parte de los paquetes de datos de la red serán encaminados a través de él.

La Figura 2.12 muestra un ejemplo del ataque *sinkhole*. En este caso, el nodo origen N_s envía en modo *broadcast* una solicitud RREQ (1), solicitando una ruta hacia el destino N_d. Este mensaje es retransmitido por los nodos intermedios (N_a, N_b y N_c en la Figura 2.12). Cuando este paquete RREQ alcanza el nodo malicioso N_m, este responde hacia el nodo origen N_s con un mensaje RREP falso (2a), asegurando disponer de una ruta hacia el destino N_d más corta ($HopCount = 1$) y más reciente, indicando un número de secuencia mayor que el proporcionado por otros nodos ($SeqNum = 37$). Al mismo tiempo, el nodo destino N_d responde con su propio mensaje RREP (2b) que incluye los valores $HopCount$ y $NumSeq$ legítimos (3 y 8, respectivamente).

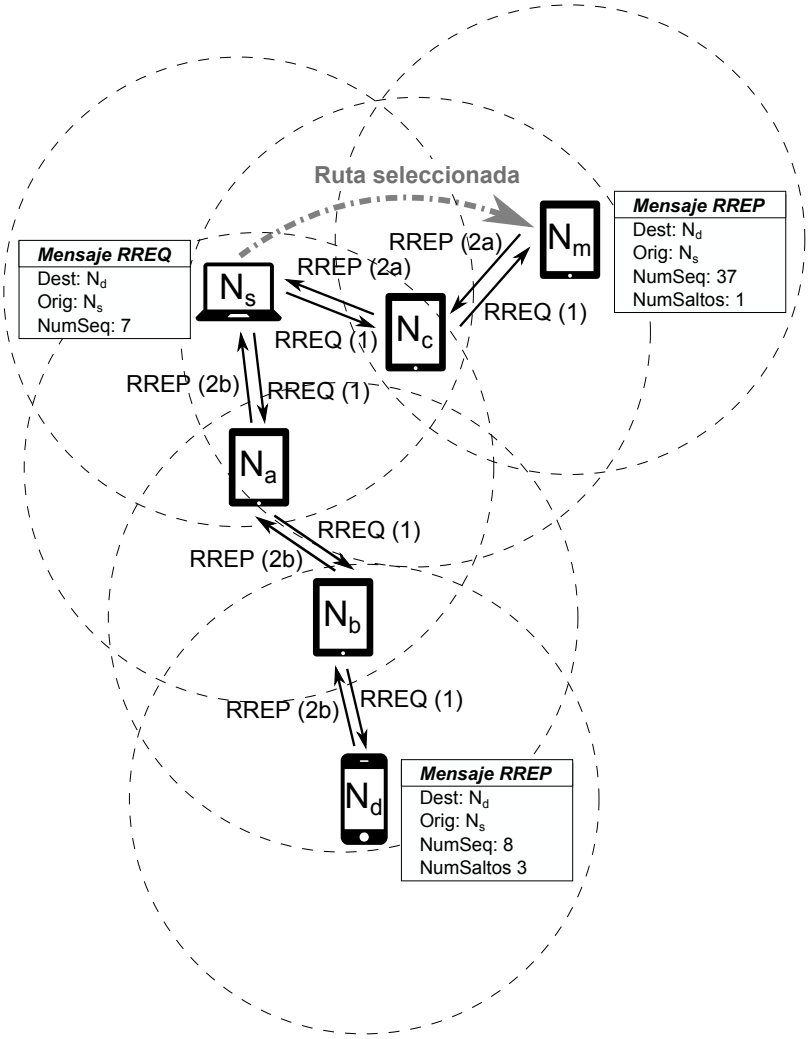


Figura 2.12: Ejemplo de nodo *sinkhole* respondiendo con un falso mensaje RREP.

Así, a pesar de recibir otras respuestas legítimas, el nodo origen N_s elegirá la ruta a través de N_c, al ser considerada como la más reciente y la que posee una métrica de distancia óptima. Finalmente, el tráfico desde el origen N_s hasta el destino N_d terminará atravesando el nodo malicioso N_m, aún cuando este no conoce realmente la ruta hacia el destino.

Tal y como se ha comentado previamente, el ataque *sinkhole* es fácilmente ejecutable en diversos protocolos, como DSR, DYMO, ODMRP, MAODV u otros, pues todos ellos comparten la característica común de emplear identificadores (números de secuencia) para evitar bucles en las rutas. Así, la ejecución del ataque se podría realizar sin más que explotar dichos identificadores de forma similar a como se realiza con los números de secuencia en AODV.

2.6. Conclusiones del capítulo

Los conceptos presentados en este capítulo constituyen la base de estudio sobre la cual se sustentan los sistemas de detección propuestos en los Capítulos 4 y 5. Así, en este capítulo se incide en la relevancia que está cobrando la seguridad en las redes de comunicación y, en particular, en las redes MANET, hecho que se demuestra mediante el gran número y tipos de amenazas que pueden surgir en este tipo de entornos.

Se ha presentado así una revisión de los ataques en redes MANET existentes en la actualidad, proponiendo una nueva taxonomía cuya aplicación agrupa y organiza dichos ataques en clases genéricas que comparten una serie de características comunes, permitiendo el diseño y desarrollo potencial de mecanismos defensivos más prácticos y eficientes.

Posteriormente, y con el fin de afrontar con mayores garantías la comprensión de las contribuciones presentadas con posterioridad en este trabajo de tesis, se detallan algunos de los fundamentos básicos de dos de los protocolos más empleados en redes MANET para la gestión de las capas MAC y de red: IEEE 802.11 y AODV, respectivamente.

Finalmente se ha presentado el funcionamiento e implementación específicas de los ataques de *dropping* y *sinkhole*, dos de las amenazas más perjudiciales en entornos MANET, en cuya detección se centrará la parte principal de este trabajo de tesis.

Publicaciones relacionadas

A pesar de que ya han sido indicadas en el Capítulo 1, se presentan aquí las publicaciones directamente relacionadas y realizadas en el marco de este capítulo. Estas son:

- P. García-Teodoro, **L. Sánchez-Casado** y G. Maciá-Fernández. "Taxonomy and Holistic Detection of Security Attacks in MANETs". *Security for Multihop Wireless Networks*, S. Khan y J. Lloret (Eds.), CRC Press, pp. 1-12, 2014.

Parte II

Detección de Ataques en Redes MANET

Capítulo 3

Defensas en redes MANET

Debido a su propia naturaleza, las redes ad hoc tienen una serie de problemas de seguridad inherentes asociados. La proliferación de estos sistemas en estos últimos años ha provocado una magnificación de los efectos y la relevancia del asunto, siendo imprescindible que dichos problemas sean tratados convenientemente. En este sentido, se han llevado a cabo por parte de la comunidad investigadora multitud de esfuerzos con el objetivo de prevenir, detectar y, consecuentemente, responder frente a estos ataques, dando como resultado un elevado incremento en el número de publicaciones relacionadas con ellos.

En este capítulo se presenta una revisión de las contribuciones más relevantes encontradas en la literatura especializada y centradas en dar solución a los ataques de *dropping* y *sinkhole* en redes MANET, por ser estos los ataques sobre los que se centrarán los sistemas de detección propuestos. La revisión bibliográfica presentada se divide en tres categorías, en las que son clasificados los trabajos según la línea defensiva a que estos se refieren: prevención, detección y respuesta.

El resto del capítulo se estructura de la siguiente forma. En primer lugar, en la Sección 3.1 se indica la problemática actual que suponen los ataques de *dropping* y *sinkhole* y se motiva la necesidad de disponer de sistemas de defensa frente a estos. Seguidamente se realiza una revisión bibliográfica de los trabajos más relevantes en el campo de la defensa en redes MANET. En la Sección 3.2 se presentan los esquemas de prevención estudiados, clasificados en distintas categorías según su mecanismo de actuación. De modo similar, las soluciones de detección se estudian y categorizan en distintas clases en la Sección 3.3. La Sección 3.4 introduce los sistemas

defensivos destinados a ejecutar una respuesta frente al ataque detectado. Por último, las conclusiones principales del presente capítulo se exponen en la Sección 3.6.

3.1. Motivación

Las redes inalámbricas, y en particular las redes MANET, son vulnerables a diversas amenazas de seguridad específicas [53] [54] [55] [56] [57], de entre las que destacan dos: los ataques de *dropping* y *sinkhole*.

Como se vio en la Sección 2.5, existen diversas *especies* de ataques que se basan en el descarte de paquetes (como *blackhole*, *greyhole* o *selfish*), que pueden ser agrupados en una *clase* común, denominada *dropping*, y que constituye una de las mayores amenazas en este tipo de entornos [52] [58].

Además, como paso previo a la ejecución por parte de un nodo malicioso de un ataque de *dropping*, es común que dicho nodo intente introducirse en las rutas origen-destino mediante la publicación de información de rutas falsa, con la intención de atraer el tráfico circundante hacia él, y así causar un mayor impacto sobre el rendimiento de la red.

Viendo el importante impacto de estos ataques en la red, es de destacar el enorme esfuerzo que se está realizando por parte de la comunidad investigadora para atajar este problema, incrementándose continuamente el número de propuestas publicadas en la literatura especializada.

Sin embargo, se pueden encontrar algunas limitaciones en la práctica totalidad de las contribuciones en este campo existentes. Primeramente, muchos de los trabajos publicados tratan únicamente aspectos parciales del problema. Por otro lado, algunos de ellos están limitados únicamente a un tipo particular de *dropping* (*blackhole*, *greyhole*, *selfish*), en vez de estudiarlos como una tipología global. Además, la mayoría de los trabajos se centran únicamente en dar respuesta a la prevención, detección o respuesta ante estos ataques, obviando las otras líneas de defensa.

Otro problema importante que generalmente afecta a los trabajos de seguridad en redes MANET, y que ya fue mencionado en el Capítulo 2, es la existencia de una cierta confusión en las especificaciones de los ataques, siendo la definición y el ámbito de cada ataque en particular erróneos en ocasiones o, cuando menos, "confusos".

Con el fin de intentar solventar algunas de estas limitaciones, aquí se presenta un detallado estado del arte que cubre las propuestas más relevantes existentes en la literatura en el campo de la defensa frente a los ataques de *dropping* y *sinkhole*. Cabe destacar que, aunque la contribución principal de este trabajo de tesis está relacionada con el desarrollo e implementación de esquemas de detección, una revisión

bibliográfica exhaustiva no debería obviar otras propuestas defensivas encaminadas a la prevención o a la respuesta. Así, con objeto de proporcionar una visión completa y organizada de la materia, el estudio se presenta en función de la línea de defensa específica considerada en cada caso. A pesar de que pueden proponerse organizaciones alternativas (*p.ej.*, la capa o protocolo afectado por el ataque), se considera la aproximación tradicional como una buena opción.

Así mismo, es destacable que algunas de las soluciones estudiadas, por su naturaleza, pueden encajar adecuadamente en más de una de las líneas de defensa propuestas. Por regla general, estas soluciones *híbridas* serán detalladas en cada una de las líneas sobre las que puedan ser consideradas, haciendo énfasis en el método llevado a cabo en cada una de ellas. Un caso particular es el de los esquemas basados en *reputación*, tal y como se detallará en las secciones correspondientes.

La Figura 3.1 muestra un esquema resumido de las distintas aproximaciones que se presentarán a continuación.

3.2. Revisión bibliográfica de sistemas de prevención

En esta sección se presentan los principales trabajos relacionados con el desarrollo de técnicas de prevención ante varios tipos de ataques, en particular ante *dropping* y *sinkhole*. Entre otros, los algoritmos criptográficos y los basados en créditos son dos de las aproximaciones más ampliamente usadas en la actualidad [24], aunque no las únicas.

En relación con algunos de los esquemas de prevención propuestos en la literatura, surge un aspecto de interés conectado con el propio concepto de prevención. En un sentido estricto, solo aquellos mecanismos o métodos que evitan la potencial aparición del ataque deben ser considerados como preventivos. Sin embargo, son numerosos los autores que etiquetan sus aproximaciones como preventivas a pesar de que realmente estén basadas en una fase de detección, por lo que estas serán incluidas en la categoría correspondiente. Por otro lado, según [59], una red ad hoc debería: "(*i*) *proporcionar mecanismos de seguridad efectivos que traten con los nodos maliciosos de la red*, y (*ii*) *fomentar la cooperación entre los nodos de la red ...*". Así, existen diversas aproximaciones que, a pesar de que fomentan o estimulan la cooperación de los nodos en el proceso de retransmisión pero no evitan la aparición de los ataques, serán incluidas en la categoría de prevención, al no involucrar mecanismos de detección o respuesta.

A partir de todo lo anterior se pueden identificar cuatro categorías de mecanismos preventivos, en función del método principal utilizado para lograr dicho objetivo: (*i*)

mecanismos basados en autenticación, (*ii*) basados en modificaciones del protocolo, (*iii*) basados en reputación y (*iv*) basados en créditos.

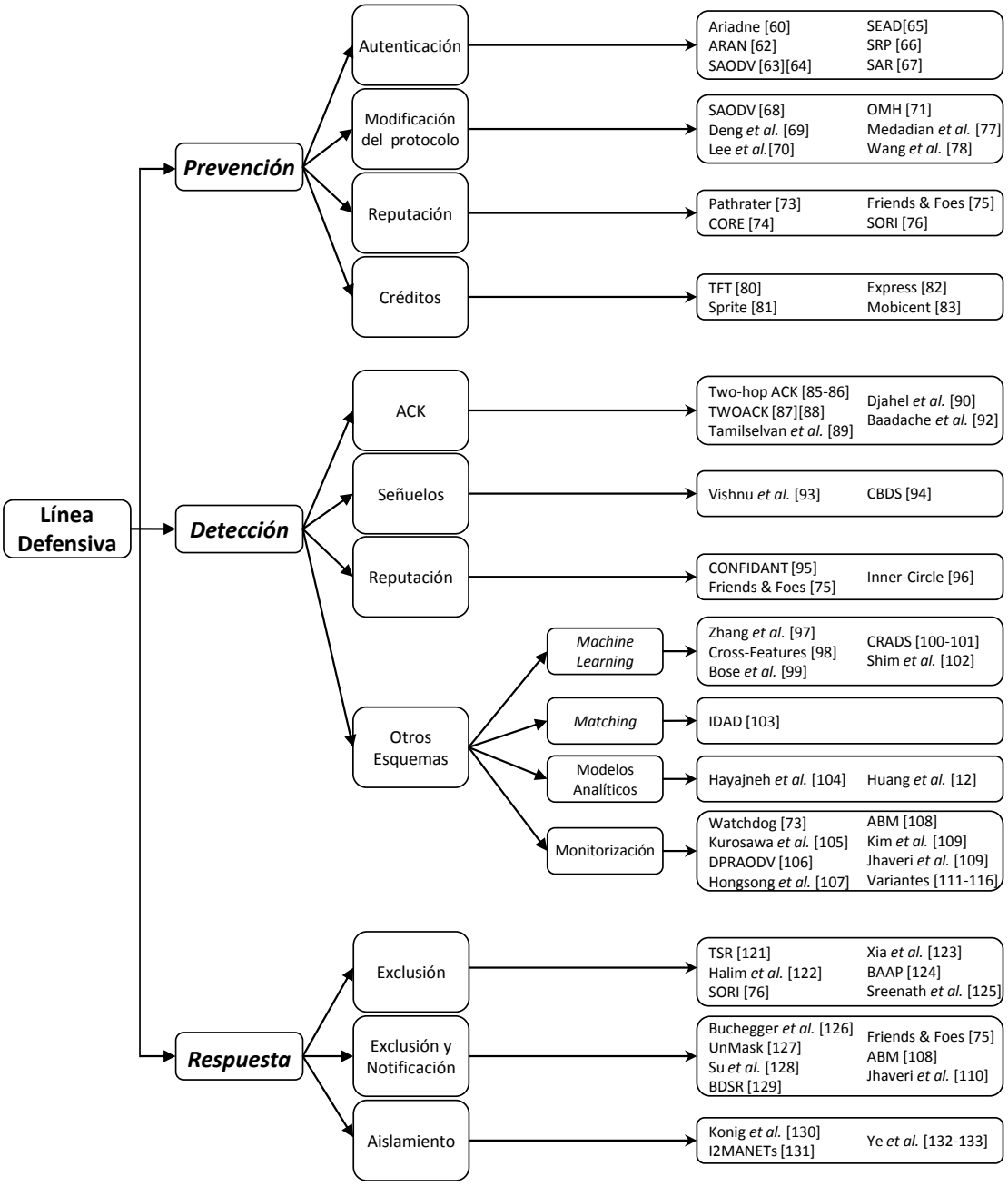


Figura 3.1: Esquemas defensivos en redes MANET.

3.2.1. Esquemas basados en *autenticación*

Los métodos incluidos en esta categoría proporcionan seguridad generalmente en base a proteger el protocolo o procedimiento de encaminamiento, *i.e.*, garantizando la corrección de las rutas anunciadas. Para ello se emplean principalmente técnicas de cifrado y gestión de claves, con la intención de asegurar la identidad de los nodos involucrados en la comunicación. Por tanto, su propósito primordial es evitar que nodos no autorizados puedan unirse a la red, lo que constituye una defensa frente a ataques externos.

La mayoría de técnicas en esta categoría se basan en autenticar el proceso de encaminamiento, como por ejemplo Ariadne [60], una extensión segura de DSR que permite la autenticación extremo-a-extremo de los mensajes de *routing* mediante el empleo de uno de tres posibles esquemas: claves compartidas entre pares de nodos, autenticación *broadcast* mediante TESLA [61] o firmas digitales.

En ARAN (*Authenticated Routing for Ad hoc Networks*) [62] se propone un esquema que utiliza autenticación salto-a-salto basada en criptografía de clave pública para asegurar que cada nodo conozca el siguiente salto en la ruta hacia un destino dado de forma correcta, obligando a los nodos a firmar los mensajes de solicitud y de respuesta de ruta. Para su correcto funcionamiento se requiere de la presencia de un servidor de certificados de confianza, de modo que los nodos deben solicitar un certificado a dicho servidor antes de acceder a la red.

Otra opción es SAODV (*Secure AODV*) [63] [64], una extensión de AODV basada en criptografía de clave pública que permite el firmado digital de los paquetes de AODV para garantizar su autenticidad e integridad. Además, incluye una *doble firma*, permitiendo que los nodos intermedios puedan responder a los mensajes RREQ.

Solo por citar algunos trabajos adicionales en esta línea, caben destacar SEAD (*Secure Efficient Ad hoc Distance*) [65], SRP (*Secure Routing Protocol*) [66] y SAR (*Security-Aware ad hoc Routing*) [67].

Este tipo de técnicas son adecuadas para evitar que nodos externos no autorizados sean capaces de interrumpir el funcionamiento de la red mediante el envío de información de encaminamiento falsa. Sin embargo, estos esquemas fallan cuando se produce un ataque "interno". Además, el uso de técnicas criptográficas constituye un inconveniente desde el punto de vista del consumo de recursos, tanto energéticos como computacionales. Si bien los esquemas basados en criptografía de clave pública evitan el intercambio de claves entre los nodos necesario en los esquemas simétricos, el coste y tiempo de procesamiento de estos es superior. Es por ello que suele optarse por aproximaciones híbridas, en las que el empleo de la criptografía asimétrica se limita a la generación de una clave privada de sesión, combinándose así las ventajas de ambos tipos de esquemas, de clave pública y privada.

3.2.2. Esquemas basados en *modificación del protocolo*

Gran parte de los protocolos de encaminamiento diseñados para entornos MANET no fueron originalmente concebidos teniendo en cuenta consideraciones de seguridad. Así, otra categoría de soluciones preventivas se basa en la introducción de los cambios necesarios en estos protocolos con el fin de solucionar las vulnerabilidades que posibilitan la ejecución de los ataques. Debe indicarse que, aunque en algunos casos se haga uso de cifrado, el objetivo principal en este tipo de aproximaciones no es garantizar la identidad de los nodos, sino la corrección de las rutas.

Existen diversas propuestas así en la literatura basadas en la solicitud explícita de confirmación de las rutas por parte de los nodos intermedios o de destino, o del envío de información adicional sobre estas para realizar comprobaciones. Un ejemplo es el caso de SAODV [68] (no confundir con la extensión SAODV propuesta en [63]), encaminado a contrarrestar ataques *sinkhole*. SAODV está basado en el empleo de dos nuevos paquetes, SRREQ (*Secure RREQ*) y SRREP (*Secure RREP*), y de un código secreto usado cada vez que se recibe un mensaje RREP. Cuando el nodo origen recibe al menos dos paquetes SRREP, selecciona la ruta más corta como la ruta segura hacia el destino.

Otro procedimiento similar se introduce en [69], el cual verifica la seguridad de la ruta solicitando a los nodos intermedios que envíen información del siguiente salto en los paquetes RREP. Así, el nodo que recibe el RREP puede enviar un paquete FRq (*Further Request*) al siguiente salto para verificar que este posee realmente rutas hacia el nodo intermedio por el que se pregunta y hacia el destino. El siguiente salto responderá con un mensaje FRp (*Further Reply*), que será ignorado por el nodo solicitante si se recibe desde el nodo intermedio por el que se pregunta, evitando así la recepción de mensajes FRp suplantados.

Alternativamente, la propuesta de Lee *et al.*[70] emplea paquetes CREQ (*Confirmation REQuest*) y CREP (*Confirmation REPly*) en DSR para confirmar la validez de la ruta. Los nodos intermedios que envían un RREP de vuelta al origen también deben enviar un mensaje CREQ a su siguiente salto hacia el destino. Este nodo, tras recibir el CREQ, también enviará un mensaje CREP de vuelta al origen, confirmando que posee una ruta válida hacia el destino, por lo que el nodo origen puede confirmar la validez del camino comparando la información recibida en los mensajes RREP y CREP.

Estos tres esquemas anteriores emplean realmente una combinación de detección y prevención. Su aproximación se basa en la detección de potenciales ataques *sinkhole*, previniendo de este modo la aparición de ataques de *dropping*. De este modo, es de destacar que esta aproximación híbrida bien podría tener cabida en la sección dedicada a las técnicas de detección.

Una aproximación alternativa se basa en "forzar" de algún modo la cooperación entre los nodos a través de la inclusión de nuevos mecanismos en el protocolo. Así, OMH (*One More Hop*) [71] emplea claves asimétricas para cifrar los paquetes, de modo que únicamente el siguiente salto a un nodo dado en la ruta es capaz de saber si el destino del paquete era el nodo previo o no. Por tanto, cada nodo que recibe un paquete tiene que retransmitirlo si desea ser informado por parte del siguiente salto sobre las claves necesarias para descifrar el paquete, en caso de ser el destino final del mismo.

La principal desventaja de todo este tipo de soluciones descritas es la alta sobrecarga introducida, especialmente en términos de mensajes necesarios, así como el elevado número de nodos involucrados en la comunicación. Por otro lado, y como se ha visto previamente, el uso de sistemas criptográficos introduce un problema adicional desde el punto de vista energético.

3.2.3. Esquemas basados en *reputación*

Los esquemas basados en reputación monitorizan el comportamiento de los nodos durante la operación de la red para asignarles un nivel de reputación o confianza. De este modo, solo aquellos nodos con un nivel de reputación adecuado podrán ser considerados válidos para realizar la retransmisión de los paquetes. Según Liu *et al.* [72], "*la reputación de un agente es una percepción relacionada con la normalidad de su comportamiento, obtenida por otros agentes en base a la experiencia y a observaciones de su comportamiento pasado*." Por tanto, para asignar un valor de reputación es necesario considerar dos componentes: (*i*) un modelo del comportamiento normal (o apropiado) de los nodos cuando retransmiten mensajes, y (*ii*) un modo de observar, medir y almacenar dichos valores de reputación.

Adicionalmente, hay otra cuestión que debe ser abordada al respecto de este tipo de procedimientos. Aunque en muchos de ellos no hay una fase explícita de detección de nodos maliciosos, el nivel de reputación de un nodo puede ser usado como un indicador de su comportamiento y, en consecuencia, es sencillo realizar una categorización de los nodos en legítimos o maliciosos. Es más, puesto que los nodos con valores de reputación insuficientes serán evitados en el proceso de retransmisión, estos procedimientos pueden ser también considerados en cierta medida como esquemas de respuesta. Sin embargo, se ha decidido incluir estas aproximaciones en la categoría de prevención dado que muchos de los esquemas se limitan únicamente a incluir los nodos que han demostrado ser confiables en los algoritmos de selección de ruta.

Una de las primeras propuestas en usar modelos de reputación es *Pathrater* [73], que fomenta la retransmisión de los paquetes por parte de los nodos para que se incre-

mente su calificación. Este esquema está basado en la detección de nodos maliciosos mediante el algoritmo *Watchdog*, que será explicado en secciones posteriores.

Algunas soluciones hacen uso de una autoridad de confianza para almacenar los valores de reputación, mientras que otras adoptan una aproximación de gestión descentralizada e introducen protocolos de recomendación para intercambiar la información de reputación. Como ejemplo, en CORE [74] cada nodo mantiene una tabla de reputación obtenida a través de la observación del comportamiento de los nodos vecinos y del intercambio de información con los nodos involucrados en cada operación. Cuando un nodo recibe un paquete que debe retransmitir, lo hará únicamente si el nodo que lo originó tiene reputación positiva.

Una aproximación similar es *Friends and Foes* [75], basado en el principio social de que la gente accede a cooperar en una determinada obligación siempre que considere que hay una distribución de tareas justa. Así, para construir una opinión para un nodo, cada nodo participante publica su conjunto de amigos (*friends*) y oponentes (*foes*), es decir, aquellos nodos a los que está o no dispuesto a ayudar en el proceso de retransmisión de paquetes.

Otro esquema descentralizado es SORI (*Secure and Objective Reputation-based Incentive*) [76], que propone un mecanismo de incentivos basado en reputación con el fin de evitar comportamientos egoístas y en el que los nodos intercambian la información de reputación únicamente con sus vecinos. Para ello, SORI descarta con cierta probabilidad los paquetes generados por un nodo egoísta. Así, valores pequeños de reputación causan mayores probabilidades de descarte, limitando a lo largo del tiempo la capacidad de transmisión de los nodos egoístas.

Un esquema híbrido es el introducido por Medadian *et al.* [77], que propone un protocolo de encaminamiento que incluye un método basado en reputación donde el origen genera una opinión acerca de los vecinos que han respondido con paquetes RREP.

En [78] los autores proponen un sistema de reputación para obtener rutas confiables en función de la confiabilidad de la comunicación y de la longitud de la ruta. Para ello, se evalúa el nivel de reputación en base a una serie de atributos (rango de transmisión, velocidad y número de nodos) y se introduce un modelo de recomendación para encontrar nodos confiables entre aquellos que son desconocidos, basado en el concepto de "similitud entre atributos".

3.2.4. Esquemas basados en *créditos*

En los esquemas basados en créditos cada nodo recibe un micro-pago por su cooperación en el proceso de retransmisión de paquetes, mientras que, a su vez,

deberá "pagar" para que otros nodos retransmitan sus mensajes. Se pueden aplicar dos modelos: uno basado en "monederos" de mensajes y otro basado en el "comercio" de mensajes [79]. En el primer modelo es el nodo origen el que tiene que pagar a los nodos intermedios por sus servicios de retransmisión, por lo que debe tener suficientes créditos para iniciar una nueva transmisión. Por el contrario, en el segundo modelo son los mensajes los considerados como mercancía, y por tanto, es el nodo destino el que paga a los nodos intermedios y al origen.

Uno de los esquemas de créditos más sencillos es TFT (*Tit-For-Tat*) [80], en el que dos nodos vecinos intercambian la misma cantidad de mensajes, distinguiéndose dos tipos de mensajes: primarios y secundarios. Los mensajes primarios son aquellos en los que el nodo está directamente interesado, es decir, el nodo es el origen o el destino del paquete. Los mensajes secundarios son aquellos en los que el nodo no está interesado, es decir, no es ni su origen ni su destino. La clave está en involucrar a los nodos en el proceso de retransmisión de mensajes secundarios con el fin de poder obtener créditos para la transmisión o recepción de paquetes primarios. Este método no requiere contabilidad de créditos o autoridad de confianza, pero su aplicación suele limitarse a redes tolerantes a retardos o DTN (*Delay Tolerant Network*), puesto que los mensajes deberán esperar en una cola hasta que el nodo tenga suficientes créditos.

Sprite [81] es una propuesta anti-fraude que emplea firmas digitales para cada transacción. Existe una autoridad central de confianza, denominada *Credit Clearance Service*, responsable de la contabilidad de los créditos. Además de la necesidad de dicha autoridad central, la principal limitación de este esquema es el uso de firmas digitales, operaciones computacionalmente costosas que deben ser llevadas a cabo además en todos los nodos involucrados en la retransmisión. Una mejora de Sprite, denominada Express [82], se basa en la sustitución de las firmas por cadenas *hash*, reduciendo los costes de procesamiento en los nodos.

Mobicent [83] es una solución más reciente en la que un banco virtual lleva a cabo los procesos de cobro y gratificación, encargándose además de repartir lo cobrado a un nodo por la transmisión de sus paquetes de forma equitativa entre los nodos intermedios involucrados en dicha transacción.

Los esquemas basados en créditos incentivan realmente la cooperación de los nodos pues deben ganar créditos para sus propias transmisiones, lo que es especialmente relevante a la hora de evitar comportamientos egoístas. Sin embargo, uno de los principales problemas de estos esquemas es la correcta gestión de los créditos. Así, para evitar trampas y engaños es necesaria la existencia de una autoridad central de confianza que asegure los pagos y garantice la correcta retransmisión de los paquetes, lo cual no es siempre práctico en entornos MANET. Además, estos esquemas pueden resultar injustos en los casos en los que no todos los nodos de la red envían cantidades similares de información.

3.3. Revisión bibliográfica de sistemas de detección

A continuación se discuten algunas de las técnicas de detección mas empleadas en la actualidad y existentes en la bibliografía. Estas, como ya se ha indicado, son necesarias por cuanto que las medidas preventivas no garantizan totalmente la no ocurrencia eventual de actividades maliciosas. A este respecto, en la literatura especializada se puede encontrar gran cantidad de soluciones de detección [52], generalmente basadas en la observación de la ocurrencia de eventos intrusivos o comportamientos anómalos en el entorno monitorizado. En este punto, cabe distinguir entre los sistemas de detección basados en firmas (*signature-based*) y los basados en anomalías (*anomaly-based*) [84].

Los primeros se basan en analizar el tráfico de red, comparando secuencias de bytes o mensajes con patrones preconfigurados y parametrizados (denominados firmas), de los que se conoce *a priori* que pertenecen a eventos maliciosos. La principal ventaja de estos esquemas es su sencillez y rapidez de ejecución, siendo mínimo el procesamiento requerido para realizar las comprobaciones sobre un conjunto limitado de firmas. Su eficacia frente a ataques conocidos es alta pero, a su vez, la rigidez funcional de estos provoca una escasa capacidad para detectar ataques cuyas firmas sean desconocidas. Además, se hace necesario actualizar regularmente las bases de datos de firmas para mantener las capacidades del sistema, teniendo en cuenta que ligeras variaciones en la implementación de los ataques evitarán (o al menos dificultarán) su detección. Dichas variaciones deberán ser incluidas, a su vez, en las bases de datos de firmas, lo que influirá negativamente en la escalabilidad de este tipo de esquemas.

Los sistemas de detección basados en anomalías son, por su propia naturaleza, más complejos de implementar, pues se basan en heurísticas o reglas para la determinación previa del comportamiento normal y aceptable de la red. Mediante la estimación de desviaciones o variaciones que superen un determinado umbral respecto del citado comportamiento normal, podrán clasificarse dichas desviaciones como eventos anómalos/maliciosos. Así, los esquemas basados en anomalías son capaces de detectar potenciales ataques aún desconocidos o en sus primeras fases (*zero-day attacks*). Además, una vez "construido" el modelo de normalidad del sistema, su escalabilidad se ve mejorada. Sin embargo, los sistemas basados en anomalías también son más propensos a realizar clasificaciones incorrectas, particularmente en lo que se refiere a la clasificación como maliciosos de eventos que, aún siendo anómalos, no constituyen amenazas reales.

Aunque la presencia de los sistemas basados en anomalías es mayoritaria, existen algunas aproximaciones que intentan realizar la detección en base al reconocimiento de patrones específicos. Sin embargo, la aplicabilidad de sistemas basados en firmas se presenta más compleja en entornos MANET.

Las soluciones estudiadas se han organizado en cuatro categorías básicas, según las bases de su funcionamiento: (*i*) esquemas basados en confirmaciones o ACK, (*ii*) basados en señuelos, (*iii*) basados en reputación y (*iv*) otros esquemas, principalmente técnicas de detección de intrusiones.

3.3.1. Esquemas basados en *ACK*

En esta tipología de esquemas los nodos solicitan explícitamente una confirmación por parte de otros nodos (vecinos, destino, a dos saltos, etc.) que indique la correcta recepción del paquete que ha sido enviado.

Un esquema basado en confirmaciones a dos saltos se presenta en [85], en el que cada nodo solicita a sus vecinos a dos saltos de distancia un mensaje de confirmación para detectar nodos maliciosos, empleando esquemas de autenticación para evitar así que el siguiente salto pueda enviar mensajes ACK suplantando la identidad de los nodos a dos saltos. Con la intención de reducir la carga de tráfico introducida, los autores proponen una mejora del mecanismo en [86], en el que los nodos preguntan aleatoriamente en vez de hacerlo de forma continuada. Sin embargo, estos esquemas fallan cuando alguno de los vecinos a dos saltos se niega a enviar de vuelta un paquete ACK, por lo que el nodo solicitante es incapaz de determinar cuál es el nodo malicioso.

Para superar esta posible ambigüedad en la detección de nodos maliciosos, Balakrishnan *et al.* [87] proponen TWOACK, un sistema para detectar *enlaces* maliciosos en vez de nodos. La idea principal consiste en el envío de mensajes de confirmación a dos saltos de distancia en la dirección opuesta a la ruta. En este esquema cada nodo que envía datos mantiene una lista con los mensajes de datos enviados pero que todavía no han sido confirmados, un contador de los mensajes de datos retransmitidos y un contador con los mensajes perdidos. Al igual que en los esquemas previos, los autores también proponen una mejora del sistema que incurre en un menor *overhead* [88], en la que únicamente una fracción de los mensajes debe ser confirmada, de acuerdo al valor de una *ratio de confirmación* (*acknowledgement ratio*).

En [89] se introduce una modificación del protocolo AODV para detectar *sinkholes* en grupo. El esquema utiliza una tabla que proporciona un valor de fidelidad a cada nodo participante. Cuando un destino recibe un paquete correctamente envía un paquete ACK de vuelta al origen, y por tanto el valor de fidelidad de los nodos intermedios se verá incrementado. Si no se recibe confirmación, el valor se verá reducido. En caso de que dicho parámetro alcance el valor cero, el nodo será clasificado como malicioso. El principal problema de esta aproximación es el retardo que introduce en la red.

Djahel *et al.* [90] investigan los efectos causados por *sinkholes* cooperativos en OLSR. Para mitigar la pérdida de información provocada por el descarte de paquetes TC (*Topology Control*), los autores proponen un esquema de confirmación a tres saltos, añadiendo dos paquetes de control extra. Así, los nodos emplean paquetes *3hop_ACK* para confirmar la recepción de los mensajes TC desde tres saltos de distancia, mientras que los paquetes *HELLO_rep* anuncian los vecinos a dos saltos a los nodos MPR (*MultiPoint Relay*). Si el número de paquetes TC o *3hop_ACK* descartados supera un determinado umbral, el nodo MPR será considerado malicioso.

Los autores de [85] y [86] completan estos trabajos en [91], en el que sugieren una solución modular que emplea confirmaciones criptográficas a dos saltos para los paquetes *unicast* y un mecanismo de realimentación pasiva basado en monitorizar los paquetes *broadcast*. La información recopilada se utiliza como base de un mecanismo colaborativo de acusación empleado para detectar los nodos atacantes.

La idea principal en [92] es el uso de árboles *Merkle*, un tipo de árboles binarios en los que cada nodo del árbol tiene un valor dado y los valores de los nodos interiores se obtienen a partir una función *hash* de un solo sentido de los valores de los nodos hijos. Para detectar ataques, cada nodo de la red contiene un *hash* que es combinación de su propia identidad y de una clave secreta que solo él conoce. A partir de ello, cada nodo en la ruta confirma la recepción del mensaje hacia el origen, que se encarga de construir un árbol *Merkle* cuyas hojas son las confirmaciones, obteniendo un valor raíz que es comparado con un valor precalculado obtenido durante un proceso de inicialización. Si los valores son iguales, la ruta es segura. Además, debido al enorme *overhead* introducido, los autores proponen dos versiones: confirmación total o aleatoria.

El principal problema de todas estas aproximaciones es la elevada sobrecarga introducida en la red, debido a las continuas confirmaciones (en muchas ocasiones multi-salto) que deben ser transmitidas.

3.3.2. Esquemas basados en *señuelos*

La idea en la que se basan estas técnicas de detección es intentar "engañar" a los nodos *sinkhole*, enviando algún tipo de información "cebo" con la que tentarlos.

Los autores en [93] proponen emplear una red troncal formada por un grupo de nodos *backbone*, o BBN (*BackBone Node*), que tienen permitido asignar direcciones IP restringidas, RIP (*Restricted IP*). Así, cuando un nodo dado quiere comunicarse con un destino, primero le pregunta al BBN más cercano por una RIP no usada aún; después, envía un mensaje RREQ preguntando tanto por el destino como por la RIP. Puesto que las direcciones RIP son restringidas y no son conocidas por los nodos que no sean BBN, si el nodo solicitante recibe un mensaje RREP que proporciona

información sobre la dirección RIP, quiere decir que existe un nodo *sinkhole* en la ruta. En ese momento, el origen empieza a enviar falsos paquetes de datos hacia el destino y los vecinos entran en modo promiscuo, monitorizando las pérdidas e intentando determinar la identidad del nodo malicioso.

Una aproximación similar propuesta para el protocolo DSR es CBDS (*Cooperative Bait Detection Scheme*) [94], en la que el nodo origen selecciona un nodo adyacente de forma estocástica y coopera con él, tomando su dirección IP como la dirección por la que preguntará en un mensaje RREQ señuelo. Si el nodo *sinkhole* "pica el anzuelo", se lanza un proceso de trazado para detectar el nodo malicioso.

La principal desventaja de estos esquemas es que requieren una tercera parte de confianza, lo que no siempre puede ser posible. Además, es necesario que los nodos envíen paquetes de datos inútiles y que entren en modo promiscuo para monitorizar el entorno en busca de los nodos que están descartando paquetes.

3.3.3. Esquemas basados en *reputación*

A pesar de estar incluidos dentro de la Sección 3.2, algunos de los esquemas que emplean métodos de reputación para establecer cooperativamente un nivel de confianza también realizan colectivamente un proceso de detección, declarando como maliciosos a aquellos nodos con niveles bajos de reputación.

En [95] se presenta el protocolo CONFIDANT (*Cooperation Of Nodes: Fairness In Dynamic Ad hoc NeTworks*) para la detección de nodos maliciosos, que se compone de cuatro componentes. Un *módulo monitor* supervisa mediante un mecanismo de escucha pasiva el comportamiento de los vecinos a un salto. Si se detecta un evento sospechoso, los detalles son enviados a un *módulo de reputación* encargado de gestionar una tabla con la valoración de todos los nodos conocidos. Dependiendo de la frecuencia o significancia del evento, la valoración de un nodo se actualizará, pudiendo este ser etiquetado como malicioso. El uso del *gestor de confianza* y del *gestor de rutas* será explicado posteriormente en la sección de respuesta (Sección 3.4.2).

En el esquema *Friends and Foes* explicado anteriormente [75], cada nodo realiza la detección mediante un procedimiento de realimentación pasiva y manteniendo *créditos* para otros nodos, que indican el número de paquetes que estos han retransmitido. Así, el nodo clasifica al resto de nodos en tres categorías que se actualizan periódicamente: amigos (*friends*), para los que el nodo acepta retransmitir paquetes; oponentes (*foes*), para los que no se proporciona ningún servicio; y egoístas (*selfish*), que son aquellos que consideran al nodo como enemigo, aunque este no considera como tales a aquellos. Cuando un nodo tiene que enviar un paquete, busca una ruta en la que el siguiente salto sea amigo. A su vez, cuando un nodo es requerido para retransmitir un paquete, solo lo hará si el solicitante es amigo.

Los autores en [96] adoptan el concepto de *consistencia* dentro del *círculo íntimo* para identificar respuestas falsificadas y prevenir ataques *sinkhole*. La idea consiste en permitir que cada nodo descubra su vecindario a k saltos de distancia. Todos estos nodos formarán su *círculo íntimo*, responsables de votar y filtrar paquetes falsos procedentes del nodo. En particular, los mensajes RREP tienen que ser aprobados por los nodos del círculo, que verifican la validez de los mismos. Si una respuesta contiene información de encaminamiento falsa para atraer paquetes, el ataque será determinado como tal en base a la votación realizada por el círculo íntimo.

La principal desventaja de estos sistemas es el excesivo tráfico de red necesario para compartir la información de reputación entre los distintos nodos.

3.3.4. Otros esquemas de detección

Las técnicas de detección de intrusiones han sido usadas de forma recurrente en la literatura para determinar la potencial existencia de eventos ilegítimos en un entorno de comunicación [84]. Así, los trabajos que siguen a continuación se han agrupado en distintas clases según los principios básicos de la técnica empleada para realizar la detección de intrusos.

Uso de técnicas de *machine learning*

Para realizar la detección se han aplicado en numerosas ocasiones técnicas de *machine learning*, bien sean supervisadas o no supervisadas. Por ejemplo, Zhang *et al.* [97] introducen un esquema local y cooperativo en el que cada nodo incorpora un agente IDS basado en SVM (*Support Vector Machine*), que monitoriza las trazas locales y es responsable de detectar, de forma local e independiente, señales de posibles ataques. Sin embargo, si se detecta una anomalía o si una evidencia es no concluyente y necesita una mayor investigación, los agentes IDS vecinos investigan de forma colaborativa, participando en el proceso de detección global y cooperativo que es ejecutado.

Un método denominado *cross-feature* se describe en [98], en el que se definen un total de 141 características topológicas y relacionadas con el tráfico de red. Este método ejecuta un análisis de minería de datos para extraer correlaciones e intercorrelaciones entre estas, reduciendo así el espacio de características. Para realizar el procedimiento de detección se utiliza un clasificador, como C4.5, RIPPER (*Repeated Incremental Pruning to Produce Error Reduction*) o Naïve-Bayes.

Los autores en [99] introducen una aproximación multi-capa compuesta de tres subsistemas diferentes: un clasificador Bayesiano para la detección de intrusiones en capa MAC, cadenas de Markov para la detección en la capa de *routing* y un

algoritmo de asociación de reglas para la capa de aplicación. Los resultados de estos tres subsistemas se integran en un módulo local, siendo este resultado final enviado a su vez a un módulo global.

CRADS (*Cross-layer Routing Attack Detection System*) [100] es una aproximación de detección que combina el uso de un detector basado en un modelo SVM no lineal y diversas técnicas de reducción de datos para disminuir el tamaño del conjunto de características, minimizando así la carga computacional ocasionada por el aprendizaje. En una línea similar, los mismos autores usan en [101] un algoritmo de clasificación lineal denominado FDA (*Fisher Discriminant Analysis*) para eliminar datos con un contenido bajo en información de utilidad, permitiendo que el uso de clasificadores SVM sea viable en redes MANET.

La propuesta de Shim *et al.* [102] realiza un proceso de *clustering* en DSR para separar los mensajes RREQ normales de los mensajes falsos. Para ello emplea cinco características propias del protocolo de *routing*: la tasa de paquetes RREQ descartados debido a la recepción previa de un mensaje RREQ con un número de secuencia mayor, la misma tasa pero obtenida para descartes causados por números de secuencia iguales, el propio número de secuencia, el grado de divergencia en la identidad del nodo origen recibida en los mensajes RREQ y el grado de divergencia para la identidad del destino.

Sin embargo, la sobrecarga computacional introducida por la ejecución de este tipo de técnicas basadas en aprendizaje automático hace que estas soluciones sean poco apropiadas en entornos de recursos limitados.

Uso de técnicas de *matching*

Otras aproximaciones se basan en técnicas de *matching*. Por ejemplo, IDAD (*Intrusion Detection based on Anomaly Detection*) [103] es una solución IDS que compara la actividad del nodo con un conjunto previamente recolectado de actividades anómalas y maliciosas, denominado *de auditoría*. Los parámetros usados para la detección se obtienen a partir de cada mensaje RREP anómalo y son: número de secuencia destino, número de saltos, tiempo de vida de la ruta, dirección IP destino y marca de tiempo. De este modo, IDAD es capaz de diferenciar paquetes RREP anómalos sin más que comprobar si son similares a los del conjunto *de auditoría*, en cuyo caso el nodo que envía el paquete RREP será considerado malicioso.

Uso de modelos analíticos

Algunos trabajos emplean modelos analíticos para representar la dinámica de un determinado protocolo, detectando inconsistencias durante su operación. Así, en

[12] los autores obtienen el *autómata de estados finitos extendido* o EFSA (*Extended Finite State Automaton*) para el protocolo AODV, modelando el estado normal del mismo y proponiendo una detección tanto basada en especificaciones como de tipo estadístico. La primera aproximación detecta eventos anómalos que son violaciones directas de las especificaciones definidas por el EFSA. La segunda define un conjunto de características estadísticas relativas a eventos anómalos asociados con diversos ataques, definiendo también un segundo conjunto que define el estado normal. Tras ello se emplea un clasificador basado en reglas (RIPPER) para procesar dichos conjuntos y generar una colección de reglas de detección útiles para detectar ataques.

En [104] se propone un modelo analítico teórico para las distintas causas de pérdidas de paquetes, capaz de detectar ataques de *dropping* en redes DSR y de distinguir dichos ataques de otras circunstancias legítimas, como colisiones o errores en el canal. Sin embargo, el trabajo estudia una topología muy limitada y no tiene en cuenta los posibles aspectos de movilidad de los nodos.

Uso de técnicas de monitorización

Algunas técnicas simplemente realizan una monitorización del entorno objetivo, comparando el valor de los parámetros recopilados con un valor umbral dado, el cual puede ser adaptativo o no. Como ya fue explicado en la Sección 3.2.3, Martí *et al.*presentan en [73] el trabajo pionero *Watchdog*. Este se basa en el empleo de un nodo monitor, encargado de llevar la cuenta de los paquetes enviados recientemente por él y de escuchar los paquetes retransmitidos por el siguiente salto. Si un paquete enviado no es retransmitido por el siguiente salto después de un determinado tiempo, se incrementa el valor de una *cuenta de fallo* para el siguiente salto. En el caso de que la cuenta exceda un determinado umbral, el nodo siguiente en cuestión será clasificado como malicioso.

Kurosawa *et al.* [105] tratan con ataques *sinkhole* mediante la introducción de un esquema de detección de anomalías que emplea un método de entrenamiento dinámico. Los autores consideran para expresar el estado de la red un conjunto de características: el número de paquetes RREQ enviados, el de paquetes RREP recibidos y la diferencia media entre el número de secuencia enviado en los paquetes RREQ y el recibido en los RREP. A partir de ello, se emplea un conjunto de entrenamiento de dichas características para calcular un umbral de detección basado en el estado normal de la red, que se actualiza dinámicamente a intervalos regulares para mejorar la precisión en la detección. En el proceso de detección posterior se compara cada muestra con el umbral calculado y se detectan posibles desviaciones respecto del comportamiento normal de la red.

En [106] los autores proponen una solución denominada DPRAODV (*Detection, Prevention and Reactive AODV*), en la que los nodos que reciben mensajes RREP desde

nodos intermedios comprueban que el número de secuencia no exceda un determinado umbral. Además, para evitar inexactitudes que puedan provocar falsas alarmas, dicho umbral se actualiza dinámicamente en cada intervalo temporal considerado. En caso de que el número de secuencia sea superior al límite impuesto, el nodo intermedio se considera sospechoso de ser malicioso.

También han sido empleadas soluciones multi-agente para realizar la detección de ataques. Hongsong *et al.* [107] proponen un esquema de detección de intrusiones basado en un ciclo de vida dinámico de los agentes, capaz de detectar ataques *sinkhole* y DoS. Además, los agentes confiables pueden cambiar su estado según evoluciona el proceso de encaminamiento en AODV para ahorrar energía.

En [108] se propone un IDS denominado ABM (*Anti-Blackhole Mechanism*), que es ejecutado en modo promiscuo en cada nodo de la red. ABM estima el valor de sospecha de un nodo vecino de acuerdo a la diferencia entre los mensajes RREQ y RREP transmitidos desde este. Imponiendo como requisito que los nodos intermedios no puedan responder a los mensajes RREQ, si un nodo intermedio no es el destino final de una comunicación y no reenvía los mensajes RREQ para una ruta específica, pero reenvía los mensajes RREP, su valor de sospecha aumenta en las tablas de sospecha de los ABM vecinos. Cuando dicho valor de sospecha supera un umbral dado, el nodo es clasificado como malicioso.

Una aproximación similar desarrollada para el protocolo DSR es la presentada por Kim *et al.* en [109]. En ella, cada nodo examina los mensajes RREQ recibidos, comprobando si su propia identidad ya aparece en la lista de nodos incluida en el campo *source route*. En caso de ser así, el nodo compara el número de secuencia recibido en el mensaje con el suyo propio almacenado. Si el valor recibido es superior, se concluye que hay algún nodo enviando mensajes de control falsos.

En [110] los nodos intermedios reaccionan descartando los paquetes RREP de un determinado nodo si una cierta variable supera un valor umbral. Esta variable se calcula en función del número de secuencia recibido en el mensaje RREP, el número de secuencia almacenado en la propia tabla de rutas del nodo intermedio y el número de mensajes RREP recibidos.

Existen también diversos trabajos que presentan ligeras variaciones a la aproximación de comparar el número de secuencia recibido en el mensaje RREP con el enviado en el RREQ. Algunos de ellos se pueden encontrar en [111] [112] [113] [114] [115] y [116]. Estos esquemas únicamente tienen en cuenta el "comportamiento" de los números de secuencia de forma local, *i.e.*, sin considerar información de la vecindad, lo que podría mejorar las capacidades de detección, tal y como se verá en el esquema que se ha propuesto en el Capítulo 5.

Es de destacar que existen numerosos trabajos cuyo objetivo es la detección de los ataques de *dropping* y *sinkhole* en diferentes protocolos e incluso en distintas (pero

relacionadas) tecnologías, lo que evidencia el impacto y la relevancia de estos ataques. Sin embargo, estas aproximaciones alternativas se encuentran fuera del ámbito de estudio de este trabajo de tesis, puesto que se proponen para otras tecnologías, como WSN (*Wireless Sensor Network*) [117] [30] [118] [119] [120].

3.4. Revisión bibliográfica de sistemas de respuesta

Además de la existencia de numerosas soluciones destinadas a la prevención y detección de ataques, es imprescindible también la ejecución de medidas de respuesta adecuadas frente a las amenazas detectadas, haciéndose necesaria la presencia de esquemas capaces de mitigar las consecuencias no deseadas ocasionadas por la aparición de estos ataques. Así, esta sección está dedicada a analizar los esquemas de seguridad relacionados con la respuesta/reacción. Es de destacar que la gran mayoría de las soluciones estudiadas no tienen en cuenta una serie de factores adicionales a la hora de ejecutar los mecanismos propuestos. Dichas consideraciones están relacionadas principalmente con los posibles efectos negativos que pueden aparecer como resultado de las acciones llevadas a cabo. En entornos MANET, la adopción de contramedidas inapropiadas puede ocasionar particiones inesperadas de la red, produciendo más daño del que se intenta evitar. Así mismo, existen diversas cuestiones relacionadas con la legalidad de las actuaciones de respuesta que, en gran medida, son obviadas a la hora de desarrollar estos sistemas.

Así, los esquemas propuestos están generalmente orientados a conseguir el aislamiento del nodo malicioso para preservar el correcto funcionamiento de la red y sus servicios. En otros casos, los esquemas reactivos están destinados a servir como mecanismo de realimentación o *feedback*, permitiendo robustecer la red mediante la adaptación de los mecanismos de seguridad considerados a las condiciones particulares observadas. Por tanto, y aunque no es tarea sencilla proporcionar una clasificación definitiva de los distintos mecanismos de respuesta, se propone una clasificación tentativa en tres posibles categorías: (*i*) esquemas basados solamente en la exclusión del nodo, (*ii*) basados en exclusión y notificación y (*iii*) basados en aislamiento.

3.4.1. Esquemas basados solamente en *exclusión*

Este tipo de técnicas de reacción tienen como objetivo evitar que los nodos maliciosos puedan actuar como nodos intermedios en las rutas origen-destino. En estos esquemas solo los nodos que pertenecen a la vecindad del nodo malicioso son conscientes de la presencia del mismo y, como acción subsecuente, dichos vecinos tratarán de eludir aquellas rutas que incluyan al nodo malicioso.

Por ejemplo, los autores en [121] proponen TSR (*Trust-based Secure Routing*), una extensión del protocolo DSR que emplea modelos ocultos de Markov o HMM (*Hidden Markov Models*) para calcular un valor de confianza para cada nodo. Así, TSR actúa contra comportamientos maliciosos eligiendo aquellas rutas cuyos nodos tienen los mayores valores de confianza, eludiendo de esta forma los nodos maliciosos.

En [122] se presenta una modificación de DSR que dota de dos agentes a cada nodo de la red: un agente monitor, MOA (*MOnitoring Agent*), y un agente de encaminamiento, ROA (*ROuting Agent*). El primero monitoriza el comportamiento del nodo para asignarle un valor de confianza, que decrece cuando se detecta el nodo como malicioso. Tras ello, el agente ROA selecciona una ruta confiable, descartando los nodos con valores más bajos.

Otros mecanismos emplean información proporcionada por esquemas de reputación para desencadenar la respuesta. Por ejemplo, SORI [76] descarta los paquetes generados por nodos egoístas con una probabilidad dada, la cual aumenta cuanto menor es el valor de reputación del nodo egoísta, limitando así su capacidad de transmisión. Recientemente, en [123] se propone un esquema de reputación como mecanismo de respuesta. En este caso, la confianza futura de cada nodo se evalúa por medio de un algoritmo de predicción dinámica que tiene en cuenta el comportamiento histórico del nodo.

En [124] cada nodo crea una tabla de "legitimidad" durante la fase de establecimiento de ruta, cuyas entradas (una por cada nodo de la red) se calculan usando dos factores: el número de veces que el nodo ha sido seleccionado como nodo intermedio y el número de veces que el destino se alcanza realmente utilizando dicho nodo intermedio. Al detectar un nodo malicioso, su valor de "legitimidad" decrece. Finalmente, solo aquellos nodos con valores elevados serán elegidos como nodos intermedios en las rutas, limitando así el impacto de los nodos maliciosos.

Otros sistemas propuestos en la literatura ejecutan simplemente la respuesta cuando alguna característica supera un determinado valor umbral. Por ejemplo, en [125] se propone un sistema que bloquea a los nodos maliciosos en las tablas de rutas de los nodos origen si el número de secuencia devuelto en el mensaje RREP tiene un valor muy elevado.

3.4.2. Esquemas basados en *exclusión* y *notificación*

Una mejora a los esquemas de respuesta presentados en la sección previa es la notificación de la existencia de nodos maliciosos al resto de la red mediante el empleo de diversos mensajes. De este modo, cualquier nodo de la red es capaz de evitar rutas que incluyan nodos maliciosos entre los nodos intermedios, dando lugar a mecanismos de respuesta más globales.

En [126] (trabajo posterior a [95]) se describe un mecanismo basado en reputación, el cual proporciona una respuesta cooperativa mediante el uso de los módulos de *gestión de confianza* y de *gestión de rutas*. Cuando se supera un determinado umbral de detección por parte de un *módulo de reputación* previo, se genera una notificación que se envía al módulo de gestión de rutas, encargado de eliminar al nodo malicioso de las rutas. Adicionalmente, se envían mensajes de alarma por parte del módulo de gestión de confianza a la vecindad. Cada mensaje de alarma recibido en un nodo dado se pasa al gestor de confianza de dicho nodo, con el fin de determinar si el nodo malicioso acusado ha sido evaluado de forma similar en otros nodos confiables, en cuyo caso se considera que existen suficientes evidencias acerca de la malicia del nodo.

El trabajo presentado en [127] propone un mecanismo de respuesta aplicable de dos formas: directa o indirectamente. En el primer caso, cada nodo es responsable de eliminar los nodos detectados como maliciosos de su propia tabla de rutas. Por el contrario, en la segunda opción un nodo monitor envía mensajes de alerta a su vecindad de manera que, dependiendo del número de mensajes de alerta recibidos en un nodo dado, este eliminará o no al nodo malicioso de su tabla de rutas.

Los autores en [128] introducen un mecanismo de respuesta basado en bloqueo, en el que existe un conjunto de agentes que monitorizan la red, con capacidad para comunicarse entre ellos. Si un nodo malicioso es detectado, se lanza una acción de respuesta. Primero se envía un mensaje de bloqueo a los nodos asociados al agente que ha descubierto el ataque. Después, este mensaje de bloqueo se disemina entre el resto de agentes de forma que el nodo malicioso pueda ser evitado.

Una modificación del protocolo DSR es presentada en [129], en la que se crea una lista negra de nodos y se disemina por toda la red, de modo que todos los nodos conozcan la identidad de los potenciales nodos maliciosos y, en consecuencia, no procesen mensajes recibidos de ellos. De forma similar, en *Friends and Foes* [75] cada nodo transmite dos listas, la de sus amigos y la de sus oponentes. De este modo, los nodos legítimos rechazarán paquetes de control de nodos maliciosos, forzando el establecimiento de rutas alternativas.

En el sistema propuesto en [108], cuando el valor de sospecha supera un determinado umbral, se envía mediante *broadcast* un mensaje de bloqueo a todos los nodos de la red, excluyendo al nodo malicioso de forma colaborativa. El mensaje de bloqueo contiene la identidad del IDS detector, la del nodo malicioso y un *timestamp* de la detección. Así, al recibir el mensaje, todos los nodos incluyen al nodo malicioso en una lista negra.

En [110], la superación de un umbral por parte de una variable dada, calculada en función del número de secuencia obtenido del mensaje RREP, el almacenado en la tabla de rutas y el número de mensajes RREP recibidos, desencadena que la

identidad del nodo se disemine por la red mediante la inclusión de dicha información acerca del nodo malicioso en los mensajes RREP.

3.4.3. Esquemas basados en *aislamiento*

En esta categoría se incluyen aquellas soluciones que intentan aislar al nodo malicioso de forma activa. En este tipo de mecanismos de respuesta, el nodo malicioso es asediado o rodeado por otros nodos encargados de bloquear las comunicaciones de este, tanto entrantes como salientes. Por tanto, estas soluciones no realizan una mera "exclusión" pasiva del nodo.

Un mecanismo multi-capa es propuesto en [130]. Aunque el ataque ocurre en la capa de red, la respuesta es ejecutada en la capa física mediante la creación de una zona de cuarentena en un radio dado alrededor del atacante. Así, los nodos que se encuentren dentro de la zona de cuarentena no pueden enviar o recibir paquetes. Este esquema necesita de un sistema de posicionamiento que proporcione la localización de los nodos a lo largo del tiempo.

Otras técnicas de reacción se basan en la utilización de agentes autónomos. En [131] los autores introducen un esquema que imita el sistema inmune humano. Existe un agente inmune, IA (*Immune Agent*), distribuido por toda la red y encargado de detectar, clasificar, aislar y recuperar el sistema frente a ataques. Un nodo dado es aislado del resto de la red cuando ha llevado a cabo un determinado número de ataques. Además, el nodo aislado puede "recuperarse" y volver a ser clasificado como legítimo cuando ya no existe amenaza en el entorno.

Una solución similar se propone en [132], en la que se consideran dos tipos de agentes en el sistema: agentes de detección y de contraataque. Cuando se detecta una amenaza, se envía un mensaje de activación mediante *broadcast* a los agentes de contraataque, y solo se activarán aquellos que se encuentran en la vecindad del atacante. Tras activarse, bloquearán cualquier paquete originado o destinado al nodo malicioso.

En [133] se particiona la red en distintos *clusters*, en cada uno de los cuales hay un CH (*Cluster Head*) que supervisa los nodos correspondientes. Cuando el CH detecta un nodo malicioso se crea un agente de acción o AA (*Action Agent*), clonándolo y posicionándolo en cada nodo vecino. Entonces, cada AA comprueba si el nodo malicioso se encuentra localizado a un salto de distancia, en cuyo caso el AA permanece en el nodo. En caso contrario, se auto-clona y se posiciona a su vez en su propia vecindad. Repitiendo el proceso, el nodo malicioso termina completamente rodeado, permitiéndose distintas respuestas: aislamiento de la red, eliminación de las tablas de rutas, bloqueo del tráfico hacia y desde el nodo, reducción de su nivel de confianza para que no pueda introducirse en nuevas rutas, etc.

3.5. Líneas de investigación futura y retos

En este capítulo se ha mostrado la existencia de una vasta literatura y un gran número de propuestas existentes en el campo de la seguridad en redes MANET. Sin embargo, a pesar de los muchos esfuerzos realizados, nuestro estudio revela que todavía es necesario fortalecer las tecnologías actuales y estimular el desarrollo de nuevas aproximaciones defensivas si se desea mejorar el rendimiento de los sistemas actuales y, principalmente, la confianza de los usuarios en el uso de este tipo de entornos. Así, a continuación se presentan algunos de los principales retos a los que se enfrenta la comunidad investigadora y de las nuevas líneas de investigación que pueden surgir.

Algunos autores defienden la necesidad de diseñar nuevos protocolos y procedimientos que refuercen algunos de los aspectos tradicionales de la seguridad, como la autenticación. En este sentido, se están desarrollando protocolos más robustos y procedimientos colaborativos con el objetivo de fortalecer la confiabilidad, *p.ej.*, [134] [135] [136]. Aunque dichos mecanismos pueden usarse de forma dinámica en multitud de tareas (control de acceso, confianza, reputación, etc.), generalmente están relacionados con una perspectiva preventiva de la seguridad. Dicho con otras palabras, la continua aparición de nuevos ataques da como resultado la evidente necesidad de mejorar las condiciones iniciales de seguridad a ser consideradas en la red.

Además, a medida que aparecen nuevos tipos y variantes de ataques, es necesario disponer de esquemas de detección más potentes, robustos y fiables. Frente a ello, la respuesta usual de la comunidad investigadora es el desarrollo de aproximaciones más especializadas. Sin embargo, esta diversificación o especialización da lugar a dos consecuencias: por un lado, se obtienen mejores rendimientos en términos de detección, pero por otro lado tiene como resultado un incremento significativo en los costes de detección si el número de ataques y variantes a detectar se amplía. Con el fin de evitar este inconveniente sin afectar a la precisión en la detección, apuntamos aquí la conveniencia de desarrollar esquemas de detección holísticos. Así, la construcción de modelos semánticos ayudará a la implementación de nuevos paradigmas de detección capaces de superar las particularidades propias de un ataque para proporcionar capacidades de detección más globales.

Además, sería recomendable concebir e idear nuevos esquemas de reacción que garanticen la continuidad y la supervivencia del sistema de comunicación monitorizado. Al contrario de los esquemas de respuesta actuales, que se suelen llevar a cabo de forma local o parcialmente colaborativa, sería deseable la existencia de esquemas de reacción globales basados en la cooperación de toda la red en el proceso, pues de otro modo la posible respuesta implementada podría ser inútil. Por ejemplo, consideremos el caso en el que un nodo malicioso es aislado por sus vecinos. Dicho nodo

podría evitar la restricción impuesta sin más que desplazarse a una zona distinta de la red e infectar a otros nodos.

Otro de los principales retos desde nuestro punto de vista es la necesidad de desarrollar e implementar mecanismos de defensa integrales, *i.e.*, la necesidad de combinar mecanismos de prevención, detección y respuesta de tal forma que el sistema de seguridad actúe como un único ente, y no como la mera suma de las distintas partes, permitiendo así la adaptación dinámica, supervisada o no, del sistema. Esta adaptación global debe converger hacia soluciones estables y óptimas que, de hecho, deben ser controladas por parte del propio sistema de seguridad. Así, sería deseable que cada elemento funcional estuviese convenientemente interrelacionado e interoperase con el resto para proporcionar una solución de seguridad global. Por ejemplo, si un nuevo ataque ha sido detectado, su riesgo real debería ser evaluado con el fin de ejecutar una respuesta adecuada o, en caso necesario, desplegar un nuevo esquema preventivo que proteja nuestro entorno de ataques futuros similares al detectado. Adicionalmente, el modelo de detección debería poder reestimarse y adaptarse a las condiciones cambiantes del entorno a lo largo del tiempo.

Una de las principales consecuencias de las líneas de investigación mencionadas es la necesidad de colaboración intra-nodo e inter-nodo. Sin embargo, esto implica un mayor nivel de complejidad y, consecuentemente, un mayor consumo de recursos. Puesto que la disposición de dichos recursos (batería, espacio en disco, capacidad de cómputo, etc.) es limitada en este tipo de dispositivos, entornos y aplicaciones, se hace necesario imponer un compromiso entre seguridad y coste de la misma. En esta línea vuelve a demostrarse recomendable el desarrollo de aproximaciones holísticas, capaces de tratar con diversas amenazas de seguridad al tiempo que se ahorra en recursos. Este compromiso también es relevante desde el punto de vista de la calidad de servicio, o QoS (*Quality of Service*) en las comunicaciones. Así, desde una perspectiva práctica, gran parte de los esquemas propuestos en la actualidad no son apropiados para este tipo de entornos, pues se obvia su elevado consumo de recursos o su impacto real en el rendimiento de la red. Resulta, por tanto, necesario el desarrollo de nuevas alternativas adecuadas a estas características.

3.6. Conclusiones del capítulo

Debido a la vulnerabilidad de las redes MANET frente a los ataques de *dropping* y *sinkhole*, y al considerable interés de la comunidad investigadora en este campo, se considera necesario realizar un detallado estudio de las principales propuestas existentes en la literatura relacionadas con la defensa frente a los mencionados ataques.

Este capítulo proporciona así una visión general de las soluciones defensivas existentes en la actualidad en el campo de la seguridad en redes MANET frente a los ataques de *dropping* y *sinkhole*. Se presenta un estado del arte de las diferentes aproximaciones, categorizando las mismas en función de la línea de defensa sobre la que se despliegan y, posteriormente, en función del mecanismo o procedimiento llevado a cabo. Así, los diversos esquemas propuestos en la literatura específica durante los últimos años, tanto de prevención cómo de detección y de respuesta, son descritos aquí con la intención de organizar el conocimiento existente en este ámbito.

Finalmente, el estudio de las contribuciones en el campo de la seguridad en MANET revela los principales retos aún abiertos y las potenciales líneas de investigación que siguen precisando de atención por parte de la comunidad investigadora, apuntándose lo que consideramos será el futuro próximo en el tema de estudio. En relación con estas líneas de trabajo propuestas, el resto de capítulos de este trabajo de tesis se incardinan en algunas de las actuaciones anteriormente apuntadas.

En este sentido, en los Capítulos 4 y 5 (Parte II) se realizan dos propuestas de detección de ataques. La primera de ellas desarrollada para detectar ataques de la clase *dropping*, independientemente de la especie en particular de los mismos (*blackhole*, *greyhole*, *selfish*, ...). La segunda propuesta se centra en un tipo específico de ataque de *poisoning*, el ataque *sinkhole*, estando basada la detección propuesta en el cálculo de una heurística obtenida a partir de una característica ampliamente empleada por distintos protocolos de encaminamiento en redes MANET.

Por otro lado, las contribuciones propuestas en los Capítulos 6 y 7 (Parte III) se justifican desde el punto de vista del desarrollo de sistemas de defensa integrales. Para ello, se propone en primer lugar un protocolo de notificación e intercambio de información específicamente diseñado para su empleo en entornos MANET, cuyo principal objetivo es proporcionar un mecanismo de interoperación entre las distintas líneas defensivas, e incluso entre módulos de detección/respuesta colaborativos y/o distribuidos. Una última contribución en esta línea es el desarrollo de un *framework* de seguridad genérico, diseñado con una arquitectura lo suficientemente flexible como para permitir la integración tanto de módulos que implementen diversos ataques como de módulos que desarrollen soluciones defensivas, independientes o integrales.

Publicaciones relacionadas

Para concluir este capítulo se indican las publicaciones directamente relacionadas con el ámbito de estudio del mismo. Estas son:

- **L. Sánchez-Casado**, R. Magán-Carrión, P. García-Teodoro y J. E. Díaz-Verdejo. "Defenses Against Packet Dropping Attacks in Wireless Multihop Ad Hoc Networks". *Security for Multihop Wireless Networks*, S. Khan y J. Lloret (Eds.), CRC Press, pp. 377-400, 2014.

Capítulo 4

Detección de ataques de *dropping* en MANET

Como ya se ha mencionado, el uso y estudio de las redes MANET ha experimentado un enorme crecimiento en los últimos años, lo que ha llevado aparejada la aparición de numerosas amenazas de seguridad en dichos entornos. De entre las distintas amenazas destacan los ataques de *dropping*, por su sencilla implementación y el alto impacto que tienen en el rendimiento de la red. Por este motivo, el interés de la comunidad investigadora por encontrar soluciones defensivas (y en particular, soluciones de detección) frente a este tipo de ataques ha aumentado también de forma considerable.

En este contexto, y considerando el marco general que al respecto de la detección de ataques se ha establecido en el Capítulo 3, en el presente tema se describe un novedoso sistema de detección de ataques de *dropping* para redes MANET basado en un modelo analítico del proceso de retransmisión llevado a cabo por los nodos en este tipo de entornos. Siguiendo una aproximación multi-capa (*cross-layer*), el IDS propuesto recopila características de las capas MAC y de red, con el fin de calcular una heurística sencilla que permita distinguir ataques de *dropping* reales de otras circunstancias que pueden dar lugar a descartes de paquetes legítimos, como colisiones, errores en el canal o situaciones de movilidad. Así, a pesar de que existen otras alternativas de detección de este tipo de ataques en la literatura especializada, nuestra contribución presenta algunas diferencias fundamentales respecto de otros esquemas. Primeramente, las distintas razones legítimas para el descarte de paquetes están explícitamente incluidas en el modelo analítico que ilustra el proceso de retransmisión en entornos MANET. Además, se propone un nuevo método de

enventanado para la recolección y análisis de las observaciones que soportan la toma de decisiones. Por último, se proponen dos posibles implementaciones para la recopilación de las observaciones: una implementación local autónoma y otra distribuida.

Estos aspectos diferenciados resultan en una mayor efectividad global del sistema propuesto, que se ve validada por la experimentación realizada. Los resultados obtenidos muestran que, a pesar de la sencillez de la heurística empleada, las capacidades del sistema son muy prometedoras, tanto en términos de detección como desde el punto de vista de la sencillez y la reducida carga impuesta al sistema.

El resto del capítulo ha sido estructurado en las siguientes secciones. En la Sección 4.1 se motiva la necesidad de diseñar y desarrollar procedimientos de detección frente a ataques de *dropping*, mostrándose en la Sección 4.2 las métricas involucradas en la evaluación general de la calidad de los sistemas de detección. En la Sección 4.3 se presenta el modelo analítico para el proceso de retransmisión en MANET usado como base para el sistema de detección propuesto. Así mismo, en dicha sección se discuten algunas asunciones subyacentes del modelo. La estimación de los parámetros y el potencial uso del modelo propuesto como sistema de detección de ataques de *dropping* se presentan en la Sección 4.4, mientras que las implementaciones concretas del IDS multi-capa se detallan en la Sección 4.5. Por su parte, la Sección 4.6 describe el entorno de experimentación, discutiendo los resultados obtenidos. Finalmente, las principales conclusiones de este capítulo se presentan en la Sección 4.7.

4.1. Motivación

Como se ha visto a lo largo de los Capítulos 2 y 3, los ataques pertenecientes a la clase *dropping* (*blackhole*, *greyhole*, *selfish*, etc.) constituyen una de las principales preocupaciones relacionadas con la seguridad de los entornos MANET [52] [58].

Sin embargo, la detección de descartes maliciosos puede verse dificultada en entornos ad hoc por la existencia de diversas circunstancias que pueden provocar eliminaciones legítimas de paquetes. Algunas de estas causas legítimas de descarte pueden ser las siguientes:

- *Colisiones*, producidas por intentos de acceso simultáneos al canal por parte de distintos nodos contendientes.
- *Errores*, que dan lugar a la corrupción de los paquetes. Estos errores se deben a la propia propagación de la señal durante su transmisión por el canal, cuyas características propias pueden provocar alteraciones en la información.

- *Situaciones de movilidad*, que pueden provocar que el mecanismo RTS/CTS falle al desplazarse algún nodo fuera del rango de cobertura, dando lugar a pérdidas de conectividad y "rotura" de enlaces. Este hecho puede tener como consecuencia descartes masivos de mensajes, pues hasta que dicha situación sea convenientemente reportada a todos los nodos involucrados en la ruta invalidada, algunos de ellos podrían eliminar paquetes.

Estos descartes legítimos introducen de hecho un grave problema, ya que pueden causar un gran número de clasificaciones incorrectas si no se tratan de forma adecuada por parte del sistema de detección. Por lo tanto, reconocer la causa real del descarte de paquetes es un reto en el campo de la seguridad en redes MANET, reto que aún sigue abierto.

A pesar de la existencia en la literatura especializada de numerosas propuestas dirigidas a luchar contra este tipo de ataques [51] (véase el Capítulo 3), lo cierto es que la gran mayoría de ellas no consideran las circunstancias antes referidas de forma explícita. Algunas soluciones sí contemplan la movilidad de los modos en sus aproximaciones de detección, pero generalmente estas se basan en técnicas de *data mining*, las cuales introducen una alta complejidad computacional, y funcionan como un modelo de *caja negra* que no permite comprender exactamente cómo se incorpora la información de movilidad y sus efectos. Uno de los pocos trabajos que tiene en cuenta situaciones de descarte no malicioso es el propuesto en [104], en el que se desarrolla un modelo teórico capaz de distinguir entre ataques de *dropping* y causas de descarte legítimas como colisiones o errores en el canal. Sin embargo, el trabajo no considera la posibilidad de que los nodos dispongan de capacidades de movilidad.

Así, tomando como foco la detección de esta tipología de ataques, en este capítulo se introduce un nuevo sistema de detección de intrusiones sustentado en una metodología multi-capa. De este modo, se recopilan y analizan distintas características (u observaciones) tanto de la capa MAC como de la capa de red. Empleando dichas observaciones, y en base a un modelo analítico que recoge el proceso de retransmisión de paquetes en redes MANET, es posible calcular una heurística sencilla que permite la detección de comportamientos maliciosos de *dropping*. Además, puesto que el modelo analítico incluye de forma nativa las distintas causas legítimas que también pueden ocasionar descartes de paquetes, nuestra aproximación es capaz de distinguir dichas circunstancias frente a ataques reales, mejorando así las capacidades de detección del sistema, especialmente en lo referido a las clasificaciones incorrectas de eventos legítimos como maliciosos.

Considerando la implementación práctica del sistema, se proponen dos posibles aproximaciones para la recopilación de las observaciones. En una primera aproximación se introduce una metodología de recolección autónoma o *stand-alone*, en la que cada nodo recopila, de forma directa, únicamente información local relativa a sí mismo, con el fin de realizar detecciones de comportamientos maliciosos. La segunda es

una aproximación de recopilación indirecta y distribuida, en la que distintos nodos de la red, o incluso todos ellos, monitorizan a sus vecinos con el fin de obtener las observaciones requeridas, compartiendo la información relativa al nodo dado sobre el que se está realizando el proceso de detección.

Nótese que el proceso de recopilación de las observaciones es independiente del proceso de evaluación de las mismas para el cálculo de los parámetros involucrados en la detección. Así, por ejemplo, diversos nodos podrían realizar una recopilación directa y local de información relativa a sí mismos, al tiempo que, por otro lado, un grupo de nodos monitorizando el entorno podrían recopilar información relacionada con un nodo dado de forma indirecta. Posteriormente, el proceso de detección realizado en base a la información recopilada podría ser llevado a cabo de distintas formas: bien por parte de cada uno de los nodos de forma autónoma e independiente, bien por parte de un nodo centralizado en el que se reúne toda la información obtenida, o bien de forma colaborativa entre los distintos nodos tras la compartición de la información correspondiente. De este modo, es necesario disociar el proceso de captura de los parámetros del proceso de detección en sí.

Además, una aportación adicional de este trabajo es el empleo de un enventanado para la recolección y análisis de las observaciones basado en eventos, en vez del tradicional uso de un enventanado temporal. Como se verá en la Sección 4.4.2, el enventanado por eventos introduce una serie de ventajas adicionales en nuestro sistema.

Puesto que es el desarrollo de mecanismos de detección lo que conformará la parte fundamental de este trabajo de tesis, y antes de presentar la propuesta principal de este capítulo, es necesario discutir algunos conceptos importantes necesarios para determinar y contrastar las bondades de un esquema de detección. Ello constituirá la base sobre la que llevar a cabo la provisión y análisis de resultados.

4.2. Evaluación de los sistemas de detección

Para evaluar la calidad de un sistema de detección es necesario realizar una comparación de los resultados que este devuelve aplicándolo a situaciones de red contrastables. Esta validación se lleva a cabo inicialmente aplicando el procedimiento sobre un conjunto de datos dispuestos a tal efecto, etiquetados con la clase a la que pertenecen (legítimo, sospechoso, malicioso, etc.). Este conjunto de datos etiquetados es considerado como la referencia (*ground truth*) con respecto a la que comparar los resultados del sistema de detección a evaluar. En la Figura 4.1a se muestra esquemáticamente dicho conjunto de datos de referencia. Los resultados obtenidos por el sistema de detección se representan en la Figura 4.1b, los cuales evidencian clasificaciones válidas y no válidas, pues algunas de las trazas identificadas corres-

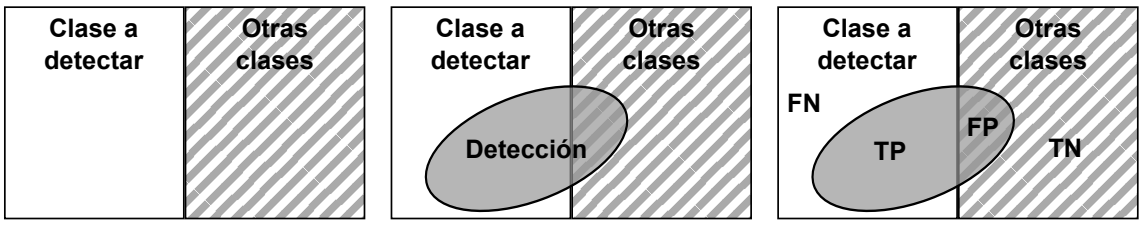


Figura 4.1: Proceso de detección: (a) conjuntos a clasificar; (b) resultados del proceso de detección; (c) medidas derivadas de la clasificación.

ponderán a la clase a detectar, mientras que otras habrán sido etiquetadas de forma incorrecta. Finalmente, en la Figura 4.1c se representa la correspondencia entre los conjuntos de datos y las medidas más comunes para indicar la calidad de los resultados de detección. Estas se derivan de la clasificación realizada y del conocimiento del conjunto de referencia, y son las siguientes:

- TP (*True Positives*), o verdaderos positivos. Son los datos detectados por el sistema de detección que, según la referencia, pertenecen realmente a la clase a detectar.
- FP (*False Positives*), o falsos positivos. Estos son datos clasificados como de la clase a detectar, pero que realmente no pertenecen a ella. Esto es, FP se refiere a una clasificación incorrecta por parte del sistema.
- TN (*True Negatives*), o verdaderos negativos. Los datos que no son detectados como pertenecientes a la clase a detectar y que realmente no pertenecen a ella.
- FN (*False Negatives*), o falsos negativos. Los datos de la clase a detectar que no son detectados por el sistema de detección. Como FP, FN se refiere a una clasificación errónea.

En base a estas cuatro medidas se definen las siguientes tasas, que permiten evaluar la efectividad global del sistema de detección:

- TPR (*True Positives Rate*), o tasa de verdaderos positivos. También conocida como sensibilidad, exhaustividad o *recall*, se calcula como:

$$TPR = \frac{TP}{TP + FN} \tag{4.1}$$

- FPR (*False Positives Rate*), o tasa de falsos positivos o *fall-out*:

$$FPR = \frac{FP}{FP + TN} \tag{4.2}$$

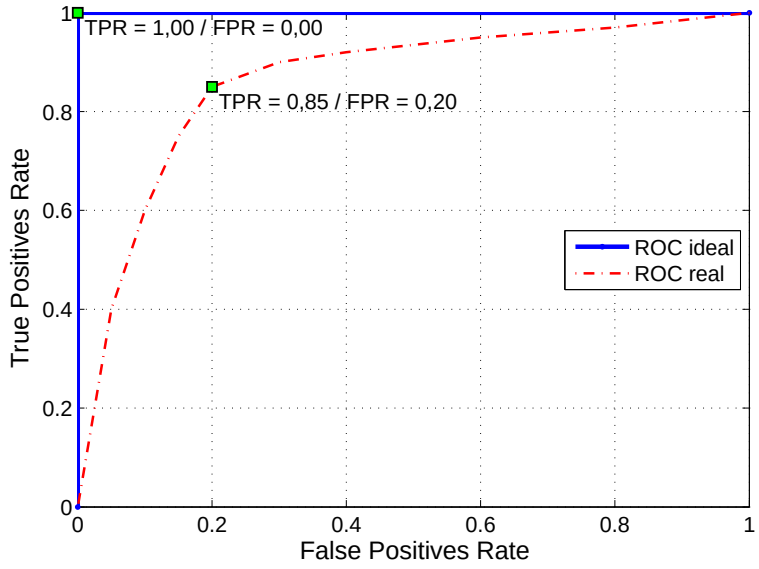


Figura 4.2: Curva ROC ideal y ejemplo de una curva ROC real de un sistema de detección.

A partir de todo lo anterior, una representación muy común acerca de la calidad de un sistema de detección es la curva ROC (*Receiver Operating Characteristic*), que muestra la evolución conjunta de las tasas FPR y TPR en función de la variación de los parámetros del sistema de detección, como por ejemplo, un estadístico en base al cual decidir si una observación dada es de una clase o de otra. Una vez obtenidos los distintos puntos que conforman la curva ROC, su representación permite determinar de forma rápida y visual el punto de operación óptimo para el que el sistema consigue los mejores resultados.

En la Figura 4.2 se muestra un ejemplo de curva ROC ideal de un sistema de detección. En dicha curva, el sistema no presenta ningún FP, mientras que es capaz de detectar correctamente todos los datos de la clase objetivo, independientemente de los parámetros seleccionados. Así, el punto óptimo de operación es el cuadrado en la Figura 4.2 con $TPR = 1$ y $FPR = 0$. Como contrapunto, en la misma figura se muestra la curva ROC más propia de un sistema de detección real, en el que las tasas de TPR y FPR no son ideales y los resultados de detección erróneos (FP y FN) deben ser tratados convenientemente. El punto de operación óptimo en este caso corresponde a la coordenada de la curva ROC más próxima a ($TPR = 1$, $FPR = 0$); en el caso de la Figura 4.2, relativo al punto $TPR = 0,85$ y $FPR = 0,20$.

Una vez definidas brevemente las diferentes métricas usadas para comparar las bondades de un sistema de detección, a continuación se presenta la contribución principal de este capítulo, analizando y comparando los resultados por ella proporcionados con los de otras soluciones actualmente existentes.

4.3. Proceso de retransmisión en entornos MANET

Esta sección está dedicada a modelar analíticamente el proceso de retransmisión de paquetes que sigue un nodo dado en redes MANET. El modelo propuesto considera distintas circunstancias legítimas que pueden ocurrir durante las comunicaciones (colisiones, errores en el canal, movilidad de los nodos, ...), así como comportamientos de *dropping* maliciosos, y permite inferir cómo todas estas situaciones afectan a la operación del proceso de retransmisión global. Antes de comenzar con la descripción del modelo, es necesario discutir una serie de asunciones relevantes en relación con la aplicación del mismo, a fin de contextualizar adecuadamente la propuesta y clarificar sus principales implicaciones.

4.3.1. Escenario de estudio

Se considerará que en los escenarios propuestos existen N_L nodos legítimos, geográficamente distribuidos en un área dada, y moviéndose con una cierta velocidad siguiendo una trayectoria aleatoria. Se asume también la utilización de IEEE 802.11 como protocolo para la capa de acceso al medio, así como el empleo del mecanismo RTS/CTS para el envío de los paquetes (véase Sección 2.4.1). Esta asunción es coherente con la movilidad de los nodos, pues la ausencia de una detección de portadora virtual en este tipo de escenarios con movilidad podría implicar la aparición de multitud de colisiones debido al problema de la estación oculta. Además, los nodos se comunican entre sí empleando algún protocolo de encaminamiento ad hoc, al tiempo que pueden generar distintos flujos de tráfico, como CBR (*Constant Bit Rate*) o VBR (*Variable Bit Rate*).

En este escenario general se considera adicionalmente la existencia de N_M nodos maliciosos, cuyo comportamiento difiere del de los nodos legítimos en el hecho de que descartan paquetes recibidos en vez de retransmitirlos. Por simplicidad, se asume un modelo de ataque en el que los nodos maliciosos actúan de forma individual, sin confabularse con otros nodos, es decir, los distintos nodos maliciosos no cooperan entre sí para evadir el sistema de detección. Una extensión de este trabajo implicaría la combinación y evaluación de la técnica de detección aquí propuesta con otros mecanismos que traten con ataques en confabulación de forma específica. Por ejemplo, nuestro esquema podría complementarse mediante la realización de comprobaciones extremo-a-extremo, como la propuesta en [137]. Dicha comprobación se encarga de determinar si los paquetes de datos han llegado realmente a su destino, siendo capaz por tanto de detectar "cadenas" de nodos atacantes actuando en confabulación. Otras propuestas relacionadas con la detección de ataques realizados en complot entre varios nodos son la adoptada en [89], en la que los nodos destino, tras la correcta recepción de un paquete, envían mensajes ACK de vuelta al origen; o la de [92], donde

se emplean árboles Merkle cuyas hojas se corresponden con las confirmaciones para verificar que la ruta sea segura.

En una primera etapa, asumimos por simplicidad también que los nodos son confiables, *i.e.*, la información proporcionada por ellos no ha sido interferida, creada artificialmente o modificada por un atacante. Por tanto, asumimos la existencia de algún sistema de gestión de confianza o reputación, bien sea basado en soluciones de infraestructura de clave pública o PKI (*Public Key Infrastructure*), bien mediante mecanismos de compartición de claves secretas o utilizando cualquier otra metodología posible.

Sin embargo, esta restricción inicial de confiabilidad se relajará en una fase posterior, en la que se propone un algoritmo distribuido en el que un subconjunto de los nodos de la red son los encargados de recopilar de forma indirecta los parámetros necesarias y, sobre la información obtenida a partir de ellos, construir el esquema de detección propuesto (véase Sección 4.5.2). De forma general, podría asumirse que dichos nodos son considerados confiables, siendo esta una asunción habitualmente adoptada por otros trabajos similares en la bibliografía. Por ejemplo el propuesto en [137], donde se discute la necesidad de disponer de una red *backbone* de nodos confiables con unas capacidades mayores para realizar funciones de detección con confiabilidad. Otra posibilidad interesante, por el contrario, consistiría en asumir la no confiabilidad de los nodos, empleándose entonces algún tipo de esquemas basados en votación o incluso esquemas basados en el funcionamiento de un tribunal, como el propuesto en [138], en el que, tras dividir de forma segura la red en distintos *clusters*, el nodo responsable de cada *cluster*, comúnmente denominado CH (*Cluster Head*), asume la responsabilidad de actuar como jurado, actuando como la "entidad confiable" previamente indicada.

4.3.2. Modelo analítico para el proceso de retransmisión

Considerando el escenario genérico previamente descrito, en esta sección se modela analíticamente el proceso de retransmisión que siguen los nodos en redes MANET. Este modelo constituye la base sobre la que se desarrollará el esquema de detección de ataques de *dropping* propuesto.

Así, la operación por la que un nodo retransmite los paquetes de datos recibidos hacia el siguiente salto en la ruta multi-salto implica distintos pasos, que se explican a continuación (véase el diagrama de flujo de la Figura 4.3).

Después de que un paquete es correctamente recibido por un nodo, deben ocurrir necesariamente varios eventos sucesivos para que dicho paquete sea retransmitido en su camino hacia el destino último:

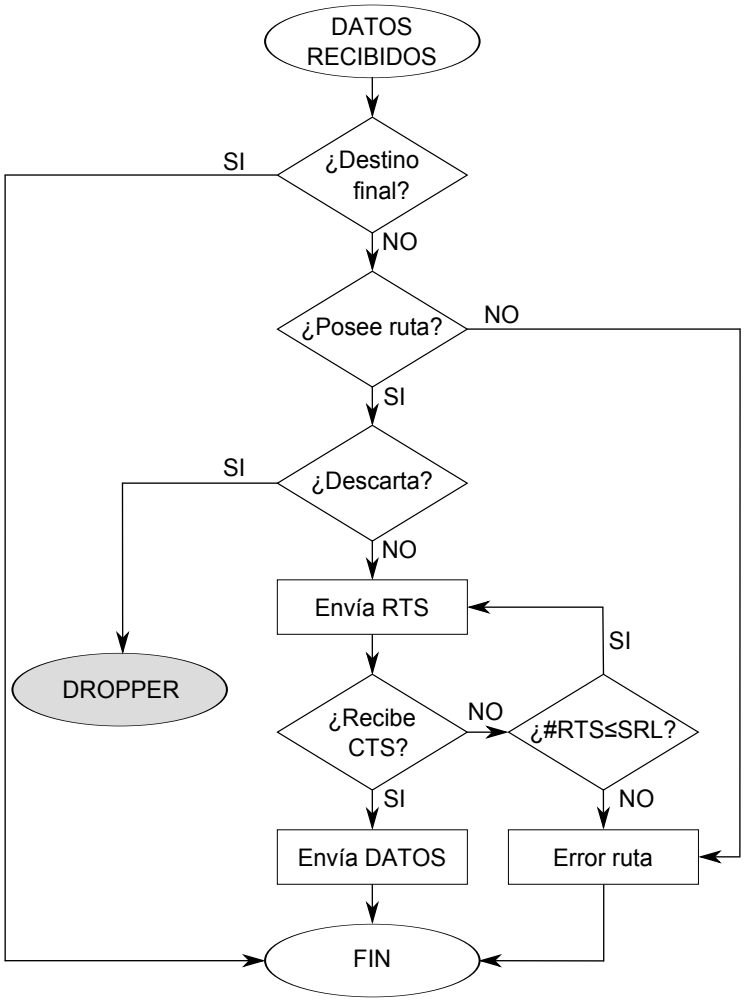


Figura 4.3: Diagrama de flujo para el proceso de retransmisión en redes MANET.

1. Que el nodo no sea el destino final del paquete recibido, lo que denominaremos como evento $\overline{dest}$.

2. Que el nodo disponga de una ruta válida a través de la cual retransmitir el paquete hacia el destino deseado. Este evento se expresará como evento $rout$.

3. Que el nodo no actúe como un atacante malicioso, en cuyo caso el paquete sería descartado en vez de ser retransmitido. Este será calificado como evento $\overline{drop}$.

Únicamente en el caso de que todos los eventos anteriores ocurran, el nodo retransmitirá el paquete. Para ello, dos eventos consecutivos tienen lugar:

- Primero, el nodo intentará enviar un mensaje RTS (se alcanza el cuadro "Envía RTS" en la Figura 4.3). Este evento se denominará evento RTS, y la probabilidad de que suceda, dados los eventos $\overline{dest}$ y $rout$, es P_{RTS}.
- Tras ello, el nodo debería recibir un mensaje CTS (se alcanza la pregunta "¿Recibe CTS?" en el flujograma de la Figura 4.3). Este mensaje CTS se recibe desde el siguiente salto en la ruta cuando el paquete RTS correspondiente ha alcanzado correctamente el destino y el paquete CTS ha sido devuelto con éxito. Denominaremos a este evento como evento CTS, siendo P_{CTS} la probabilidad de que ocurra.

Para obtener la probabilidad de que un nodo esté actuando como *dropper*, las probabilidades anteriores se estimarán como sigue. La probabilidad P_{RTS} se obtiene teniendo en cuenta la ocurrencia de los eventos del 1 al 3, es decir, se enviará un mensaje RTS si existe una ruta hacia el destino y el nodo no es ni el destino final ni un nodo malicioso. De este modo, P_{RTS} se calcula como:

$$P_{RTS} = \Pr\left(RTS \mid \overline{dest}, rout\right) = (1 - P_{DROP}) \tag{4.3}$$

donde P_{DROP} se refiere a la probabilidad de que un paquete sea descartado de forma maliciosa por el nodo. Nótese que el evento $\overline{drop}$ es modelado como una probabilidad, mientras que los eventos $\overline{dest}$ y $rout$ no lo son. En realidad, ambas condiciones se pueden verificar de forma sencilla mediante la inspección del paquete recibido en el nodo. Así, para el cálculo de la probabilidad condicional propuesta en (4.3) solo se tienen en cuenta aquellos paquetes que cumplen las condiciones $\overline{dest}$ y $rout$.

Los mensajes RTS y CTS pueden perderse por distintas causas legítimas tras su envío. Por ejemplo, un paquete RTS podría sufrir una colisión si un nodo dentro del rango de cobertura del nodo origen envía un RTS al mismo tiempo. Por su parte, los mensajes CTS pueden colisionar si un nodo oculto, localizado dentro del rango de comunicación del nodo receptor, pero fuera del rango del emisor, emite el mensaje al mismo tiempo. Además, ambos mensajes pueden verse afectados por errores en el canal, evitando que estos alcancen su destino. Otra causa por la que se pueden descartar los paquetes se produce si los nodos se encuentran fuera del rango de comunicación a consecuencia de su propio movimiento y no han tenido tiempo suficiente de actualizar sus tablas de rutas, por lo que no podrán comunicarse entre sí.

Todas estas circunstancias pueden ocasionar que los mensajes se pierdan y, en consecuencia, que el mensaje CTS no sea recibido correctamente, dando lugar a la retransmisión del mensaje RTS. Recordemos de la Sección 2.4.2 que el mecanismo IEEE 802.11 RTS/CTS permite un máximo número de intentos de retransmitir los paquetes RTS, es decir, si un emisor no recibe un mensaje CTS como respuesta tras

un límite predefinido de retransmisiones de paquetes RTS, el proceso de envío se considera fallido. Este límite superior es denominado SRL, y está definido a valor 7 por defecto en el protocolo. Una vez que dicho límite se ha superado, el paquete de datos correspondiente es descartado y el nodo emisor asume que el enlace se ha roto y el siguiente salto ya no se encuentra accesible.

Por tanto, la probabilidad de que un mensaje CTS sea recibido correctamente en el emisor (evento CTS) se puede aproximar como sigue. En nuestro modelo la probabilidad P_{CTS} se ha dividido en dos términos independientes. El primero de ellos está relacionado con colisiones o errores en el canal, y tiene en cuenta aquellas situaciones en las que se producen retransmisiones de paquetes RTS sin superar el límite SRL. El segundo término está asociado con situaciones de movilidad que pueden causar que el número de retransmisiones supere el valor SRL, por lo que el enlace se considerará roto[1]. Por tanto, el paquete CTS será recibido si ninguna de las dos situaciones mencionadas ocurre. La probabilidad del evento CTS, dado que ha ocurrido el evento RTS previamente, se obtendrá como:

$$P_{CTS} = \Pr(CTS \mid RTS) = 1 - (P_{COL} + P_{MOB}) \tag{4.4}$$

siendo P_{COL} la probabilidad de que los paquetes RTS o CTS se hayan perdido a causa de colisiones o errores en el canal, y P_{MOB} la probabilidad relacionada con las pérdidas de paquetes ocasionadas por la rotura de un enlace.

Finalmente, si el emisor captura el medio transmitirá los datos deseados, es decir, el paquete será retransmitido por el nodo (evento FWD). Dado que para proceder con la retransmisión del paquete los eventos RTS y CTS deben finalizar correctamente (Figura 4.3), la probabilidad para el proceso completo de retransmisión, P_{FWD}, es calculada como:

$$\begin{aligned} P_{FWD} &= \Pr\left(CTS, RTS \mid \overline{dest}, rout\right) = \Pr(CTS \mid RTS) \cdot \Pr\left(RTS \mid \overline{dest}, rout\right) \\ &= (1 - P_{DROP}) \cdot [1 - (P_{COL} + P_{MOB})] \end{aligned} \tag{4.5}$$

Como en el cálculo de P_{RTS}, esta probabilidad se obtendrá únicamente teniendo en cuenta aquellos paquetes que cumplan los eventos $\overline{dest}$ y $rout$.

Aunque el estudio ha sido realizado explícitamente para el proceso de retransmisión de paquetes de datos, el modelo analítico resultante es aplicable a otros tipos

[1] A pesar de que existen otras circunstancias que pueden provocar la rotura de los enlaces, como fallos en los nodos, congestión u otras, en este trabajo se emplearán por simplicidad indistintamente los términos *movilidad* y *enlace roto* para abarcar todas estas situaciones. Por supuesto, este aspecto no afecta a los fundamentos de la propuesta.

de paquetes en redes MANET, como son los paquetes de control. O incluso a otros protocolos, bien sean reactivos o proactivos. La única restricción es que el proceso de retransmisión asociado debe emplear el mecanismo RTS/CTS, como se discutió en la Sección 4.3.1. En consecuencia, la aproximación de detección de comportamientos de *dropping* desarrollada y presentada a continuación es aplicable, con ligeras modificaciones, a otros casos distintos del descarte de paquetes de datos.

4.4. Detección de ataques de *dropping*

En esta sección presentamos la nueva metodología propuesta en este trabajo de tesis para la detección de ataques de *dropping* en entornos MANET.

Como en otras propuestas de detección, nuestra aproximación sigue un proceso de enventanado, de modo que el nodo será considerado o no como malicioso de forma discreta a lo largo del tiempo. De esta forma se obtiene, para cada nodo, un conjunto de observaciones de red durante cada ventana de análisis. A partir de dichas observaciones se estiman los parámetros (probabilidades) previamente indicados en la Sección 4.3 (P_{RTS}, P_{CTS} y demás). Finalmente se obtiene la decisión final acerca del comportamiento del nodo.

La probabilidad de que el nodo esté realizando comportamientos de descarte maliciosos se obtiene a partir de (4.5) como:

$$P_{DROP} = 1 - \frac{P_{FWD}}{[1 - (P_{COL} + P_{MOB})]} \tag{4.6}$$

Esta probabilidad de *dropping* será posteriormente comparada con un determinado umbral de detección, θ. Si el valor de la probabilidad es superior al umbral, entonces se puede concluir que el nodo analizado es malicioso, considerándose como legítimo en caso contrario:

$$nodo = \begin{cases} \text{malicioso}, & \text{si } P_{DROP} \geq \theta \\ \text{legítimo}, & \text{en caso contrario} \end{cases} \tag{4.7}$$

Como es evidente, el punto de operación del sistema depende del valor seleccionado para el umbral θ. Si θ es fijado a un valor bajo, serán más los nodos detectados como maliciosos en la red, pero también serán más los nodos legítimos incorrectamente clasificados como maliciosos. Por otro lado, la selección de un valor elevado para θ resultará en la detección de menos nodos maliciosos, produciéndose también menos falsos positivos. De este modo, y como se verá en la sección de experimenta-

ción, un valor de compromiso entre estas dos situaciones es, típicamente, la mejor elección.

Tal y como se ha indicado en secciones previas, nuestra aproximación de detección tiene en consideración distintas causas por las que un paquete podría no ser retransmitido: bien por causas legítimas, como colisiones, errores o movilidad, o bien porque ha sido descartado de forma maliciosa. Sin embargo, estimar numéricamente dichos efectos mediante la monitorización de distintos parámetros de red no es una tarea trivial. En la siguiente subsección se discute cómo se calculan las probabilidades involucradas en nuestro modelo analítico a partir de las distintas medidas obtenidas de la red y, a partir de ello, la decisión final acerca del carácter malicioso (*dropper*) o no de un nodo.

4.4.1. Estimación de los parámetros

Los parámetros principales a ser estimados para calcular P_{DROP} en (4.6) son P_{FWD}, P_{COL} y P_{MOB}. Utilizaremos una aproximación empírica para estimar tanto P_{FWD} como P_{COL}.

P_{FWD} puede obtenerse como el porcentaje de paquetes de datos retransmitidos por un determinado nodo en relación con aquellos que han sido recibidos. Para ello, el IDS monitoriza los paquetes de datos recibidos cuyo destino no es el propio nodo analizado. El estimador para esta probabilidad, $\hat{P}_{FWD}$, es:

$$\hat{P}_{FWD} = \frac{\#DATA_{FWD}}{\#DATA_{RECV}} \tag{4.8}$$

Nótese que un paquete recibido será tenido en cuenta en la estadística $\#DATA_{RECV}$ únicamente si el nodo no es el destino final del paquete y si existe una ruta válida para dicho paquete hasta su destino final.

Respecto a los descartes legítimos de paquetes, recordemos que nuestro modelo distingue dos posibles situaciones: (*i*) la producida por colisiones o errores en el canal, que considera aquellas retransmisiones de paquetes RTS que no superan el valor SRL ($\#RTS$ consecutivos sin respuesta $\leq SRL$) y que contribuye a la obtención de P_{COL}; y (*ii*) la debida a enlaces rotos, que tiene en cuenta las retransmisiones de paquetes RTS superando el umbral SRL ($\#RTS$ consecutivos sin respuesta $> SRL$) y que contribuye al cálculo de P_{MOB}.

En relación con P_{COL}, puesto que el efecto asociado está relacionado con la carga de tráfico, se calculará el número de mensajes RTS enviados por el nodo sin haber recibido una respuesta CTS apropiada ($\#RTS_{SENT} - \#CTS_{RECV}$), así como el número total de intentos de reserva del canal. Como se ha dicho, solo aquellos paquetes que

no están directamente relacionados con situaciones de enlaces rotos serán tenidos en cuenta, es decir, aquellas retransmisiones RTS que no excedan el límite SRL. En definitiva, el estimador para la probabilidad de colisión y error en el canal, $\hat{P}_{COL}$, puede calcularse como:

$$\hat{P}_{COL} = \frac{\#RTS_{SENT} - \#CTS_{RECV}}{\#RTS_{SENT}} \tag{4.9}$$

Finalmente, veamos cómo estimar P_{MOB}. El estimador propuesto para el cálculo de la probabilidad de un enlace roto se puede obtener de forma sencilla, puesto que tomará solo dos valores. Esta estimación, $\hat{P}_{MOB}$, se fijará a 1 cuando el número de retransmisiones RTS supere el límite SRL, dado que el nodo considerará que ha dejado de tener conectividad con el siguiente salto. En cambio, el estimador tomará un valor cero en cualquier otro caso, al no considerarse el enlace como roto. De este modo:

$$\hat{P}_{MOB} = \begin{cases} 1, & \text{si } \#RTS_{SENT} > SRL \\ 0, & \text{en caso contrario} \end{cases} \tag{4.10}$$

Puesto que las situaciones de movilidad pueden tener ciertas particularidades, es conveniente dedicar una discusión más detallada a estudiar la estimación de la movilidad y qué circunstancias pueden tener lugar en ello. Recuérdese de la Sección 2.4.2 que, en AODV, cuando el mantenimiento de las rutas falla y el enlace es considerado como roto ($\#RTS_{SENT} > SRL$), se pueden dar dos situaciones:

- *Escenario 1*: si el enlace roto se encuentra más cerca del nodo origen que del nodo destino, el nodo intermedio marcará la ruta como inválida y enviará hacia el origen un mensaje RERR alertando a sus precursores, que dejarán de enviarle paquetes.
- *Escenario 2*: en caso de que el enlace fallido se encuentre más cerca del nodo destino, el nodo intermedio intentará realizar una *reparación local* de la ruta, enviando un mensaje RREQ. Si tras un cierto tiempo la ruta no ha podido ser reparada, el nodo la marcará como inválida y actuará de forma análoga a como actúa en el *escenario 1*.

Nótese que, durante un cierto tiempo, el nodo que ha detectado el enlace roto continuará recibiendo mensajes que será incapaz de retransmitir, *i.e.*, el nodo se comportará de forma similar a como lo haría un nodo malicioso, descartando paquetes. Este período de tiempo será considerablemente mayor en el caso de producirse la situación explicada en el *escenario 2*, puesto que el proceso de mantenimiento de ruta

puede durar hasta docenas de segundos antes de que se genere un mensaje RERR y los precursores sean alertados para que dejen de enviar paquetes.

Por tanto, es importante distinguir qué situación concreta ha tenido lugar, pues la decisión acerca de durante cuánto tiempo la probabilidad $\hat{P}_{MOB}$ será considerada 1 (y por tanto, P_{DROP} considerada 0 y el nodo considerado como legítimo) no es trivial. Para distinguir los escenarios, el IDS monitoriza también si se ha enviado algún mensaje RREQ por el nodo tras la detección del enlace roto. En tal caso, $\hat{P}_{MOB}$ se fijará a 1 durante un cierto tiempo T. La elección del valor de T se justificará en la Sección 4.6.

4.4.2. Enventanado basado en eventos para la recolección y el análisis de observaciones

Tal y como ha sido establecido previamente, la estimación de la naturaleza maliciosa de un nodo determinado involucra la estimación de los parámetros a través de las ecuaciones (4.7), (4.8), (4.9) y (4.10). Estas ecuaciones hacen la estimación a partir de las observaciones RTS_{SENT}, CTS_{RECV}, $DATA_{RECV}$ y $DATA_{FWD}$. Como suele ser común en numerosas aproximaciones de detección, el modo en el que se adquieren y posteriormente se analizan estas observaciones se basa en considerar observaciones temporales sobre ventanas de análisis consecutivas y (habitualmente) no solapadas de duración fija. Sin embargo, esta metodología presenta algunos inconvenientes:

- Un primer inconveniente está relacionado con aquellas situaciones en las que la ventana temporal finaliza justo tras la transmisión de un paquete RTS. En este caso no es posible conocer si el paquete será respondido correctamente por un mensaje CTS, si se producirá una colisión o si tendrá lugar una situación de movilidad. Este hecho puede ocasionar efectos indeseados debido a las discontinuidades causadas por el enventanado. La Figura 4.4a muestra un ejemplo específico de la citada situación, donde las líneas de puntos representan la finalización de las ventanas temporales. Como puede observarse, la ventana temporal podría finalizar durante la retransmisión de un mensaje RTS, *p.ej.*, justo tras el envío del RTS #5. En este caso, no podría ser "capturada" toda la circunstancia que caracteriza la situación de movilidad en ninguna de las ventanas temporales y, por tanto, los descartes legítimos producidos por dicha movilidad no serían considerados como tales, al no haber sido detectada esta situación.
- Un segundo inconveniente está relacionado con el hecho de que, incluso si durante un intervalo de tiempo no ha habido ningún parámetro a estimar, o se han producido solo unos pocos eventos, estos serán analizados de todas formas, dando como resultado la obtención de información sesgada (*biased*) que puede

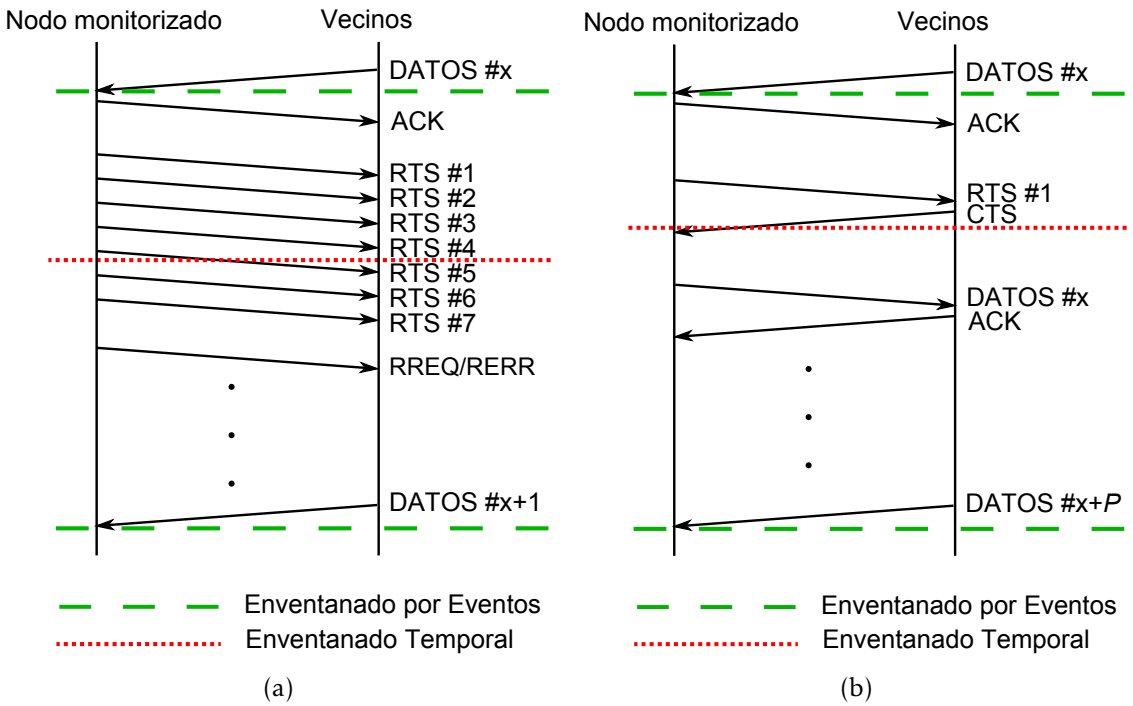


Figura 4.4: Discontinuidades (a) y aparición de información sesgada (b) debido al enventanado temporal.

dar lugar a resultados de detección incorrectos. La Figura 4.4b muestra cómo podría ocurrir dicho problema. Supongamos que en la ventana temporal solo se han recibido unos pocos paquetes de datos, o en el caso extremo mostrado, solo se ha recibido un único paquete de datos. Supongamos también que la ventana temporal finaliza antes de que se haya recibido el mensaje CTS como respuesta al paquete RTS #1. En tal caso, el modelo analítico considerará un elevado porcentaje de paquetes de datos descartados (un 100 % en el ejemplo), provocando la incorrecta clasificación del nodo como malicioso.

Para remediar estos inconvenientes proponemos aquí realizar un procedimiento de enventanado basado en eventos en vez de temporal. Es decir, las observaciones se obtienen en ventanas consecutivas y no solapadas de P paquetes de datos recibidos para cada nodo de la red. En la Figura 4.4 se muestra también la diferencia entre los modos de operación de ambos tipos de enventanado y cómo el enventanado basado en eventos solventa los inconvenientes presentes en la aproximación temporal. Las líneas discontinuas se corresponden con la finalización de las ventanas de eventos.

El primer problema sería evitado puesto que el fin de cada ventana siempre coincide con la recepción de un paquete de datos. La Figura 4.4a evidencia que, empleando el enventanado basado en eventos, nos aseguramos que las situaciones de

movilidad sean completamente detectadas. Se produzca el fin de la ventana tras la recepción del paquete $DATOS\ \#x$ o tras la recepción del paquete $DATOS\ \#x+1$, el evento de movilidad es capturado completamente.

Por otro lado, la recolección de las estadísticas siempre considerará el mismo número de eventos P, atenuando el efecto de la información sesgada. La Figura 4.4b ilustra cómo el esquema basado en eventos garantiza que se emplee siempre una cantidad de datos representativa, minimizando potenciales clasificaciones erróneas.

Además, se debe mencionar una ventaja adicional significativa que aparece como consecuencia del empleo del esquema basado en eventos. Esta se refiere al hecho de que si un nodo dado no está recibiendo tráfico, no tiene sentido que realice un proceso de detección cada cierto tiempo, lo que únicamente conllevaría un gasto innecesario de recursos. Así, el empleo del enventanado propuesto ayuda además a ahorrar recursos en nodos con escasa o nula actividad.

4.4.3. Esquema general del sistema de detección

Para determinar si un nodo N_i dado está descartando de forma maliciosa o no paquetes es necesario obtener las siguientes observaciones:

- $\#RTS_{SENT,i}$: número total de mensajes RTS enviados por el nodo N_i a cualquier otro nodo de su vecindad.
- $\#CTS_{RECV,i}$: número total de mensajes CTS recibidos por el nodo N_i desde sus vecinos.
- $\#DATA_{RECV,i}$: número total de paquetes de datos recibidos por el nodo N_i de parte de sus vecinos.
- $\#DATA_{FWD,i}$: número total de paquetes de datos retransmitidos por el nodo N_i a sus vecinos.
- $RREQ_i$: parámetro *booleano* que toma un valor $TRUE$ si se ha transmitido algún mensaje RREQ por parte del nodo N_i, y $FALSE$ en caso contrario. Este parámetro se usa para determinar el tiempo durante el cual se fijará $\hat{P}_{MOB}$ al valor 1.

Teniendo en cuenta todo lo anterior, y en base a las cinco observaciones indicadas ($\#RTS_{SENT}$, $\#CTS_{RECV}$, $\#DATA_{RECV}$, $\#DATA_{FWD}$ y $RREQ$), podemos derivar finalmente la probabilidad de la ocurrencia de un ataque de *dropping*, P_{DROP}, reduciendo el criterio (4.6) a la siguiente expresión:

Algoritmo 4.1 Pseudo-código para el proceso de detección de ataques de *dropping*.

para cada ventana ω en el tiempo de monitorización **hacer**
  para cada nodo N_i en la red **hacer**
    Obtener $\hat{P}_{FWD}$ usando (4.8)
    Estimar $\hat{P}_{COL}$ con (4.9)
    Extraer $\hat{P}_{MOB}$ de (4.10)
    Calcular P_{DROP} mediante (4.11)
    si $P_{DROP} < \theta$ **entonces**
      si $RREQ == FALSE$ **entonces**
        Nodo N_i es legítimo durante la ventana ω.
      si no
        Nodo N_i es legítimo durante cada ventana ω en el tiempo T.
      fin si
    si no
      Nodo N_i es malicioso durante la ventana ω.
    fin si
  fin para
fin para

$$P_{DROP} = \begin{cases} 0, & \text{si } \hat{P}_{MOB} = 1 \\ 1 - \frac{\hat{P}_{FWD}}{1-\hat{P}_{COL}}, & \text{en caso contrario} \end{cases} \tag{4.11}$$

La descripción detallada del algoritmo de detección completo se muestra en el Algoritmo 4.1:

Nótese que la propuesta presentada en este trabajo de tesis se basa en un modelo analítico que emplea parámetros sencillos para llevar a cabo el proceso de detección. El uso de esta metodología incurre en un menor *overhead* en comparación con otras técnicas más sofisticadas basadas en algoritmos de minería de datos o de aprendizaje máquina, que requieren un mayor coste computacional. Además, el modelo propuesto evita la necesidad de realizar una fase de entrenamiento, minimizando también los grandes conjuntos de datos (etiquetados o no) que deben ser utilizados en las otras aproximaciones.

4.5. Implementación práctica del esquema de detección

Más allá del desarrollo teórico del método de detección multi-capa propuesto, a continuación se detallarán dos posibles implementaciones prácticas del mismo para su despliegue real en redes MANET.

Ambas implementaciones hacen uso del mismo proceso general de detección de *dropping* descrito en la Sección 4.4. Sin embargo, difieren en el modo en que se recopilan las observaciones de red utilizadas para caracterizar el comportamiento de los nodos.

El primer esquema es una aproximación local autónoma (*stand-alone*), en la que un nodo se monitorizaría a sí mismo, de modo similar a como lo haría un sistema antivirus en un ordenador tradicional. El problema con este tipo de esquema es el hecho de que un nodo comprometido podría también engañar a estos sistemas de seguridad, emitiendo información falsa sobre sí mismo. En todo caso, el sistema autónomo tiene sentido como prueba de concepto para la solución IDS planteada, pues proporcionará los límites teóricos del rendimiento del sistema.

Alternativamente a ella, a modo de solución de detección realista, proponemos en segundo lugar una arquitectura distribuida, donde las observaciones empleadas para estimar el potencial comportamiento malicioso de un nodo dado se obtienen de forma indirecta por parte del resto de nodos (generalmente por los vecinos del nodo monitorizado) y no por él mismo.

4.5.1. Aproximación local autónoma

En esta aproximación, el proceso de recolección de las observaciones se realiza de forma local en el propio nodo monitorizado. Así, el IDS puede acceder a toda la información necesaria para realizar una detección más precisa. Por ejemplo, un nodo podrá tener acceso al número de paquetes que ha recibido correctamente; sin embargo, sus vecinos no podrán conocer exactamente si un paquete enviado hacia él fue realmente recibido correctamente o no. De este modo, las observaciones necesarias para estimar los parámetros del sistema de detección ($\#RTS_{SENT}$, $\#CTS_{RECV}$, $\#DATA_{RECV}$, $\#DATA_{FWD}$ y $RREQ$) se pueden realizar y emplear de forma directa, siendo por tanto el proceso de detección realizado también de forma autónoma e independiente en cada nodo.

La principal ventaja de este esquema es su simplicidad, pues únicamente requiere la instalación de un agente IDS local en los distintos nodos de la red. Además, al tratarse de agentes independientes y autónomos, se puede seleccionar de forma rápida y sencilla en qué nodos se desea que se realice el proceso de detección, evitándose así un consumo de recursos innecesario.

Sin embargo, como ya se ha mencionado, la aproximación propuesta tiene una limitación principal: la fiabilidad de los nodos. Puesto que este tipo de arquitecturas autónomas carece de ningún mecanismo para la gestión de la confianza/reputación, la información que caracteriza el comportamiento de un nodo dado podría no ser completamente confiable o precisa. Por ejemplo, en el caso de un nodo malicioso, es

bastante probable que el propio nodo intente falsificar sus estadísticas con el objetivo de evadir el proceso de detección.

4.5.2. Aproximación distribuida

Asumiendo la falta de confianza en la información extraída de los nodos de la red, algunas de las tareas (por no decir todas) requeridas para el proceso de detección de intrusiones descrito anteriormente deberían ser ejecutadas de forma distribuida y cooperativa.

En esta línea, proponemos una segunda implementación que se basa en la necesidad de emplear nodos (específicos o no) que cooperen entre sí para proporcionar un proceso colaborativo para obtener las observaciones. Estos nodos, denominados monitores, deben actuar en modo promiscuo, recopilando y analizando las observaciones de interés dentro de su área de comunicación. Cuando el modo promiscuo está activo, el nodo es capaz de capturar todas las tramas de información enviadas en su vecindad, independientemente de su destino; sin embargo, no puede conocer con absoluta certeza si dichos paquetes han sido correctamente recibidos o no por la interfaz inalámbrica del siguiente salto. Por tanto, dos de las observaciones necesarias en nuestro sistema IDS deben ser reemplazadas con otras similares, esta vez estimadas a partir de la observación de paquetes enviados. Específicamente, las observaciones $\#CTS_{RECV,i}$ y $\#DATA_{RECV,i}$ son reemplazadas por las siguientes:

- $\#CTS_{SENT,i}$: el número total de paquetes CTS enviados por los nodos vecinos al nodo N_i.
- $\#DATA_{SENT,i}$: el número total de paquetes de datos enviados hacia el nodo N_i por el resto de nodos de su vecindad.

Nótese que el uso de estas dos nuevas observaciones es solo una aproximación, dado que los paquetes enviados pueden no llegar a recibirse correctamente por diversas razones, *p.ej.*, ocurrencia de errores en el canal. Sin embargo, en la experimentación presentada seguidamente en la Sección 4.6 se demuestra que el efecto de emplear esta estimación no degrada significativamente las capacidades del modelo.

Así mismo, en esta aproximación distribuida también hay que tratar adecuadamente la información redundante recopilada. Al ser varios los nodos monitores que pueden estar escuchando la misma información, esta debe ser filtrada de forma apropiada para evitar valores replicados de las observaciones de interés. Dado que para cada tipo de observación ($\#RTS_{SENT}$, $\#CTS_{SENT}$, $\#DATA_{SENT}$, $\#DATA_{FWD}$ y $RREQ$) se conoce el tiempo de transmisión necesario para que la información sea recopilada por un nodo monitor que se encuentre a la máxima distancia del nodo

monitorizado, es posible agrupar las observaciones que han sido recibidas en el intervalo definido por dicho tiempo máximo de transmisión. Si además el campo de dirección (origen/destino, capa MAC/red, según corresponda al tipo de observación concreto) es coincidente, las observaciones se considerarán duplicadas, por lo que habrá que filtrarlas, manteniendo únicamente una de ellas.

Una vez que los nodos monitores han recopilado toda la información necesaria acerca de un nodo dado, el comportamiento de este último es estimado usando la heurística propuesta. Recordemos que el proceso de detección es independiente del proceso de recopilación de información. En este trabajo, para la implementación práctica del sistema hemos optado, por simplicidad, por una aproximación de detección centralizada, asumiendo la existencia de un nodo central que reúne toda la información recopilada por los nodos monitores y que realiza el cálculo de la heurística y la posterior clasificación del nodo monitorizado en malicioso o no.

Nótese en este punto que el uso de nodos monitores asume de forma implícita la existencia de soporte para la gestión de confianza, lo que permite que estos puedan considerarse confiables. Sin embargo, esto no es estrictamente necesario, pues podría implementarse alternativamente algún tipo de procedimiento de votación que permita decidir acerca de potenciales diferencias en los valores de las observaciones recopiladas, ocasionadas por la existencia de uno o más nodos maliciosos intentando evadir el proceso de detección. En este caso, los nodos monitores confiables podrían ser sustituidos por los propios nodos vecinos de la red.

4.6. Resultados experimentales

Esta sección presenta, en primer lugar, la descripción del entorno experimental utilizado para evaluar el sistema de detección de ataques *dropping* propuesto. Posteriormente se realizan distintas pruebas para verificar el correcto funcionamiento del mismo con las dos implementaciones indicadas y, tras ello, se analizan y discuten los resultados de detección obtenidos bajo distintas condiciones.

4.6.1. Descripción del entorno experimental

Hemos hecho uso de la popular herramienta NS-2 (*Network Simulator 2*) [139] para simular diversos despliegues MANET, al ser este uno de los simuladores de red más ampliamente utilizados por la comunidad investigadora.

El área de simulación se restringe a un cuadrado de 1.000 x 1.000 metros. Se han seleccionado AODV e IEEE 802.11b como protocolos de *routing* y capa MAC respectivamente, empleándose también el mecanismo RTS/CTS para la transmisión de los

Tabla 4.1: Parámetros de configuración en NS-2.

Parámetro	**Valor**	**Parámetro**	**Valor**
Modelo radio	*TwoRayGround*	Tipo MAC	*802_11*
Canal	*WirelessChannel*	$CW_{min/max}$	31/1023 *slots*
Antena	*OmniAntenna*	Tiempo ranura	20 μs
Ganancia Tx/Rx	1	SIFS	10 μs
Altura	1,5 m	Tasa datos	11 Mb
NIC	*WirelessPhy*	Tasa básica	2 Mb
Umbral captura	10 dB	Tasa PLCP	1 Mb
Umbral portadora	1.5e-11 W $\approx$ 550 m	*Short Retry Limit*	7
Umbral Rx	3.6e-10 W $\approx$ 250 m	*Long Retry Limit*	4
Potencia Tx	0.2818 W $\approx$ 250 m	Umbral RTS	0 *bytes*
Frecuencia	914 MHz	Tipo cola	*PriQueue*
Factor pérdidas	1	Tamaño	50

Tabla 4.2: Parámetros de AODV en NS-2.

Parámetro	**Valor**	**Parámetro**	**Valor**
Active Route Timeout	10 s	*RREP Wait Time*	1 s
Reverse Route Life	6 s	#Retransmisiones RREQ	3
Max. RREQ Timeout	10 s	Detección capa enlace	si

paquetes. Otros parámetros de simulación se muestran en la Tabla 4.1 (parámetros generales) y en la Tabla 4.2 (parámetros de AODV).

El número total de nodos es 25, de los cuales se varía entre 1 y 20 el número de los que actúan de forma maliciosa descartando paquetes. El número de aplicaciones de tráfico se fija a 20, cada una correspondiente a una conexión CBR que envía 4 paquetes por segundo con un tamaño de paquete de 512 bytes.

El modelo de propagación considerado es *Two Ray Ground* [140], teniendo los nodos un radio de cobertura de 250 metros.

Para modelar la movilidad de los nodos se ha empleado el modelo RWP (*Random Way Point*) [141], con una velocidad mínima fija de 1 m/s y una velocidad máxima variando entre 5 y 30 m/s. El tiempo de pausa se fija a 15 s, es decir, una vez que el nodo alcanza el destino deseado, espera durante el tiempo de pausa antes de elegir un nuevo destino de forma aleatoria y repetir el proceso.

De acuerdo con una amplia investigación realizada en [142] para modelar la probabilidad de error en enlaces inalámbricos bajo diversas condiciones, se fijó el

valor de dicha probabilidad inicialmente a 0,37 %. Sin embargo, en este trabajo modificaremos este valor entre el 0,37 % y el 7 % para evaluar el sistema de detección propuesto bajo distintas circunstancias.

Por otro lado, los nodos maliciosos están configurados para que descarten únicamente el 20 % de los paquetes de datos que pasan a través de ellos y que deberían ser retransmitidos. Sin embargo, estos nodos participan normalmente en el proceso de *routing*, sin modificar o descartar los paquetes de control. Por tanto, el modelo de ataque puede considerarse como nodos *greyhole* que no intentan introducirse maliciosamente en la ruta.

El tiempo máximo que puede tardar el proceso de reparación local tras la detección de un enlace roto depende de diversos parámetros de AODV, incluyendo una cierta aleatoriedad causada por un mecanismo de retroceso exponencial binario que se utiliza para evitar congestión a causa de las retransmisiones de mensajes RREQ. A causa de esto, el mencionado límite rondará los 60 segundos, y por tanto, este será el valor temporal seleccionado para el parámetro T, tiempo durante el cual la probabilidad $\hat{P}_{MOB}$ será considerada 1 al estar llevándose a cabo el procedimiento de reparación local.

4.6.2. Resultados de detección

Como se detalló al principio del capítulo, la efectividad global del IDS propuesto es evaluada mediante dos métricas, la tasa de verdaderos positivos, o TPR, y la tasa de falsos positivos, o FPR. En esta línea se obtienen varios puntos de operación que conforman la curva ROC mediante la variación del umbral de detección θ en (4.7). Para capturar el comportamiento estadístico, la curva ROC se ha derivado repitiendo 75 veces (con distinta semilla) cada simulación, para una velocidad máxima de los nodos de 10 m/s.

La Figura 4.5 muestra la curva ROC obtenida para ambas implementaciones: local autónoma y distribuida. Los resultados obtenidos con la aproximación IDS distribuida son un poco peores que los obtenidos con la aproximación local. Esto tiene fácil explicación por el hecho de que en el esquema distribuido se están empleando aproximaciones para dos de las observaciones principales, al considerarse que cada paquete CTS y de datos enviado será correctamente recibido. Sin embargo, puesto que en realidad algunos pueden no recibirse correctamente por diversas causas, el rendimiento de este esquema se ve ligeramente deteriorado.

Como puede verse en la gráfica, si para el umbral de detección θ se selecciona un valor más elevado, el sistema obtendrá una mejor FPR, pero empeorarán los resultados de TPR. Por otro lado, la elección de un valor de θ más bajo resultará en mejores resultados de TPR, a costa de incrementar los falsos positivos, FPR.

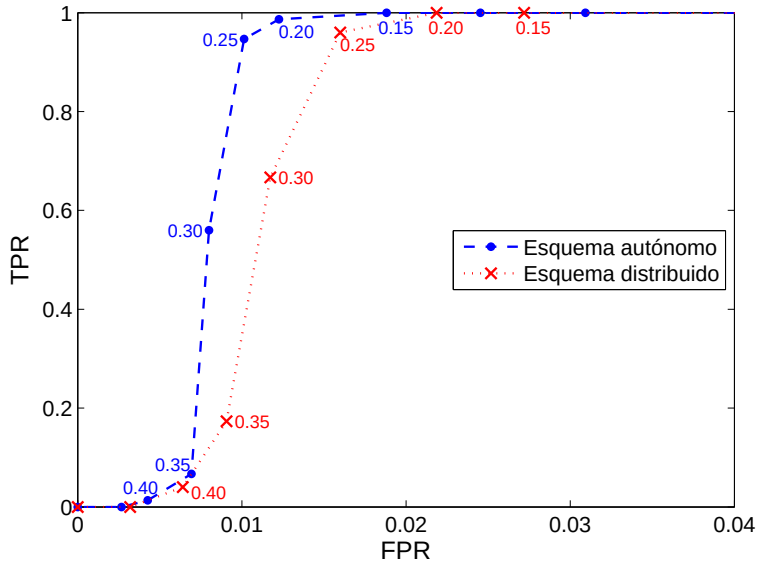


Figura 4.5: Curva ROC para el esquema local autónomo y el distribuido, modificando el umbral de detección θ.

El punto de operación óptimo del sistema se obtiene así empíricamente. Para un mejor funcionamiento, el parámetro θ se ha fijado en ambos casos a un valor 0,15, que proporciona un valor de compromiso entre las tasas de falsos positivos y de verdaderos positivos.

Influencia del tamaño de ventana

Los resultados mostrados anteriormente son los obtenidos una vez se han ajustado en el sistema algunos de los parámetros implicados en el proceso de detección; por ejemplo, el tamaño de la ventana de observación basada en eventos. Para ello, previamente se ha evaluado el tamaño óptimo de la misma. Se han realizado pruebas variando el número de paquetes recibidos, P, entre 50, 75, 100 y 125 paquetes, mostrándose los resultados en la Tabla 4.3. Como antes, se han repetido las pruebas 75 veces variando la semilla, empleando una velocidad máxima de 10 m/s para los nodos.

Es de destacar que el tiempo necesario para recopilar las estadísticas empleando un enventanado basado en eventos será dependiente de las condiciones de tráfico concretas y, en consecuencia, los retardos introducidos en el proceso de detección no serán constantes. Hemos realizado un estudio preliminar de los retardos medios introducidos por cada implementación, mostrándose los resultados también en la Tabla 4.3.

Tabla 4.3: Punto de operación y retardo de detección de los esquemas local autónomo y distribuido para distintos tamaños de ventana.

Tamaño	Esquema autónomo			Esquema distribuido		
ventana	TPR (%)	FPR (%)	Ret. (s)	TPR (%)	FPR (%)	Ret. (s)
50	100,0±0,00	6,51±1,30	10,67	100,0±0,00	8,75±1,48	10,56
75	100,0±0,00	2,99±0,77	15,70	100,0±0,00	4,85±1,06	15,54
100	100,0±0,00	1,92±0,64	20,48	100,0±0,00	2,88±0,84	20,26
125	98,67±2,61	1,17±0,49	25,06	98,67±2,61	1,82±0,67	24,84

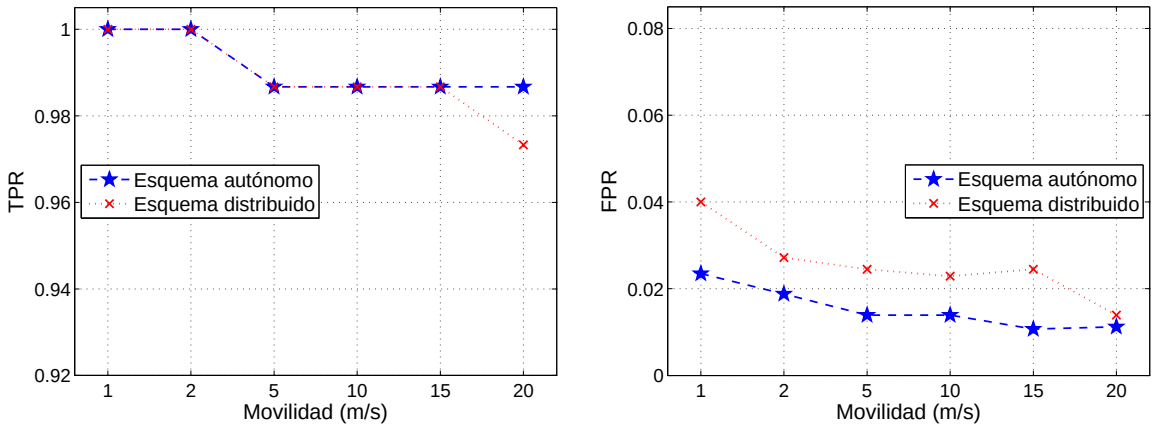
Como era de esperar, mayores tamaños de ventana producen mejores resultados en términos de FPR, al ser la cantidad de información recopilada más representativa, aunque la tasa TPR también disminuye ligeramente. Sin embargo, el tamaño de la ventana no puede ser extremadamente grande, pues ello daría lugar a un incremento en el tiempo necesario para realizar el proceso de detección. De acuerdo con los resultados obtenidos, el tamaño de ventana de ahora en adelante será considerado de 100 paquetes de datos recibidos, término medio entre el retardo introducido en la detección y las capacidades de detección del sistema.

Influencia de la movilidad

También hemos estudiado la eficiencia en la detección para distintas condiciones de movilidad de los nodos. Se han simulado seis escenarios para considerar un amplio abanico de posibilidades, variando la velocidad máxima entre 5 y 30 m/s. La Figura 4.6 muestra los resultados de TPR y FPR obtenidos por ambas implementaciones para las distintas condiciones propuestas.

Como puede observarse, tanto la aproximación local autónoma como la distribuida obtienen excelentes resultados en relación con las dos métricas consideradas. Así, la tasa TPR supera el 97% en todos los escenarios, mientras que la tasa FPR siempre permanece por debajo del 4%. Estos resultados confirman las capacidades de nuestro modelo y que, como era de esperar, estas se degradan ligeramente conforme la movilidad incrementa.

Por otro lado, y como ya se explicó en la Sección 4.6.2, los resultados de detección obtenidos para la aproximación que emplea un esquema de recolección distribuido son ligeramente inferiores que los obtenidos por la aproximación autónoma, al estar empleándose aproximaciones para dos de las observaciones principales a partir de las que se deriva el comportamiento de un nodo.



(a) TPR (b) FPR

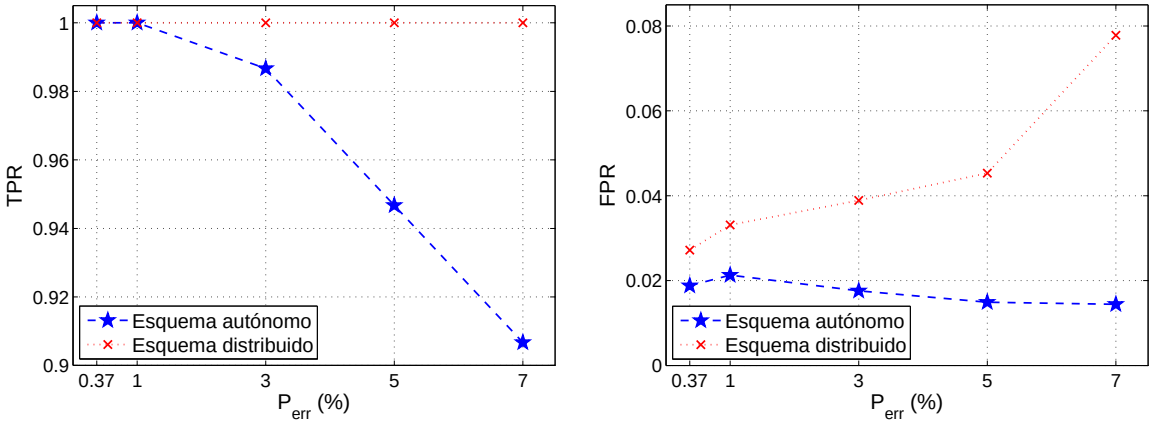
Figura 4.6: TPR (a) y FPR (b) obtenidos por los esquemas local autónomo y distribuido, para diferentes escenarios de movilidad.

Influencia de la probabilidad de error en el canal

Otro de los aspectos analizados es la variación de la eficiencia en la detección bajo distintas probabilidades de error en el canal. Aunque los resultados de [142] demuestran que una probabilidad de error en el canal del 0,37 % es una buena estimación para canales IEEE 802.11, en este trabajo hemos considerado también probabilidades de error mayores con el objetivo de evaluar el esquema de detección propuesto bajo distintas circunstancias del canal que ocasionan unas mayores pérdidas de paquetes, como apantallamientos (*shadowing*) o desvanecimientos (*fading*). La Figura 4.7 muestra gráficamente los resultados de las tasas TPR y FPR en estas situaciones.

Como se puede ver en la figura, el valor TPR se degrada en la aproximación autónoma al aumentar la probabilidad de error. Esto se debe principalmente al hecho de que, por cada paquete recibido, será más probable que el nodo tenga que retransmitirlo en más ocasiones a causa de las pérdidas, al emplearse en IEEE 802.11 las confirmaciones positivas, o ACK. Por tanto, estas múltiples retransmisiones pueden "ocultar" el comportamiento malicioso de los nodos, especialmente cuando la tasa de descarte no es demasiado alta, como ocurre con nuestro modelo de ataque, en el que la tasa de descarte es del 20 %. Como resultado, la tasa FPR también decrecerá.

Para el caso de la aproximación distribuida, aunque un nodo puede recibir paquetes con errores que no deben ser retransmitidos hasta que sean primero recibidos correctamente, es probable que alguno de los nodos monitores considere que los paquetes han sido correctamente recibidos, por lo que los tratará como si fuesen



(a) TPR (b) FPR

Figura 4.7: TPR (a) y FPR (b) obtenidos por los esquemas local autónomo y distribuido, para diferentes probabilidades de error en el canal.

paquetes descartados. Consecuentemente, la tasa TPR tiende a aumentar, haciéndolo también la tasa FPR.

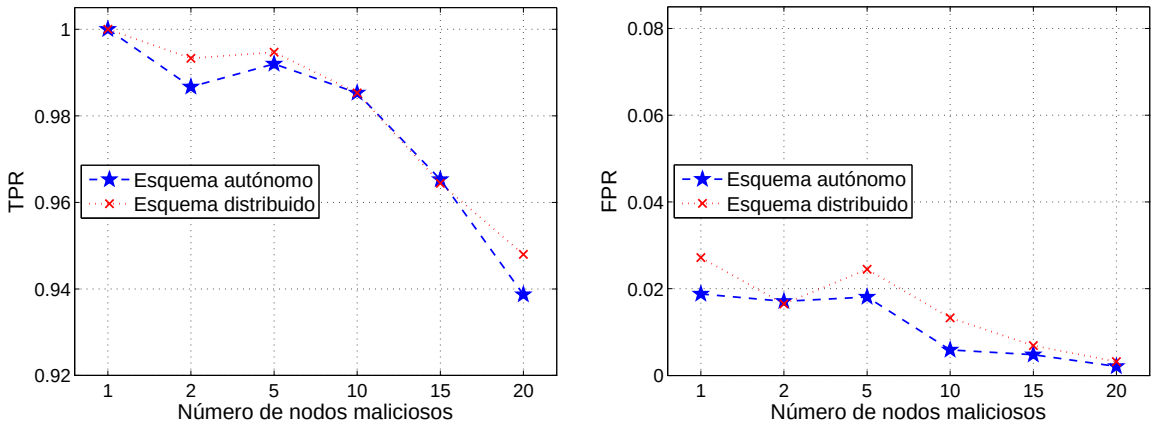
Influencia del número de nodos maliciosos

Otro conjunto de experimentos ha consistido en analizar el rendimiento de ambos esquemas de detección para un número creciente de nodos maliciosos, con el objetivo de demostrar que las capacidades de detección de nuestra propuesta no se ven degradadas de forma severa aunque varios nodos de la red hayan sido comprometidos. Los resultados de estas pruebas se representan en la Figura 4.8.

Estos resultados revelan que, a pesar de existir un elevado número de nodos maliciosos en la red, el esquema propuesto continúa siendo preciso en la detección, manteniéndose el valor de FPR por debajo del 3 %, esto es, nuestra propuesta obtiene muy buenos resultados. Por otra parte, debe notarse que, incluso en el peor caso, en el que el 80 % de los nodos de la red son *droppers*, la tasa de detección se mantiene por encima del 93 %.

4.6.3. Discusión de los resultados de detección

En resumen, es evidente a partir de todos los resultados previos obtenidos que el IDS propuesto para la detección de ataques de *dropping*, independientemente de la implementación realizada para la recopilación de las observaciones (local autónoma o distribuida), puede detectar eficientemente los nodos maliciosos, con una precisión



(a) TPR (b) TPR

Figura 4.8: TPR (a) y FPR (b) obtenidos por los esquemas local autónomo y distribuido, para distinto número de nodos maliciosos.

superior al 93 %. Además, y no menos importante que lo anterior, el sistema obtiene una tasa de falsos positivos muy baja, menor del 4 % en cualquier situación.

Se ha llevado a cabo también la comparación entre estos resultados y los obtenidos por otros esquemas similares que pueden encontrarse en la literatura, como [99], [101] o [105]. Recordemos, a partir de lo explicado en la Sección 3.3.4, las bases de estas otras aproximaciones. En [99] se introduce una aproximación multi-capa compuesta por tres subsistemas distintos que emplean una clasificador Bayesiano, cadenas de Markov y un algoritmo de asociación de reglas para la detección en capa MAC, de red y de aplicación, respectivamente. Los resultados de estos tres subsistemas se integran en un módulo local, enviándose el resultado final a un módulo global. Los autores en [101] utilizan un algoritmo de clasificación lineal denominado FDA para eliminar datos con poca información, pudiendo entonces utilizar un clasificador SVM en redes MANET. Por su parte, el esquema propuesto en [105] introduce un método de entrenamiento dinámico para realizar la detección, usando como parámetros para expresar el estado de la red el número de paquetes de control enviados y recibidos, así como la diferencia media entre el número de secuencia enviado en los paquetes RREQ y el recibido en los RREP. El estado es actualizado dinámicamente para poder detectar de forma más precisa las potenciales desviaciones del mismo.

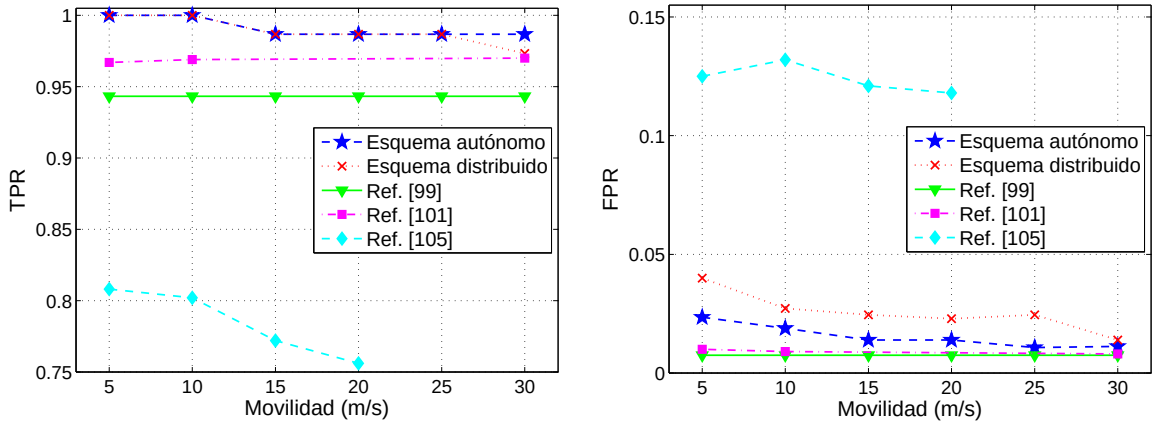
Para mostrar la oportunidad de la comparación de las prestaciones de detección, la Tabla 4.4 presenta algunos parámetros que definen los escenarios considerados para el análisis en cada uno de los esquemas. Estas similitudes entre todos ellos validan la adecuación de esta comparativa. Nótese que el trabajo propuesto en [99]

Tabla 4.4: Comparación de las características de los escenarios para distintos esquemas de detección.

Características	Esquema propuesto	Ref. [99]	Ref. [101]	Ref. [105]
Número de nodos	25	30	30-50	30
Número de atacantes	1-20	1	3	1
Densidad de tráfico	≈ 80 %	-	≈ 60 %	≈ 100 %
Modelo de movilidad	RWP (5-30 m/s)	RWP (-)	RWP (0-30 m/s)	RWP (1-20 m/s)

no proporciona información sobre la movilidad de los nodos, por lo que se ha asumido que los resultados de detección son independientes de la movilidad en este esquema.

La Figura 4.9 muestra cómo los resultados de nuestro esquema superan los obtenidos por las otras aproximaciones. Por ejemplo, los resultados de TPR en [105] no superan el 80 %, con un 12 % mínimo para la tasa FPR. Por otro lado, los resultados de detección obtenidos en [99] son similares a los nuestros, al obtener un menor valor de TPR pero también uno menor de FPR. Sin embargo, este esquema integra tres subsistemas diferentes (clasificador Bayesiano, cadenas de Markov y algoritmo de asociación de reglas), siendo por tanto una aproximación mucho más compleja y costosa. De forma análoga, los resultados obtenidos en [101] son comparables con los obtenidos por nuestro esquema, pero el sistema que se propone incurre en un mayor *overhead* debido al uso de clasificadores SVM no lineales y del algoritmo FDA.



(a) TPR (b) FPR

Figura 4.9: TPR (a) y FPR (b) obtenidos por los esquemas local autónomo y distribuido, comparados con otros esquemas similares.

Resumiendo, nuestra propuesta mejora claramente (en cualquiera de sus versiones) las prestaciones de otras soluciones actualmente disponibles para la detección de *droppers*.

4.7. Conclusiones del capítulo

En este capítulo se aborda la problemática de la detección de ataques de *dropping* en redes MANET. Para este fin se propone una nueva metodología multi-capa, en la que se recopilan estadísticas de la capa MAC y de red. La aproximación propuesta se basa en el desarrollo de un modelo analítico que representa el proceso de retransmisión de información en redes ad hoc, incluyendo de forma nativa las distintas circunstancias que pueden ocasionar descartes legítimos de paquetes, como colisiones, errores en el canal o situaciones de movilidad. El uso de una heurística sencilla en nuestro esquema introduce un menor *overhead* en relación con otras técnicas más sofisticadas encontradas en la literatura, generalmente basadas en algoritmos de *data mining*. Además, al incluirse en el modelo las citadas situaciones de descartes legítimos, que usualmente no son tenidas en cuenta en trabajos previos, los resultados de detección mejoran especialmente en términos de falsos positivos.

El desarrollo realizado se sustenta en un nuevo método de enventanado para la recolección y análisis de las observaciones de red empleadas. Este enventanado, basado en eventos, elimina algunas de las limitaciones existentes en el enventanado temporal tradicional, además de resultar una mejor aproximación en aquellos nodos con baja o nula actividad, resultando así además en un menor consumo de recursos.

Se han propuesto dos posibles implementaciones prácticas del IDS: un método local autónomo (*stand-alone*), en el que cada nodo se encarga de recopilar información sobre sus propias características, y un esquema distribuido, que se sustenta en nodos trabajando en modo promiscuo para realizar esta recolección de modo colaborativo.

Se ha verificado el correcto funcionamiento del sistema mediante la simulación de distintos escenarios y despliegues MANET. Los resultados obtenidos resaltan las bondades de la aproximación IDS propuesta, que consigue un TPR medio del 93 %, con un FPR medio inferior al 4 %, lo que supera ampliamente los resultados proporcionados por otros esquemas similares. Se ha realizado además una exhaustiva experimentación, evaluando las capacidades del sistema frente a numerosas circunstancias y situaciones.

Publicaciones relacionadas

Para finalizar este tema se presentan de nuevo las publicaciones derivadas y relacionadas con el ámbito de estudio objeto de discusión. Son:

- *Enviada* → **L. Sánchez-Casado**, G. Maciá-Fernández, y P. García-Teodoro. "A Model of Data Forwarding in MANETs for Lightweight Detection of Malicious Packet Dropping". *The Scientific World Journal (Hindawi)*, 25 páginas, 2014.
- **L. Sánchez-Casado**, G. Maciá-Fernández y P. García-Teodoro. "An Efficient Cross-Layer Approach for Malicious Packet Dropping Detection in MANETs". *11th IEEE International Conference on Trust, Security and Privacy in Computing and Communications* (Trustcom), pp. 231-238, 2012.
- **L. Sánchez-Casado**, G. Maciá-Fernández y P. García-Teodoro. "Multi-Layer Information for Detecting Malicious Packet Dropping Behaviors in MANETs". *Reunión Española sobre Criptología y Seguridad de la Información* (RECSI), pp. 57-62, 2012.
- **L. Sánchez-Casado**, G. Maciá-Fernández y P. García-Teodoro. "Caracterización de Servicios en Redes Ad Hoc Inalámbricas mediante Métricas Cross-Layer". *X Jornadas de Ingeniería Telemática* (JITEL), pp. 381-384, 2011.

Capítulo 5

Detección de ataques *sinkhole* en redes MANET

En el capítulo anterior se presentó un esquema de detección de ataques de *dropping*. En el presente nos centraremos en otra de las principales amenazas existentes en las redes MANET: los ataques de *route poisoning* y, en particular dentro de esta categoría, en los ataques *sinkhole*. Estos últimos son usualmente empleados como paso previo para realizar ataques más complejos (en particular, ataques de *dropping* entre otros muchos), siendo por tanto su ejecución ampliamente extendida en entornos MANET. Por este motivo, el estudio de posibles soluciones de detección para este tipo de ataques sigue siendo un reto abierto en el campo de la seguridad.

Se propone y discute en el presente capítulo un sistema de detección de ataques *sinkhole* basado en la existencia de lo que denominamos "bordes de contaminación", esto es, nodos legítimos bajo la influencia del ataque y que a su vez son vecinos de nodos que no están afectados por este. El detector propuesto recopila como parámetros para llevar a cabo la detección el número de secuencia de las distintas rutas en estos nodos pertenecientes a los bordes de contaminación además de en sus vecinos. Mediante el cálculo de una heurística sencilla con toda la información recopilada se pueden encontrar incoherencias en las rutas, pudiéndose detectar de forma simple y efectiva comportamientos *sinkhole* maliciosos. A pesar de la existencia de diversas aproximaciones en la literatura que emplean los números de secuencia como parámetros para llevar a cabo la detección de estos comportamientos, la contribución aquí presentada difiere de estas en varios aspectos fundamentales. La diferencia más importante reside en el cálculo de la heurística en base a la obtención colaborativa de la información, es decir, la heurística se calcula no solo

a partir de información local del nodo que realiza el proceso de detección, sino incorporando información proporcionada por los vecinos de este. Además, como una fase previa destinada a reducir la sobrecarga de tráfico introducida por esta detección colaborativa, se ejecuta también una pre-detección local.

Estos aspectos diferenciados derivan en la obtención de unas excelentes capacidades de detección por parte del IDS propuesto, siendo dichas capacidades verificadas mediante la experimentación realizada. Los resultados obtenidos demuestran la eficacia del sistema en términos de detección además de su potencial aplicabilidad en escenarios ad hoc, al introducirse un *overhead* de comunicación asumible en dichos entornos.

En este marco de trabajo, el resto del capítulo se organiza como sigue. En la Sección 5.1 se motiva la necesidad de desarrollar mecanismos de detección de ataques *sinkhole* en redes MANET. La existencia de las denominadas "zonas de contaminación" y "bordes de contaminación", así como su potencial utilidad como base de nuestro sistema de detección, se introducen en la Sección 5.2. La propuesta específica planteada para la detección de ataques *sinkhole* se explica en la Sección 5.3, detallándose también las posibilidades de comunicación que pueden emplearse para una fase de detección colaborativa. En la Sección 5.4 se describe el entorno de simulación desplegado para evaluar la aproximación de detección y los resultados en él obtenidos, discutiéndose también algunas consideraciones acerca del esquema de detección propuesto. Finalmente, en la Sección 5.5 se presentan las principales conclusiones de este capítulo.

5.1. Motivación

En los Capítulos 2 y 3 se ha puesto de manifiesto cómo los ataques de *route poisoning* [9] son (junto con los de *dropping*) una de las principales amenazas en entornos MANET. Esta tipología de ataques consiste en la modificación, creación o eliminación de paquetes de encaminamiento con el objetivo de modificar el funcionamiento normal del protocolo de encaminamiento empleado y, consecuentemente, alterar la operación normal de la red y de los servicios proporcionados. Esta categoría general incluye diversos ataques, como son *sinkhole*, *wormhole*, *link spoofing* o *route cache poisoning*, entre otros.

El presente trabajo se centra en el estudio del ataque *sinkhole*, posiblemente uno de los más representativos y ampliamente ejecutados de los ataques de *route poisoning*. En este ataque los nodos maliciosos intentan falsificar las rutas origen-destino multi-salto con el objetivo de atraer el tráfico de la red a través de ellos. Para esto, los nodos *sinkhole* modifican los paquetes de control mediante la publicación de información de encaminamiento falsa (como el número de saltos, el número de

secuencia de la ruta, la calidad del enlace, etc.) con el fin de introducirse en las rutas, logrando ser considerados por parte de los nodos legítimos como el mejor camino hacia el destino. De esta manera, los nodos *sinkhole* son elegidos como el siguiente salto en la ruta comprometida. En este punto, como se ha indicado con anterioridad, los nodos maliciosos podrán llevar a cabo subsecuentemente una serie de acciones maliciosas adicionales (*p.ej.*, descarte de paquetes).

Tomando como eje la detección de este tipo de ataques *sinkhole*, en este trabajo se propone un nuevo esquema de detección sustentado en la existencia de lo que hemos denominado "bordes de contaminación". Dichos bordes están formados por aquellos nodos legítimos de la red que se encuentran bajo la influencia del ataque *sinkhole* (*i.e.*, poseen información de encaminamiento falsificada) y, al mismo tiempo, son vecinos de otros nodos legítimos que no han sido contaminados aún. A este respecto se plantea la hipótesis de que, en dichos nodos frontera, la información de encaminamiento para las distintas rutas es más inconsistente y, por tanto, es posible detectar comportamientos anómalos. De este modo, recopilando y analizando la información de *routing* en dichos nodos frontera, así como la información de su vecindad, estos "bordes de contaminación" podrán determinar de forma más precisa la existencia de comportamientos *sinkhole* en la red.

Basándonos en esta hipótesis de partida se propone un esquema colaborativo de detección de ataques *sinkhole* en dos fases. La primera fase consiste en un proceso de pre-detección local, principalmente destinado a reducir el tráfico introducido por el proceso de detección llevado a cabo en la segunda fase. Solo cuando este primer proceso lanza una alarma, el IDS inicia una segunda fase colaborativa en la que se recopilan parámetros de la capa de red de los nodos vecinos para estimar el potencial comportamiento malicioso de un nodo dado. Esta aproximación en dos fases comporta dos beneficios principales: (*i*) el *overhead* se ve reducido como consecuencia de emplear una primera fase de pre-detección, pudiendo ser empleado el IDS en entornos de recursos limitados, y (*ii*) las capacidades globales del sistema se ven mejoradas respecto a las logradas por otros esquemas de detección propuestos en la literatura gracias al empleo de información distribuida y el uso de una heurística diferenciadora.

Estas capacidades han sido probadas para el protocolo AODV [15], uno de los más empleados en redes MANET. Sin embargo, tal y como se explicó en la Sección 2.4.2, los ataques *sinkhole* pueden ser fácilmente implementados en otros protocolos similares, como DYMO [43] o DSR [42]. De este modo, aunque hay algunas diferencias en el funcionamiento de AODV con respecto a otros protocolos, la gran mayoría de ellos emplea algún tipo de identificador para las rutas similar a los números de secuencia usados en AODV, pudiendo ser dichos identificadores explotados por parte de los nodos maliciosos para llevar a cabo ataques *sinkhole*. Así, el esquema de detección presentado en este capítulo puede ser fácilmente extendido a otros protocolos sin más que realizar algunas ligeras modificaciones en el cálculo de la heurística.

Además, al igual que ocurre con el sistema de detección presentado en el Capítulo 4, es necesario disociar el proceso de recopilación de parámetros del proceso de detección como tal. En este trabajo, la información necesaria para la detección se obtiene de forma colaborativa entre el nodo que realiza el proceso de detección y su vecindad, por lo que puede considerarse que el proceso de obtención de los parámetros es un proceso también distribuido. Sin embargo, una vez que el nodo que está ejecutando el IDS ha reunido toda la información necesaria, el cálculo de la heurística y la toma de decisiones se realizan de forma aislada e independiente por parte de dicho nodo.

A continuación se discute la hipótesis de partida sobre la que residen las bases del IDS propuesto: la existencia de "bordes de contaminación".

5.2. "Bordes de contaminación" en el ataque *sinkhole*

En esta sección se introduce la existencia de lo que denominamos "bordes de contaminación" [143], mostrándose también cómo los nodos que los componen pueden actuar a su vez, de forma no intencionada, como nodos *sinkhole*. También se discute la utilidad de dichos nodos *frontera* como base de nuestra aproximación de detección.

Consideremos la existencia de una red MANET compuesta por L nodos legítimos $\{N_1,...,N_L\}$, geográficamente distribuidos en un área dada y moviéndose a una cierta velocidad y con una cierta trayectoria. Para cada nodo N_i de la red se extraen los parámetros de interés siguiendo un procedimiento temporal que considera instantes discretos cada W segundos: $t_1,\ldots,t_N$, donde $t_k = k{\cdot}W$ segundos, con $k = 1,\ldots,N$. Dado que se está asumiendo la movilidad de los nodos de la red, cada nodo N_i tendrá su propio conjunto de vecinos NB_i^t, formado por aquellos nodos que comparten un enlace inalámbrico con N_i en el instante t.

Los nodos pueden generar distintos tipos de flujos de tráfico (CBR, VBR, ...), y se comunican entre sí empleando AODV como protocolo de *routing*. De este modo, la tabla de rutas de cada nodo contendrá las entradas correspondientes a cada ruta aprendida, definiéndose $R_{i,j}^t$ como la ruta aprendida por el nodo N_i hacia un destino dado N_j en un instante t. Las rutas están compuestas, entre otros campos, por la siguiente información: $R_{i,j}^t = \left\{SN_{i,j}^t, NH_{i,j}^t\right\}$, donde $SN_{i,j}^t$ es el número de secuencia aprendido para la ruta $N_i \rightarrow N_j$ y $NH_{i,j}^t$ representa el siguiente salto hacia el destino en el instante de observación t.

En este escenario general se considerará también la existencia de M nodos maliciosos comportándose como nodos *sinkhole*, *i.e.*, nodos que responden a las solicitudes

RREQ con falsos mensajes RREP, con la intención de introducirse como siguiente salto en la ruta hacia un destino dado (véase Sección 2.5.2).

5.2.1. Existencia de "bordes de contaminación"

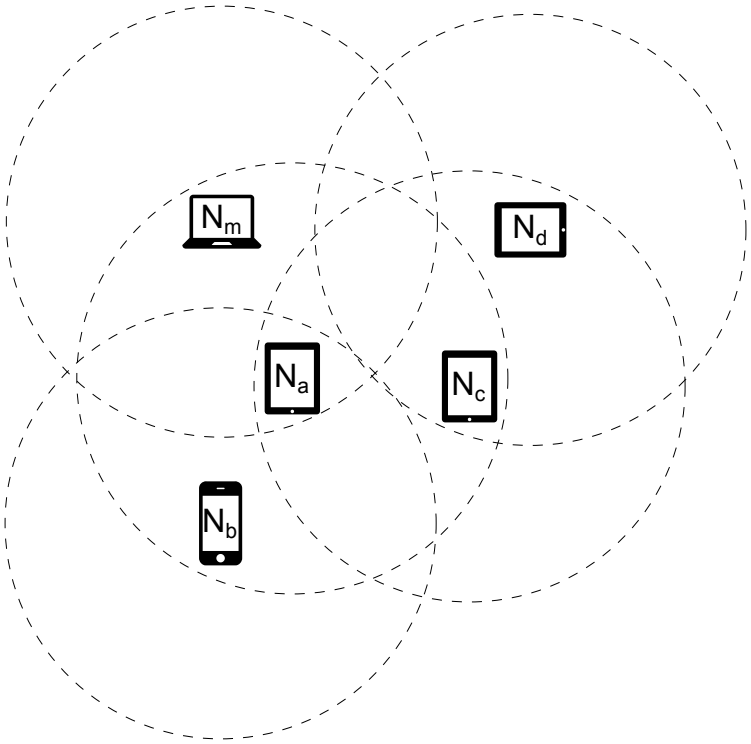
En el escenario general presentado previamente nuestra aproximación de detección se basa en la existencia de zonas de contaminación, formadas por nodos legítimos que se encuentran bajo la influencia de un ataque. De este modo, aquellos nodos que poseen información falsa para alguna de sus rutas se considerarán contaminados. Algunos de estos nodos pertenecientes a una zona de contaminación conformarán el "borde de contaminación". La peculiaridad de estos últimos nodos es que son vecinos, de forma simultánea, de nodos contaminados y de nodos que no están bajo la influencia del ataque (es decir, nodos que potencialmente pueden tener el conocimiento sobre las rutas legítimas).

Los nodos en las zonas de contaminación retransmitirán el tráfico a través del nodo *sinkhole*. Al mismo tiempo, cuando un nodo no contaminado pregunta a alguno de los bordes de contaminación por una ruta que ha sido previamente comprometida, este último responderá con la información falsa que él posee, *i.e.*, el nodo frontera publicará involuntariamente las rutas falsas que haya aprendido cuando sea preguntado. En dicha situación, estos nodos frontera se comportarán de forma análoga a como lo haría un nodo malicioso, actuando a su vez como nodos *sinkhole*.

Esta idea puede clarificarse con el ejemplo mostrado en la Figura 5.1. En un instante dado t_0, el nodo N_c posee una ruta legítima hacia el destino N_d con número de secuencia 35. En t_1, N_b necesita una ruta hacia N_d y genera una solicitud RREQ, que es retransmitida por N_a. Como consecuencia, en t_2, N_m responde con un falso mensaje RREP que incluye un número de secuencia artificialmente aumentado (*p.ej.*, 100), y N_c responde con un mensaje RREP legítimo. Puesto que el número de secuencia enviado por N_c será menor, el nodo N_a aprenderá la ruta a través de N_m y retransmitirá la respuesta hacia N_b. Las rutas son actualizadas en t_3.

En tal situación, N_a se incorporará al borde de contaminación, al enviar un falso mensaje RREP sin intenciones maliciosas. Así, la zona contaminada estará formada por los nodos N_a y N_b, siendo N_m el nodo malicioso. Por su parte, los nodos N_c y N_d permanecerán sin contaminar.

Bajo estas circunstancias, la única diferencia entre un nodo *sinkhole* y un nodo contaminado es que el primero intentará atraer de forma deliberada la mayor parte del tráfico circundante, mientras que el nodo contaminado únicamente actuará como nodo *sinkhole* para aquellas solicitudes relacionadas con rutas contaminadas aprendidas, y no para cada solicitud que reciba y, por supuesto, sin intenciones maliciosas.



	N_a				N_b				N_c			
T	***Dst***	***NH***	***NS***	***Est***	***Dst***	***NH***	***NS***	***Est***	***Dst***	***NH***	***NS***	***Est***
t_0	-	-	-	-	-	-	-	-	N_d	N_d	35	VAL
t_1	Reenvía RREQ				Genera RREQ				N_d	N_d	35	VAL
t_2	Recibe RREP				Recibe RREP				N_d	N_d	35	VAL
t_3	N_d	N_m	100	VAL	N_d	N_a	100	VAL	N_d	N_d	35	VAL

Figura 5.1: Existencia de zonas y bordes de contaminación.

De este modo, se puede concluir:

Corolario. *Los nodos en la "zona de contaminación" se comportan como nodos* sinkhole *para las rutas contaminadas, pero la existencia de vecinos no contaminados permitirá la detección de incoherencias en dichas rutas.*

Este ejemplo se puede extender a una situación más compleja y realista. En la Figura 5.2 es posible observar un ejemplo de la evolución de las zonas y bordes de contaminación a lo largo del tiempo. El ejemplo representa un escenario estático con 25 nodos (sin movilidad) aleatoriamente distribuidos. Los nodos legítimos, de N_0 a N_{23}, se representan con círculos azules; el nodo *sinkhole* N_m con un triángulo rojo, mientras que los nodos legítimos que han sido contaminados se representan mediante cuadrados magenta, conectados por líneas azules discontinuas al siguiente salto de la ruta contaminada. Entre t_0 y t_2, cada nodo legítimo inicia un flujo de aplicación hacia un destino elegido de forma aleatoria.

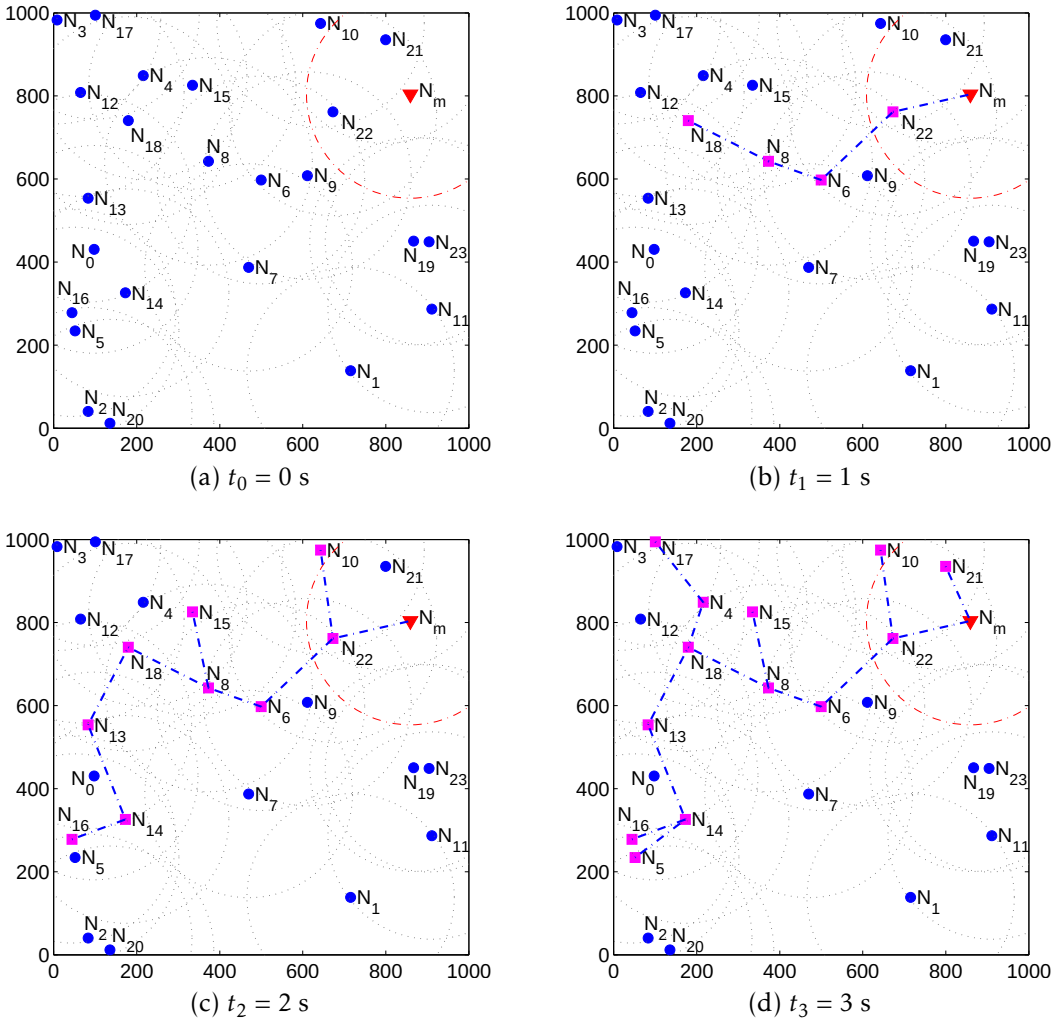


Figura 5.2: Evolución de las zonas y bordes de contaminación.

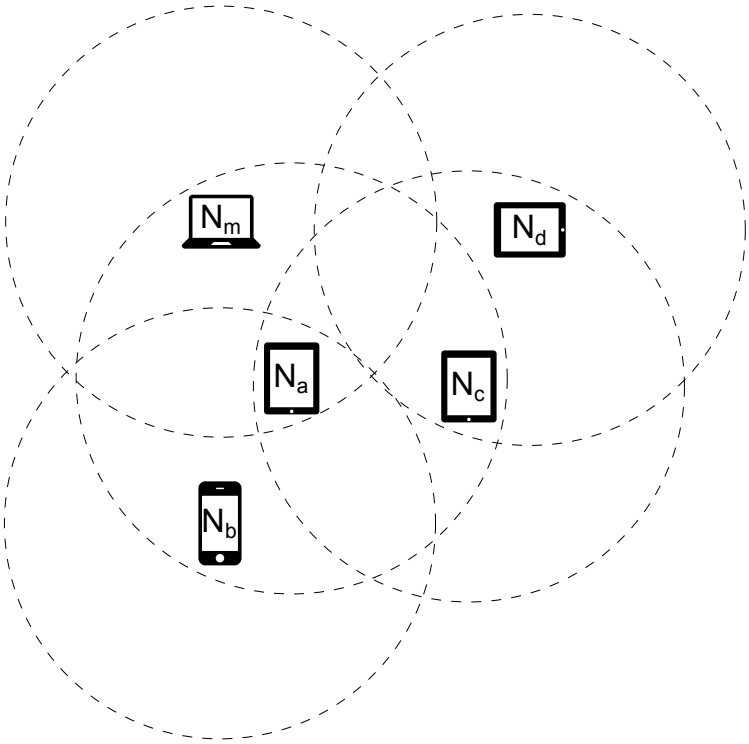
Podemos observar que al comienzo de la simulación (t_0) únicamente N_m está actuando como nodo *sinkhole*. En el instante t_1 algunos de los nodos legítimos ya han sido contaminados, es decir, contienen al menos una ruta falsa debido a la acción del nodo *sinkhole*. En esta situación, el nodo N_{22} se convertirá en un borde de contaminación, al encontrarse bajo la influencia del *sinkhole* N_m y, al mismo tiempo, ser vecino de nodos legítimos no contaminados, como el nodo N_9. Como se explicará en la siguiente subsección, utilizando conjuntamente su propia información y la proporcionada por el nodo N_9, es más probable que el nodo N_{22} sea capaz de clasificar al nodo N_m como malicioso. En t_3, tras solo 3 segundos de simulación, más de la mitad de los nodos legítimos de la red se encuentran dentro de la zona de contaminación.

5.2.2. Detección de nodos *sinkhole* basada en "bordes de contaminación"

Como se ha indicado en la Sección 5.1, la mayoría de las aproximaciones actuales de detección frente a ataques *sinkhole* consideran únicamente la información directamente accesible por parte del nodo que está realizando el proceso de detección. En particular, este tipo de esquemas emplean alguna métrica relacionada con la diferencia entre el número de secuencia enviado y recibido por el nodo detector. Sin embargo, en esta sección mostraremos cómo este tipo de aproximaciones presentan ciertos inconvenientes que pueden derivar en errores en el proceso de detección y cómo el esquema aquí propuesto puede superar dichas limitaciones.

El primer inconveniente está relacionado con el hecho de que las aproximaciones existentes pueden proporcionar buenos resultados siempre y cuando los números de secuencia publicados por los nodos *sinkhole* sean elevados. Es decir, siempre y cuando la diferencia entre el número de secuencia recibido en el mensaje RREP por el nodo detector y el número de secuencia enviado en el mensaje RREQ sea suficientemente elevada. Sin embargo, el problema aparece cuando los nodos *sinkhole* son suficientemente "inteligentes", en cuyo caso intentarán publicar números de secuencia falsos moderadamente altos, asegurándose que estos son lo bastante elevados como para hacer que el nodo *sinkhole* sea seleccionado como el siguiente salto en la ruta, pero que al mismo tiempo sean lo suficientemente bajos como para dificultar el proceso de detección. Por otro lado, los nodos legítimos contaminados, al haber aprendido rutas falsas podrían publicarlas cuando se les pregunte por ellas, ayudando sin darse cuenta al nodo *sinkhole* en su objetivo de atraer tráfico. Y por si fuera poco, dichos nodos legítimos contaminados son, por su parte, propensos a ser también incorrectamente detectados como nodos *sinkhole* maliciosos. Por tanto, ambos hechos pueden ocasionar incrementos en las clasificaciones erróneas de nodos legítimos como maliciosos si se requiere un determinado nivel de detección, degradándose por tanto las capacidades de este tipo de esquemas de detección.

Asumiendo que los "bordes de contaminación" se comportan como nodos *sinkhole* para las rutas contaminadas y que pueden detectar incoherencias en las rutas gracias a la existencia de nodos vecinos no contaminados (véase el Corolario previo en Sección 5.2.1), hemos desarrollado un método para la detección de nodos *sinkhole* basado en el análisis de la evolución de los números de secuencia para las rutas. En nuestra aproximación, cada nodo compara periódicamente el número de secuencia para una ruta dada con el valor almacenado en la última comparación (tal y como se hace en las aproximaciones disponibles mencionadas). Pero, además, cada nodo realiza la misma comparación también con los números de secuencia almacenados para dicha ruta por sus vecinos. Siguiendo nuestra hipótesis y, tal y como se mostrará en la sección experimental, los nodos en los "bordes de contaminación" obtendrán mayores diferencias en estas comparaciones. Cuando el valor de dicha diferencia



	N_a				N_b				N_c			
T	***Dst***	***NH***	***NS***	***Est***	***Dst***	***NH***	***NS***	***Est***	***Dst***	***NH***	***NS***	***Est***
t_0	N_d	N_m	100	VAL	N_d	N_a	100	VAL	N_d	N_d	35	VAL
t_1	N_d	N_m	100	VAL	N_d	N_a	101	INV	N_d	N_d	35	VAL
t_2	Reenvía RREQ				Genera RREQ				N_d	N_d	35	VAL
t_3	Recibe RREP				Recibe RREP				N_d	N_d	35	VAL
t_4	N_d	N_m	121	VAL	N_d	N_a	121	VAL	N_d	N_d	35	VAL

Figura 5.3: Utilidad de un nodo perteneciente al borde de contaminación, N_a, en el proceso de detección de nodos *sinkhole*.

para los números de secuencia sea suficientemente elevado, el siguiente salto en la ruta será etiquetado como malicioso.

El ejemplo sencillo propuesto en la Figura 5.3 puede ayudar en la comprensión de la aproximación desarrollada. En el instante t_0 el nodo N_a posee una ruta falsa hacia el destino N_d a través del nodo *sinkhole* N_m, con número de secuencia 100, dado que dicha ruta fue falsificada con anterioridad (véase el ejemplo de la Figura 5.1). El nodo N_b también posee una ruta hacia N_d con número de secuencia 100, cuyo siguiente salto es el nodo N_a. Además, el nodo N_c conoce una ruta no contaminada hacia N_d, con un número de secuencia legítimo igual a 35. En el tiempo t_1 la ruta hacia el destino N_d en el nodo N_b caduca, por lo que este la marca como inválida (INV) e incrementa el número de secuencia en una unidad (101). En t_2 el nodo N_b vuelve a necesitar comunicarse con el destino N_d y, por lo tanto, genera un mensaje

de solicitud RREQ que será retransmitido por parte del nodo N_a. Como consecuencia, en t_3 el nodo malicioso N_m responderá con un falso mensaje RREP que incluirá un número de secuencia artificialmente incrementado (por ejemplo, 121). Sin embargo, puesto que el número de secuencia para la ruta solicitada en la tabla de rutas del nodo N_c es menor que el incluido en el mensaje RREQ, N_c no responderá a dicha solicitud.

En este punto, asumamos que el esquema de detección en el nodo N_a calcula los valores de los indicadores de detección en el instante t_4, cuando las rutas ya han sido actualizadas. En aproximaciones anteriores, como [112] o [114], el nodo N_a obtendría como indicador la diferencia entre el número de secuencia recibido en el mensaje RREP y el enviado en el mensaje RREQ, dando como resultado $121 - 101 = 20$ unidades, que podría ser un valor suficientemente elevado como para atraer el tráfico hacia el *sinkhole*, pero no como para que este sea detectado por el nodo N_a. Empleando nuestra aproximación, el nodo N_a calcula el valor de la diferencia entre el número de secuencia recibido en el mensaje RREP y el mínimo número de secuencia para la ruta requerida entre los valores almacenados por sus vecinos, dando como resultado $121 - 35 = 86$ unidades, que es un valor indicador claramente mayor que el obtenido con las aproximaciones previas.

Así, utilizando esta aproximación somos capaces de detectar rutas contaminadas e identificar aquellos nodos que están propagando información falsa (nodos contaminados). Sin embargo, nuestro interés real es identificar el nodo *sinkhole* malicioso a partir del conjunto de nodos contaminados. Para hacer esto posible asumimos también que, mientras que los nodos contaminados publican información falsa solamente para aquellas rutas que han aprendido como tales, es más probable que los nodos *sinkhole* envíen dicha información falsa para muchas más rutas, o incluso para todas ellas. Por tanto, en nuestro sistema asignaremos una mayor probabilidad de ser clasificado como malicioso a todos aquellos nodos que han sido etiquetados como sospechosos para muchas rutas. Nótese también que, siguiendo nuestra metodología, únicamente los nodos pertenecientes a los bordes de contaminación que sean vecinos de nodos maliciosos reales serán capaces de identificar a dichos nodos.

A continuación, en la Sección 5.3, se proporcionan todos los detalles acerca de la implementación de este método de detección propuesto.

5.3. Implementación práctica del esquema de detección

En esta sección se presenta la implementación específica del detector de ataques *sinkhole* propuesto, que emplea una heurística sencilla basada en parámetros de la capa de red para obtener un valor indicador para la detección de nodos maliciosos. El detector calcula la heurística recopilando información relativa a las tablas de rutas

del propio nodo que ejecuta el detector y, en caso necesario, de las tablas de rutas pertenecientes a sus vecinos. Así, aunque el proceso de detección es realizado de forma local y autónoma por parte de cada nodo que ejecuta el sistema de detección, los parámetros involucrados en dicho proceso se recopilarían en caso necesario de forma colaborativa por parte de la vecindad del nodo.

5.3.1. Especificación del esquema de detección

Como se dijo en la Sección 5.2, nuestra aproximación sigue un procedimiento basado en un muestreo temporal para detectar nodos maliciosos de forma discreta a lo largo del tiempo. Cada nodo N_i ejecuta un procedimiento de detección local, comprobando periódicamente (con período W) si alguno de sus vecinos se está comportando de forma maliciosa o no. Así, por cada nodo que esté presente como siguiente salto (*next hop*, NH) en la tabla de rutas del nodo N_i se recopilan los siguientes parámetros:

- $\boldsymbol{D^t_{i,NH}}$: conjunto de todos los destinos en la tabla de rutas de N_i que utilizan al nodo NH como siguiente salto, en el instante t. Solo aquellas rutas válidas con $HopCount$ mayor que 1 se deben tener en cuenta para la extracción de dichos destinos, puesto que las rutas con $HopCount = 1$ (nodos vecinos) se aprenden de forma automática y no tienen por qué haber sido publicadas, por lo que no proporcionan información acerca de si un nodo dado está publicando o no rutas falsas.
- $\boldsymbol{SN^t_{i,j}}$: número de secuencia en el nodo N_i para el destino N_j, en el instante t.
- $\boldsymbol{NB^t_i}$: conjunto de vecinos del nodo N_i, en el instante t.

Sobre esta información básica aplicamos una heurística para obtener un valor indicador sobre el comportamiento *sinkhole* o no de un cierto nodo monitorizado. El proceso de decisión se realiza en dos fases. La primera fase (*fase de pre-detección*) está principalmente destinada a detectar sospechas locales sobre la actividad de los nodos como nodos maliciosos. Solo en el caso de que un nodo sea considerado como sospechoso, el nodo detector iniciará la segunda fase (*fase colaborativa*). De este modo, esta aproximación en dos fases consigue reducir la sobrecarga del proceso. En resumen, se ejecuta el siguiente proceso de detección global:

Fase de pre-detección

1) Inicialmente, cada nodo N_i obtiene, para cada nodo NH en su tabla de rutas, un conjunto de valores de sospecha local $LSV_{i,j}$, uno por cada posible destino

N_j en $D^t_{i,NH}$. Cada valor de sospecha local se calcula a lo largo del tiempo como la diferencia entre el número de secuencia de la ruta en el instante de cálculo (t) y en el instante anterior ($t-1$):

$$LSV^t_{i,j} = SN^t_{i,j} - SN^{t-1}_{i,j} \tag{5.1}$$

2) Si, para alguna ruta de las evaluadas, existe al menos un valor de sospecha local mayor que un determinado umbral, θ_s, el nodo NH correspondiente a dicha ruta se considera sospechoso de ser un nodo malicioso:

$$NH = \begin{cases} \text{sospechoso}, & \text{si} \quad \exists\, N_j \in D^t_{i,NH} \,/\, LSV^t_{i,j} \;\geq\; \theta_s \\ \text{legítimo}, & \text{en caso contrario} \end{cases} \tag{5.2}$$

Solo si el nodo NH es clasificado como sospechoso (denotado entonces como NH^*) en la primera fase, se lanzará la fase de detección colaborativa.

Fase de detección colaborativa

3) El detector en el nodo N_i extrae, para cada siguiente salto sospechoso NH^* en su tabla de rutas, el conjunto de destinos D^t_{i,NH^*}, es decir, todos los posibles destinos que se supone están comprometidos.

4) Tras ello, N_i difunde un mensaje solicitando a sus vecinos (NB^t_i) el número de secuencia para cada uno de los destinos N_j en D^t_{i,NH^*}. En la Sección 5.3.2 se discutirá cómo se realiza esta comunicación.

5) Tras recopilar las respuestas enviadas por todos los vecinos, el nodo N_i obtiene el mínimo número de secuencia de entre todos los recibidos para cada destino N_j, y calcula la diferencia entre sus propios números de secuencia almacenados y los respectivos valores mínimos obtenidos:

$$\Delta SN^t_{i,j} = SN^t_{i,j} - \min_{n \in NB^t_i} \{SN^t_{n,j}\} \tag{5.3}$$

6) El siguiente paso es calcular un valor de sospecha global para el nodo NH^*, GSV^t_{i,NH^*}, indicador de la probabilidad de que NH^* sea un nodo malicioso, obtenido como el producto de las diferencias previamente calculadas para cada destino N_j. A través del producto se considera que los nodos NH^* que aparecen en más rutas tienen una mayor probabilidad de ser nodos maliciosos y no simplemente nodos contaminados:

$$GSV^t_{i,NH^*} = \prod_{N_j \in D^t_{i,NH^*}} \left(1 + \Delta SN^t_{i,j}\right) \tag{5.4}$$

Nótese que se añade una unidad a los factores puesto que, para un destino comprometido, la diferencia calculada entre los números de secuencia podría dar como resultado un valor cero.

7) Después del cálculo de GSV^t_{i,NH^*}, si el indicador supera un umbral determinado, θ_d, el nodo NH^* es finalmente clasificado como nodo *sinkhole* malicioso:

$$NH^* = \begin{cases} \text{malicioso}, & \text{si } GSV^t_{i,NH^*} \geq \theta_d \\ \text{legítimo}, & \text{en caso contrario} \end{cases} \tag{5.5}$$

8) Como consecuencia de la clasificación del nodo NH^* como *sinkhole*, el nodo monitor N_i podría aplicar algún mecanismo de respuesta, como la inclusión de NH^* en una lista negra (*blacklist*) o la notificación a todos los nodos de la red acerca del comportamiento malicioso de NH^* mediante la difusión de mensajes de alerta.

La descripción detallada del proceso de detección llevado a cabo se muestra en el Algoritmo 5.1. Se puede observar cómo el cálculo del comportamiento malicioso de un nodo dado es un proceso sencillo con un bajo coste computacional una vez que la información necesaria de los vecinos ha sido recopilada.

El punto de operación del sistema de detección depende de los valores seleccionados para los umbrales θ_s y θ_d. La idea principal es fijar θ_s a un valor bajo, dado que su utilidad es reducir el *overhead* mediante la determinación de un valor de sospecha local. Además, si hacemos θ_s demasiado alto, disminuirá la tasa TPR. En relación con θ_d, fijar este umbral a un valor bajo dará lugar a más nodos *sinkhole* detectados, pero también a más nodos legítimos erróneamente clasificados como maliciosos. Por otro lado, valores elevados de θ_d producirán menos falsos positivos, a expensas de, como para θ_s, reducir el número de nodos maliciosos detectados. Como se verá en la Sección 5.4, típicamente la mejor opción es seleccionar para los umbrales un valor de compromiso entre ambas situaciones.

5.3.2. Protocolo de comunicación y detección colaborativa

Como se explicó con anterioridad, la fase colaborativa del proceso de detección implica la comunicación entre distintos nodos para obtener la información sobre sus números de secuencia para las rutas sospechosas (pasos 4) y 5) previos). Para dicha comunicación pueden emplearse dos posibilidades: (*i*) usar los propios mensajes definidos en AODV, que es una solución más sencilla pero que implica un mayor *overhead* en número de mensajes de control necesarios para realizar la detección (como se discutirá a continuación), o (*ii*) diseñar e implementar nuevos mensajes, bien

Algoritmo 5.1 Pseudo-código para el proceso de detección de ataques *sinkhole*.

```
1: para cada instante t = k·W con k = 1,...,N en el tiempo de monitorización hacer
2:   para cada nodo N_i en la red hacer
3:     para cada siguiente salto NH en la tabla de rutas del nodo N_i hacer
4:       Obtener D^t_{i,NH}
5:       para cada destino N_j ∈ D^t_{i,NH} hacer
6:         Obtener LSV^t_{i,j} usando (5.1)
7:         si LSV^t_{i,j} ≥ θ_s entonces
8:           marcar NH como sospechoso (de acuerdo con (5.2))
9:         fin si
10:      fin para
11:      para cada NH marcado como sospechoso, NH*, hacer
12:        para cada nodo vecino N_v ∈ NB^t_i hacer
13:          para cada destino N_j ∈ D^t_{i,NH*} hacer
14:            Solicitar SN^t_{v,j}
15:          fin para
16:        fin para
17:        Calcular GSV^t_{i,NH*} empleando (5.4)
18:        si GSV^t_{i,NH*} ≥ θ_d entonces
19:          marcar NH* como malicioso (según (5.5))
20:        fin si
21:      fin para
22:    fin para
23:  fin para
24: fin para
```

para el propio protocolo AODV o bien como un protocolo específico independiente para reducir el número de mensajes involucrados en dicho proceso de detección.

Uso de mensajes de AODV

La primera opción es emplear los propios mensajes RREQ y RREP del protocolo AODV. Cuando el sistema de detección en el nodo N_i clasifica un determinado nodo NH^* como sospechoso, se podría enviar simplemente en modo *broadcast* un mensaje RREQ solicitando a los vecinos el número de secuencia para cada destino que emplee NH^* como siguiente salto. Fijando el *número de secuencia destino* del mensaje RREQ al valor mínimo nos aseguramos de que todos los nodos que reciban el mensaje y que posean un número de secuencia válido transmitan de vuelta un mensaje RREP que incluya dicho número de secuencia. Fijando el campo TTL de la cabecera IP al valor 1 nos aseguramos también que el mensaje no se difunda por la red más

allá de la vecindad, es decir, que sea respondido únicamente por los vecinos del nodo solicitante.

La principal ventaja de este esquema es su simplicidad, siendo su uso directamente compatible con el protocolo AODV ya desplegado en la red, *i.e.*, dado que se están empleando los propios mensajes de AODV, no hay necesidad de modificar el protocolo. Sin embargo, el problema de usar los mensajes RREQ/RREP es que, para cada posible destino N_j que utilice al nodo NH^* como siguiente salto, se requiere un intercambio de mensajes entre el nodo N_i y sus vecinos. Dependiendo del número de vecinos y también del número de destinos a consultar, el consumo del ancho de banda podría ser elevado (se analizará este aspecto más adelante).

Uso de nuevos mensajes

Una segunda posibilidad es definir nuevos mensajes específicamente diseñados para el sistema desarrollado. Estos mensajes pueden establecerse como un nuevo tipo de mensajes en AODV o como mensajes propios de un protocolo de comunicación diferente. En este trabajo hemos optado por la segunda opción, tal y como se verá en el Capítulo 6, en el que se define un protocolo específico de notificación y alerta de eventos de seguridad para redes MANET, siendo los nuevos mensajes aquí detallados una particularización de los que han sido diseñados para el intercambio de información de seguridad entre entidades a través del citado nuevo protocolo.

Así, cuando el sistema de detección en el nodo N_i considera un nodo NH^* como sospechoso, difunde un mensaje de solicitud de información que incluye las direcciones IP de los destinos D^t_{i,NH^*} para los que se requiere el número de secuencia, empleando la dirección IP de *broadcast* y fijando el campo TTL de la cabecera IP a valor 1. De este modo, los nodos a un salto de distancia del nodo N_i (nodos vecinos) recibirán dicha solicitud. El formato de los mensajes de solicitud se ilustra en la Figura 5.4. A continuación se proporciona una breve descripción de los distintos campos, aunque estos serán explicados en mayor detalle en la Sección 6.3.3:

- *Tipo mensaje*: tipo del mensaje, solicitud de información de seguridad en este caso.
- # *Nodos*: correspondiente al número de nodos para los que se requiere el valor del número de secuencia asociado a un destino.
- # *Variables*: cantidad de parámetros que se solicitan para todos los nodos solicitados. En este caso particular el campo se fijará al valor 1, pues el único parámetro solicitado es el número de secuencia de la ruta.
- *Longitud total*: longitud del mensaje, en palabras de 32 bits.

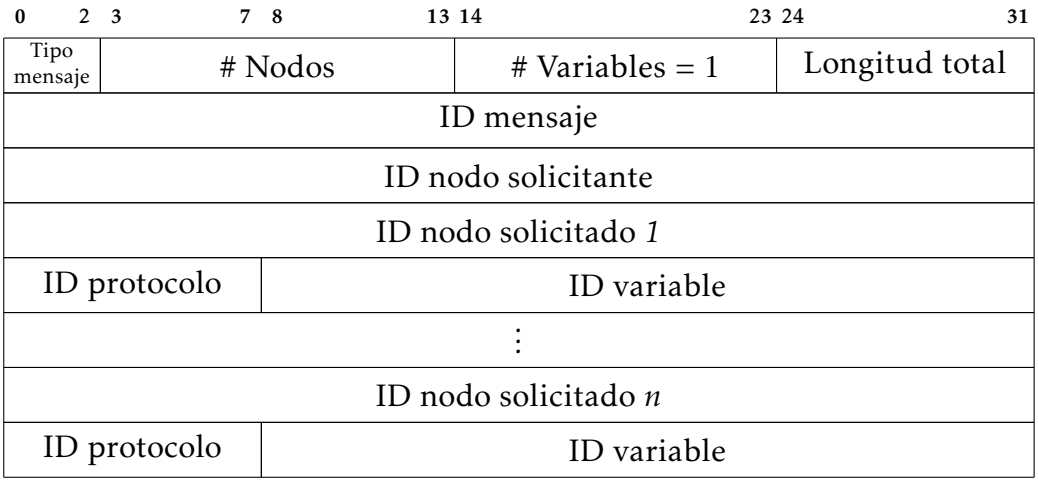


Figura 5.4: Formato de los mensajes de solicitud de información de seguridad, particularizado para la solicitud de una única variable (véase Figura 6.2).

- *ID mensaje*: número monótonamente creciente identificativo del mensaje para hacer corresponder solicitudes con respuestas y, entre otros fines, robustecer el protocolo ante ataques de repetición.
- *ID nodo solicitante*: identifica unívocamente el nodo que solicita la información y que, en definitiva, se prevé llevará a cabo el proceso de detección posterior.
- *ID nodo solicitado* 1...*n*: identificador unívoco de cada uno de los nodos de los que se solicita información.
- *Variable i*: campo de 32 bits a través del cual se identifica la variable de la que se pide información para cada nodo solicitado *i*. Como se verá en la Sección 6.3.3, una variable queda definida por dos campos: *protocolo/procedimiento* al que hace referencia e *identificador* dentro del mismo.

Los nodos vecinos que reciben el mensaje de solicitud responden enviando un mensaje de respuesta *unicast* de vuelta al nodo solicitante. El formato de los mensajes de respuesta se muestra en la Figura 5.5, el cual es una particularización de los de respuesta mostrados en la Figura 6.3. Los campos empleados por dichos mensajes son los siguientes:

- *Tipo mensaje*: respuesta a solicitud de información de seguridad.
- *# Nodos*: correspondiente al número de nodos para los que se proporciona el valor del número de secuencia.
- *# Variables*: número de parámetros proporcionados. En el caso que nos ocupa, un único parámetro.

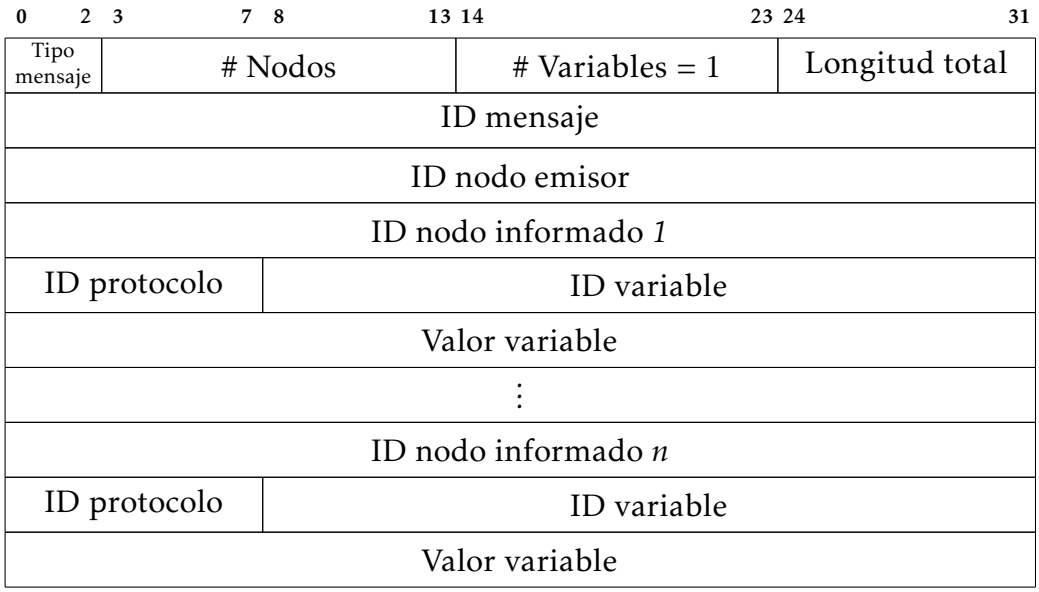


Figura 5.5: Formato de los mensajes de respuesta de información de seguridad, particularizado para la respuesta sobre una única variable (véase Figura 6.3).

- *Longitud total*: longitud del mensaje, en palabras de 32 bits.
- *ID mensaje*: como antes, para hacer corresponder solicitudes con respuestas.
- *ID nodo emisor*: para identificar unívocamente el nodo que envía la información.
- *ID nodo informado* 1...*n*: identificador unívoco de cada uno de los nodos de los que se proporciona información.
- *Variable i*: campo de 64 bits a través del cual se identifica e informa de la variable de la que se requirió información para cada nodo solicitado i. Tras ser identificada cada variable (con 32 bits, 8 de los cuales se usan para indicar el protocolo/procedimiento al que se refiere), seguidamente se especificará su valor mediante un campo de 32 bits.

Mediante el uso de estos mensajes específicamente diseñados se reduce el número de paquetes de control involucrados en el proceso de detección, puesto que se necesita un único flujo de comunicación entre el nodo solicitante N_i y sus vecinos. Esta aproximación permite solicitar información acerca de todos los posibles destinos que emplean al nodo NH^* como siguiente salto en un único mensaje de solicitud. De forma similar, también permite que los nodos vecinos envíen toda la información al nodo solicitante transmitiendo un único mensaje de respuesta. Sin embargo, la desventaja de emplear esta aproximación es la necesidad de modificar convenientemente el protocolo AODV para la correcta gestión de estos nuevos mensajes o, como

proponemos aquí, la necesidad de diseñar y disponer un nuevo protocolo específico para el intercambio de información entre entidades de detección.

Comparativa de los protocolos de comunicación propuestos

A continuación se presenta una breve discusión acerca del *overhead O* (expresado en bytes), introducido por cada una de las aproximaciones de comunicación previamente presentadas. Para ello se hace necesario definir algunas variables adicionales, así como su notación:

- $\boldsymbol{E[NB_i^t]}$: número esperado de vecinos del nodo N_i, en el instante de estudio t.
- $\boldsymbol{E[D_{i,NH^*}^t]}$: número esperado de rutas en el nodo N_i que emplean al nodo sospechoso NH^* como siguiente salto, en el instante t.
- $\boldsymbol{PS^{rreq/rrep}}$: tamaño de los mensajes RREQ/RREP del protocolo AODV. Dicho tamaño (considerando que no se incluyen extensiones) es de 24 y 20 bytes, respectivamente.
- $\boldsymbol{PS^{new\_rq/new\_rp,t}}$: tamaño de los nuevos paquetes empleados. Depende del número de posibles destinos que usan NH^* como siguiente salto y por los que se desea solicitar el número de secuencia, siendo el tamaño de los paquetes de $12+8\cdot E[D_{i,NH^*}^t]$ bytes para los de solicitud y $12+12\cdot E[D_{i,NH^*}^t]$ bytes para los de respuesta.
- $\boldsymbol{p\left(I_j^t\right)}$: probabilidad de que un nodo de la red tomado al azar posea una ruta válida (y consecuentemente, un número de secuencia válido) para un destino N_j solicitado, en el instante t.

Empleando los mensajes RREQ/RREP de AODV se necesita un intercambio de mensajes de solicitud y respuesta entre el nodo N_i y sus vecinos por cada posible destino en la tabla de rutas de N_i que emplee NH^* como siguiente salto en la ruta. Así, en el instante t, el *overhead* esperado introducido por el proceso de detección iniciado por N_i contra NH^*, $O_{i,NH^*}^{aodv,t}$ (en bytes), es:

$$E[O_{i,NH^*}^{aodv,t}] = E[D_{i,NH^*}^t]\cdot\left(PS^{rreq}+PS^{rrep}\cdot E[NB_i^t]\cdot p\left(I_j^t\right)\right) \tag{5.6}$$

Nótese que el término $p\left(I_j^t\right)$ considera el hecho de que si un nodo recibe un mensaje de solicitud y no posee la ruta solicitada no devolverá un mensaje de respuesta.

Respecto de la segunda aproximación, que emplea los nuevos mensajes definidos, se requiere un único intercambio de mensajes entre el nodo N_i y sus vecinos cuando NH es clasificado como sospechoso, NH^*. El intercambio comienza con un mensaje *broadcast* a un único salto, solicitando la información de los números de secuencia de todos los posibles destinos a los vecinos. Aquellos que conozcan la información solicitada responderán con un mensaje de respuesta *unicast* enviado de vuelta hacia el nodo solicitante. Por tanto, en el instante de estudio t, el *overhead* esperado introducido por el nuevo esquema, $O^{new,t}_{i,NH^*}$ (en bytes), es:

$$E[O^{new,t}_{i,NH^*}] = PS^{new\_rq,t} + PS^{new\_rp,t} \cdot E[NB^t_i] \cdot p\left(I^t_j\right) \tag{5.7}$$

Cabe esperar que, aunque el tamaño de los nuevos mensajes pueda ser mayor que el de los mensajes de AODV (dependiendo del número esperado de destinos sobre los que se solicite información), la reducción en el número de mensajes enviados si se emplea la segunda aproximación resulte en un menor *overhead* en bytes. Para estimar en qué situaciones el *overhead* introducido (en bytes) será realmente menor empleando el nuevo esquema, se considera el caso en el que $E[O^{aodv,t}_{i,NH^*}] > E[O^{new,t}_{i,NH^*}]$. A partir de la Ec. (5.6) y de la Ec. (5.7) se puede obtener una expresión en función de $E[D^t_{i,NH^*}]$ que permita calcular el número esperado de posibles destinos sobre los que solicitar información, a partir del cual merece la pena emplear el segundo modelo de comunicación si quiere decrementarse la sobrecarga en bytes. Expandiendo la desigualdad, el *overhead* se reduce utilizando nuestra aproximación si

$$E[D^t_{i,NH^*}] > \frac{12 \cdot E[NB^t_i] \cdot p\left(I^t_j\right) + 12}{8 \cdot E[NB^t_i] \cdot p\left(I^t_j\right) + 16} \tag{5.8}$$

Teniendo en cuenta que $p\left(I^t_j\right)$ es un valor de probabilidad en el rango [0,1], es posible obtener los valores máximo y mínimo de $E[D^t_{i,NH^*}]$ tomando límites cuando $E[NB^t_i]$ se aproxima a 0 y a ∞, respectivamente:

$$\begin{aligned} \lim_{E[NB^t_i]\to 0} E[D^t_{i,NH^*}] &= 3/4 = 0{,}75 \\ \lim_{E[NB^t_i]\to \infty} E[D^t_{i,NH^*}] &= 3/2 = 1{,}5 \end{aligned} \tag{5.9}$$

Por tanto, puesto que el límite superior de $E[D^t_{i,NH^*}]$ es 1,5, se demuestra que el nuevo esquema de comunicación diseñado será más eficiente que el uso de los propios mensajes de AODV siempre y cuando el número esperado de destinos que emplean a NH^* sea de dos o más. Estos resultados teóricos serán posteriormente confirmados en la Sección 5.4.2.

5.4. Resultados experimentales

En esta sección se presenta en primer lugar una descripción del entorno experimental utilizado para testar el esquema de detección desarrollado y, posteriormente, se realizan distintos experimentos para evaluar el sistema de detección bajo distintas condiciones, analizándose los resultados obtenidos.

5.4.1. Descripción del entorno experimental

En esta ocasión se han simulado distintos despliegues MANET empleando el popular simulador OMNeT++ (*Objective Modular Network Test-bed in C++*) [144]. Para simular los nodos *sinkhole* maliciosos se ha empleado NETA (*NETwork Attacks*) [145], un *framework* construido sobre OMNeT++ que permite la simulación de distintos ataques de red de forma sencilla [146] y que será explicado detalladamente en el Capítulo 7. Los parámetros comunes empleados en todos los escenarios considerados se explican a continuación.

El área de simulación queda restringida a un cuadrado de 1.000 x 1.000 metros, seleccionándose IEEE 802.11g y AODV como protocolos de capa MAC y de red respectivamente, empleándose el mecanismo RTS/CTS para las transmisiones (véanse Secciones 2.4.1 y 2.4.2).

El número total de nodos es 25, 24 de ellos legítimos y 1 actuando como *sinkhole* malicioso. El ataque *sinkhole* se lleva a cabo durante todo el tiempo de simulación, siendo la tasa de ataque del 100 %. Es decir, el nodo malicioso siempre responde con un falso mensaje RREP a cualquier mensaje RREQ recibido, incluso si el nodo malicioso no posee una ruta válida hacia el destino solicitado. No obstante ello, también se han realizado experimentos en la Sección 5.4.4 para distintas tasas de ataque.

Para realizar el ataque el nodo *sinkhole* genera un número aleatorio entre 20 y 30 unidades, siguiendo una distribución uniforme discreta. Dicho valor generado se añade al número de secuencia observado en el mensaje RREQ, dando como resultado un número de secuencia artificialmente incrementado en el mensaje RREP falsificado. Nótese que la mayoría de los trabajos en la literatura simplemente fijan el falso número de secuencia al valor máximo posible ($2^{31} - 1 = 4294967295$), mientras que otros trabajos añaden al número de secuencia valores relativamente altos; por ejemplo, generados por una distribución uniforme entre 15 y 200 unidades. En este trabajo consideramos un comportamiento *sinkhole* más realista e "inteligente", en el que el atacante intenta dificultar el proceso de detección mientras que se asegura ser seleccionado como siguiente salto en la ruta.

El modelo de propagación considerado inicialmente es *Two Ray Ground* [140], estableciéndose el radio de cobertura de los nodos a 250 metros. En nuestra experimentación también evaluaremos la efectividad del sistema cuando otros modelos son utilizados.

Para modelar la movilidad de los nodos podría emplearse RWP [141], tal y como se hizo en el Capítulo 4. Sin embargo, este modelo sufre de algunas limitaciones a ser consideradas, como la gran densidad de nodos en el área central o la movilidad decreciente [147], [148], lo que podría dar lugar a escenarios poco realistas. Por esta razón aquí se empleará un modelo más sofisticado, RPGM (*Reference Point Group Mobility*) [149]. En este, los nodos pertenecen a un grupo determinado y se distribuyen de forma aleatoria alrededor del centro lógico de dicho grupo, denominado *líder del grupo*. Este líder define un vector de movimiento aleatorio que representa el desplazamiento general del grupo. Por su parte, cada nodo del grupo tiene su propio movimiento que se ve afectado en cierta medida por el vector de movilidad grupal, es decir, los miembros usan su propio modelo de movilidad que se añade al punto de referencia, encargado de guiarlos en la dirección del grupo. Cada nodo se desplazará entonces a algún punto en la vecindad de dicho punto de referencia. El modelo RPGM se usa para simular multitud de aplicaciones, como soldados pertenecientes a escuadrones militares en entornos tácticos o equipos de rescate cooperando durante situaciones de catástrofe. En este caso se ha considerado el tamaño del grupo de 5 nodos, con una distancia máxima entre cada nodo y su correspondiente centro de 250 metros. Para generar los patrones de movilidad se ha empleado una herramienta ampliamente conocida en este campo, el software *Bonnmotion* [150].

La velocidad mínima de los nodos se fija a 0,5 m/s, mientras que la velocidad máxima varía entre 3 y 10 m/s, con un tiempo de pausa de 15 s. Estas velocidades máximas (3 - 10 m/s ≡ 10,8 - 38 km/h) cubren el rango de velocidades desde aquella que alcanzaría un peatón, hasta una velocidad moderada que podría llevar un vehículo.

Cada nodo legítimo transmite su propio flujo de aplicación, que simula llamadas de voz punto-a-punto en tiempo real. Por cada flujo se obtiene distinto número de llamadas modelando el tiempo de pausa entre dos llamadas, IAT (*Inter-Arrival Time*), y la duración de cada llamada, CHT (*Call Holding Time*). IAT se describe mediante una distribución exponencial con λ igual a 7,5 segundos y CHT mediante una distribución *lognormal* de media μ fija a 3,287 y desviación estándar σ 0,891 [151]. Para cada llamada se elige de forma aleatoria un destino distinto de entre los nodos legítimos, siendo las llamadas tratadas como conexiones CBR, con un tráfico de 4 paquetes por segundo y un *payload* de 512 bytes.

La duración de los intervalos de muestreo W empleados para la recopilación de los parámetros se estudiará en una subsección aparte más adelante.

5.4.2. Resultados de detección

A continuación evaluamos la efectividad global del sistema de detección propuesto mediante la simulación de diversas pruebas. Al igual que en el capítulo previo, las capacidades de detección del sistema se miden mediante dos métricas, la tasa de verdaderos positivos, o TPR, y la de falsos positivos, o FPR.

En esta línea, y analizando la influencia de los distintos parámetros, se obtienen varios puntos de operación que conforman el espacio ROC mediante la variación del umbral de detección θ_d en la Ec. (5.5). Es importante destacar que, para capturar el comportamiento estadístico, todos los resultados se han derivado repitiendo cada simulación 50 veces, con distintas semillas.

La Figura 5.6 muestra la curva ROC obtenida para ambas movilidades, 3 y 10 m/s, tras el ajuste de los parámetros W y θ_s, cuyos valores, tras una cierta experimentación que se presentará más adelante, se han fijado a 5 segundos y 20 unidades respectivamente. Como puede observarse en la gráfica, cuanto mayor es el umbral de detección θ_d mejor FPR se obtiene, pero a costa de reducir el valor de TPR. Por otro lado, valores más bajos de dicho umbral resultarán en mejores capacidades de detección en términos de TPR, pero a expensas de aumentar la tasa FPR. Se puede observar también que a mayor velocidad de los nodos, mejores resultados se obtienen. Esto se debe principalmente al hecho de que velocidades superiores de los nodos implican más situaciones en las que los "bordes de contaminación" serán vecinos de nodos legítimos no contaminados que conocen las rutas válidas, *i.e.*, a mayor

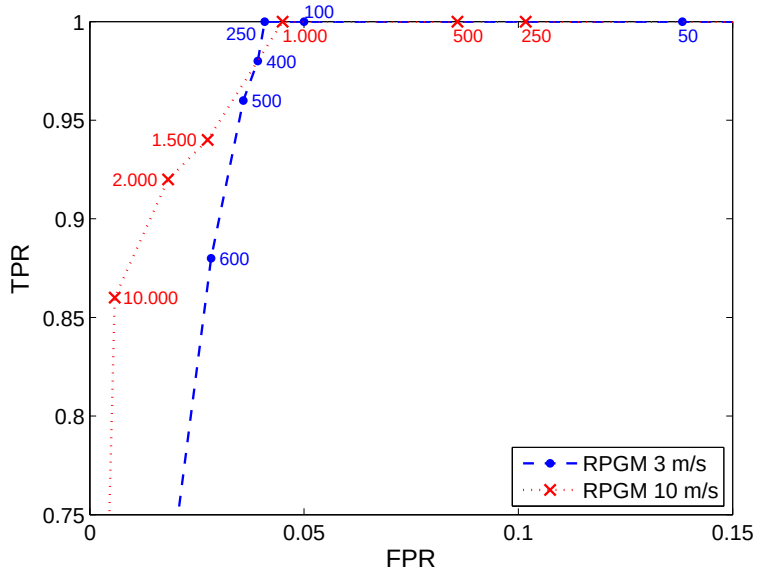


Figura 5.6: Curva ROC variando el umbral de detección θ_d, con $W = 5$ segundos y $\theta_s = 20$ unidades.

movilidad, más posibilidades de obtener información de nodos no contaminados, resultando en más oportunidades de detectar al nodo *sinkhole*.

Así, el punto de operación óptimo del esquema de detección propuesto se puede obtener de forma empírica, siendo este fuertemente dependiente de las condiciones específicas de la red y, en particular, de la movilidad. Para un mejor rendimiento, el umbral θ_d se ha fijado al valor 250 para una movilidad de 3 m/s, y al valor 1.000 para 10 m/s. Dichos valores proporcionan una buena relación entre la tasa de falsos positivos y la de falsos negativos, manteniendo la tasa TPR al 100 % y la tasa FPR por debajo del 5 %.

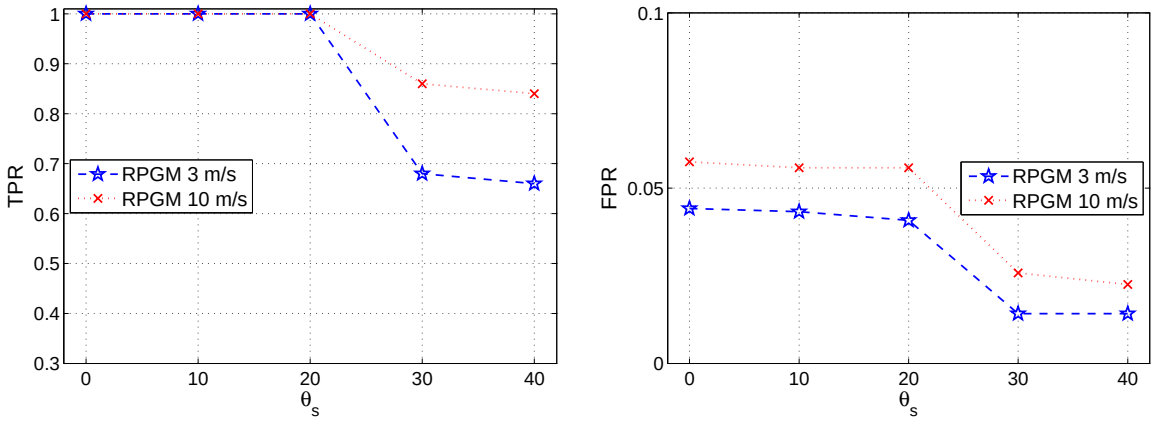
Influencia del umbral de sospecha local, θ_s

Los experimentos previos se han realizado para un valor de θ_s igual a 20 unidades. A continuación discutiremos la influencia real del umbral de sospecha local θ_s en las capacidades de detección de nuestro sistema. Como se explicó en la Sección 5.3, el propósito principal de dicho umbral es ser utilizado en una fase de pre-detección para reducir la sobrecarga introducida por el intercambio de información implicado en el proceso de detección colaborativo subsiguiente. Sin embargo, el umbral θ_s también tiene una influencia importante en las capacidades de detección del sistema. Por ejemplo, la elección de un valor demasiado elevado para θ_s podría implicar que muy pocos nodos sean clasificados como sospechosos y, por tanto, se reduciría tanto la tasa TPR como la FPR. De este modo, hemos evaluado distintos valores para θ_s, desde 0 a 40 unidades. Al igual que antes, se han realizado 50 repeticiones con distintas semillas.

Como se muestra en la Figura 5.7, mientras que el valor de θ_s permanezca por debajo o cercano al valor 20, las capacidades de detección del sistema no se ven negativamente afectadas. Sin embargo, como se previó anteriormente, valores superiores a 20 unidades degradan la efectividad del sistema en términos de TPR (Figura 5.7a). Por tanto, para experimentos posteriores se ha seleccionado $\theta_s = 20$ unidades, al ser este el valor más elevado para el umbral de sospecha local que no deteriora excesivamente las capacidades de detección, al tiempo que introduce una menor sobrecarga de tráfico como se probará en experimentos posteriores.

Influencia del tamaño del intervalo de muestreo, W

También se ha estudiado experimentalmente cómo afecta el tamaño del intervalo de muestreo para la recopilación de los parámetros en la eficiencia de la detección, eligiéndose el valor óptimo a partir de dicho análisis. Se han realizado pruebas para valores de W de 1, 5, 10, 30 y 60 segundos, mientras que el umbral de sospecha

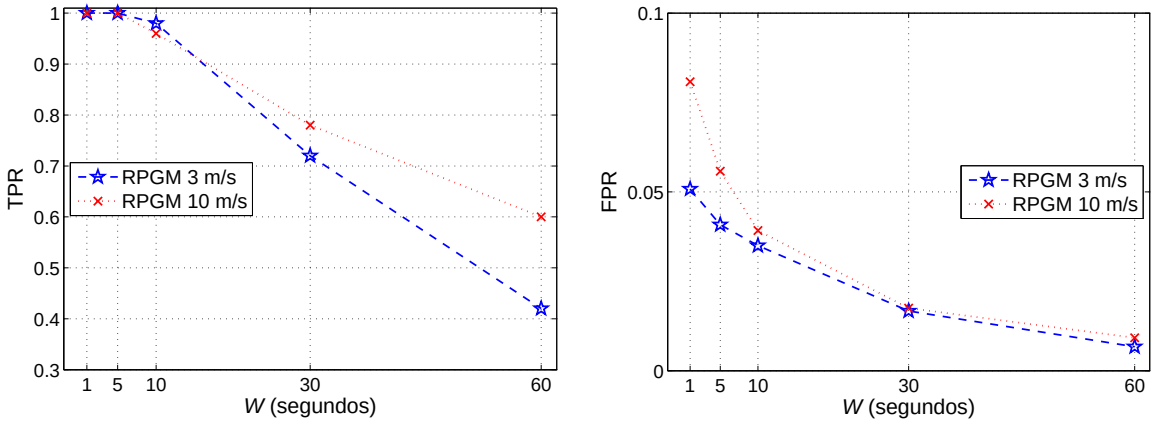


(a) TPR (b) FPR

Figura 5.7: Dependencia de TPR (a) y FPR (b) con el umbral de sospecha local θ_s, con $W = 5$ segundos y $\theta_d = 250/1.000$ unidades respectivamente.

local θ_s se ha fijado a 20 unidades. La Figura 5.8 muestra gráficamente tanto la tasa TPR como la FPR para los distintos intervalos de muestreo indicados.

De forma similar a como ocurre con el umbral de sospecha local θ_s, valores elevados de W permitirán reducir el *overhead* introducido por el esquema de detección, al producirse la transmisión de la información y el consecuente cálculo de la heurística menos frecuentemente. Sin embargo, el tamaño del intervalo no puede crecer de forma indefinida pues se produce un empeoramiento en los resultados de detección, al reducirse considerablemente el valor de TPR.



(a) TPR (b) FPR

Figura 5.8: Dependencia de TPR (a) y FPR (b) con el tamaño de los intervalos de muestreo W, con $\theta_s = 5$ unidades y $\theta_d = 250/1.000$ unidades respectivamente.

De acuerdo con los resultados obtenidos, el tamaño de intervalo seleccionado como punto óptimo de operación es de 5 segundos, considerando que dicho valor proporciona unos buenos resultados de detección al tiempo que reduce el ancho de banda consumido por el proceso de detección. Tamaños de intervalo superiores ya comienzan a degradar las capacidades de detección del sistema, en particular la tasa TPR.

Influencia del modelo de propagación

Más allá de los buenos resultados obtenidos por nuestro esquema, aquí se presenta otra serie de experimentos destinados a demostrar el funcionamiento de nuestra aproximación de detección bajo un modelo de propagación más realista que el modelo *Two Ray Ground* considerado hasta ahora. Para ello se ha seleccionado el modelo de propagación *Nakagami* [152], un modelo genérico probabilístico donde la potencia de recepción sigue una distribución *gamma*, reflejando de forma precisa diferentes circunstancias del entorno, como desvanecimientos ocasionados por multi-trayectos de la señal transmitida (*multipath fading*). El modelo emplea el parámetro *m* para especificar la intensidad del efecto de desvanecimiento, cubriendo un amplio rango de fluctuaciones de intensidad. El sistema de detección propuesto es evaluado así incluyendo dicho modelo *Nakagami*, fijándose el valor de *m* a 4. En la Figura 5.9 se pueden observar los resultados de detección obtenidos para los escenarios descritos, asumiendo una movilidad de los nodos de 3 y 10 m/s. Aunque en términos de TPR los resultados empeoran ligeramente con respecto a los obtenidos considerando *Two*

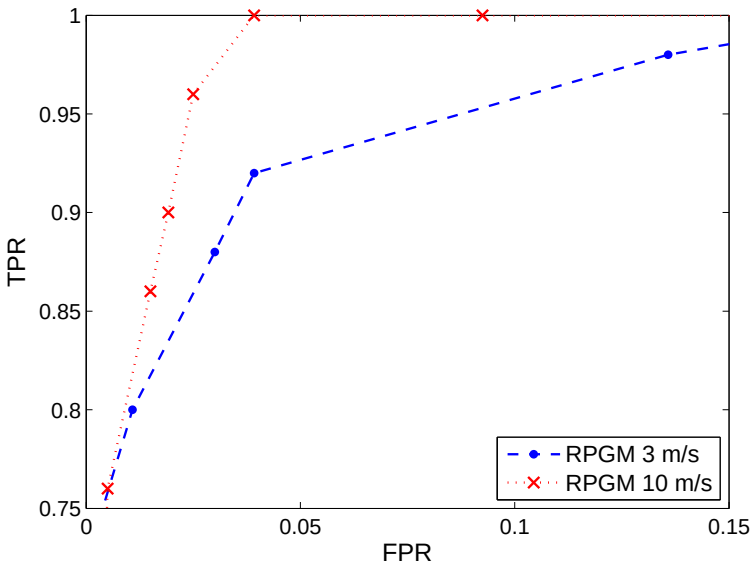


Figura 5.9: Curva ROC de nuestro sistema de detección de *sinkhole* considerando el modelo de propagación *Nakagami*.

Ray Ground (cuyos resultados de detección se proporcionaron en la Figura 5.6), cabe destacar que todavía se consiguen unas excelentes capacidades, obteniéndose una tasa TPR superior al 90 % y manteniendo el valor FPR por debajo del 5 %.

Es importante indicar que, en entornos con tan alto grado de pérdidas ocasionadas por problemas asociados a la propagación de la señal (reflexión, desvanecimiento, dispersión, etc.), el uso de protocolos reactivos como AODV no es lo más apropiado. En estos casos se recomienda la utilización de protocolos proactivos que empleen algún tipo de métrica que considere el estado del enlace, como puede ser OLSR. Sin embargo, podemos comprobar que, incluso bajo estas circunstancias, el detector propuesto logra notables resultados en términos de TPR y FPR en nuestro escenario.

Overhead de comunicación

Otra serie de experimentos se ha realizado para estudiar al sobrecarga de tráfico introducida por los mensajes generados por el sistema de detección para adquirir los números de secuencia de los vecinos durante la fase de detección colaborativa. Se estudian ambas aproximaciones de comunicación, *i.e.*, aquella que emplea los propios mensajes del protocolo AODV y la que emplea los nuevos mensajes propuestos.

Para comparar dicha sobrecarga introducida, en vez de hablar en términos absolutos sobre paquetes o bytes transmitidos se calculará una métrica más intuitiva: el ancho de banda consumido por la transmisión de los mensajes, tanto en paquetes por segundo como en bytes por segundo. Esta métrica se obtiene como la relación entre el número de paquetes/bytes transmitidos y el tiempo total de simulación:

$$BW_{pkt/s} = \frac{\text{\#Paquetes enviados a causa de la detección}}{\text{Tiempo total de simulación}} \text{ (paquetes/s)} \qquad (5.10)$$

$$BW_{B/s} = \frac{\text{\#Bytes transmitidos a causa de la detección}}{\text{Tiempo total de simulación}} \text{ (bytes/s)} \qquad (5.11)$$

La Tabla 5.1 y la Tabla 5.2 muestran los resultados del ancho de banda consumido por ambas aproximaciones de comunicación, para las dos situaciones de movilidad propuestas, 3 y 10 m/s, respectivamente.

Como se postuló en la Sección 5.3.2, el uso de los nuevos mensajes definidos siempre reduce el ancho de banda en términos de paquetes/s con respecto al uso de los mensajes de AODV. Sin embargo, el mayor tamaño de los nuevos mensajes definidos implica un *overhead* ligeramente mayor en términos de bytes transmitidos.

Tabla 5.1: Ancho de banda para ambas aproximaciones de comunicación, AODV y nuevos mensajes, para una velocidad máxima de los nodos de 3 m/s.

Intervalo muestreo (s)	θ_s	$\mathbf{BW}^{\mathbf{aodv}}_{\mathbf{pkt/s}}$	$\mathbf{BW}^{\mathbf{new}}_{\mathbf{pkt/s}}$	$\mathbf{BW}^{\mathbf{aodv}}_{\mathbf{B/s}}$	$\mathbf{BW}^{\mathbf{new}}_{\mathbf{B/s}}$
$W = 1$	0	41,56	34,83	843,03	904,78
	10	15,12	12,13	306,67	322,81
	20	13,09	10,56	265,49	280,10
	30	3,08	2,44	62,56	65,36
	40	2,33	1,83	47,44	49,35
$W = 5$	0	36,40	31,21	738,37	801,01
	10	13,72	11,10	278,24	293,95
	20	11,99	9,74	243,22	257,36
	30	3,06	2,42	62,08	64,83
	40	2,32	1,81	47,00	48,82
$W = 10$	0	24,39	21,04	494,66	538,35
	10	11,08	9,07	224,71	238,68
	20	9,76	7,97	197,93	209,99
	30	2,69	2,10	54,54	56,69
	40	1,99	1,55	40,33	41,92
$W = 30$	0	11,32	9,98	229,47	252,35
	10	6,27	5,26	127,08	136,57
	20	5,65	4,74	114,65	123,14
	30	2,23	1,80	45,23	47,76
	40	1,61	1,27	32,61	34,02
$W = 60$	0	5,84	5,24	118,43	131,41
	10	3,65	3,15	74,19	80,69
	20	3,34	2,86	67,78	73,52
	30	1,83	1,51	37,08	39,60
	40	1,22	0,98	24,77	26,05

Esto se debe a que, a lo largo de la simulación, el número de destinos acerca de los que se solicita el número de secuencia es menor a 1,5 en media, por lo que la segunda aproximación no supone una reducción del ancho de banda efectiva en bytes/s.

Como era de esperar también, cuanto mayor sea el tamaño del intervalo W y el umbral de sospecha local θ_s, menor será el *overhead* introducido. Sin embargo, las capacidades de detección deberían ser, por regla general, más importantes que el ancho de banda consumido y, en consecuencia, los valores óptimos de ambos parámetros se han obtenido en secciones previas, siendo fijados a 5 segundos y 20

Tabla 5.2: Ancho de banda para ambas aproximaciones de comunicación, AODV y nuevos mensajes, para una velocidad máxima de los nodos de 10 m/s.

Intervalo muestreo (s)	θ_s	$\mathbf{BW}^{\mathbf{aodv}}_{\mathbf{pkt/s}}$	$\mathbf{BW}^{\mathbf{new}}_{\mathbf{pkt/s}}$	$\mathbf{BW}^{\mathbf{aodv}}_{\mathbf{B/s}}$	$\mathbf{BW}^{\mathbf{new}}_{\mathbf{B/s}}$
$W = 1$	0	84,84	66,86	1721,91	1795,30
	10	34,43	26,13	698,67	716,73
	20	31,17	23,58	632,55	648,00
	30	9,52	7,01	193,29	195,63
	40	8,13	5,94	165,07	166,46
$W = 5$	0	67,97	55,86	1379,20	1466,16
	10	29,61	22,99	600,78	622,54
	20	27,03	20,87	548,41	566,89
	30	9,27	6,86	188,19	190,81
	40	7,71	5,68	156,53	158,39
$W = 10$	0	44,48	37,18	902,33	967,05
	10	22,38	17,65	454,10	473,89
	20	20,58	16,15	417,51	434,81
	30	7,66	5,69	155,41	157,88
	40	6,41	4,73	130,19	131,92
$W = 30$	0	18,12	15,66	367,49	400,09
	10	11,13	9,19	225,75	240,65
	20	10,46	8,58	212,21	225,45
	30	6,04	4,73	122,53	127,44
	40	4,83	3,72	98,02	101,13
$W = 60$	0	8,38	7,64	177,51	194,21
	10	6,04	5,08	122,62	131,82
	20	5,71	4,77	115,85	124,11
	30	3,97	3,22	80,59	85,19
	40	3,14	2,50	63,81	66,80

unidades respectivamente. Estos valores permiten obtener excelentes resultados en términos de TPR y FPR, manteniendo la sobrecarga de tráfico en niveles razonables.

5.4.3. Discusión de los resultados de detección

Hemos llevado a cabo también una comparativa entre los resultados obtenidos por nuestra aproximación colaborativa y los exhibidos por otras aproximaciones pro-

puestas en la literatura que calculan una heurística local considerando los números de secuencia enviados y recibidos en el nodo encargado de realizar el proceso de detección, como las introducidas en [112] y [114]. Estos esquemas equivaldrían a la fase de pre-detección de nuestro esquema, dado que calculan localmente la diferencia entre el número de secuencia enviado en el mensaje RREQ y el recibido en el mensaje RREP.

La Figura 5.10 presenta la curva ROC obtenida para las dos condiciones de movilidad (3 y 10 m/s) para ambas aproximaciones, colaborativa y local. Como se puede comprobar, la inclusión de información procedente de los vecinos en el proceso de detección permite obtener unos resultados claramente mejores que los logrados por la aproximación local propuesta en otros esquemas previos.

Además de ello, se presenta otro conjunto de experimentos destinado a comparar el rendimiento del detector propuesto frente a otras aproximaciones locales bajo distintas severidades del ataque *sinkhole*, para una velocidad máxima de 10 m/s. En la Figura 5.11 se proporcionan las curvas ROC para dos tasas de ataque distintas (80 % y 50 %), así como para la situación extrema en la que el nodo *sinkhole* únicamente envía mensajes RREP falsificados cuando es preguntado por una ruta en particular, es decir, el nodo *sinkhole* contamina una única ruta.

Se puede observar que el detector propuesto es capaz de alcanzar un valor TPR de casi el 90 % con una tasa FPR del 10 % en la situación extrema en la que una única ruta es contaminada. Además, como se explicó en la Sección 5.2.2, los esquemas que siguen una aproximación de detección local presentan usualmente una elevada tasa FPR, al ser los nodos legítimos contaminados propensos a ser clasificados como

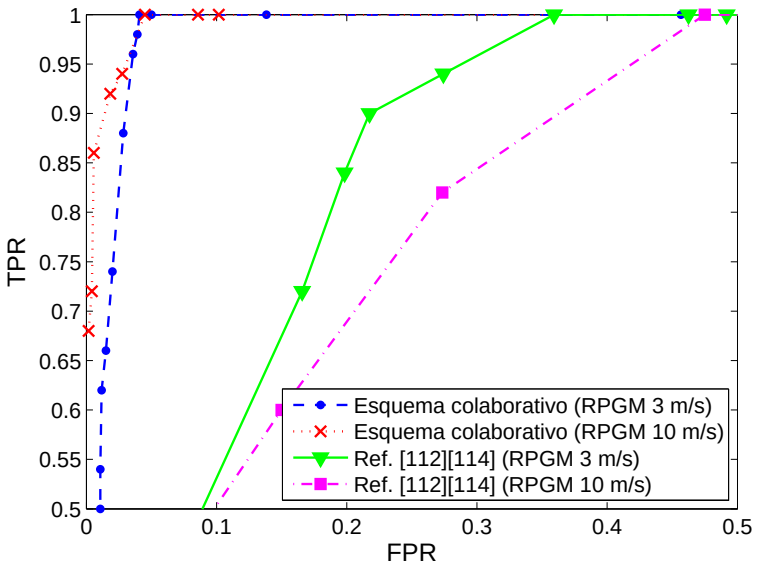


Figura 5.10: Curvas ROC para diferentes esquemas de detección de ataques *sinkhole*.

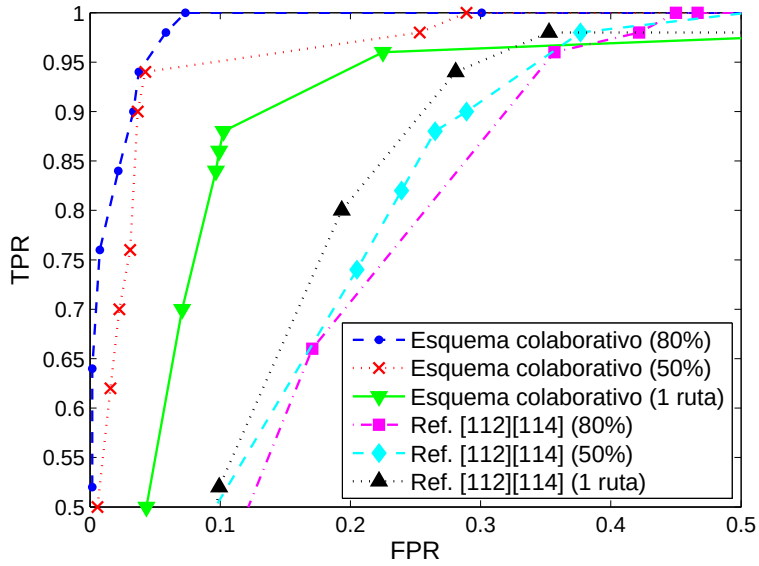


Figura 5.11: Curvas ROC para diferentes esquemas de detección variando la severidad del ataque, para una movilidad de 10 m/s.

maliciosos. Así, si la severidad del ataque es menor, también será menor la zona de contaminación y, en consecuencia, el valor de FPR puede ser mejorado, dado que será inferior el número de nodos legítimos contaminados clasificados erróneamente como nodos *sinkhole*.

En resumen, podemos concluir que el sistema de detección de ataques *sinkhole* propuesto obtiene excelentes resultados al respecto de las dos métricas consideradas, TPR y FPR. Seleccionando el punto de operación óptimo se puede lograr una tasa TPR del 100 % manteniendo el valor FPR por debajo del 5 %, por lo que dichos resultados confirman las bondades del sistema. Además, el tráfico introducido por la comunicación requerida para la fase de detección colaborativa es aceptable para este tipo de entornos ad hoc.

5.4.4. Consideraciones acerca del esquema propuesto

En este punto es necesario introducir una breve discusión acerca de algunos aspectos que deberían ser tenidos en cuenta para el esquema propuesto, principalmente referidos al entrenamiento del sistema y a la posible aparición de nodos maliciosos trabajando en confabulación.

Consideraciones sobre el entrenamiento

A pesar de que el sistema de detección propuesto presenta un alto rendimiento tanto en términos de detección como de *overhead* introducido, no se resuelve un problema común presente en la mayoría de los esquemas de detección: la necesidad de ajustar los parámetros del sistema para obtener el punto de operación óptimo. Por tanto, sigue siendo necesario llevar a cabo un entrenamiento del sistema en función del escenario específico o de las condiciones particulares de la red.

De este modo, sería muy interesante diseñar un método efectivo que permita determinar de forma automática y dinámica, en base a alguna métrica específica, los valores óptimos de los umbrales θ_s y θ_d. Por ejemplo, algunos trabajos como [105] o [106] adaptan el umbral de detección calculando la diferencia media entre los números de secuencia enviados y recibidos en los intervalos temporales previos. Sin embargo, se podrían aplicar otros métodos para la adaptación dinámica de los umbrales, *p.ej.*, mediante la obtención de algún tipo de media ponderada de los valores umbrales previos, calculando los umbrales en función de la movilidad de los nodos, etc. Este diseño de métodos para la adaptación dinámica de los umbrales constituye un interesante tema de estudio futuro.

Consideraciones sobre confabulación en el ataque *sinkhole*

Así mismo, se deben realizar algunas aclaraciones a la hora de considerar la posibilidad de que varios nodos maliciosos actúen confabulados. Esta confabulación puede verse motivada por distintas razones: que el ataque tenga un mayor impacto en el rendimiento de la red, intentar evadir el proceso de detección, etc.

Para el ataque *sinkhole* la primera motivación no parece tener mucho sentido, puesto que la existencia de varios nodos *sinkhole* daría como resultado que estos compitieran entre sí para atraer el tráfico circundante, reduciéndose entonces el impacto de cada nodo malicioso en la red. En cualquier caso, estos nodos maliciosos en confabulación podrían ser considerados como nodos *sinkhole* individuales, pues continúan teniendo que contaminar las rutas para tener éxito en su ataque.

Por tanto, el esquema de confabulación más típico es aquel en el que un único nodo actúa como *sinkhole* real, mientras que el resto de nodos maliciosos (generalmente vecinos de este) intentan evitar la detección del primero. Sin embargo, si se desea asegurar que el nodo *sinkhole* continúe atrayendo tráfico, no es adecuado que los nodos confabulados envíen por su parte información falsa para dificultar la detección, bien sea información que intente incriminar a otros nodos, bien sea información que altere las rutas. Así, los nodos confabulados son generalmente responsables de responder a mensajes de solicitud de confirmación (como los mensajes FRq o CREQ propuestos por distintas aproximaciones de detección vistas en el Capítulo 3), logrando que el

sistema de detección considere que el nodo *sinkhole* se está comportando de forma apropiada.

Puesto que nuestra aproximación de detección se basa en una heurística distribuida obtenida a partir del mínimo número de secuencia almacenado entre los nodos vecinos, puede considerarse inherentemente resistente a la confabulación. Ello siempre y cuando exista al menos un nodo legítimo no contaminado que sea vecino de un "borde de contaminación" y conozca la ruta valida, el número de secuencia mínimo no podrá ser falsificado de forma efectiva, por lo que no será posible para los nodos maliciosos confabularse para intentar evadir la detección. El único modelo de confabulación posible contra nuestro sistema sería que los nodos maliciosos enviasen falsos mensajes de alerta incriminando a nodos legítimos. Este tipo de incriminación debe distinguirse de aquella que podría ser ocasionada por el envío de información de encaminamiento falsa y que, como se ha visto, carece de mucho sentido.

Una extensión de este trabajo en la que se considere este último tipo de confabulación implicaría la combinación y evaluación de nuestra técnica con algún tipo de solución basada en reputación, en la que se trate de forma específica con problemas de falsas acusaciones o votos contra la detección/revocación de nodos maliciosos.

5.5. Conclusiones del capítulo

En el presente capítulo se aborda el problema de detectar ataques *sinkhole* en redes MANET. Para ello se propone una nueva metodología de detección, en la que se emplean los números de secuencia en las tablas de *routing* como parámetros base. El esquema desarrollado se basa en la hipótesis de que existen "zonas de contaminación" y "nodos frontera", *i.e.*, nodos legítimos bajo la influencia del ataque *sinkhole* pero que a su vez son vecinos de nodos legítimos no afectados por el ataque. En base a esto se plantea una heurística sencilla que se calcula como la diferencia entre el número de secuencia en estos "bordes de contaminación" y el mínimo obtenido de entre sus nodos vecinos para las mismas rutas. Esta heurística permite estimar incoherencias en las rutas, detectando así comportamientos *sinkhole* maliciosos en los que los nodos envían mensajes RREP falsificados como respuesta a solicitudes RREQ legítimas, con la intención de atraer el tráfico circundante hacia los nodos *sinkhole*.

Se ha verificado mediante la simulación de diversos despliegues MANET el correcto funcionamiento del sistema bajo diversas situaciones y circunstancias. Los resultados obtenidos prueban las excelentes capacidades de detección del sistema propuesto, capaz de alcanzar una tasa TPR del 100 % con un valor FPR potencialmente inferior al 5 %, lo que mejora sensiblemente los resultados obtenidos por otros esquemas propuestos en la literatura.

Además, como se muestra también en la sección experimental, el ancho de banda introducido por el intercambio de información necesario para calcular la heurística en la fase de detección colaborativa es asumible en este tipo de entornos MANET, permitiendo una mejora sustancial en las capacidades de detección del sistema.

Publicaciones relacionadas

Para finalizar el tema se presentan las publicaciones directamente relacionadas con la detección de ataques *sinkhole* en redes MANET. Estas son:

- *Enviada* → **L. Sánchez-Casado**, G. Maciá-Fernández, P. García-Teodoro y N. Aschenbruck. "Identification of Contamination Zones for Sinkhole Detection in MANETs". *Journal of Network and Computer Applications (Elsevier)*, 20 páginas, 2014.
- **L. Sánchez-Casado**, G. Maciá-Fernández, P. García-Teodoro y N. Aschenbruck. "A Novel Collaborative Approach for Sinkhole Detection in MANETs". *Workshop on Security in Ad Hoc Networks* (SecAN), pp. 42-55, 2014.
- **L. Sánchez-Casado**, G. Maciá-Fernández y P. García-Teodoro. "Indicadores de Ataques Sinkhole en MANETs". *XI Jornadas de Ingeniería Telemática* (JITEL), pp. 475-480, 2013.

Parte III

Integración de Soluciones de Seguridad

Capítulo 6

Protocolo para la notificación y alerta de eventos de seguridad en MANET

Para conseguir la protección integral de una red de comunicación frente a las distintas amenazas de seguridad existentes, es necesario desplegar mecanismos que implementen las tradicionales líneas defensivas ya conocidas y comentadas al principio de esta memoria. Así, una vez abordado el problema de la detección de ataques en redes MANET en capítulos anteriores, se hace oportuna la adopción de medidas de respuesta concretas frente a estas amenazas, teniendo que habilitarse por tanto algún procedimiento para la notificación al resto de la red acerca de la existencia del evento intrusivo previamente detectado.

Con este fin, el presente capítulo aborda el estudio y desarrollo de un protocolo de notificación y alerta de eventos de seguridad cuyo fin principal es servir de interfaz entre los módulos de detección y respuesta. Ideado específicamente para redes ad hoc, su uso posibilita poner en conocimiento de los elementos constitutivos del entorno monitorizado la ocurrencia de un cierto comportamiento malicioso detectado. Este conocimiento será clave para la ejecución posterior de los mecanismos de respuesta oportunos. Un diseño inteligente del protocolo permitirá su empleo para un conjunto más amplio de actuaciones, como por ejemplo la intercomunicación de módulos de detección distribuidos y colaborativos como los apuntados previamente para el ataque *sinkhole*.

El resto del capítulo se estructura de la siguiente forma. La Sección 6.1 motiva la necesidad de desarrollar protocolos para la integración de soluciones de seguridad. En la Sección 6.2 se discuten algunas propuestas en la línea aquí planteada existentes

en la bibliografía especializada. Tras ello, y habida cuenta de la baja idoneidad de las mismas para entornos ad hoc, en la Sección 6.3 se presenta nuestra propuesta concreta y se discute su uso con varios fines relacionados. Seguidamente, la Sección 6.4 se dedica a un breve análisis de prestaciones del protocolo introducido desde el punto de vista del impacto que tiene su uso sobre las comunicaciones del entorno. Finalmente, en la Sección 6.5 se concluye con los aspectos más relevantes de la propuesta realizada y se apuntan brevemente algunas actuaciones de futuro.

6.1. Motivación

Como se vio en el Capítulo 1, para conseguir una seguridad más robusta en redes de comunicaciones, tradicionalmente se han propuesto y desplegado tres líneas defensivas claramente diferenciadas: (*i*) prevención, mediante la adopción de medidas que eviten la ocurrencia de actuaciones maliciosas en el entorno, (*ii*) detección, necesaria ante la imposibilidad de garantizar de forma absoluta la no aparición de dichos eventos maliciosos en el entorno monitorizado, y (*iii*) respuesta, a través de la aplicación de las medidas oportunas necesarias frente a los eventos previamente detectados. Además, es altamente recomendable la realimentación del proceso con el fin de obtener sistemas de seguridad integrales, efectivos y robustos (véase Figura 1.3).

En la literatura se encuentran desarrollados numerosos esquemas de prevención, detección y respuesta. Sin embargo, a pesar de que en teoría dichos módulos defensivos deben interoperar entre sí, por lo general son planteados y adoptados como soluciones independientes. De esta manera, las soluciones de seguridad habitualmente disponibles son parciales por cuanto que solo se centran en uno de los tres aspectos citados y, sobre todo, porque se desarrollan obviando la necesidad de disponer de procedimientos efectivos de comunicación entre los distintos módulos. Este hecho es especialmente crítico en redes red ad hoc [14], en las que se evidencia una alta carencia de propuestas específicas orientadas a la interoperación de estos módulos o soluciones de seguridad.

Frente a esta falta de soluciones de interoperación, en este trabajo de tesis se ha diseñado y desarrollado un protocolo específico para la notificación y alerta de eventos de seguridad en redes ad hoc. Además de servir para la interoperación de los módulos de seguridad, la propuesta planteada también es susceptible de ser usada para la distribución de información en procesos de detección y/o respuesta colaborativos, viniendo a cubrir una carencia manifiesta en el campo objeto de estudio.

En este ámbito, y con el fin de proporcionar un intercambio seguro de esta información de naturaleza sensible entre distintas entidades de detección de intrusiones,

el grupo IDWG (*Intrusion Detection Working Group*) de la IETF especifica en el RFC 4766 ("Intrusion Detection Message Exchange Requirements") una serie de requisitos que deberían cumplirse en relación con el formato de mensajes y con el protocolo de comunicación empleado entre las entidades de detección para lograr un intercambio seguro de información, garantizando la separación entre la semántica utilizada y el mecanismo de comunicación empleado.

En particular, se proponen ocho requisitos para el protocolo de comunicación: (*i*) transmisión confiable, (*ii*) interacción con *firewalls*, (*iii*) autenticación mutua, (*iv*) confidencialidad del mensaje, (*v*) integridad del mensaje, (*vi*) autenticación independiente en cada fuente, (*vii*) protección frente a denegación de servicio, y (*viii*) protección frente a duplicado de mensajes.

Sin embargo, dichos requisitos consideran implícitamente que el despliegue de los sistemas de detección a comunicarse se realizará en redes cableadas o con infraestructura, por lo que algunos de los requisitos no son realistas en otro tipo de entornos, especialmente en redes ad hoc, en las que, debido a su propia naturaleza, el cumplimiento de los citados requisitos puede relajarse en cierto sentido, siendo algunos difícilmente alcanzables. Por ejemplo, no suele ser posible garantizar la confiabilidad de la transmisión, al emplearse generalmente protocolos de transporte no orientados a conexión, como UDP (*User Datagram Protocol*). Esto se debe al hecho de que el empleo del protocolo TCP es poco apropiado en redes MANET, donde la existencia de numerosas pérdidas ocasionadas por colisiones o situaciones de movilidad puede provocar comportamientos "anómalos" en la ventana de congestión, dando lugar a rendimientos poco deseables del protocolo [153], [154], [155]. Así mismo, en este tipo de entornos también es complejo asegurar la protección frente a ataques DoS, pues un simple ataque de *jamming* podría interrumpir el funcionamiento de la red. Por otro lado, otros requisitos son generalmente innecesarios, como los relacionados con la interacción con *firewalls*, al ser el despliegue de estos escaso en entornos ad hoc.

En cualquier caso, para el diseño y desarrollo de nuestra propuesta asumiremos la existencia de sistemas de gestión de confianza subyacentes, *p.ej.*, basados en soluciones de infraestructura de clave pública o PKI, en mecanismos de compartición de claves secretas o en cualquier otra metodología posible. De este modo, la mayoría de los requisitos indicados todavía son satisfechos (autenticación, integridad, confidencialidad, etc.). Esta asunción, por otra parte, es usualmente aceptada por la mayoría de soluciones de seguridad donde la "seguridad básica" no es el objetivo principal perseguido.

En este contexto de comunicación entre módulos defensivos, la propuesta de un protocolo de comunicación específicamente diseñado para notificar eventos de seguridad en redes MANET queda justificada. De este modo, aquí estudiamos y desarrollamos un nuevo protocolo con dicho propósito, que pueda ser empleado

como una interfaz de comunicación entre los módulos de detección y respuesta, así como para la distribución de información de seguridad entre módulos en procesos de detección y/o respuesta colaborativos.

6.2. Trabajos relacionados

En la literatura existen diversos esquemas que permiten el intercambio de información entre distintas entidades, siendo interesante distinguir, por un lado, las propuestas de formatos de mensajes y, por otro, las definiciones de los propios protocolos empleados para la transmisión de dicha información. Así, es posible encontrar diversas soluciones genéricas desarrolladas para permitir dicho intercambio, pudiendo mencionarse algunos protocolos tales como XMPP (*eXtensible Messaging and Presence Protocol*) [156] [157] [158], un protocolo abierto y extensible ideado originalmente para mensajería instantánea que establece una plataforma para el intercambio de información XML (*eXtensible Markup Language*). Sin embargo, XMPP genera una gran sobrecarga debido al envío de datos de presencia, lo que resulta poco escalable, además de funcionar sobre TCP, lo que se ha visto que no es especialmente apropiado en entornos MANET.

Otra posibilidad para el manejo de información relacionada con una red es el servicio *syslog* [159]. Desarrollado para trazar los eventos de un sistema, *syslog* permite la separación del software que genera los mensajes, del sistema que los almacena y del software que los analiza. Los mensajes *syslog* están etiquetados con un código indicativo del tipo de software que los generó (FTP, *email*, etc.) y un grado de severidad (desde *Emergency*, el más alto, hasta *Debug*, el más bajo). Aunque puede utilizarse para la gestión de eventos de seguridad, la complejidad de *syslog* (derivada de su amplia versatilidad) hace que este estándar no resulte el mejor candidato para el fin que aquí perseguimos. Y ello en especial para redes ad hoc, donde, según lo ya apuntado en la Sección 6.1, interesaría la adopción de soluciones específicas y, en consecuencia, ligeras desde el punto de vista del coste y carga implicados.

Al margen de las propuestas genéricas antes comentadas, existen también otras diversas propuestas específicas de esquemas de notificación de incidentes de seguridad en la literatura. Por un lado, son varios los formatos de mensaje desarrollados para el intercambio de información de seguridad. Uno de los más conocidos y empleados es el formato IDMEF (*Intrusion Detection Message Exchange Format*) [160], definido por la IETF para estandarizar el formato de los datos intercambiados entre las entidades IDS. Independiente del protocolo de comunicación y basado en una especificación XML, el formato proporciona una gran flexibilidad y versatilidad. Sin embargo, estas características implican una mayor complejidad y sobrecarga, siendo estas sus mayores desventajas a la hora de emplearlo en entornos de recursos limitados, como pueden ser las redes MANET, en las que se requieren formatos más simples pero al

mismo tiempo flexibles y versátiles. Adicionalmente a ello, IDMEF está centrado en el manejo de información proporcionada por sistemas IDS, no siendo especialmente adecuado para la gestión de datos relacionados con respuesta a incidentes en un contexto más general [161].

El formato IODEF (*Incident Object Description and Exchange Format*) [162] puede verse como una extensión de IDMEF. También basado en una representación de información de incidentes de seguridad de tipo XML, IODEF no ha sido adoptado de forma masiva debido a los requerimientos en cuanto a las herramientas necesarias para soportarlo.

IRMEF (*Intrusion Response Message Exchange Format*) [163] es una extensión de IDMEF que añade una nueva clase *respuesta* a las previstas por este último, con el fin de definir mensajes de alerta que permitan el inicio de las oportunas medidas de respuesta. Para reducir el ancho de banda involucrado, los autores proponen convertir los mensajes XML a un formato intermedio y encapsularlos en el protocolo SNMP (*Simple Network Management Protocol*) para su transporte.

Otra posibilidad es el formato de mensaje X-ARF (*eXtended Abuse Reporting Format*) [164], que se compone de distintos *contenedores* independientes entre sí, basados en MIME (*Multipurpose Internet Mail Extensions*) para su transporte mediante correo electrónico y que pueden incluir contenidos legibles tanto por humanos, incluyendo información de tipo textual para su comprensión por parte de administradores de sistemas, como por sistemas automáticos, conteniendo información estructurada mediante YAML (*YAML Ain't Markup Language*). Sin embargo, el empleo del *email* como transporte no es muy apropiado en entornos ad hoc.

Sea como fuere, la escasa generalización alcanzada por las propuestas mencionadas hace que las recomendaciones acerca de soluciones de gestión de información de incidentes de seguridad tiendan hacia el uso de ficheros de tipo texto, extendiéndose el empleo de formatos ligeros como CSV (*Comma Separated Value*).

Por otro lado, cabe destacar la existencia de diversos protocolos para la comunicación entre entidades de detección. En concreto, es de reseñar el protocolo IDXP (*Intrusion Detection eXchange Protocol*) [165], desarrollado por la IETF en conjunción con el formato IDMEF. Dicho protocolo está especificado como un "perfil" de la *suite* BEEP (*Blocks Extensible Exchange Protocol*) [166], un protocolo de aplicación genérico para conexiones TCP *unicast* que proporciona diversas capacidades como autenticación o integridad, lo que permite que IDXP satisfaga los requisitos indicados en el RFC 4766. Sin embargo, esta *suite* fue originalmente diseñada teniendo en consideración redes cableadas, por lo que introduce un *overhead* relativamente alto. Además, como ya ha sido indicado, el uso de TCP suele estar desaconsejado en redes MANET, en las que no se pueden garantizar conexiones de larga duración estables [167]. Otra de las principales desventajas de IDXP es que no permite transmisiones *broadcast* o *multicast* autenticadas. Como se verá a lo largo del capítulo,

ambos modos de comunicación deberían estar permitidos para dar soporte a algunos de los posibles usos del protocolo propuesto.

Los autores de IRMEF [163] proponen el uso del protocolo SNMP, en su versión tercera, para la transmisión de los mensajes. Sin embargo, la complejidad del mismo y la necesidad de mantener la información en árboles MIB (*Management Information Base*) hacen que SNMP no sea el mejor candidato para las funcionalidades aquí requeridas. Así mismo, tal y como ocurre en IDXP, SNMP tiene la desventaja adicional de que la transmisión de mensajes *broadcast* o *multicast* tampoco está soportada.

De modo similar, IDIP (*Intrusion Detection and Isolation Protocol*) [168] presenta una infraestructura para la cooperación entre distintos componentes con el fin de detectar y aislar intrusiones o comportamientos anómalos. Dichos componentes pueden ser sistemas IDS, *firewalls*, *routers* o componentes de gestión de red, entre otros. Sin embargo, la arquitectura propuesta es compleja, no siendo apropiada en entornos MANET.

Ninguna de las soluciones citadas resuelve, pues, de modo efectivo la problemática abordada en este trabajo: el establecimiento de procedimientos para la notificación de eventos de seguridad a todas las entidades presentes en redes ad hoc, especialmente a aquellas encargadas de ejecutar las medidas de respuesta oportunas. Por esta razón, seguidamente se describe la propuesta de notificación y alerta adoptada en nuestro caso. Esta, frente a las anteriores, está específicamente diseñada para su uso en entornos ad hoc, de manera que resulte lo menos costosa posible desde el punto de vista de los recursos requeridos. Y ello al tiempo que se le dota de una cierta versatilidad y flexibilidad de uso.

6.3. Definición del protocolo

Como ya se ha comentado anteriormente, no existen reportadas en la literatura soluciones adecuadas para la notificación de eventos de seguridad en redes ad hoc. Ideada tomando como base el protocolo de AODV, y diferenciada de este (véase Sección 5.3.2), la propuesta particular aquí desarrollada presenta las siguientes características principales:

- Versátil, al implementarse sobre la capa de aplicación.
- Rápida y eficiente, definiéndose sobre UDP para reducir retardos y consumo de recursos.
- Flexible, ya que posibilita su uso con diversos fines, contemplándose en la versión actual tres principales: notificación de alertas de seguridad, intercambio

de información de detección y/o respuesta colaborativa y notificación asíncrona de información de seguridad.

- El envío de estos mensajes se prevé en tres variantes: *unicast*, inundación a toda la red y *broadcast* a los vecinos a k saltos de distancia (TTL=k), dependiendo del uso y tipo de mensaje concreto de que se trate.

6.3.1. Funcionalidades y usos

Como se ha comentado desde el principio, se propone un diseño flexible del protocolo a través de la especificación de diversos tipos de mensajes, siendo posible su adopción para diversos objetivos especialmente interesantes en el contexto de la seguridad. Así, más allá de la indudable utilidad de la notificación de alertas, es manifiesto el posible empleo del protocolo de notificación ideado para otros fines.

De este modo, se plantean inicialmente tres posibles usos: (*i*) la difusión de alertas de seguridad ante la detección de ciertos incidentes en el entorno, (*ii*) el intercambio de información de seguridad entre los nodos del entorno para, por ejemplo, posibilitar una detección de eventos maliciosos de forma colaborativa o una respuesta coordinada frente a los mismos, y (*iii*) la notificación de dicha información de seguridad de forma asíncrona. Las distintas funcionalidades se detallan a continuación.

Notificación de alertas

Siendo manifiesta la ausencia de procedimientos efectivos de alertas de seguridad en entornos ad hoc, la primera funcionalidad a desarrollar es un procedimiento para la notificación de alertas de seguridad ante la constatación de ciertos incidentes en el entorno monitorizado. Tomando como base los desarrollos IDS realizados por los autores en [58] y [143], al tiempo que la experiencia en esquemas de respuesta dadas en [169] y [170], se plantea un modelo de comunicación que permitirá, en su caso, el despliegue posterior de potenciales medidas de respuesta orientadas a dar solución a los incidentes reportados.

Así, el propósito de dicha funcionalidad es la difusión o compartición de información importante acerca del evento intrusivo detectado, como por ejemplo un nodo particular clasificado como malicioso, el nodo que ha realizado la detección, o cierta información adicional de interés, como la severidad del ataque o el grado de confianza en el proceso de detección. Ello posibilitará el estudio y adopción final de esquemas de respuesta alternativos.

Intercambio de información de seguridad

Esta funcionalidad, al margen de la evidente similitud con esquemas como IDMEF o *syslog*, surge principalmente de los trabajos [58] y [143], donde se plantean esquemas IDS colaborativos fundamentados en el intercambio de información entre nodos (principalmente vecinos). Estos, frente a los de naturaleza aislada, donde cada nodo implementa su propio IDS a partir de información adquirida exclusivamente de forma local, persiguen la adopción de decisiones de detección más globales y, como tales, más robustas y fiables. La aplicabilidad del citado intercambio comprende también IDS centralizados donde se precisa la adquisición de información de toda la red por parte de un solo nodo central.

En uno u otro caso, centralizado y distribuido, el esquema de intercambio es totalmente análogo: existe un nodo (que implementa un IDS) que solicita información acerca de otro cierto nodo (*p.ej.*, porque el IDS local del solicitante ha disparado una alarma de sospecha para él) a otros nodos de la red (todos, su vecindad, etc.), en respuesta a lo cual se proporciona la información específica solicitada para facilitar la posterior decisión de detección y/o respuesta.

En particular, en este trabajo se ha considerado que esta funcionalidad es empleada para realizar un intercambio de información de seguridad entre el módulo IDS presente en un nodo solicitante y los módulos de detección de los nodos vecinos a este. Dicho intercambio de información se produce como consecuencia de la detección local por parte del IDS solicitante de un evento sospechoso que requiere una investigación más global y colaborativa. Este es el caso particular del esquema de detección *sinkhole* planteado en el Capítulo 5.

Notificación asíncrona de información de seguridad

Por último, una funcionalidad adicional definida en nuestro protocolo es la notificación de información de seguridad de forma asíncrona, en la que no se precisa una solicitud previa para la transmisión de los datos. En este caso, un nodo dado, ante la eventual aparición o medición de una circunstancia anómala relevante, podría determinar que es particularmente útil enviar dicha información a otros nodos de la red para que estos realicen una investigación más detallada.

Si bien esta funcionalidad es muy similar al intercambio de información de seguridad, su implementación difiere ligeramente de la implementación previa. En particular, el modo de distribución de la información a publicar podrá venir definido por las propias necesidades requeridas. En este trabajo hemos considerado que esta será enviada a los módulos IDS de los nodos vecinos aunque, como se ha hecho notar, se podrían especificar modos de transmisión alternativos.

La información exacta proporcionada/intercambiada mediante cada una de las funcionalidades propuestas, especificada a través del formato de los mensajes necesarios, se detallará en la Sección 6.3.3.

6.3.2. Operación del protocolo

Como se ha indicado previamente, el protocolo de notificación y alertas se desarrolla como un protocolo de aplicación sobre de UDP, siendo empleado por todos aquellos nodos de la red que implementan a su vez algún tipo de sistema defensivo (bien sea de prevención, detección o respuesta). De este modo, las entidades a comunicarse serán precisamente dichos módulos de defensa.

Cada una de estas entidades puede actuar como origen de la comunicación así como también de destino de la misma. Por ejemplo, un módulo de detección podría necesitar información, para lo cual enviaría una solicitud que será respondida por los módulos de detección presentes en otros nodos. Con posterioridad, dicho módulo de detección podría enviar un mensaje de notificación de alerta, el cual sería recibido por los módulos de reacción/respuesta para llevar a cabo alguna de las medidas oportunas para la subsanación del incidente de seguridad detectado. Así mismo, estos módulos de respuesta podrían comunicarse con los módulos preventivos a fin de adaptar el funcionamiento de estos últimos a las condiciones particulares del entorno.

Tipos de mensajes

Para dar soporte al correcto funcionamiento de las distintas funcionalidades propuestas para el protocolo se definen cuatro tipos principales de mensajes: (*i*) de notificación de alertas, (*ii*) de solicitud de información de seguridad, (*iii*) de respuesta de información de seguridad, y (*iv*) de notificación asíncrona de información de seguridad.

Los mensajes de *notificación de alerta* se envían ante la potencial detección de un evento intrusivo por parte de un módulo de detección. Dichos mensajes se difundirán por toda la red proporcionando información importante sobre la actividad maliciosa detectada.

Los mensajes de *solicitud* y *respuesta de información* son intercambiados ante la necesidad, por parte de un nodo dado, de incorporar información más global a la que ha recopilado de forma local, con el fin de llevar a cabo procedimientos de detección y/o respuesta cooperativos y, en consecuencia, más fiables. Para no sobrecargar la red, no se define en el protocolo un mensaje para la confirmación de la correcta

recepción de los mensajes de respuesta. Sin embargo, la flexibilidad pretendida por el protocolo permite su inclusión en caso necesario.

Por último, se define un mensaje de *notificación asíncrona de información de seguridad* muy similar al mensaje de respuesta de información previo. De este modo, los módulos defensivos de la red serán capaces de proporcionar, de forma proactiva, información considerada de interés para incrementar la seguridad del sistema.

Distribución de mensajes

Un aspecto importante en el diseño de todo procedimiento de notificación de alertas e intercambio de información es el esquema a emplear para la transmisión o envío de la información correspondiente, pues el objetivo es que los recursos implicados, y con ello el impacto sobre las comunicaciones globales, sean los menores posibles. Distintas posibilidades son contempladas para ello en la bibliografía: inundaciones, encaminamiento selectivo, agrupamiento, publicación/suscripción [171]. En nuestro caso, vamos a considerar las siguientes posibilidades en función de la aplicación y del tipo de mensaje:

- *Broadcast* a toda la red, para la notificación de alertas ante la detección de incidentes. Para ello, los nodos que reciban dichos mensajes deberán, a su vez, retransmitir la eventualidad reportada para su distribución a toda la red. Para limitar la carga, cada mensaje incorporará un identificador único para evitar retransmisiones duplicadas.
- *Broadcast* a los vecinos a k saltos, para los mensajes de solicitud de información y para los mensajes de notificación asíncrona de información de seguridad. Para ello, estos paquetes serán enviados sobre la red con el campo TTL del paquete IP sobre el que se encapsulan a valor k. Hay que destacar que este tipo de transmisión es más eficiente que el anterior, por cuanto que se limita la transmisión de los mensajes a un número de nodos menor. En las funcionalidades propuestas hasta la fecha se ha considerado $k = 1$, aunque la flexibilidad del protocolo permite la elección de otros valores mayores para k en caso de requerirse dicha información.
- *Unicast*, para los mensajes de respuesta hacia el nodo solicitante. También podría considerarse el envío *unicast* en los mensajes de solicitud de datos, dependiendo del deseo del nodo emisor en cuanto a información pretendida y procedencia de la misma.

Requisitos criptográficos

Teniendo en consideración que la información a transmitir por el protocolo será de naturaleza sensible, es importante notar también la necesidad de incorporar mecanismos criptográficos que aseguren el cumplimiento de algunos de los requisitos indicados en la Sección 6.1 por parte del protocolo propuesto.

De este modo, y como ya se apuntó, se asume la existencia de algún mecanismo de gestión de la confianza subyacente que permita asegurar la disponibilidad del material criptográfico siempre que sea necesario, típicamente mediante algún esquema de distribución de claves [172]. Además, es de destacar la conveniencia de incorporar mecanismos para proporcionar seguridad en las transmisiones *broadcast* [61], una de las principales carencias de muchas de las soluciones propuestas en la literatura. Así, la existencia de dichos esquemas posibilitará el cumplimiento de varios de los requisitos propuestos en el RFC 4766, como la integridad y confidencialidad de los mensajes, o la autenticación de las entidades. Sin embargo, el tratamiento de dichos mecanismos, considerados como "básicos", queda fuera del ámbito de este trabajo de tesis.

6.3.3. Formato de los mensajes

Llegados a este punto, es evidente la necesidad de definir los mensajes específicos que darán soporte a las funcionalidades mencionadas para nuestro protocolo. Seguidamente se discute en detalle todo ello.

Mensajes de notificación

El formato de los mensajes de notificación de alertas de incidentes de seguridad propuesto es el mostrado en la Figura 6.1. En ella se indican los campos de que constan, junto con los bits asignados a cada uno ellos. Es de significar que algunos de los campos se proponen con una longitud superior a la estrictamente necesaria en este punto para posibilitar la expansión futura del protocolo. Los campos principales desde el punto de la funcionalidad pretendida son:

- *Tipo mensaje*: necesario para diferenciar entre los distintos usos ya apuntados para el protocolo pretendido, notificación de alertas, intercambio de información y notificación asíncrona de información.
- *Evento*: codifica el tipo de ataque o evento de seguridad notificado en el mensaje. Teniendo presentes las distintas tipologías de ataque existentes (*dropping*,

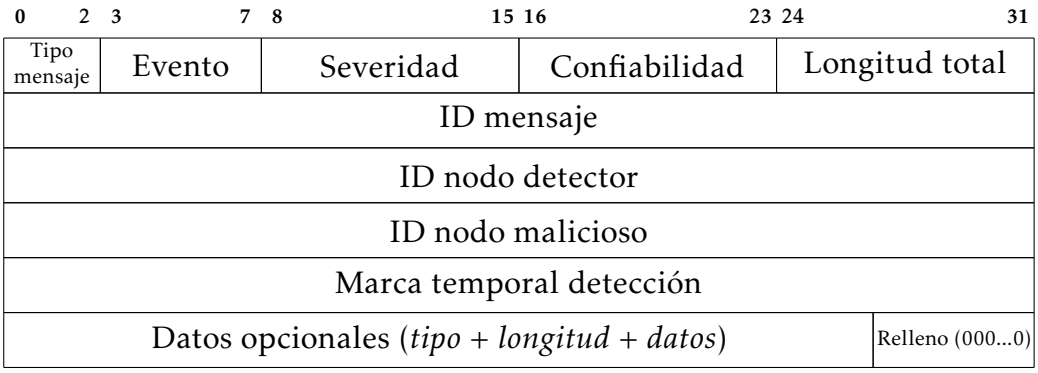


Figura 6.1: Formato de mensajes de notificación de alertas.

sinkhole, etc. [9]), parece evidente que los posibles mecanismos de respuesta a desplegar dependerán del tipo concreto de ataque detectado.

- *Severidad*: grado de afectación del evento. Por ejemplo, no es lo mismo un ataque de *dropping* donde se descarte un 20 % de los paquetes a retransmitir que uno donde se descarten todos ellos.
- *Confiabilidad*: grado de certeza con el que se concluye el proceso de detección. Por ejemplo, no es comparable la detección de un ataque fundamentada en la observación de un patrón conocido (basada en firmas o *misuse*) que una derivada de la desviación del comportamiento del sistema analizado (detección basada en anomalías). Es evidente que en el primer caso la confiabilidad será en torno al 100 %, mientras que en el segundo será función (previsiblemente) del grado de desviación observado [84].
- *ID nodo detector*: identifica el nodo que detectó el incidente reportado y que corresponde con el nodo emisor del mensaje de notificación. Esta identidad se refiere típicamente a la dirección IP del nodo detector.
- *ID nodo malicioso*: similar a la anterior, pero referida al nodo reportado como malicioso. Esta información es necesaria para la adopción de ciertos mecanismos de respuesta específicos (*p.ej.*, el aislamiento del nodo en cuestión).
- *Marca temporal detección*: identificativa del momento temporal en el que se produjo la observación del incidente de seguridad reportado. Esta información puede resultar útil de cara a la correlación de eventos.

Adicionalmente a la información principal anterior, centrada en el evento de seguridad específico detectado, otra información oportuna a considerar en los mensajes es la siguiente:

- *ID mensaje*: como es habitual en numerosos protocolos de comunicaciones, este valor se refiere a un número monótonamente creciente identificativo del mensaje para evitar la retransmisión de mensajes duplicados en la notificación de las alertas. También permite, entre otros fines, robustecer el protocolo ante ataques de repetición.
- *Longitud total*: debido principalmente al campo último que sigue abajo, es precisa la indicación expresa de la longitud total (en este caso en palabras de 32 bits) del mensaje.
- *Datos (opcional)*: aunque en la versión actual no está definido, sería interesante la inclusión de otra posible información útil varia. Por ejemplo, la localización exacta del nodo malicioso para solucionar ataques de *jamming*. Sean cuales fueren estos posibles usos futuros, el formato de este campo debe ser:

$$< tipo\_datos > \ < longitud\_octetos\_datos > \ < datos >$$

  Gracias al campo de *longitud total* previo referido, resulta posible el uso secuenciado de información extra diversa. También es de señalar la necesidad de, con objeto de que el mensaje sea múltiplo de 32 bits, contemplar un campo de *relleno* (*padding*) consistente en todo ceros, localizado (en su caso) al final del campo de información opcional.

Mensajes de solicitud y respuesta de información

Frente a la anterior tipología, en la Figura 6.2 se muestra el formato específico de los mensajes de solicitud involucrados en el intercambio de información. Los campos definidos en este caso son:

- *Tipo mensaje*: a través de este se indica la semántica del paquete. Solicitud de información de seguridad en este caso.
- # *Nodos*: correspondiente al número de nodos acerca de los que se solicita información.
- # *Variables*: cantidad de variables de información cuyo valor se solicita. Como se indica más adelante, cada variable se codifica con 32 bits.
- *Longitud total*: longitud total (en palabras de 32 bits) del mensaje enviado.
- *ID solicitud*: como en los mensajes de notificación de alertas, este campo se emplea para evitar retransmisiones repetidas de los mensajes de solicitud. En este caso también se utilizará para hacer corresponder solicitudes con respuestas.

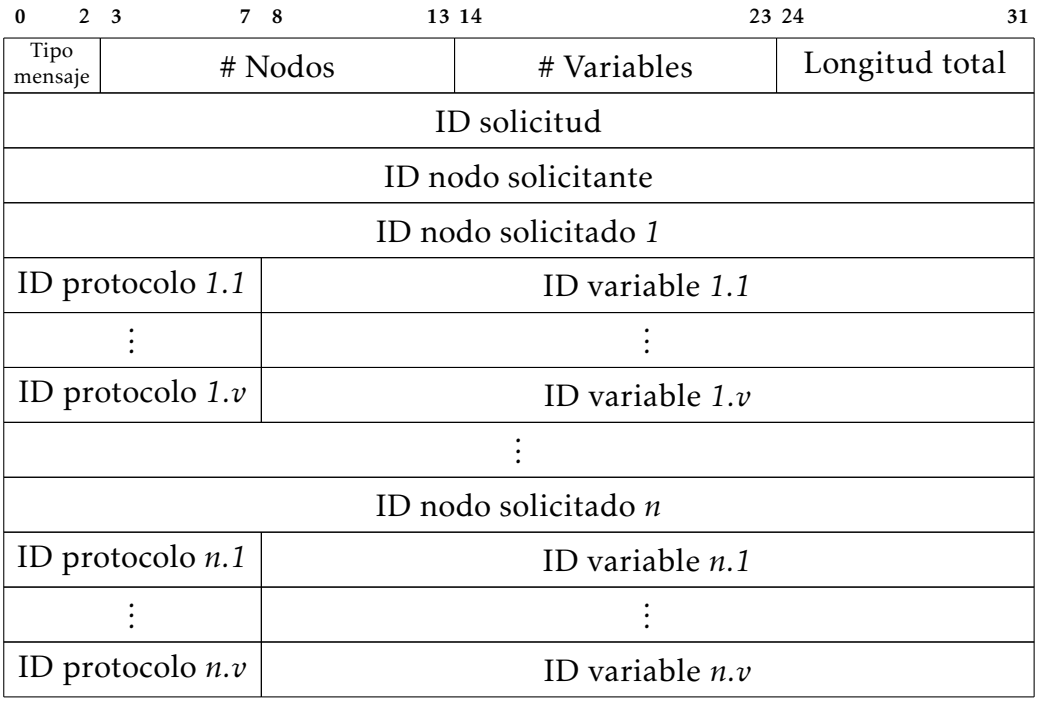


Figura 6.2: Formato de los mensajes de solicitud de información de seguridad.

- *ID nodo solicitante*: para identificar unívocamente el nodo que solicita la información y que, en definitiva, se prevé llevará a cabo el proceso de detección posterior.
- *ID nodo solicitado 1...n*: para identificar unívocamente cada uno de los n nodos (donde n es el valor indicado en el campo # *Nodos*) de los que se requiere la información.
- *Variable 1...v*: campos sucesivos de 32 bits de longitud a través de los cuales se identifica cada una de las v variables (donde v es el valor indicado en el campo # *Variables*) de las que se pide información para cada nodo solicitado. Cada variable queda definida a partir de dos campos: *ID protocolo* (o procedimiento) al que hace referencia la información requerida –*p.ej.*, IP, ICMP (*Internet Control Message Protocol*), IEEE 802.11, etc.– e *ID variable* dentro del mismo –*p.ej.*, número de paquetes RTS enviados en IEEE 802.11–.

Al respecto de los mensajes de respuesta de información de seguridad derivados de los anteriores, el formato de los mismos se muestra en la Figura 6.3, siendo los campos incluidos en el mensaje los siguientes:

- *Tipo mensaje*: respuesta a solicitud de información.

- # *Nodos*: correspondiente al número de nodos acerca de los que se proporciona información.
- # *Variables*: cantidad de variables cuyo valor se indica en el mensaje. Como se describe más adelante, cada variable implica el uso de 64 bits, 32 para su identificación y 32 para su valor.
- *Longitud total*: especifica la longitud total (en palabras de 32 bits) del mensaje enviado.
- *ID solicitud*: para hacer corresponder solicitudes con respuestas.
- *ID nodo emisor*: para identificar unívocamente el nodo que envía la información.
- *ID nodo informado 1...n*: para identificar unívocamente cada uno de los nodos de los que se comunica la información solicitada.

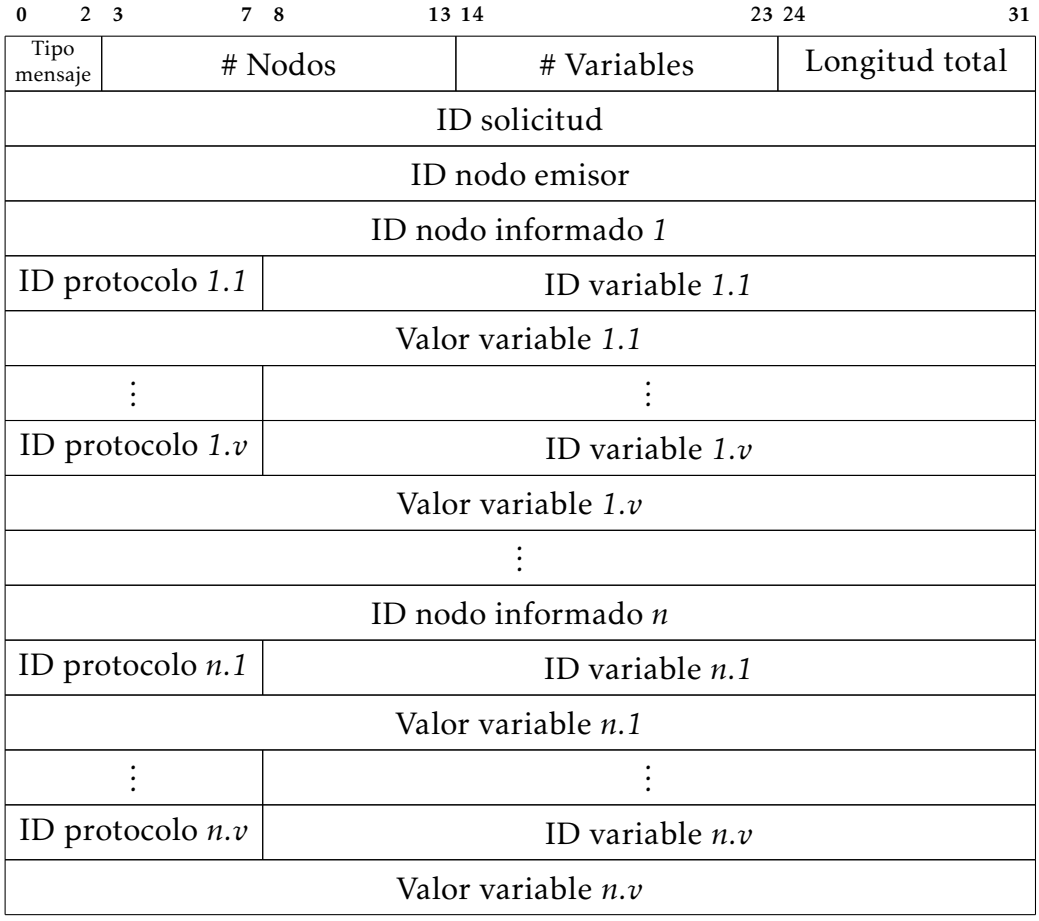


Figura 6.3: Formato de los mensajes de respuesta de información de seguridad.

- *Variable 1...v*: campos sucesivos de 64 bits de longitud a través de los cuales se identifica e informa de cada una de las v variables (donde v es el valor indicado en el campo # *Variables*) de las que se requirió información para cada nodo solicitado en el mensaje de petición. Tras ser identificada cada variable (con 32 bits como se ha establecido antes, 8 de los cuales son para indicar el protocolo/procedimiento al que se refiere), seguidamente se especificará su valor mediante un campo de 32 bits.

Aunque la funcionalidad descrita de intercambio de información objeto de estudio no se ha desarrollado completamente en la práctica en el momento de la elaboración de esta memoria, en la Tabla 6.1 se indican algunas de las variables consideradas y que son utilizadas en los IDS desplegados hasta la fecha por los autores, además de ser de amplio uso para este fin en la literatura. Ha de hacerse notar en particular la variable *NumSeq* (de AODV) intercambiada en la detección de nodos *sinkhole* en el Capítulo 5 (véase Sección 5.3.2).

En resumen, el protocolo propuesto proporciona una gran flexibilidad, posibilitando la extensión del mismo mediante la definición e inclusión de nuevas variables. En

Tabla 6.1: Ejemplo de variables para intercambio de información.

Protocolo / Procedimiento	**Variable**	**Observaciones**
Miscelánea	Periodo muestreo	En s
Topología	Velocidad	En m/s
	Aceleración	En m/s^2
	Localización	Posición GPS
Física	RSSI	De 0 a −80 dBm
MAC IEEE 802.11	$\#P_{RTS}$	Paquetes RTS / CTS; enviados y recibidos
	$\#P_{CTS}$	
AODV	$\#P_{HELLO}$	Paquetes HELLO / RREQ / RREP; enviados, recibidos, retransmitidos y descartados
	$\#P_{RREQ}$	
	$\#P_{RREP}$	
	NumSeq	Número de secuencia
	HopCount	Número de saltos
Aplicación	$\#P_{datos}$	Paquetes de datos; enviados, recibidos y perdidos
	#Sesiones	Número de sesiones

concreto, habida cuenta que los esquemas IDS propuestos por los autores son multicapa (acceso al canal, capa de red, etc.), la información requerida se va a identificar en los mensajes intercambiados organizada en base a los protocolos/procedimientos específicos a los que aquella se refiere.

Mensajes de notificación asíncrona de información de seguridad

Por último, en relación con los mensajes de notificación asíncrona de información de seguridad, su formato es el mismo que el mostrado en la Figura 6.3 y empleado por las respuestas de intercambio de información, con los siguientes matices:

- *Tipo mensaje*: notificación asíncrona de información de seguridad.
- *ID mensaje*: identificativo del paquete en sí y no para hacer corresponder solicitudes con respuestas.

Todas las cuestiones previamente abordadas, así como la propia especificación de los distintos mensajes, tienen un impacto directo sobre las prestaciones de las comunicaciones del entorno monitorizado. Ello es estudiado brevemente en el siguiente apartado.

6.4. Análisis de prestaciones

En esta sección se realizará un breve análisis teórico de prestaciones del protocolo propuesto. Para ello, más allá de la comparativa realizada con el uso de AODV en la Sección 5.3.2, se estimará el ancho de banda AB (en bytes/s) consumido por la transmisión de los mensajes previamente especificados.

Consideremos una red MANET compuesta de L nodos legítimos $\{N_1, ..., N_L\}$ con un rango de cobertura de r metros, y que se encuentran distribuidos uniformemente en un área de $a \times b$ metros^2, con $a, b \gg r$. Asumiendo la existencia de movilidad, cada nodo N_i tendrá su propio conjunto de vecinos NB_i. En este escenario general, consideramos adicionalmente la existencia de M nodos maliciosos. Dichos nodos serán excluidos de los cálculos, pues es de suponer que estos no participarán en actuaciones que tienen como objetivo su propia detección y/o aislamiento de la red.

Para el cálculo del ancho de banda consumido será necesario definir una serie de parámetros de interés, así como sus notaciones.

- $f_i^{a/s/n}$: representa la frecuencia (en transmisiones por segundo) con la que un nodo N_i envía mensajes (de alerta, de solicitud de información de seguridad o

de notificación asíncrona) acerca de una serie de nodos. Dicha frecuencia de transmisión vendrá determinada, entre otros, por el procedimiento de detección subyacente implementado.

- $\boldsymbol{E[NB_i]}$: denota el número esperado de vecinos del nodo N_i. Dada L/ab la densidad de nodos en el área total, y $(L/ab)\pi r^2$ el número esperado de nodos en el área de cobertura de N_i, es evidente que, restando el propio nodo:

$$E[NB_i] = \frac{(L-1)\pi r^2}{ab} \tag{6.1}$$

- $\boldsymbol{E[n]}$: número esperado de nodos sobre los que se requiere/proporciona información de seguridad.
- $\boldsymbol{E[v]}$: número esperado de variables solicitadas/proporcionadas para cada uno de los nodos indicados en el mensaje.
- $\boldsymbol{PS^{a/s/r/n}}$: representa el tamaño de los paquetes transmitidos (alerta, solicitud o respuesta de información o notificación asíncrona de información). El tamaño de los mensajes de alerta, considerando que no existen datos adicionales, es de 20 bytes, mientras que el tamaño de los mensajes de solicitud, respuesta y notificación asíncrona de información depende del número nodos por los que se pregunta/informa y del número de variables solicitadas/informadas, siendo $12+4{\cdot}E[n]+4{\cdot}E[n]{\cdot}E[v]$ el tamaño en bytes de los primeros y $12+4{\cdot}E[n]+8{\cdot}E[n]{\cdot}E[v]$ el tamaño en bytes de los dos últimos.
- $\boldsymbol{p(I)}$: representa la probabilidad de que un nodo dado de la red conozca la información requerida relativa a los nodos solicitados y, en consecuencia, pueda responder con un mensaje (*unicast*) a la solicitud recibida.

Una vez definida la notación, se ha de distinguir la aplicación concreta para la que se está empleando el protocolo pues tanto el número como el tamaño de los paquetes intercambiados (y con ello el ancho de banda) serán dependientes del uso.

6.4.1. Notificación de alertas

Para calcular el ancho de banda consumido por la notificación de alertas debemos considerar el peor escenario, es decir, aquel en el que todos los nodos de la red tienen conectividad con al menos otro de los nodos. Puesto que la idea es notificar a todos los nodos la existencia del nodo malicioso, este proceso se realizará mediante una difusión a toda la red, donde cada nodo retransmitirá a su vez el mensaje de alerta recibido. En esta situación, el número de paquetes de alerta propagados por la red

para la notificación iniciada por el nodo N_i relativa al nodo malicioso N_j será, como máximo, de L paquetes (siendo L el número de nodos legítimos en la red).

En consecuencia, el valor esperado del ancho de banda (en bytes/s) para las situaciones de alerta iniciadas por el nodo N_i respecto a otro N_j, $AB^a_{i,j}$, será:

$$E[AB^a_{i,j}] = f^a_i \cdot PS^a \cdot L \tag{6.2}$$

6.4.2. Intercambio de información de seguridad

Con respecto a la segunda funcionalidad aquí prevista, el intercambio de información se producirá cada vez que un nodo N_i precise conseguir información de seguridad acerca de una serie de nodos para, por ejemplo, determinar su comportamiento de forma cooperativa. Dicho flujo se inicia con un mensaje de solicitud *broadcast* a los vecinos, que será respondido únicamente por aquellos que conozcan la información solicitada por el solicitante. Las respuestas serán enviadas en mensajes de respuesta *unicast*.

En consecuencia, el valor esperado del ancho de banda consumido ante las posibles peticiones de información de un nodo N_i a sus vecinos respecto de un conjunto de nodos, AB^{int}_i (en bytes/s), será:

$$E[AB^{int}_i] = f^s_i \cdot \left(PS^s + PS^r \cdot E[NB_i] \cdot p(I) \right) \tag{6.3}$$

La Figura 6.4 muestra una comparación entre el ancho de banda consumido por la funcionalidad de intercambio de información de seguridad a partir de la Ec. (6.3), variando el número esperado de vecinos, $E[NB_i]$, y para dos valores distintos del número esperado de nodos solicitados: $E[n] = 3$ y 10. Por simplicidad, se considerará también que se está solicitando información sobre una única variable (*i.e.*, $E[v] = 1$), y que nos encontramos en el peor escenario, es decir, aquel en el que todos los nodos vecinos conocen la información solicitada (*i.e.*, $p\left(I\right) = 1$). Además, también se estudia la frecuencia de las transmisiones, considerándose un escenario especialmente desfavorable (y poco realista) en el que se requiere un intercambio de información entre los nodos a intervalos de 1 s ($f^s_i = 1$), y un segundo escenario en el que el intercambio de información solo es necesario cada 10 s ($f^s_i = 0{,}1$).

Puede observarse que, incluso para un número elevado de variables solicitadas y de vecinos involucrados en el intercambio de información de seguridad propuesto para esquemas de detección y/o respuesta colaborativos, nuestro protocolo de comunicación introduce una baja sobrecarga. Esta experimentación teórica se ha realizado únicamente para la funcionalidad que introduce un mayor consumo de ancho de

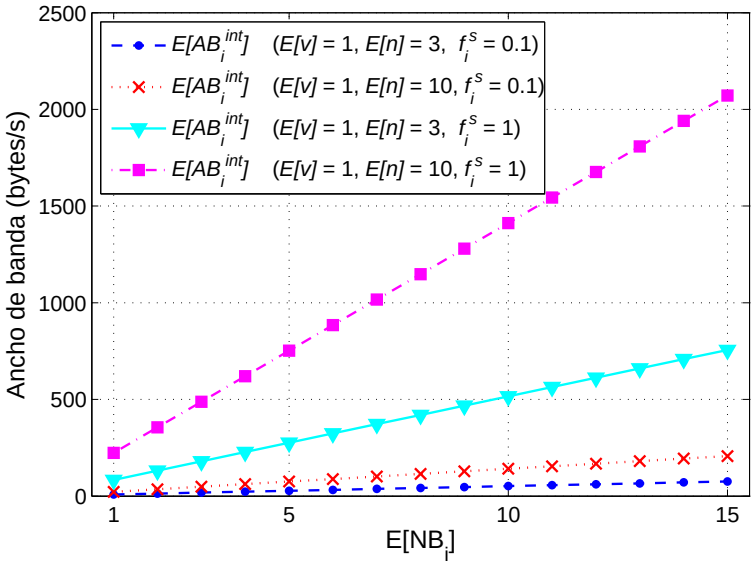


Figura 6.4: Ancho de banda consumido por el intercambio de información de seguridad para distintos parámetros.

banda, el intercambio de información de seguridad. Así, dado que el consumo en este caso puede considerarse aceptable en entornos MANET, se puede asegurar que el ancho de banda introducido por las otras dos funcionalidades es también asumible.

6.4.3. Notificación asíncrona de información de seguridad

En relación con la última aplicación propuesta para nuestro protocolo, la notificación de información se producirá cada vez que un nodo N_i considere oportuno proporcionar información de distintas variables acerca de una serie de nodos (entre los que puede incluirse a sí mismo). Al tratarse de una notificación asíncrona, la comunicación consistirá únicamente en el envío de un mensaje *broadcast* a los vecinos, sin que exista necesidad de solicitud previa ni de una posterior respuesta.

El valor esperado del ancho de banda introducido por la notificación de información de seguridad por parte de un nodo N_i a sus vecinos acerca de una serie de nodos y variables, AB_i^n (en bytes/s), será:

$$E[AB_i^n] = f_i^n \cdot PS^n \tag{6.4}$$

Concluido este breve análisis teórico, una vez que se complete la implementación efectiva del protocolo propuesto, estos estudios deberán concretarse sobre escenarios prácticos a fin de ser conscientes de los requisitos reales involucrados.

6.5. Conclusiones del capítulo

En este capítulo se ha propuesto y discutido un protocolo de notificación y alerta de eventos de seguridad ideado para la comunicación de actividades maliciosas contra la seguridad de un entorno de red. El procedimiento permite proporcionar diversa información útil, de cara a la adopción de medidas reactivas subsiguientes a la detección. Esta notificación es realizada de forma distribuida a fin de permitir su uso en entornos no centralizados como son las redes ad hoc. Además, el protocolo puede ser usado también como mecanismo de intercambio de información de seguridad entre nodos en este tipo de entornos con objeto de posibilitar una detección colaborativa, así como para la notificación asíncrona de dicha información de seguridad.

Si bien las bondades de la propuesta han sido evidenciadas a nivel teórico en el documento, es objetivo inmediato de los autores la implementación efectiva del protocolo y evaluación de prestaciones del mismo en escenarios experimentales de simulación. Este desarrollo prevé incorporarse al *framework* NETA [146], diseñado y desarrollado por el grupo de investigación NESG (*Network Engineering & Security Group*) (http://nesg.ugr.es), de la Universidad de Granada, y que será explicado en el capítulo siguiente.

Publicaciones relacionadas

Para concluir el capítulo se presentan las publicaciones directamente relacionadas con el ámbito de estudio abordado. Estas son:

- **L. Sánchez-Casado**, R. Magán-Carrión, P. Garrido-Sánchez y P. García-Teodoro. "Protocolo para la Notificación y Alerta de Eventos de Seguridad en Redes Ad hoc". Aceptado en *Reunión Española sobre Criptología y Seguridad de la Información* (RECSI), 6 páginas, 2014.

Capítulo 7

NETA: *framework* de seguridad para desplegar y analizar ataques en redes

Como se ha puesto de manifiesto a lo largo de los diferentes capítulos de este trabajo de tesis, la seguridad es uno de los principales problemas a la hora de desarrollar nuevas tecnologías y servicios en redes de comunicaciones. De este modo, y con el propósito de integrar y reunir distintos desarrollos defensivos a fin de lograr una seguridad integral, es especialmente interesante el diseño de entornos de seguridad genéricos que faciliten la tarea de desplegar y estudiar nuevas soluciones defensivas frente a gran variedad de ataques, amenazas o vulnerabilidades.

A este respecto, y en la línea ya comenzada en el Capítulo 6 con el desarrollo de un protocolo de notificación y alertas, en este capítulo se presenta NETA (*NETwork Attacks*), un *framework* para la simulación de ataques y defensas en redes de comunicación desarrollado sobre el *framework* INET y el simulador OMNeT++. Su diseño flexible y versátil es apropiado para el despliegue de multitud de ataques, permitiendo implementar, integrar y comparar de forma fiable y bajo condiciones controladas de simulación nuevas técnicas de prevención, detección y/o respuesta.

Se pretende aportar así un marco de referencia para analizar el impacto de diferentes ataques y diversos esquemas defensivos, en múltiples tecnologías, protocolos y escenarios, que constituya una herramienta útil a la comunidad investigadora centrada en el campo de la seguridad en redes. Como prueba de concepto se han implementado tres ataques diferentes hasta la presente: *dropping*, *delay* y *sinkhole*, siendo las capacidades de NETA puestas de manifiesto mediante la evaluación del funcionamiento de estos tres ataques sobre distintos despliegues MANET.

El resto del capítulo se organiza de la siguiente forma. En la Sección 7.1 se motiva el interés de desarrollar un marco de simulación genérico de seguridad integral. En la Sección 7.2 se proporciona un análisis del estado del arte, describiéndose distintos simuladores, así como otras propuestas similares a la presentada en este trabajo. La arquitectura general del *framework* NETA se presenta en la Sección 7.3, donde se explican los principales componentes y las reglas de diseño. En la Sección 7.4 se describen los ataques implementados en esta primera versión. La Sección 7.5 detalla los escenarios de estudio, así como el entorno de experimentación y los resultados obtenidos. Finalmente, la Sección 7.6 expone las conclusiones.

7.1. Motivación

Como se ha visto a lo largo de todo el trabajo de tesis, la provisión de seguridad se está convirtiendo en uno de los principales retos a la hora de desarrollar nuevas tecnologías y servicios en redes de telecomunicaciones. Las metodologías utilizadas por los *hackers* evolucionan constantemente y a gran velocidad hacia nuevas técnicas de ataque y objetivos [173], [30], dificultando enormemente el desarrollo de mecanismos de defensa.

En este contexto, se han llevado a cabo numerosos esfuerzos por parte de la comunidad investigadora para desplegar nuevas técnicas de defensa destinadas a frustrar los ataques a la seguridad en redes. El ciclo es casi siempre el mismo: cada vez que se descubre una nueva vulnerabilidad o técnica de ataque, se implementa una prueba de concepto específica, se evalúan las capacidades de dicha técnica y se proponen nuevas técnicas de defensa. Sin embargo, y aunque numerosos investigadores contribuyen con el código fuente de sus ataques, el resultado de esta metodología de investigación es que no existen implementaciones de ataques aceptadas por la mayoría de la comunidad investigadora que permitan la comparación de las soluciones propuestas frente a dichos ataques en las mismas condiciones.

Por tanto, resulta deseable la existencia de un *framework* común que posibilite el desarrollo de implementaciones de ataques y de sus respectivas defensas. Ello permitiría combinar la ejecución de todos los ataques implementados, de forma similar a como lo haría un *hacker*, así como analizar su impacto en múltiples tecnologías, protocolos y escenarios.

Son diversas las técnicas que se emplean hoy día a la hora de analizar el funcionamiento de una red [174]; por ejemplo, modelado analítico, despliegues reales y simulación. El modelado analítico no es una solución viable en sistemas altamente dinámicos cuyas propiedades son difíciles de capturar en un análisis matemático, mientras que los despliegues reales son aproximaciones complejas, costosas y que consumen mucho tiempo, especialmente en entornos ad hoc, que generalmente im-

plican el despliegue de multitud de nodos en áreas de difícil acceso. Por ello, la mejor opción para la evaluación de este tipo de entornos suele ser el empleo de herramientas de simulación. Así, la simulación se usa generalmente con la intención de analizar protocolos, algoritmos o sistemas complejos en sus distintas etapas (diseño, desarrollo, implementación, ...), ofreciendo un compromiso adecuado entre coste y complejidad [175]. Optando por esta solución, en la siguiente sección se estudiarán algunos de los principales simuladores existentes, seleccionándose aquel que mejor se adapta a las necesidades requeridas para el desarrollo del entorno de seguridad aquí pretendido.

La contribución principal resultante de este trabajo es NETA, un *framework* de ataques que pretende proporcionar un marco base de referencia con el que unificar el desarrollo y simulación de ataques. NETA es extensible y ofrece un alto grado de versatilidad para el desarrollo de nuevos ataques y soluciones defensivas. Su objetivo es minimizar los esfuerzos en el proceso de creación de ataques con el propósito de probar y evaluar distintas soluciones de seguridad. NETA está disponible al público para su descarga en http://nesg.ugr.es/index.php/en/neta, así como en la lista de *frameworks* ofrecida en la propia web de OMNeT++.

7.2. Trabajos relacionados

Como se ha motivado en la sección anterior, las herramientas de simulación son la alternativa óptima para el desarrollo de entornos de red, y en particular, de seguridad en estos sistemas. Sin embargo, la elección del simulador más apropiado no es una tarea sencilla, pues requiere de un estudio previo que considere las distintas ventajas y desventajas de los distintos existentes. Algunos de los simuladores más utilizados en el campo de las comunicaciones [176], [177] se presentan a continuación.

NS-2 (*Network Simulator 2*) [139] es un simulador de eventos discretos para redes de comunicaciones desarrollado usando la combinación de dos lenguajes, *C++* y OTcl (*Object-oriented Tool command language*) [178], un lenguaje de *scripting* orientado a objetos. NS-2 es uno de los simuladores más conocidos y empleados debido a la gran variedad de protocolos que ofrece. Sin embargo, no posee una interfaz amigable, es poco escalable y además requiere de gran esfuerzo para su aprendizaje. Debido a ello, en los últimos años su uso está siendo reemplazado por otras herramientas, en particular en lo relativo a la simulación de redes ad hoc.

Diseñado con el objetivo de reemplazar a NS-2, aunque no compatible con este, uno de los simuladores cuyo empleo está creciendo más en los últimos tiempos es NS-3 (*Network Simulator 3*) [179]. Escrito en *C++* y con una interfaz *Python*, NS-3 mejora en diversos aspectos el funcionamiento de NS-2. Sin embargo, la variedad de

protocolos disponibles (aún) no es tan elevada como la de otros simuladores, siendo esta su mayor limitación.

GloMoSim (*Global Mobile system Simulator*) [180] es un entorno de simulación escalable para redes cableadas e inalámbricas diseñado usando las capacidades proporcionadas por *PARSEC* [181], una variante del lenguaje *C* para la ejecución secuencial y paralela de modelos de simulación de eventos discretos. Sin embargo, la principal desventaja de GloMoSim es su limitación a redes IP y su mantenimiento no continuado.

OPNET (*OPtimized Network Engineering Tools*) [182] es un simulador de eventos discretos orientado a objetos, diseñado inicialmente con propósitos militares. OPNET es una herramienta de uso comercial (aunque ofrece versiones limitadas de uso académico), por lo que ofrece una gran potencia y robustez, siendo destacable su capacidad de ejecutar y gestionar de forma concurrente y rápida simulaciones a gran escala. No obstante lo anterior, no dispone de muchos modelos disponibles para la simulación de redes inalámbricas.

Además de los anteriores, cabe destacar OMNeT++ (*Objective Modular Network Test-bed in C++*) [144], una plataforma de simulación con una arquitectura genérica y flexible implementada en *C++* que proporciona la infraestructura necesaria para desarrollar diferentes simulaciones. OMNeT++ se está convirtiendo en la actualidad en uno de los simuladores más empleados, principalmente debido a la amplia variedad de *frameworks* (INET, MiXiM, Castalia, etc.) que ofrece una gran flexibilidad y la inclusión de una interfaz gráfica fácil de usar, entre otras muchas ventajas. Por estas razones, el desarrollo del entorno de seguridad propuesto se realizará sobre este simulador.

En cuanto al diseño y simulación de ataques tomando como base este u otros simuladores, los autores implementan generalmente ataques específicos, con el propósito de usarlos para probar propuestas de seguridad, rendimiento de protocolos, etc. [183]. Sin embargo, estas implementaciones suelen ser privadas y, por tanto, no es posible comparar distintas propuestas de defensa con la misma implementación del ataque, haciendo que dichas comparativas sean poco precisas y fiables.

Otro caso es el de [184], donde se proporciona un *framework* basado en OMNeT++ para simular patrones de tráfico y ataques DoS sobre redes IP. Sin embargo, solo implementan un tipo específico de ataque y su propuesta no es extensible a otros tipos. Otro *framework* de simulación de ataques, esta vez aplicado a WSN, se propone en [185]. Los autores presentan un procedimiento para simular ataques basado en un lenguaje particular que describe el comportamiento de los mismos. El *framework* parece ser extensible, pero no se encuentra disponible públicamente y no es aplicable a otros entornos distintos de las redes de sensores.

Por todas estas razones, consideramos necesario un *framework* de ataques general, extensible y versátil, que aborde y dé solución a los inconvenientes citados. Con este fin, en este trabajo se propone y desarrolla NETA.

7.3. NETA: *framework* para la simulación de ataques

NETA se ha desarrollado como un *framework* de OMNeT++, construido sobre INET. Considerando que OMNeT++ es una de las herramientas de simulación más usadas en el ámbito de la simulación de redes de comunicación, se pretende extender el uso de NETA entre la comunidad investigadora. A continuación se detallan algunos fundamentos básicos acerca del funcionamiento general de OMNeT++, como paso previo a la introducción y descripción de NETA.

7.3.1. Fundamentos de OMNeT++

La plataforma OMNeT++ divide el proceso de simulación en varios **estadios**: descripción de funcionalidad, ensamblado de módulos y simulación. Estos son como sigue:

- La descripción de la funcionalidad de un módulo simple se realiza dentro de una jerarquía de clases de *C++*. En ella, OMNeT++ proporciona el núcleo (primeros niveles de jerarquía), la funcionalidad e interfaces básicas.
- Para crear sistemas complejos donde existe interconectividad e interoperación entre diferentes módulos simples o complejos, OMNeT++ proporciona un lenguaje de alto nivel llamado NED (*NEtwork Description*).
- Para realizar una simulación, OMNeT++ espera una arquitectura concreta descrita en NED y un fichero de descripción de simulación `.ini` donde se configuran los parámetros de la simulación y se concretan los elementos parametrizables de la descripción NED.

OMNeT++ permite definir **módulos** simples mediante una descripción `.ned` que indica, entre otras características, las puertas (o *gates*) de entrada y salida. A esta descripción se le añade la funcionalidad deseada mediante una clase de *C++* que extenderá a las clases básicas de la jerarquía de OMNeT++ para asegurar la implementación de los métodos básicos de funcionamiento. En una descripción `.ned` también pueden definirse parámetros generales, que se concretarán en el momento de simular. A partir de estos módulos simples cuyo funcionamiento básico está establecido, se definen módulos que heredan dichas funciones siguiendo una jerarquía

típica orientada a objetos. A nivel de descripciones `.ned`, estos módulos hijos también son simples ya que su funcionalidad, aunque heredada, sigue estando implementada en *C++*.

El siguiente nivel de abstracción son los módulos complejos, formados a partir de la interconexión de módulos simples. De la misma manera que estos, en los módulos complejos también se definen puertas de entrada y salida, así como distintos parámetros. Al estar formados por módulos simples, la descripción de cada módulo complejo ha de incluir el modo en que sus componentes simples se interconectan entre sí y con las puertas de entrada y salida externas del módulo complejo. La jerarquía a partir de este punto es ilimitada, pudiendo crear tantos niveles de módulos complejos como sean necesarios. A modo de ejemplo, los nodos de la red están definidos como módulos complejos, formados a su vez por distintos módulos simples y/o complejos que implementan los distintos protocolos o procedimientos.

En relación con la **comunicación** entre módulos, esta se puede realizar en base a dos procedimientos:

1. *Puertas*: este procedimiento es empleado generalmente para proporcionar comunicación entre nodos, y es llevado a cabo mediante el uso de las puertas de entrada y/o salida definidas, que se conectan entre sí mediante enlaces (clases que modelan las características de un canal) y transmiten objetos de tipo mensaje. Estos objetos de tipo mensaje parten de una clase básica de la jerarquía de OMNeT++ y son extensibles. De esta manera, se pueden implementar mensajes con diferentes campos y con diferentes métodos.

2. *Paso de mensajes*: aunque la comunicación entre módulos puede producirse mediante puertas, es también usual llevarla a cabo mediante un procedimiento de llamadas a funciones de *C++*, también denominado *paso de mensajes*. Esto es así porque no debe olvidarse que bajo los procedimientos propuestos por OMNeT++ existe un conjunto de clases instanciadas en objetos que se ejecutan durante la simulación y se pueden comunicar entre ellas a nivel de programa.

Las comunicaciones basadas en puertas modelan el tiempo de simulación que se emplea en la transmisión del mensaje, es decir, estas transmisiones no son inmediatas. Por ejemplo, la transmisión de una trama *Ethernet* por un enlace entre dos nodos tendrá en cuenta el tiempo de propagación y demás retardos existentes en las transmisiones. Por su parte, las comunicaciones basadas en paso de mensajes son inmediatas.

Las **simulaciones** en OMNeT++ se definen mediante dos ficheros. El primero de ellos describe la topología del sistema o de la arquitectura mediante una descripción `.ned`. Por otra parte, un fichero `.ini` define los parámetros de cada uno de los módulos, pudiendo ser accedidos estos parámetros por las clases que implementan

el funcionamiento, lo que posibilita la parametrización de las simulaciones sin necesidad de recompilar los ficheros fuente. De este modo, OMNeT++ permite una separación completa entre diseño y simulación. Por una parte, un conjunto de clases y descripciones `.ned` de módulos simples y complejos conforma el modelado o diseño del sistema. Por otro lado, un conjunto de descripciones `.ned`, junto con sus archivos de parametrización `.ini`, conforma las simulaciones del proyecto.

En otro orden de cosas, INET es una librería o *framework* de OMNeT++ que implementa modelos para multitud de tecnologías y protocolos presentes en arquitecturas de redes de comunicaciones cableadas e inalámbricas, tales como IEEE 802.11, IP, UDP, TCP. Además, INET presenta implementaciones para diversos modelos de movilidad o de propagación. Sobre todo ello, NETA añadirá funcionalidades extra en algunos de los módulos de INET con el fin de permitir la ejecución de distintos ataques o soluciones defensivas.

Una vez explicados los fundamentos básicos de OMNeT++, se detalla seguidamente la arquitectura del *framework* propuesto: NETA.

7.3.2. Arquitectura de NETA

NETA se basa en la misma idea que OMNeT++, *i.e.*, módulos que se comunican entre sí mediante el procedimiento de llamadas a funciones o paso de mensajes.

La idea general es desarrollar en OMNeT++ nuevos nodos que puedan ejecutar ataques, esto es, *nodos atacantes*. Para llevar a cabo y configurar los ataques se hace uso de los denominados *controladores de ataque*. Dichos controladores gestionan uno o varios módulos de NETA mediante el envío de *mensajes de control*. Estos mensajes viajan desde los controladores de ataque hacia módulos específicos que previamente han sido modificados para implementar el comportamiento del ataque. Estos módulos se denominan *módulos hackeados*. Para implementar el citado comportamiento, los módulos *hackeados* heredan o replican el código de módulos de INET que, posteriormente, modifican conveniente para obedecer las órdenes indicadas por los controladores.

Los principios de diseño de NETA siguen dos reglas principales:

Regla 1. *Cualquier* framework *base utilizado no debe ser modificado en modo alguno. Por ejemplo, cuando se utilizan módulos de INET, estos deben permanecer como los originales.* Esta regla pretende facilitar la compatibilidad con futuras versiones de INET y otras implementaciones. Para lograr este objetivo, simplemente se importa la versión más reciente de INET y no se lleva a cabo ninguna modificación sobre ella.

Regla 2. *Modificar lo mínimo posible el código original de los módulos* hackeados. Obviamente, para implementar los ataques deseados es necesario realizar

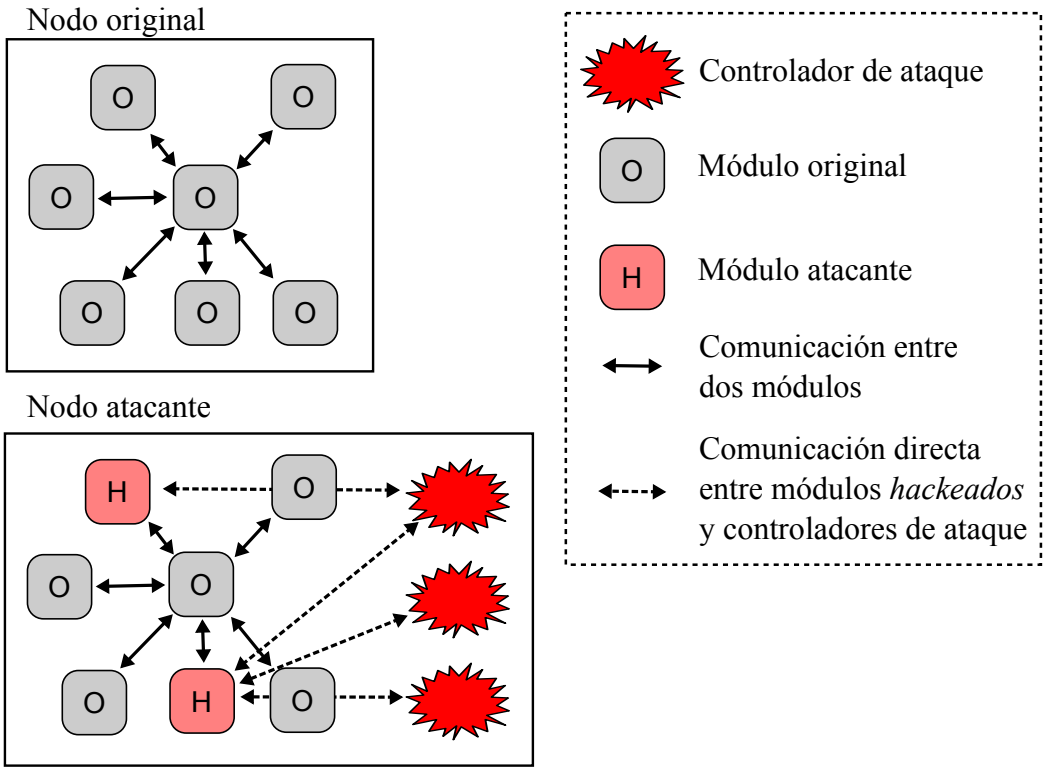


Figura 7.1: Esquema comparativo entre un nodo original y el correspondiente nodo atacante en NETA.

modificaciones en el comportamiento de los módulos que se convertirán en módulos *hackeados*. Sin embargo, esta regla pretende minimizar estas modificaciones tanto como sea posible.

En consecuencia, la creación de un nodo atacante puede resumirse en los siguientes pasos: (*i*) añadir al archivo `.ned` asociado los controladores relacionados con los ataques a ejecutar, (*ii*) crear los mensajes de control asociados y (*iii*) sustituir los módulos requeridos por parte de los controladores de ataque por los módulos *hackeados* correspondientes.

La Figura 7.1 muestra las diferencias entre un nodo normal y un nodo atacante. El nodo normal se compone de módulos simples y compuestos comunicándose entre sí. El nodo atacante se compone, por su parte, del mismo número de módulos, a los que se añaden los correspondientes controladores. Además, algunos de los módulos originales son reemplazados por los módulos *hackeados* para permitir la ejecución del ataque cuando este sea iniciado por los controladores de ataque.

A continuación se describen los componentes principales de un ataque en nuestro *framework*: (*i*) *controladores de ataque*, (*ii*) *mensajes de control*, y (*iii*) *módulos hackeados*.

Controladores de ataque

Los módulos que controlan la ejecución de los ataques poseen las siguientes propiedades:

- `attackType`: nombre proporcionado para diferenciar un ataque del resto.
- `active`: indica si el ataque se encuentra o no activo durante la simulación.
- `startTime`: tiempo en el que el ataque comienza a lo largo de la simulación.
- `endTime`: tiempo en el que cesa el ataque.
- `parametros_especificos_de_ataque`: diferentes parámetros de configuración que dependen de las funcionalidades específicas del ataque.

El proceso llevado a cabo por un controlador de ataque para un ataque dado en un nodo atacante puede resumirse como sigue:

1. Obtener los diferentes módulos *hackeados* involucrados en la ejecución del ataque deseado.
2. Activar aquellos módulos *hackeados* en el nodo atacante, enviando mensajes de activación que también pueden contener información de configuración.
3. Desactivar los módulos *hackeados* en el nodo atacante, enviando un mensaje de desactivación.

Mensajes de control

Estos son mensajes enviados desde los controladores de ataque a los módulos *hackeados* involucrados en la ejecución del ataque. Dichos mensajes contienen la información necesaria para la activación y desactivación de los ataques. Además, estos mensajes pueden incluir la información de configuración necesaria para la ejecución de los ataques.

Es importante remarcar que los mensajes de control se envían directamente a los módulos *hackeados*. Esta es la mejor opción encontrada para cumplir con la segunda regla de nuestros principios de diseño: "Minimizar la modificación en el código original de los módulos *hackeados*".

Módulos *hackeados*

Estos módulos se refieren a aquellos cuyo comportamiento ha sido modificado para ejecutar un determinado ataque. Por ejemplo, un ataque que descarte paquetes (*dropping*) usualmente requiere la modificación del módulo encargado del reenvío a nivel IP. Por tanto, la implementación de dicho ataque implica la modificación del módulo `IPv4` en NETA, que se comportará así como un módulo *hackeado*.

Es importante resaltar que solo existe un módulo *hackeado* por módulo modificado, en lugar de un módulo *hackeado* por cada implementación de un ataque. Si dos ataques diferentes necesitan modificar el mismo módulo, solo existirá un único módulo *hackeado*. Por ejemplo, como se mostrará en la sección siguiente, tanto el ataque de *dropping* como el de *delay* están relacionados con el módulo `IPv4`. Sin embargo, solo se necesita un módulo `IPv4` *hackeado* para la implementación de ambos ataques. Este diseño tiene como objetivo mejorar la flexibilidad del *framework*, permitiendo la ejecución de más de un ataque simultáneamente, *p.ej.*, los ataques de *dropping* y *delay* pueden lanzarse en un mismo nodo sin más que incluir sus correspondientes controladores de ataque.

7.4. Ataques implementados

En esta sección se exponen los ataques implementados como prueba de concepto sobre el *framework* NETA. En las subsecciones siguientes se describirán, para cada ataque: (*i*) el comportamiento de este y (*ii*) los parámetros que pueden modificarse para configurar dicho ataque.

7.4.1. Ataque de *dropping* en IP

Como ya sabemos, en este ataque, los nodos que exhiben dicho comportamiento descartan, de forma intencionada y con una cierta probabilidad, los paquetes de datos recibidos, en vez de retransmitirlos. De este modo se ve interrumpido el funcionamiento normal de la red. Según la aplicación afectada, el resultado puede ser una ralentización de la red debido a numerosas retransmisiones, un excesivo consumo de energía en los nodos, etc. Los principales parámetros disponibles en la implementación del ataque son:

- `droppingAttackProbability`: es la probabilidad de descartar un paquete de datos, definida entre 0 y 1. Por defecto está fijada a 0, lo que indica que el nodo atacante se comporta de forma legítima, *i.e.*, sin descartar paquetes.

7.4.2. Ataque de *delay* en IP

En el ataque de *delay* los nodos retrasan los paquetes de datos IP durante un cierto tiempo. Esto puede afectar a distintos parámetros de QoS (retardo extremo-a-extremo, *jitter*, etc.), dando como resultado un rendimiento de la red pobre. La lista de parámetros de nuestra implementación es:

- `delayAttackProbability`: la probabilidad de retardar un paquete de datos, definida entre 0 y 1. Por defecto está fijada a 0, lo que implica un comportamiento normal del nodo atacante, *i.e.*, no se aplica ningún retardo extra.
- `delayAttackValue`: el tiempo de retardo específico aplicado a cada paquete. Este parámetro puede especificarse de acuerdo a una distribución estadística. Por esta razón, el parámetro está definido como volátil, *i.e.*, es modificado cada vez que se accede a él. Por defecto, sigue una distribución normal con media 1 segundo y desviación estándar 0,1 segundos.

7.4.3. Ataque *sinkhole* en AODV

En un ataque *sinkhole* los nodos atacantes envían información falsa de *routing*, anunciando que tienen una ruta óptima hacia un destino y provocando que otros nodos encaminen los paquetes de datos a través de los atacantes. Aquí, los atacantes falsifican los mensajes de respuesta RREP para atraer tráfico. Los parámetros del *sinkhole* son:

- `sinkholeAttackProbability`: es la probabilidad de responder a un mensaje de solicitud RREQ con una respuesta RREP falsa, entre 0 y 1. Por defecto está fijada a 0, lo que implica un comportamiento normal del protocolo AODV.
- `sinkOnlyWhenRouteInTable`: si está fijado a *true*, el *sinkhole* solo envía falsos RREP a solicitudes para las que el atacante tenga una ruta válida, *i.e.*, rutas existentes en su tabla de *routing*. En caso contrario (valor *false*), el nodo envía RREP falsos a cualquier mensaje RREQ que le llegue, incluso si no tiene una ruta válida.
- `seqnoAdded`: número falso de secuencia generado por el nodo atacante, el cual es añadido al número de secuencia observado en la solicitud. Puede ser distinto en cada ocasión si está especificado como una distribución estadística. Por defecto, sigue una distribución uniforme con valores entre 20 y 30.
- `numHops`: número falso de saltos devuelto por el atacante. Por defecto está fijado a 1, indicando que el atacante alcanza el destino de la comunicación en un único salto.

7.5. Resultados experimentales

En esta sección se presenta el entorno experimental utilizado para evaluar los ataques presentados en la sección anterior. Además, se han realizado distintos tests con objeto de verificar el funcionamiento correcto de cada ataque, midiendo su impacto en la red en base a distintas métricas.

Con esta evaluación se pretende presentar la funcionalidad de simulación de NETA, poniendo de manifiesto su capacidad para facilitar la extracción de información sobre el funcionamiento de los ataques.

7.5.1. Descripción del entorno experimental

Como caso de estudio se simulan una serie de despliegues MANET, cuyos parámetros comunes se describen a continuación.

El área de simulación se restringe a un cuadrado de 1.000x1.000 metros^2. Cada nodo tiene una cobertura de 250 metros. El tiempo de simulación se fija a 300 segundos. Los resultados obtenidos se derivan promediando (con distintas semillas) 50 repeticiones de cada simulación.

Como protocolos MAC y de *routing* se han elegido IEEE 802.11g y AODV respectivamente, así como el mecanismo RTS/CTS para el envío de paquetes. Esta última asunción es coherente con la propia movilidad de los nodos, dado que el hecho de no emplear la detección por portadora virtual en escenarios de movilidad podría implicar un gran número de colisiones debido al problema de la estación oculta (véase Sección 2.4.1).

El número total de nodos es 25, variando el número de atacantes entre 1 y 3. Los ataques son ejecutados durante todo el tiempo de la simulación y su correspondiente *tasa de ataque* está fijada al 100%, siendo esta tasa la probabilidad de que un nodo atacante realice el ataque.

El número de flujos con tráfico a nivel de aplicación está fijado a 21. Cada flujo consiste en una aplicación `UDPBasicBurst` que simula una conexión CBR con una tasa de envío de 4 paquetes/segundo, teniendo cada paquete un *payload* de 512 bytes. Para cada flujo la dirección destino es elegida aleatoriamente entre todos los nodos legítimos, manteniéndose el mismo destino durante todo el tiempo de la simulación. Los flujos comienzan de forma aleatoria entre 0,5 y 1,5 segundos, y terminan entre 290 y 295 segundos.

Se utiliza el modelo RWP para simular el movimiento de los nodos. La velocidad mínima está fija a 1 m/s y la velocidad máxima varía entre 5 y 20 m/s, con un tiempo de pausa de 15 segundos.

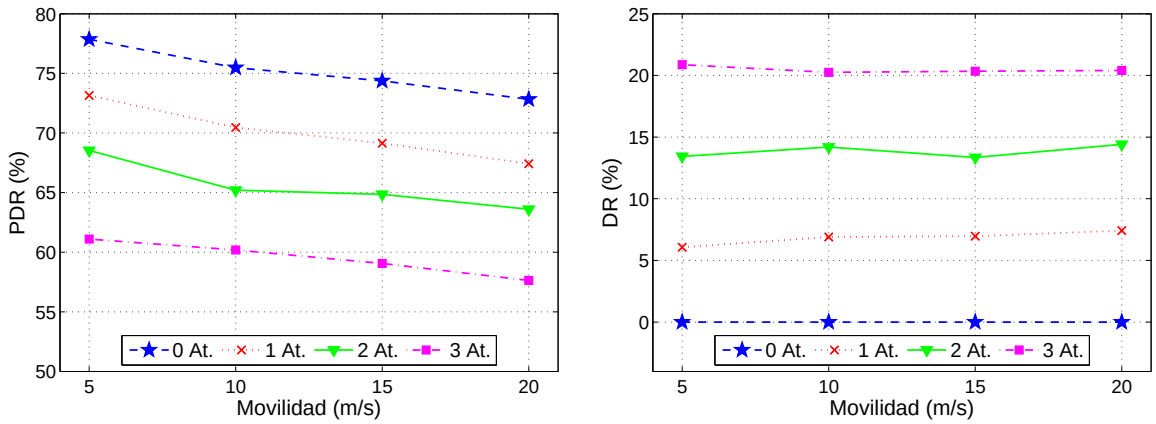
7.5.2. Evaluación del ataque de *dropping*

Para evaluar el funcionamiento del ataque de *dropping* se definen las siguientes métricas:

- PDR (*Packet Delivery Ratio*): número total de paquetes de datos entregados correctamente, dividido por el número total de paquetes de datos enviados.
- DR (*Dropping Ratio*): número total de paquetes de datos perdidos como consecuencia de la ejecución del ataque, dividido por el número total de paquetes de datos transmitidos.

Como puede verse en la Figura 7.2, si el número de atacantes aumenta, PDR se ve deteriorado, mientras que DR crece. Además, puede observarse que PDR decrece con la movilidad, mientras que DR permanece casi constante. Esto es debido a que un aumento en la movilidad implica un incremento en el número de paquetes perdidos por las colisiones y los errores del canal, mientras que el número de paquetes descartados como consecuencia del ataque permanece constante.

En resumen, los resultados experimentales demuestran la correcta implementación del ataque, cuyo impacto se ajusta a lo esperado.



(a) PDR (b) DR

Figura 7.2: PDR (a) y DR (b) en función de la movilidad y del número de atacantes.

7.5.3. Evaluación del ataque de *delay*

Se emplea la siguiente métrica de rendimiento para evaluar el funcionamiento del ataque de *delay*:

- E2ED (*End-to-End Delay*): tiempo medio (en segundos) empleado por un paquete de datos desde el inicio de su transmisión hasta que alcanza el destino, incluyendo todos los posibles retrasos debidos a descubrimiento de rutas, colas, propagación, etc. Se calcula como el promedio de los E2ED de cada paquete en cada flujo, extrayéndose de esta forma el valor medio para toda la red.

Aquí se ha evaluado el ataque de *delay* en función (*i*) del número de atacantes (Figura 7.3a), y (*ii*) del retardo aplicado por el atacante (Figura 7.3b). En el primer caso se añade un retardo fijo de 0,25 segundos, correspondiente al tiempo entre llegadas de la aplicación CBR. Como puede verse en la figura, el *delay* medio aumenta con el número de atacantes. En el segundo caso se fija la movilidad a 5 m/s y se varía el retardo introducido por los atacantes. Los resultados muestran que, incluso introduciendo retardos inferiores al tiempo entre llegadas, esto puede dar lugar a un gran E2ED medio.

En suma, el impacto del ataque corresponde con lo esperado de acuerdo a los resultados obtenidos.

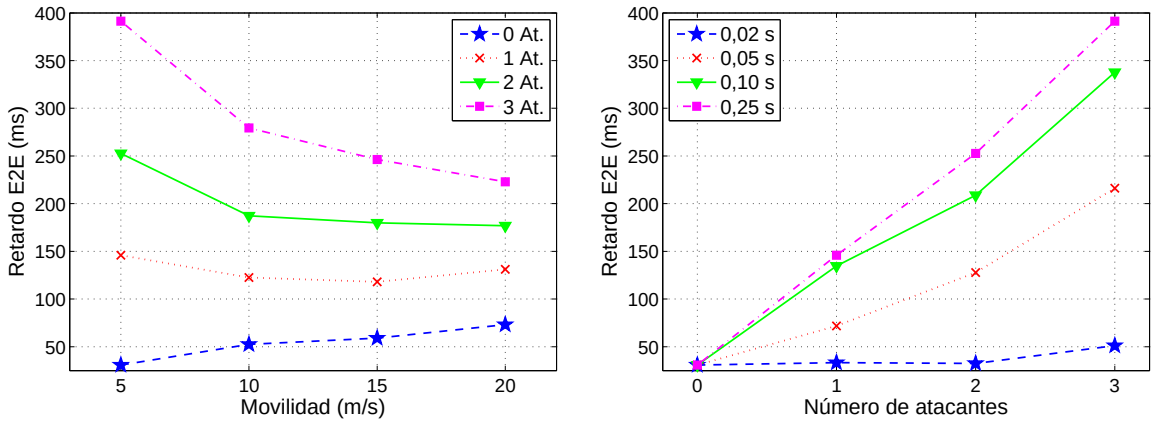


Figura 7.3: E2ED para (a) distintas velocidades y número de atacantes, aplicando un *delay* de 0,25 segundos y (b) aplicando distintos *delays*, con una velocidad fija de 5 m/s.

7.5.4. Evaluación del ataque *sinkhole*

Para caracterizar el rendimiento de los nodos *sinkhole* se define la siguiente métrica:

- AR (*Attraction Ratio*): es la capacidad de atracción de los nodos *sinkhole* respecto de la atracción de los nodos legítimos. Más específicamente, puede verse como la relación entre el número medio de paquetes recibidos por los nodos *sinkhole* y el número medio de paquetes recibidos por los nodos legítimos. AR se calcula cómo:

$$AR = \frac{\frac{1}{S}\sum_{i=1}^{S} pkt_i - \frac{1}{L}\sum_{j=1}^{L} pkt_j}{\frac{1}{L}\sum_{i=1}^{L} pkt_i} \tag{7.1}$$

  siendo S y L el número de nodos *sinkhole* y legítimos, respectivamente, y pkt_i el número total de paquetes recibidos por el nodo N_i.

La Figura 7.4 muestra cómo los nodos *sinkhole* atraen un tráfico superior al resto de nodos. Además, puede observarse que AR decrece a medida que aumenta el número de atacantes. Esto es debido a que los atacantes tienen que competir entre sí para atraer el tráfico, resultando en un menor AR. Sin embargo, el número total de paquetes atraídos por todos los nodos *sinkhole* crece con el número de atacantes.

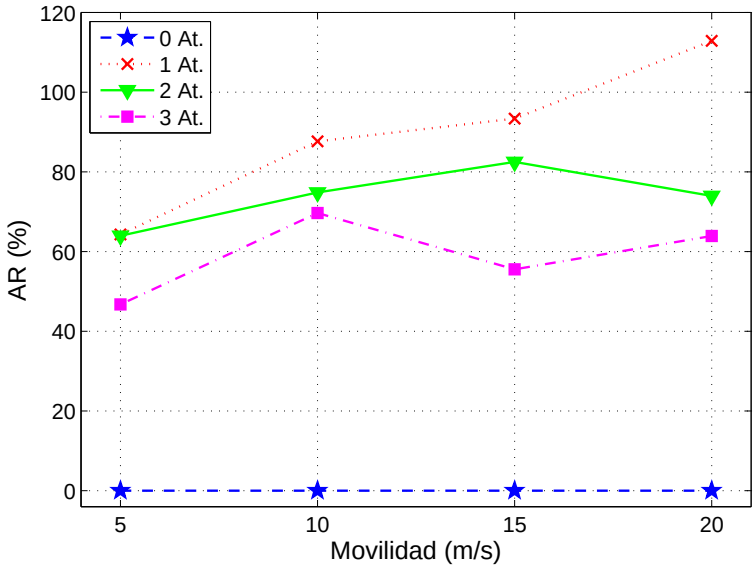


Figura 7.4: AR para distintas velocidades y número de atacantes.

De este modo, los resultados obtenidos de la evaluación de este ataque, así como de los ataques previos, demuestran el correcto funcionamiento de los mismos, lo que confirma las capacidades del *framework*.

7.6. Conclusiones del capítulo

En este trabajo se ha propuesto NETA, un *framework* para la simulación de ataques en redes de comunicación desarrollado sobre INET y el simulador OMNeT++.

NETA consta de tres componentes principales: (*i*) *controladores de ataque*, que gestionan la ejecución de los ataques, (*ii*) *módulos* hackeados, que implementan el comportamiento del ataque, y (*iii*) *mensajes de control*, a través de los que transmitir la información de activación/desactivación, así como información de configuración desde los controladores de ataque a los módulos *hackeados*. Como prueba de concepto, se han implementado tres ataques: *dropping*, *delay* y *sinkhole*.

Se han considerado escenarios de aplicación realistas, analizando una serie de despliegues MANET, de donde los resultados experimentales obtenidos corroboran el funcionamiento correcto de los ataques implementados. Adicionalmente, se ha evaluado cómo afectan los distintos ataques al rendimiento normal de la red.

Cabe destacar que, a raíz de diversos Trabajos Fin de Grado, algunas medidas defensivas ya se encuentran implementadas de forma efectiva en NETA, si bien estas han de ser aún convenientemente evaluadas. En resumen, todo lo obtenido pronostica un futuro prometedor para NETA.

Publicaciones relacionadas

Las publicaciones relacionadas con la temática abordada en este capítulo son las siguientes:

- **L. Sánchez-Casado**, R. A. Rodríguez-Gómez, R. Magán-Carrión y G. Maciá-Fernández. "NETA: Evaluating the effects of NETwork Attacks. MANETs as a case study". *Advances in Security of Information and Communication Networks*, (SecNet), pp. 1-10, 2013.
- **L. Sánchez-Casado**, R. A. Rodríguez-Gómez, R. Magán-Carrión y G. Maciá-Fernández. "NETA: un Framework para Simular y Evaluar Ataques en Redes Heterogéneas. MANETs como Caso de Estudio". *XI Jornadas de Ingeniería Telemática* (JITEL), pp. 487-492, 2013.

Conclusiones
Y
Trabajo Futuro

Capítulo 8

Conclusiones y trabajo futuro

En el presente capítulo se resumen las principales conclusiones extraídas tras la realización del trabajo completo de tesis aquí expuesto. Estas se han ido detallando de forma concreta en cada uno de los capítulos previos, presentándose en este de forma unificada y sintética. Adicionalmente, se describen brevemente los principales trabajos futuros a abordar en la línea de investigación desarrollada.

8.1. Conclusiones

Las redes MANET han evolucionado considerablemente en los últimos años, dando lugar a la aparición de distintas tecnologías, arquitecturas o aplicaciones relacionadas con ellas. Sin embargo, a causa de sus características inherentes, el problema de la seguridad en este tipo de entornos sigue siendo un reto que debe ser tratado de forma adecuada. Un primer paso para proporcionar seguridad en estas redes consiste en **estudiar y clasificar de forma apropiada las amenazas de seguridad actuales**. En relación con este aspecto podemos resaltar las siguientes conclusiones del trabajo realizado:

- Se ha introducido una nueva taxonomía para clasificar los ataques a la seguridad en redes MANET, con el propósito de organizar y clasificar estas amenazas desde una perspectiva práctica, permitiendo el diseño y desarrollo de mecanismos defensivos más flexibles y eficientes.

- Se han presentado también los principales fundamentos de comunicación en redes MANET, proporcionándose detalles acerca del funcionamiento de dos de los protocolos más ampliamente utilizados para la capa MAC y de red, IEEE 802.11 y AODV, respectivamente.
- Se ha descrito también el funcionamiento e implementación específicas de los ataques de *dropping* y *sinkhole*.

Tras este necesario *background*, se ha discutido la necesidad de desplegar **nuevos esquemas de detección para la protección de redes MANET** contra los citados ataques de seguridad. Las principales conclusiones del presente trabajo de tesis en relación con la detección de ataques se resumen a continuación:

- Se ha propuesto un modelo analítico para representar el proceso de retransmisión seguido por los nodos en entornos MANET. Este modelo incluye de forma nativa las distintas circunstancias que pueden dar lugar a descartes legítimos de paquetes, como colisiones, errores en el canal o situaciones de movilidad, permitiendo distinguir entre dichas causas legítimas y comportamientos maliciosos de descarte (*dropping*).
- En base al modelo analítico propuesto, y siguiendo una metodología multicapa, se ha desarrollado un IDS para la detección de ataques de *dropping*. Para ello se calcula una heurística sencilla que permite reconocer este tipo de comportamientos maliciosos en redes MANET.
- Para la recopilación de los parámetros se ha implementado un enventanado basado en eventos, en lugar de utilizar de la aproximación tradicional basada en ventanas temporales. De este modo, los parámetros empleados para la detección se obtienen en ventanas no solapadas de P paquetes de datos recibidos.
- El IDS propuesto puede ser desplegado en base a dos implementaciones, que difieren en el modo en el que se realiza la recopilación de los parámetros de red empleados para la detección: una aproximación local autónoma y otra distribuida.
- Se ha llevado a cabo una exhaustiva experimentación para evaluar las capacidades de detección del sistema propuesto. Distintas pruebas han sido realizadas bajo diferentes condiciones, verificándose los prometedores resultados de nuestra aproximación, que mejoran en cualquier caso los obtenidos por otros esquemas existentes.
- En relación con el ataque *sinkhole*, se ha probado la existencia de los denominados "bordes de contaminación", *i.e.*, nodos legítimos bajo la influencia de un nodo atacante que son, al mismo tiempo, vecinos de otros nodos legítimos no contaminados.

- Se ha mostrado que la información de *routing* en estos nodos frontera es más inconsistente y, por tanto, se comporta de forma anómala. Recopilando y analizando su propia información, así como aquella perteneciente a sus vecinos, estos "bordes de contaminación" pueden detectar incoherencias y, en consecuencia, determinar la existencia de ataques *sinkhole*.
- En base a ello se ha propuesto un IDS colaborativo en dos fases: la primera consistente en un proceso de pre-detección local que, en caso de dar un resultado positivo, lanza una segunda fase en la que se ejecuta un mecanismo cooperativo que recopila información de la vecindad para detectar ataques *sinkhole* de forma precisa.
- Se han propuesto dos posibles aproximaciones para llevar a cabo la comunicación inter-nodal necesaria para intercambiar los parámetros de detección entre los distintos nodos. La primera opción emplea los propios mensajes de AODV, siendo esta una solución más sencilla pero que introduce un mayor *overhead* en términos de mensajes transmitidos. La segunda aproximación utiliza mensajes específicamente diseñados, lo que implica la modificación del protocolo de encaminamiento pero también resulta en un menor número de mensajes de control transmitidos.
- Se han realizado distintas pruebas bajo diferentes condiciones para evaluar el correcto funcionamiento del esquema de detección propuesto. Los resultados obtenidos confirman las excelentes capacidades de nuestro IDS.

En relación con la **integración de soluciones de seguridad**, se destacan las siguientes conclusiones:

- Se ha contribuido con un nuevo protocolo para la notificación de eventos de seguridad en entornos MANET, que puede ser empleado como interfaz entre los diferentes módulos defensivos.
- Se han propuesto tres posibles usos para el mencionado protocolo. El primero permite notificar la ocurrencia de incidentes de seguridad a toda la red mediante un mecanismo de difusión. La segunda aplicación propuesta tiene como propósito el intercambio de información de seguridad entre las distintas entidades o módulos de defensa, permitiendo el despliegue de aproximaciones de detección y/o respuesta distribuidas. En este caso se emplean solicitudes *broadcast* enviadas a los nodos vecinos y respuestas *unicast*. Dicha aplicación ha sido utilizada para el intercambio de información necesario en el proceso de detección colaborativa de ataques *sinkhole*. La tercera aplicación permite la notificación de forma asíncrona de información de seguridad relevante.

- Un breve análisis de los requisitos involucrados evidencia que, al menos de forma teórica, el protocolo desarrollado es apto para su uso en entornos MANET.
- Se ha contribuido también el desarrollo de NETA, un *framework* para la simulación de ataques y defensas en redes MANET. El marco de simulación está desarrollado sobre el *framework* INET y sobre el simulador OMNeT++.
- La arquitectura de NETA se basa en tres componentes distintos: *controladores de ataques*, *mensajes de control* y *módulos hackeados*. Su flexibilidad y versatilidad lo hace especialmente apropiado para la evaluación de diferentes ataques y soluciones defensivas bajo las mismas condiciones.
- Como prueba de concepto se han implementado tres ataques: ataques de *dropping* en IP, ataques de *delay* en IP y ataques *sinkhole* en AODV. Se ha evaluado el correcto funcionamiento de los mismos, demostrando los resultados obtenidos la usabilidad y capacidades de NETA.
- Aunque NETA se encuentra actualmente en un estado todavía preliminar, se espera que se convierta en una herramienta útil para la comunidad investigadora en el campo de la seguridad en redes de comunicación. En esta línea, ya se encuentran integrados también algunos mecanismos defensivos.

8.2. Líneas de trabajo futuro

Las principales líneas de trabajo futuro que se extraen del presente documento se enumeran, desde las más específicas hasta las más generales, a continuación:

1. Implementar métodos efectivos para la determinación adaptativa y dinámica de los umbrales de detección considerados, dado que sus valores son fuertemente dependientes de las condiciones específicas de la red.
2. Explorar la posibilidad de incluir mecanismos de confianza o reputación en nuestras propuestas de detección. Dichos esquemas pueden ser aplicados como mecanismos de prevención o de respuesta, permitiendo la realimentación entre las tres líneas de defensa tradicionales.
3. En relación con el punto anterior, incluir, cuando sea posible y aplicable, mecanismos para tratar con ataques en confabulación, es decir, aquellas situaciones en las que dos o más atacantes cooperan para evadir el proceso de detección o para incrementar el impacto del ataque ejecutado en la red. En estas circunstancias, los citados mecanismos basados en confianza o reputación aparecen como una solución razonable.

4. Extender nuestras aproximaciones de detección a otros protocolos distintos de AODV. Puesto que la aproximación de detección de ataques de *dropping* se fundamenta en un modelo analítico para el proceso de retransmisión en redes MANET, el protocolo de encaminamiento subyacente no debería constituir una gran limitación. En relación con el esquema propuesto para la detección de ataques *sinkhole*, se ha mostrado que gran parte de los protocolos de encaminamiento emplean algún tipo de identificador similar a los números de secuencia usados por AODV, que podrán ser utilizados como parámetros base para la detección.

5. Desarrollar aproximaciones de detección holísticas en vez de esquemas específicos en función del ataque, mediante la extracción de características comunes, independientes de la función objetivo atacada (*especies*). Este paradigma permitiría, presumiblemente, la definición de sistemas de detección más completos y efectivos.

6. Incorporar la implementación real del protocolo de notificación al *framework* NETA y evaluar sus prestaciones reales, con el objetivo de confirmar su aplicabilidad en entornos MANET.

7. Unificar, en un único sistema de detección, las diferentes contribuciones propuestas en este trabajo de tesis (ataques de *dropping* y *sinkhole*, los detectores para ambos ataques, el protocolo de comunicación, etc.). La implementación del *framework* NETA constituye un gran progreso para alcanzar este objetivo.

8. Continuar con el desarrollo de NETA mediante la inclusión de nuevos ataques y técnicas defensivas.

9. Aplicar y validar todas las técnicas desarrolladas a escenarios reales, tales como entornos VANET o despliegues para la gestión de crisis.

Bibliografía

[1] T. S. Rappaport, A. Annamalai, R. M. Buehrer, and W. H. Tranter, "Wireless Communications: Past Events and a Future Perspective," *IEEE Communications Magazine*, vol. 40, pp. 148–161, May. 2002.

[2] J. Jubin and J. D. Tornow, "The DARPA packet radio network protocols," *Proceedings of the IEEE*, vol. 75, pp. 21–32, Ene. 1987.

[3] MANET IETF. [Online; Accedido 18 Julio 2014] `http://datatracker.ietf.org/wg/manet/charter`.

[4] J. He, S. Ji, Y. Pan, and Y. Li, eds., *Wireless ad-hoc and Sensor Networks: Management, Performance, and Applications*. Boca Raton, FL: CRC Press, 2014.

[5] K. I. Lakhtaria, ed., *Technological Advancements and Applications in Mobile Ad-Hoc Networks: Research Trends*. IGI Global, 2012.

[6] R. R. Roy, "Mobile Ad Hoc Networks," in *Handbook of Mobile Ad Hoc Networks for Mobility Models*, ch. 1, pp. 3–22, Springer US, Abr. 2011.

[7] I. Bekmezci, O. K. Sahingoz, and S. Temel, "Flying Ad-Hoc Networks (FANETs): A survey," *Ad Hoc Networks*, vol. 11, pp. 1254–1270, May. 2013.

[8] R. Beyah, J. McNair, and C. Corbett, *Security in Ad-hoc and Sensor Networks*. Hackensack, NJ: World Scientific, 2010.

[9] P. García-Teodoro, L. Sánchez-Casado, and G. Maciá-Fernández, "Taxonomy and Holistic Detection of Security Attacks in MANETs," in *Security for Multihop Wireless Networks* (S. Khan and J. Lloret Mauri, eds.), pp. 1–12, CRC Press, Abr. 2014.

[10] Norton Symantec, "Norton Study Calculates Cost of Global Cybercrime: $114 Billion Annually," 2011. [Online; Accedido 18 Julio 2014] `http://www.symantec.com/about/news/release/article.jsp?prid=20110907_02`.

[11] H. Bigdoli, ed., *Book of Information Security. Threats, Vulnerabilities, Prevention, Detection, and Management*, vol. 3. John Wiley & Sons, 2006.

[12] Y.-A. Huang and W. Lee, "Attack analysis and detection for ad hoc routing protocols," in *Recent Advances in Intrusion Detection* (E. Jonsson, A. Valdes, and M. Almgren, eds.), vol. 3224 of *Lecture Notes in Computer Science*, pp. 125–145, Springer Berlin Heidelberg, 2004.

[13] P. Brutch and C. Ko, "Challenges in Intrusion Detection for Wireless Ad-Hoc Networks," in *Proc. of the Symposium on Applications and the Internet Workshops (SAINT)*, pp. 368–373, Ene. 2003.

[14] A. Nadeem and M. P. Howarth, "A Survey of MANET Intrusion Detection & Prevention Approaches for Network Layer Attacks," *IEEE Communications Surveys & Tutorials*, vol. 15, pp. 2027–2045, Nov. 2013.

[15] C. E. Perkins, E. M. Belding-Royer, and S. R. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing," *IETF, RFC 3561*, Jul. 2003. [Online; Accedido 18 Julio 2014]
`http://www.rfc-editor.org/rfc/rfc3561.txt`.

[16] D. Martins and H. Guyennet, "Wireless Sensor Network Attacks and Security Mechanisms: A Short Survey," in *Proc. of the 13th International Conference on Network-Based Information Systems (NBiS)*, pp. 313–320, Sep. 2010.

[17] B. Kannhavong, H. Nakayama, Y. Nemoto, N. Kato, and A. Jamalipour, "A survey of routing attacks in mobile ad hoc networks," *IEEE Wireless Communications*, vol. 14, pp. 85–91, Oct. 2007.

[18] B. Wu, J. Chen, J. Wu, and M. Cardei, "A Survey of Attacks and Countermeasures in Mobile Ad Hoc Networks," in *Wireless Network Security* (Y. Xiao, X. Shen, and D.-Z. Du, eds.), Signals and Communication Technology, ch. 5, pp. 103–135, Springer US, 2007.

[19] R. P. Ankala, D. Kavitha, and D. Haritha, "Mobile Agent Based Routing in MANETs - Attacks & Defences," *Network Protocols and Algorithms*, vol. 3, pp. 108–121, Dic. 2011.

[20] K. Sahadevaiah and P. Reddy, "Impact of security attacks on a new security protocol for mobile ad hoc networks," *Network Protocols and Algorithms*, vol. 3, pp. 122–140, Dic. 2011.

[21] M. S. Alkatheiri, J. Liu, and A. R. Sangi, "AODV routing protocol under several routing attacks in MANETs," in *Proc. of the 13th IEEE International Conference on Communication Technology (ICCT)*, pp. 614–618, Sep. 2011.

[22] M. Salehi, H. Samavati, and M. Dehghan, "Performance assessment of OLSR protocol under routing attacks," in *Proc. of the International Conference for Internet Technology and Secured Transactions (ICITST)*, pp. 791–796, Dic. 2011.

[23] M. Salehi and H. Samavati, "DSR vs OLSR: Simulation Based Comparison of Ad Hoc Reactive and Proactive Algorithms under the Effect of New Routing Attacks," in *Proc. of the 6th International Conference on Next Generation Mobile Applications, Services and Technologies (NGMAST)*, pp. 100–105, Sep. 2012.

[24] A. El-Mousa and A. Suyyagh, "Ad Hoc networks security challenges," in *Proc. of the 7th International Multi-Conference on Systems Signals and Devices (SSD)*, pp. 1–6, Jun. 2010.

[25] R. Mishra, S. Sharma, and R. Agrawal, "Vulnerabilities and security for ad-hoc networks," in *Proc. of the International Conference on Networking and Information Technology (ICNIT)*, pp. 192–196, Jun. 2010.

[26] P. M. Jawandhiya and M. M. Ghonge, "A Survey of Mobile Ad Hoc Network Attacks," *International Journal of Engineering Science and Technology (IJEST)*, vol. 2, pp. 4063–4071, Sep. 2010.

[27] A. A. Cardenas, T. Roosta, and S. Sastry, "Rethinking Security Properties, Threat Models, and the Design Space in Sensor Networks: A Case Study in SCADA Systems," *Ad Hoc Networks*, vol. 7, pp. 1434–1447, Nov. 2009.

[28] A. K. Jain and V. Tokekar, "Classification of Denial of Service Attacks in Mobile Ad Hoc Networks," in *Proc. of the International Conference on Computational Intelligence and Communication Networks (CICN)*, pp. 256–261, Oct. 2011.

[29] X. Liao, D. Hao, and K. Sakurai, "Classification on attacks in wireless ad hoc networks: A game theoretic view," in *Proc. of the 7th International Conference on Networked Computing and Advanced Information Management (NCM)*, pp. 144–149, Jun. 2011.

[30] Y. Yu, K. Li, W. Zhou, and P. Li, "Trust mechanisms in wireless sensor networks: Attack analysis and countermeasures," *Journal of Network and Computer Applications*, vol. 35, pp. 867–880, May. 2012.

[31] "IEEE Standard for Information Technology – Local and Metropolitan Area Networks – Specific Requirements – Part 15.1: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Wireless Personal Area Networks (WPANs)," *IEEE Std 802.15.1-2005 (Revision of IEEE Std 802.15.1-2002)*, pp. 1–580, Ago. 2005.

[32] "IEEE Standard for Local and Metropolitan Area Networks – Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)," *IEEE Std 802.15.4-2011 (Revision of IEEE Std 802.15.4-2006)*, pp. 1–314, Sep. 2011.

[33] "IEEE Standard for Local and Metropolitan Area Networks – Part 16: Air Interface for Broadband Wireless Access Systems," *IEEE Std 802.16-2012 (Revision of IEEE Std 802.16-2009)*, pp. 1–2542, Ago. 2012.

[34] "IEEE Standard for Information Technology – Local and metropolitan area networks – Specific requirements – Part 22: Cognitive Wireless RAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Policies and procedures for operation in the TV Bands," *IEEE Std 802.22-2011*, pp. 1–680, Jul. 2011.

[35] "IEEE Standard for Information Technology – Local and metropolitan area networks – Specific requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *IEEE Std 802.11-2012 (Revision of IEEE Std 802.11-2007)*, pp. 1–2793, Mar. 2012.

[36] E. M. Royer and C.-K. Toh, "A review of current routing protocols for ad hoc mobile wireless networks," *IEEE Personal Communications*, vol. 6, pp. 46–55, Abr. 1999.

[37] M. Abolhasan, T. Wysocki, and E. Dutkiewicz, "A review of routing protocols for mobile ad hoc networks," *Ad Hoc Networks*, vol. 2, pp. 1–22, Ene. 2004.

[38] B. Renu, M. H. lal, and T. Pranavi, "Routing Protocols in Mobile Ad-Hoc Network: A Review," in *Quality, Reliability, Security and Robustness in Heterogeneous Networks* (K. Singh and A. Awasthi, eds.), vol. 115 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pp. 52–60, Springer Berlin Heidelberg, 2013.

[39] T. Clausen and P. Jacquet, "Optimized Link State Routing Protocol (OLSR)," *IETF, RFC 3626*, Oct. 2003. [Online; Accedido 18 Julio 2014] `http://www.rfc-editor.org/rfc/rfc3626.txt`.

[40] C. E. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," in *Proc. of the Conference on Communications Architectures, Protocols and Applications (SIGCOMM)*, pp. 234–244, Ago. 1994.

[41] S. Murthy and J. J. Garcia-Luna-Aceves, "An Efficient Routing Protocol for Wireless Networks," *Mobile Networks & Applications*, vol. 1, pp. 183–197, Oct. 1996.

[42] D. B. Johnson, Y. Hu, and D. A. Maltz, "The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4," *IETF, RFC 4728*, Feb. 2007. [Online; Accedido 18 Julio 2014] `http://www.rfc-editor.org/rfc/rfc4728.txt`.

[43] I. D. Chakeres and C. E. Perkins, "Dynamic MANET On-demand (AODVv2) Routing," *IETF Draft*, Feb. 2014. Work in progress [Online; Accedido 18 Julio 2014]
http://tools.ietf.org/html/draft-ietf-manet-aodvv2-03.

[44] Z. J. Haas and M. R. Pearlman, "The Zone Routing Protocol (ZRP) for Ad Hoc Networks," *IETF Draft*, Jul. 2002. Expired [Online; Accedido 18 Julio 2014]
http://tools.ietf.org/html/draft-ietf-manet-zone-zrp-04.

[45] M. Jiang, J. Li, and Y. C. Tay, "Cluster Based Routing Protocol (CBRP) Functional Specification," *IETF Draft*, Ago. 1999. Expired [Online; Accedido 18 Julio 2014]
http://tools.ietf.org/html/draft-ietf-manet-cbrp-spec-01.

[46] X. Hong, K. Xu, and M. Gerla, "Scalable routing protocols for mobile ad hoc networks," *IEEE Network*, vol. 16, pp. 11–21, Jul. 2002.

[47] A. Boukerche, B. Turgut, N. Aydin, M. Z. Ahmad, L. Bölöni, and D. Turgut, "Routing protocols in Ad Hoc networks: A survey," *Computer Networks*, vol. 55, pp. 3032–3080, Sep. 2011.

[48] C. Liu and J. Kaiser, "A survey of Mobile Ad Hoc network Routing Protocols," Tech. Rep. Nr. 2003-2008, The University of Magdeburg; the University of Ulm, Oct. 2005.

[49] M. Gerla, G. Pei, S.-J. Lee, and C.-C. Chiang, "On-Demand Multicast Routing Protocol (ODMRP) for Ad-Hoc Networks," *IETF Draft*, Mar. 2014. Work in progress [Online; Accedido 18 Julio 2014]
http://tools.ietf.org/html/draft-gerla-manet-odmrp-02.

[50] E. M. Royer and C. Perkins, "Multicast Ad Hoc On-Demand Distance Vector (MAODV) Routing," *IETF Draft*, Jul. 2000. Work in progress [Online; Accedido 18 Julio 2014]
http://tools.ietf.org/html/draft-ietf-manet-maodv-00.

[51] L. Sánchez-Casado, R. Magán-Carrión, P. García-Teodoro, and J. E. Díaz-Verdejo, "Defenses Against Packet Dropping Attacks in Wireless Multihop Ad Hoc Networks," in *Security for Multihop Wireless Networks* (S. Khan and J. Lloret Mauri, eds.), pp. 377–400, CRC Press, Abr. 2014.

[52] S. Djahel, F. Nait-abdesselam, and Z. Zhang, "Mitigating Packet Dropping Problem in Mobile Ad Hoc Networks: Proposals and Challenges," *IEEE Communications Surveys & Tutorials*, vol. 13, pp. 658–672, Nov. 2011.

[53] S. Dokurer, Y. M. Erten, and C. E. Acar, "Performance analysis of ad-hoc networks under black hole attacks," in *Proc. of the IEEE SoutheastCon*, pp. 148–153, Mar. 2007.

[54] S. M. Bo, H. Xiao, A. Adereti, J. A. Malcolm, and B. Christianson, "A Performance Comparison of Wireless Ad Hoc Network Routing Protocols under Security Attack," in *Proc. of the 3rd International Symposium on Information Assurance and Security (IAS)*, pp. 50–55, Ago. 2007.

[55] M. Parsons and P. Ebinger, "Performance Evaluation of the Impact of Attacks On Mobile Ad hoc Networks," in *Proc. of the 28th IEEE International Symposium on Reliable Distributed Systems (SRDS)*, Sep. 2009.

[56] H. L. Nguyen and U. T. Nguyen, "A study of different types of attacks in mobile ad hoc networks," in *Proc. of the 25th IEEE Canadian Conference on Electrical Computer Engineering (CCECE)*, pp. 1–6, Abr. 2012.

[57] L. M. T. Harb, M. Tantawy, and M. Elsoudani, "Performance of mobile ad hoc networks under attack," in *Proc. of the International Conference on Computer Applications Technology (ICCAT)*, pp. 1–8, Ene. 2013.

[58] L. Sánchez-Casado, G. Maciá-Fernández, and P. García-Teodoro, "An Efficient Cross-Layer Approach for Malicious Packet Dropping Detection in MANETs," in *Proc. of the IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pp. 231–238, Jun. 2012.

[59] R. Raghuvanshi, R. Kaushik, and J. Singhai, "A review of misbehaviour detection and avoidance scheme in ad hoc network," in *Proc. of the International Conference on Emerging Trends in Networks and Computer Communications (ETNCC)*, pp. 301–306, Abr. 2011.

[60] Y.-C. Hu, A. Perrig, and D. B. Johnson, "Ariadne: A Secure On-demand Routing Protocol for Ad Hoc Networks," in *Proc. of the 8th Annual International Conference on Mobile Computing and Networking (MobiCom)*, pp. 12–23, Sep. 2002.

[61] A. Perrig, R. Canetti, J. D. Tygar, and D. Song, "The TESLA Broadcast Authentication Protocol," *RSA CryptoBytes*, vol. 5, pp. 2–13, Nov. 2002.

[62] K. Sanzgiri, B. Dahill, B. N. Levine, C. Shields, and E. M. Belding-Royer, "A secure routing protocol for ad hoc networks," in *Proc. of the 10th IEEE International Conference on Network Protocols (ICNP)*, pp. 78–87, Nov. 2002.

[63] M. G. Zapata and N. Asokan, "Securing Ad Hoc Routing Protocols," in *Proc. of the 1st ACM Workshop on Wireless Security (WiSE)*, pp. 1–10, Sep. 2002.

[64] D. Cerri and A. Ghioni, "Securing AODV: The A-SAODV Secure Routing Prototype," *IEEE Communications Magazine*, vol. 46, pp. 120–125, Feb. 2008.

[65] Y.-C. Hu, D. B. Johnson, and A. Perrig, "SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad Hoc Networks," in *Proc. of the 4th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA)*, p. 3, Jun. 2002.

[66] P. Papadimitratos and Z. J. Haas, "Secure Routing for Mobile Ad Hoc Networks," in *Proc. of the SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS)*, pp. 193–204, Ene. 2002.

[67] S. Yi, P. Naldurg, and R. Kravets, "Security-aware Ad Hoc Routing for Wireless Networks," in *Proc. of the 2nd ACM International Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc)*, pp. 299–302, Oct. 2001.

[68] S. Lu, L. Li, K.-Y. Lam, and L. Jia, "SAODV: A MANET Routing Protocol that can Withstand Black Hole Attack," in *Proc. of the International Conference on Computational Intelligence and Security (CIS)*, vol. 2, pp. 421–425, Dic. 2009.

[69] H. Deng, W. Li, and D. P. Agrawal, "Routing Security in Wireless Ad Hoc Networks," *IEEE Communications Magazine*, vol. 40, pp. 70–75, Oct. 2002.

[70] S. Lee, B. Han, and M. Shin, "Robust Routing in Wireless Ad Hoc Networks," in *Proc. of the 31st International Conference on Parallel Processing Workshops (ICPPW)*, pp. 73–78, Ago. 2002.

[71] C. Song and Q. Zhang, "OMH – Suppressing Selfish Behavior in Ad Hoc Networks with One More Hop," *Mobile Networks and Applications*, vol. 14, pp. 178–187, Abr. 2009.

[72] J. Liu and V. Issarny, "Enhanced Reputation Mechanism for Mobile Ad Hoc Networks," in *Trust Management* (C. Jensen, S. Poslad, and T. Dimitrakos, eds.), vol. 2995 of *Lecture Notes in Computer Science*, pp. 48–62, Springer Berlin Heidelberg, 2004.

[73] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating Routing Misbehavior in Mobile Ad Hoc Networks," in *Proc. of the 6th Annual International Conference on Mobile Computing and Networking (MobiCom)*, pp. 255–265, Ago. 2000.

[74] P. Michiardi and R. Molva, "Core: A Collaborative Reputation Mechanism to Enforce Node Cooperation in Mobile Ad Hoc Networks," in *Proc. of the IFIP 6th Joint Working Conference on Communications and Multimedia Security (CMS): Advanced Communications and Multimedia Security*, pp. 107–121, Sep. 2002.

[75] H. Miranda and L. Rodrigues, "Friends and Foes: preventing selfishness in open mobile ad hoc networks," in *Proc. of the 23rd International Conference on Distributed Computing Systems Workshops (ICDCSW)*, pp. 440–445, May. 2003.

[76] Q. He, D. Wu, and P. Khosla, "SORI: a secure and objective reputation-based incentive scheme for ad-hoc networks," in *Proc. of the IEEE Wireless Communications and Networking Conference (WCNC)*, vol. 2, pp. 825–830, Mar. 2004.

[77] M. Medadian, M. H. Yektaie, and A.-M. Rahmani, "Combat with Black hole attack in AODV routing protocol in MANET," in *Proc. of the 1st Asian Himalayas International Conference on Internet (AH-ICI)*, pp. 1–5, Nov. 2009.

[78] J. Wang, Y. Liu, and Y. Jiao, "Building a trusted route in a mobile ad hoc network considering communication reliability and path length," *Journal of Network and Computer Applications*, vol. 34, pp. 1138–1149, Jul. 2011.

[79] J. Miao, O. Hasan, S. B. Mokhtar, L. Brunie, and K. Yim, "An Analysis of Strategies for Preventing Selfish Behavior in Mobile Delay Tolerant Networks," in *Proc. of the 6th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS)*, pp. 208–215, Jul. 2012.

[80] L. Buttyán, L. Dóra, M. Félegyházi, and I. Vajda, "Barter trade improves message delivery in opportunistic networks," *Ad Hoc Networks*, vol. 8, pp. 1–14, Ene. 2010.

[81] S. Zhong, J. Chen, and Y. R. Yang, "Sprite: a simple, cheat-proof, credit-based system for mobile ad-hoc networks," in *Proc. of the 22nd Annual Joint Conference of the IEEE Computer and Communications (INFOCOM)*, vol. 3, pp. 1987–1997, Mar. 2003.

[82] H. Janzadeh, K. Fayazbakhsh, M. Dehghan, and M. S. Fallah, "A Secure Credit-based Cooperation Stimulating Mechanism for MANETs Using Hash Chains," *Future Generation Computer Systems*, vol. 25, pp. 926–934, Sep. 2009.

[83] B. B. Chen and M. C. Chan, "MobiCent: a Credit-Based Incentive System for Disruption Tolerant Network," in *Proc. of the 29th Annual Joint Conference of the IEEE Computer and Communications (INFOCOM)*, pp. 1–9, Mar. 2010.

[84] P. García-Teodoro, J. E. Díaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based Network Intrusion Detection: Techniques, systems and Challenges," *Computers & Security*, vol. 28, pp. 18–28, Mar. 2009.

[85] D. Djenouri and N. Badache, "New approach for selfish nodes detection in mobile ad hoc networks," in *Proc. of the Workshop 1st International Conference on Security and Privacy for Emerging Areas in Communication Networks (SecurComm)*, pp. 288–294, Sep. 2005.

[86] D. Djenouri, N. Ouali, A. Mahmoudi, and N. Badache, "Random Feedbacks for Selfish Nodes Detection in Mobile Ad Hoc Networks," in *Operations and*

Management in IP-Based Networks (T. Magedanz, E. Madeira, and P. Dini, eds.), vol. 3751 of *Lecture Notes in Computer Science*, pp. 68–75, Springer Berlin Heidelberg, 2005.

[87] K. Balakrishnan, J. Deng, and P. K. Varshney, "TWOACK: preventing selfishness in mobile ad hoc networks," in *Proc. of the IEEE Wireless Communications and Networking Conference (WCNC)*, vol. 4, pp. 2137–2142, Mar. 2005.

[88] K. Liu, J. Deng, P. K. Varshney, and K. Balakrishnan, "An Acknowledgment-Based Approach for the Detection of Routing Misbehavior in MANETs," *IEEE Transactions on Mobile Computing*, vol. 6, pp. 536–550, May. 2007.

[89] L. Tamilselvan and V. Sankaranarayanan, "Prevention of Co-operative Black Hole Attack in MANET," *Journal of Networks*, vol. 3, pp. 13–20, May. 2008.

[90] S. Djahel, F. Nait-Abdesselam, and A. A. Khokhar, "An Acknowledgment-Based Scheme to Defend Against Cooperative Black Hole Attacks in Optimized Link State Routing Protocol," in *Proc. of the IEEE International Conference on Communications (ICC)*, pp. 2780–2785, May. 2008.

[91] D. Djenouri and N. Badache, "On eliminating packet droppers in MANET: A modular solution," *Ad Hoc Networks*, vol. 7, pp. 1243–1258, Ago. 2009.

[92] A. Baadache and A. Belmehdi, "Fighting against packet dropping misbehavior in multi-hop wireless ad hoc networks," *Journal of Network and Computer Applications*, vol. 35, pp. 1130–1139, May. 2012.

[93] K. Vishnu and A. J. Paul, "Detection and Removal of Cooperative Black/Gray hole attack in Mobile Ad Hoc Networks," *International Journal of Computer Applications*, vol. 1, no. 22, pp. 38–42, 2010.

[94] J.-M. Chang, P.-C. Tsou, H.-C. Chao, and J.-L. Chen, "CBDS: A Cooperative Bait Detection Scheme to Prevent Malicious Node for MANET Based on Hybrid Defense Architecture," in *Proc. of the 2nd International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace Electronic Systems Technology (Wireless VITAE)*, pp. 1–5, Feb. 2011.

[95] S. Buchegger and J.-Y. Le Boudec, "Performance Analysis of the CONFIDANT Protocol," in *Proc. of the 3rd ACM International Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc)*, pp. 226–236, Jun. 2002.

[96] C. Basile, Z. T. Kalbarczyk, and R. K. Iyer, "Inner-Circle Consistency for Wireless Ad Hoc Networks," *IEEE Transactions on Mobile Computing*, vol. 6, pp. 39–55, Ene. 2007.

[97] Y. Zhang, W. Lee, and Y. A. Huang, "Intrusion Detection Techniques for Mobile Wireless Networks," *Wireless Networks*, vol. 9, pp. 545–556, Sep. 2003.

[98] Y. Huang, W. Fan, W. Lee, and P. S. Yu, "Cross-Feature Analysis for Detecting Ad-Hoc Routing Anomalies," in *Proc. of the 23rd IEEE International Conference on Distributed Computing Systems (ICDCS)*, pp. 478–487, May. 2003.

[99] S. Bose, S. Bharathimurugan, and A. Kannan, "Multi-Layer Integrated Anomaly Intrusion Detection System for Mobile Adhoc Networks," in *Proc. of the International Conference on Signal Processing, Communications and Networking (ICSCN)*, pp. 360–365, Feb. 2007.

[100] J. F. C. Joseph, A. Das, B.-C. Seet, and B.-S. Lee, "CRADS: Integrated Cross Layer Approach for Detecting Routing Attacks in MANETs," in *Proc. of the IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1525–1530, Mar. 2008.

[101] J. F. C. Joseph, B.-S. Lee, A. Das, and B.-C. Seet, "Cross-Layer Detection of Sinking Behavior in Wireless Ad Hoc Networks Using SVM and FDA," *IEEE Transactions on Dependable and Secure Computing*, vol. 8, pp. 233–245, Mar. 2011.

[102] W. Shim, G. Kim, and S. Kim, "A distributed sinkhole detection method using cluster analysis," *Expert Systems with Applications*, vol. 37, pp. 8486–8491, Dic. 2010.

[103] Y. F. Alem and Z. C. Xuan, "Preventing Black Hole Attack in Mobile Ad-Hoc Networks using Anomaly Detection," in *Proc. of the 2nd International Conference on Future Computer and Communication (ICFCC)*, vol. 3, pp. 672–676, May. 2010.

[104] T. Hayajneh, P. Krishnamurthy, D. Tipper, and T. Kim, "Detecting Malicious Packet Dropping in the Presence of Collisions and Channel Errors in Wireless Ad Hoc Networks," in *Proc. of the IEEE International Conference on Communications (ICC)*, pp. 1–6, Jun. 2009.

[105] S. Kurosawa, H. Nakayama, N. Kato, A. Jamalipour, and Y. Nemoto, "Detecting Blackhole Attack on AODV-based Mobile Ad Hoc Networks by Dynamic Learning Method," *International Journal of Network Security*, vol. 5, pp. 338–346, Nov. 2007.

[106] P. N. Raj and P. B. Swadas, "DPRAODV: A Dynamic Learning System against Blackhole Attack in AODV based MANET," *International Journal of Computer Science Issues*, vol. 2, pp. 54–59, Ago. 2009.

[107] C. Hongsong, J. Zhenzhou, H. Mingzeng, F. Zhongchuan, and J. Ruixiang, "Design and performance evaluation of a multi-agent-based dynamic lifetime security scheme for AODV routing protocol," *Journal of Network and Computer Applications*, vol. 30, pp. 145–166, Ene. 2007.

[108] M.-Y. Su, "Prevention of selective black hole attacks on mobile ad hoc networks through intrusion detection systems," *Computer Communications*, vol. 34, pp. 107–117, Ene. 2011.

[109] G. Kim, Y. Han, and S. Kim, "A cooperative-sinkhole detection method for mobile ad hoc networks," {*AEU*} - *International Journal of Electronics and Communications*, vol. 64, pp. 390–397, May. 2010.

[110] R. H. Jhaveri, S. J. Patel, and D. C. Jinwala, "A Novel Approach for GrayHole and BlackHole Attacks in Mobile Ad Hoc Networks," in *Proc. of the 2nd International Conference on Advanced Computing Communication Technologies (ACCT)*, pp. 556–560, Ene. 2012.

[111] M. Al-Shurman, S.-M. Yoo, and S. Park, "Black Hole Attack in Mobile Ad Hoc Networks," in *Proc. of the 42nd Annual Southeast Regional Conference (ACM-SE)*, pp. 96–97, Abr. 2004.

[112] N. Mistry, D. C. Jinwala, and M. Zaveri, "Improving AODV Protocol against Blackhole Attacks," in *Proc. of the International MultiConference of Engineers and Computer Scientists (IMECS)*, vol. 2, pp. 96–97, Mar. 2010.

[113] S. C. Mandhata and S. N. Patro, "A counter measure to Black hole attack on AODV-based Mobile Ad-Hoc Networks," *International Journal of Computer & Communication Technology (IJCCT)*, vol. 2, pp. 37–42, Feb. 2011.

[114] L. Himral, V. Vig, and N. Chand, "Preventing AODV Routing Protocol from Black Hole Attack," *International Journal of Engineering Science and Technology (IJEST)*, vol. 3, pp. 3927–3932, May. 2011.

[115] J. V. S. Jebadurai, A. R. Melvin, and I. J. R. Jebadurai, "Sinkhole detection in mobile ad-hoc networks using mutual understanding among nodes," in *Proc. of the 3rd International Conference on Electronics Computer Technology (ICECT)*, vol. 3, pp. 321–324, Abr. 2011.

[116] N. Gandhewar and R. Patel, "Detection and Prevention of Sinkhole Attack on AODV Protocol in Mobile Adhoc Network," in *Proc. of the 4th International Conference on Computational Intelligence and Communication Networks (CICN)*, pp. 714–718, Nov. 2012.

[117] S. Sharmila and G. Umamaheswari, "Detection of Sinkhole Attack in Wireless Sensor Networks Using Message Digest Algorithms," in *Proc. of the International Conference on Process Automation, Control and Computing (PACC)*, pp. 1–6, Jul. 2011.

[118] M. A. Rassam, A. Zainal, M. A. Maarof, and M. Al-Shaboti, "A sinkhole attack detection scheme in Mintroute wireless Sensor Networks," in *Proc. of the*

International Symposium on Telecommunication Technologies (ISTT), pp. 71–75, Nov. 2012.

[119] J. Qi, T. Hong, K. Xiaohui, and L. Qiang, "Detection and defence of Sinkhole attack in Wireless Sensor Network," in *Proc. of the IEEE 14th International Conference onCommunication Technology (ICCT)*, pp. 809–813, Nov. 2012.

[120] S. A. Salehi, M. A. Razzaque, P. Naraei, and A. Farrokhtala, "Detection of sinkhole attack in wireless sensor networks," in *Proc. of the IEEE International Conference onSpace Science and Communication (IconSpace)*, pp. 361–365, Jul. 2013.

[121] M. E. G. Moe, B. E. Helvik, and S. J. Knapskog, "TSR: Trust-based Secure MANET Routing Using HMMs," in *Proc. of the 4th ACM Symposium on QoS and Security for Wireless and Mobile Networks (Q2SWinet)*, pp. 83–90, Oct. 2008.

[122] I. T. A. Halim, H. M. A. Fahmy, A. M. Bahaa El-Din, and M. H. El-Shafey, "Agent-Based Trusted On-Demand Routing Protocol for Mobile Ad Hoc Networks," in *Proc. of the 4th International Conference on Network and System Security (NSS)*, pp. 255–262, Sep. 2010.

[123] H. Xia, Z. Jia, X. Li, L. Ju, and E. H. M. Sha, "Trust prediction and trust-based source routing in mobile ad hoc networks," *Ad Hoc Networks*, vol. 11, pp. 2096–2114, Sep. 2013.

[124] S. Gupta, S. Kar, and S. Dharmaraja, "BAAP: Blackhole attack avoidance protocol for wireless network," in *Proc. of the 2nd International Conference on Computer and Communication Technology (ICCCT)*, pp. 468–473, Sep. 2011.

[125] N. Sreenath, A. Amuthan, and P. Selvigirija, "Countermeasures against Multicast attacks on Enhanced-On Demand Multicast Routing Protocol in MANETs," in *Proc. of the International Conference on Computer Communication and Informatics (ICCCI)*, pp. 1–7, Ene. 2012.

[126] S. Buchegger and J.-Y. Le Boudec, "Nodes bearing grudges: towards routing security, fairness, and robustness in mobile ad hoc networks," in *Proc. of the 10th Euromicro Workshop on Parallel, Distributed and Network-based Processing (PDP)*, pp. 403–410, Ene. 2002.

[127] I. Khalil, S. Bagchi, C. N. Rotaru, and N. B. Shroff, "UnMask: Utilizing Neighbor Monitoring for Attack Mitigation in Multihop Wireless Sensor Networks," *Ad Hoc Networks*, vol. 8, pp. 148–164, Mar. 2010.

[128] M.-Y. Su, K.-L. Chiang, and W.-C. Liao, "Mitigation of Black-Hole Nodes in Mobile Ad Hoc Networks," in *Proc. of the International Symposium on Parallel and Distributed Processing with Applications (ISPA)*, pp. 162–167, Sep. 2010.

[129] P.-C. Tsou, J.-M. Chang, Y.-H. Lin, H.-C. Chao, and J.-L. Chen, "Developing a BDSR scheme to avoid black hole attack based on proactive and reactive architecture in MANETs," in *Proc. of the 13th International Conference on Advanced Communication Technology (ICACT)*, pp. 755–760, Feb. 2011.

[130] A. Konig, M. Hollick, and R. Steinmetz, "On the Implications of Adaptive Transmission Power for Assisting MANET Security," in *Proc. of the 29th IEEE International Conference on Distributed Computing Systems Workshops (ICDCS)*, pp. 537–544, Jun. 2009.

[131] Y. A. Mohamed and A. B. Abdullah, "Immune-inspired framework for securing hybrid MANET," in *Proc. of the IEEE Symposium on Industrial Electronics Applications (ISIEA)*, vol. 1, pp. 301–306, Oct. 2009.

[132] X. Ye and J. Li, "A security architecture based on immune agents for MANET," in *Proc. of the International Conference on Wireless Communication and Sensor Computing (ICWCSC)*, pp. 1–5, Ene. 2010.

[133] X. Ye, J. Li, and R. Luo, "Hide Markov Model Based Intrusion Detection and Response for Manets," in *Proc. of the 2nd International Conference on Information Technology and Computer Science (ITCS)*, pp. 142–145, Jul. 2010.

[134] P. L. R. Chze, W. K. W. Yan, and K. S. Leong, "A User-Controllable Multi-Layer Secure Algorithm for MANET," in *Proc. of the 8th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 1080–1084, Ago. 2012.

[135] A. Nabet, R. Khatoun, L. Khoukhi, J. Dromard, and D. Gaiti, "Towards secure route discovery protocol in MANET," in *Proc. of the Global Information Infrastructure Symposium (GIIS)*, pp. 1–8, Ago. 2011.

[136] C. A. Melchor, B. A. Salem, P. Gaborit, and K. Tamine, "AntTrust: A Novel Ant Routing Protocol for Wireless Ad-hoc Network Based on Trust between Nodes," in *Proc. of the 3rd International Conference on Availability, Reliability and Security (ARES)*, pp. 1052–1059, Mar. 2008.

[137] P. Agrawal, R. K. Ghosh, and S. K. Das, "Cooperative Black and Gray Hole Attacks in Mobile Ad Hoc Networks," in *Proc. of the 2nd International Conference on Ubiquitous Information Management and Communication (ICUIMC)*, pp. 310–314, Ene. 2008.

[138] D. Zhang and C.-K. Yeo, "A Novel Architecture of Intrusion Detection System," in *Proc. of the 7th IEEE Consumer Communications and Networking Conference (CCNC)*, pp. 1–5, Ene. 2010.

[139] S. McCanne and S. Floyd, "NS Network Simulator." [Online; Accedido 18 Julio 2014]
`http://www.isi.edu/nsnam/ns/`.

[140] T. S. Rappaport, *Wireless Communications: Principles and Practice*. Piscataway, NJ, USA: IEEE Press, 1996.

[141] D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," in *Mobile Computing* (T. Imielinski and H. F. Korth, eds.), vol. 353 of *The Kluwer International Series in Engineering and Computer Science*, pp. 153–181, Springer US, 1996.

[142] J. Arauz and P. Krishnamurthy, "Markov modeling of 802.11 channels," in *Proc. of the 58th IEEE Vehicular Technology Conference (VTC-Fall)*, vol. 2, pp. 771–775, Oct. 2003.

[143] L. Sánchez-Casado, G. Maciá-Fernández, P. Garcia-Teodoro, and N. Aschenbruck, "A Novel Collaborative Approach for Sinkhole Detection in MANETs," in *Proc. of the Workshop on Security in Ad-Hoc Networks (SecAN), in conjunction with ADHOC-NOW*, pp. 42–55, Jun. 2014.

[144] A. Varga, "OMNeT++ Discrete Event Simulation System." [Online; Accedido 18 Julio 2014]
`http://www.omnetpp.org/`.

[145] Network Engineering Security Group (NESG), "NETA: NETwork Attacks Framework for OMNeT++." [Online; Accedido 18 Julio 2014]
`http://nesg.ugr.es/index.php/en/neta`.

[146] L. Sánchez-Casado, R. A. Rodríguez-Gómez, R. Magán-Carrión, and G. Maciá-Fernández, "NETA: Evaluating the Effects of NETwork Attacks. MANETs as a Case Study," in *Advances in Security of Information and Communication Networks* (A. Awad, A. Hassanien, and K. Baba, eds.), vol. 381 of *Communications in Computer and Information Science*, pp. 1–10, Springer Berlin Heidelberg, 2013.

[147] C. Bettstetter and C. Wagner, "The Spatial Node Distribution of the Random Waypoint Mobility Model," in *Proc. of the 1st German Workshop on Mobile Ad-Hoc Networks (WMAN)*, pp. 41–58, Mar. 2002.

[148] J. Yoon, M. Liu, and B. Noble, "Random Waypoint Considered Harmful," in *Proc. of the 22nd Annual Joint Conference of the IEEE Computer and Communications (INFOCOM)*, vol. 2, pp. 1312–1321, Mar. 2003.

[149] X. Hong, M. Gerla, G. Pei, and C.-C. Chiang, "A Group Mobility Model for Ad Hoc Wireless Networks," in *Proc. of the 2nd ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, pp. 53–60, Ago. 1999.

[150] N. Aschenbruck, R. Ernst, E. Gerhards-Padilla, and M. Schwamborn, "BonnMotion: A Mobility Scenario Generation and Analysis Tool," in *Proc. of the 3rd International ICST Conference on Simulation Tools and Techniques (SIMUTools)*, pp. 51:1–51:10, Mar. 2010.

[151] F. Barceló and J. Jordán, "Channel Holding Time Distribution In Cellular Telephony," in *Electronics Letters*, vol. 34, pp. 146–147, Ene. 1998.

[152] M. Nakagami, "The m-Distribution – A General Formula of Intensity Distribution of Rapid Fading," in *Statistical Methods in Radio Wave Propagation* (W. C. Hoffman, ed.), pp. 3–36, Pergamon Press Oxford, 1960.

[153] A. Al Hanbali, E. Altman, and P. Nain, "A survey of TCP over ad hoc networks," *IEEE Communications Surveys & Tutorials*, vol. 7, pp. 22–36, Mar. 2005.

[154] G. Holland and N. Vaidya, "Analysis of tcp performance over mobile ad hoc networks," *Wireless Networks*, vol. 8, pp. 275–288, Mar. 2002.

[155] V. Anantharaman, S.-J. Park, K. Sundaresan, and R. Sivakumar, "TCP Performance over Mobile Ad Hoc Networks: A Quantitative Study," *Wireless Communications & Mobile Computing*, vol. 4, pp. 203–222, Mar. 2004.

[156] P. Saint-Andre, "Extensible Messaging and Presence Protocol (XMPP): Core," *IETF, RFC 6120*, Mar. 2011. [Online; Accedido 18 Julio 2014] `http://www.rfc-editor.org/rfc/rfc6120.txt`.

[157] P. Saint-Andre, "Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence," *IETF, RFC 6121*, Mar. 2011. [Online; Accedido 18 Julio 2014] `http://www.rfc-editor.org/rfc/rfc6121.txt`.

[158] P. Saint-Andre, "Extensible Messaging and Presence Protocol (XMPP): Address Format," *IETF, RFC 6122*, Mar. 2011. [Online; Accedido 18 Julio 2014] `http://www.rfc-editor.org/rfc/rfc6122.txt`.

[159] R. Gerhards, "The Syslog Protocol," Mar. 2009.

[160] H. Debar, D. Curry, and B. Feinstein, "The Intrusion Detection Message Exchange Format (IDMEF)," *IETF, RFC 4765*, Mar. 2007. [Online; Accedido 18 Julio 2014] `http://www.rfc-editor.org/rfc/rfc4765.txt`.

[161] K. Gorzelak, T. Grudziecki, P. Jacewicz, P. Jaroszewski, Ł. Juszczyk, and P. Kijewski, "Proactive Detection of Network Security Incidents," tech. rep., CERT Polska / NASK & ENISA Report (A. Belasovs, Ed.), Dic. 2011.

[162] R. Danyliw, J. Meijer, and Y. Demchenko, "The Incident Object Description Exchange Format," *IETF, RFC 5070*, Dic. 2007. [Online; Accedido 18 Julio 2014]
`http://www.rfc-editor.org/rfc/rfc5070.txt`.

[163] G. Klein, H. Rogge, F. Schneider, J. Toelle, M. Jahnke, and S. Karsch, "Response Initiation in Distributed Intrusion Response Systems for Tactical MANETs," in *Proc. of the European Conference on Computer Network Defense (EC2ND)*, pp. 55–62, Oct. 2010.

[164] X-ARF, "Network Abuse Reporting." [Online; Accedido 18 Julio 2014]
`http://www.x-arf.org`.

[165] B. Feinstein and G. Matthews, "The Intrusion Detection Exchange Protocol (IDXP)," *IETF, RFC 4767*, Mar. 2007. [Online; Accedido 18 Julio 2014]
`http://www.rfc-editor.org/rfc/rfc4767.txt`.

[166] M. Rose, "The Blocks Extensible Exchange Protocol Core," *IETF, RFC 3080*, Mar. 2001. [Online; Accedido 18 Julio 2014]
`http://www.rfc-editor.org/rfc/rfc3080.txt`.

[167] M. Jahnke, G. Klein, A. Wenzel, N. Aschenbruck, E. Gerhards-Padilla, P. Ebinger, S. Karsch, and J. Haag, "MITE – MANET Intrusion Detection for Tactical Environments," in *Proc. of the NATO/RTO IST-076 Research Symposium on Information Assurance for Emerging and Future Military Systems*, pp. 1–17, Oct. 2008.

[168] D. Schnackenberg, K. Djahandari, and D. Sterne, "Infrastructure for Intrusion Detection and Response," in *Proc. of the DARPA Information Survivability Conference and Exposition (DISCEX)*, vol. 2, pp. 3–11, Ene. 2000.

[169] R. Magán-Carrión, F. Pulido-Pulido, J. Camacho-Páez, and P. García-Teodoro, "Tampered Data Recovery in WSNs through Dynamic PCA and Variable Routing Strategies," *Journal of Communications*, vol. 8, pp. 738–750, Nov. 2013.

[170] R. Magán-Carrión, J. Camacho-Páez, and P. García-Teodoro, "A Multiagent Self-healing System against Security Incidents in MANETs," in *Highlights of Practical Applications of Heterogeneous Multi-Agent Systems. The PAAMS Collection* (J. M. Corchado, J. Bajo, J. Kozlak, P. Pawlewski, J. Molina, B. Gaudou, V. Julian, R. Unland, F. Lopes, K. Hallenborg, and P. García-Teodoro, eds.), vol. 430 of *Communications in Computer and Information Science*, pp. 321–332, Springer International Publishing, Jun. 2014.

[171] J. Li, S. U. Khan, and Q. Li, "An efficient event delivery scheme in mobile ad hoc communities," *Journal of Communication Networks and Distributed Systems*, vol. 10, pp. 25–39, Nov. 2013.

[172] T. Aurisch, T. Ginzler, and P. Martini, "Practical efficiency analysis of a dual mode group key management," in *Proc. of the IEEE Military Communications Conference (MILCOM)*, pp. 1–7, Nov. 2008.

[173] R. H. Jhaveri, S. J. Patel, and D. C. Jinwala, "DoS Attacks in Mobile Ad Hoc Networks: A Survey," in *Proc. of the 2nd International Conference on Advanced Computing Communication Technologies (ACCT)*, pp. 535–541, Ene. 2012.

[174] M. Imran, A. M. Said, and H. Hasbullah, "A survey of simulators, emulators and testbeds for wireless sensor networks," in *Proc. of the International Symposium in Information Technology (ITSim)*, vol. 2, pp. 897–902, Jun. 2010.

[175] J. Lessmann, P. Janacik, L. Lachev, and D. Orfanus, "Comparative Study of Wireless Network Simulators," in *Proc. of the 7th International Conference on Networking (ICN)*, pp. 517–523, Abr. 2008.

[176] A. ur Rehman Khan, S. M. Bilal, and M. Othman, "A performance comparison of open source network simulators for wireless networks," in *Proc. of the IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, pp. 34–38, Nov. 2012.

[177] A. Kumar, S. K. Kaushik, R. Sharma, and P. Raj, "Simulators for Wireless Networks: A Comparative Study," in *Proc. of the International Conference on Computing Sciences (ICCS)*, pp. 338–342, Sep. 2012.

[178] D. Wetherall and C. J. Lindblad, "Extending Tcl for Dynamic Object-oriented Programming," in *Proc. of the 3rd Annual USENIX Workshop on Tcl/Tk (TCLTK)*, vol. 3, pp. 19–19, Jul. 1995.

[179] The NS-3 Project, "NS-3." [Online; Accedido 18 Julio 2014] `http://www.nsnam.org/`.

[180] X. Zeng, R. Bagrodia, and M. Gerla, "GloMoSim: a library for parallel simulation of large-scale wireless networks," in *Proc. of the 12th Workshop on Parallel and Distributed Simulation (PADS)*, pp. 154–161, May. 1998.

[181] R. Bagrodia, R. Meyer, M. Takai, Y.-A. Chen, X. Zeng, J. Martin, and H. Y. Song, "PARSEC: a parallel simulation environment for complex systems," *Computer*, vol. 31, pp. 77–85, Oct. 1998.

[182] Riverbed Technology, "OPNET Modeler." [Online; Accedido 18 Julio 2014] `http://www.opnet.com/`.

[183] H. Ehsan and F. A. Khan, "Malicious AODV: Implementation and Analysis of Routing Attacks in MANETs," in *Proc. of the IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pp. 1181–1187, Jun. 2012.

[184] T. Gamer and M. Scharf, "Realistic simulation environments for IP-based networks," in *Proc. of the 1st International ICST Conference on Simulation Tools and Techniques (SIMUTools)*, pp. 83:1–83:7, Mar. 2008.

[185] G. Dini and M. Tiloca, "ASF: An attack simulation framework for wireless sensornetworks," in *Proc. of the 8th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pp. 203–210, Oct. 2012.

Apéndices

Appendix A

Thesis Summary

To comply with the PhD normative of the University of Granada, in this Appendix we provide an extended abstract in English of the present thesis. The Appendix is organized as follows. In Section A.1, we motivate the importance of the security in MANET environments and highlight two of the most relevant types of security threats nowadays: dropping and poisoning attacks. After that, in Section A.2 we point out the specific objectives of the present work, subsequently enumerating the main contributions achieved in Section A.3. Finally, Section A.4, which presents a summary of the main part of the work, is devoted to discuss in some detail each of the contributions.

A.1. Motivation

Among the multitude of technologies currently present in the ICT (*Information & Communications Technologies*) field, wireless networks have considerably evolved during the last years [1], leading to the appearance of different technologies, architectures and applications. In particular, this work focuses on ad hoc networks, a communication paradigm of increasing deployment which relies on certain specific characteristics, like the use of wireless communications and the lack of a fixed transport infrastructure (*i.e.*, routers or access points). An ad hoc network is a particular type of network composed of autonomous devices, geographically distributed in a given area and without a fixed infrastructure or centralized administration. Nodes that are within the communication range communicate directly, while those which are out of that range make use of other nodes to relay their messages to reach their

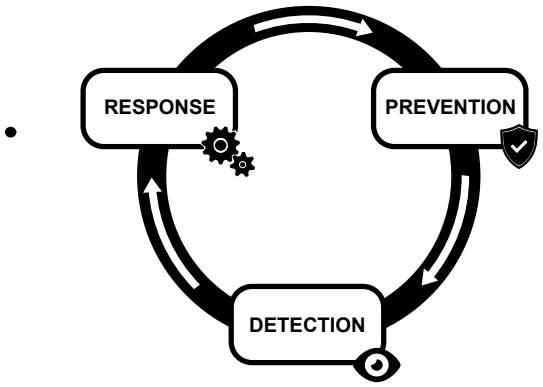


Figure A.1: Traditional defense lines against security threats.

destination (multi-hop strategy). The versatility of such a kind of environments is increased when the devices are mobile, setting up the so-called MANETs (*Mobile Ad hoc NETwork*s) [5], [6]. The inherent characteristics of these technologies make them a particularly useful candidate in certain areas, such as environmental and military applications, disaster and crisis management (earthquakes, terrorist attacks), etc. [4].

Despite the benefits of this kind of networks, they present several limitations. On the one hand, MANETs nodes are usually resource-constrained: reduced bandwidth, lifetime of the battery, power-constrained processing or limited storage capacity. On the other hand, a huge number of security issues associated with these environments emerge [8]. Due to the intrinsic characteristics of MANETs, including their open nature (wireless communications) or the lack of a transport infrastructure, they are vulnerable to several security threats which need to be properly addressed [9].

To deal with security threats, three defense lines have traditionally been proposed: prevention, detection and response (see Fig. A.1). Even though there are well-known preventive measures recommended to avoid the occurrence of malicious actions, their deployment does not completely prevent these behaviors from occurring. Consequently, it is necessary to complement these mechanisms with other techniques oriented to detect malicious events. Once those intrusive events are detected, response schemes aimed at solving these issues should be applied. Additionally, it would be desirable to include feedback mechanisms in the whole process to provide dynamic adaptability to the particular conditions of the environment [11][1]. However, although these modules should interact among them, they are usually developed as independent solutions, thus adding another problem due to the lack of interoperation schemes.

[1]Recovery is often specified as a separate additional defense line.

In this general context, the central purpose of this thesis is to strengthen the security in MANETs, mainly through the development of new detection approaches for highly disruptive attacks in this kind of networks.

A.2. Objectives and Methodology

As previously stated, this thesis is focused on *strengthening the security in MANET networks*. This general objective is divided into two more specific targets: (*i*) study and development of detection mechanisms for security attacks in MANETs, and (*ii*) development of procedures to allow the integration of defensive solutions. To achieve these goals, the following methodology has been followed.

The first step is to thoroughly **study existing attacks in MANET environments**, identifying those more relevant from the point of view of their impact. In particular, this study has shown that two of the main current security threats are dropping and poisoning attacks [12]. Indeed, the development of precise detection schemes for both types of attack is nowadays a very popular topic in the field of network security in the research community.

In order to detect the existence of attacks in a network, it is common to deploy IDSs (*Intrusion Detection Systems*) which, by means of monitoring several parameters related to the activity of the environment (*e.g.*, number of forwarded or dropped packets, information about learned routes, characteristics of the flows), are able to determine the occurrence of malicious behaviors against the system. Due to the inherent nature of MANETs and their scarcity of resources, most of the detection techniques and procedures developed for WLANs (*Wireless Local Area Network*s) and wired networks are no longer suitable for this kind of environments [13]. In this line, we have introduced new efficient an effective **mechanisms to detect attacks in MANETs**. To evaluate our specific approaches, an experimental framework has been developed. Since attacks evolve rapidly, the deployment of new defensive techniques is a hard task. This way, simulation offers a good compromise between cost and complexity and, therefore, we have used this kind of tools to test the developed detection schemes. This has allowed us to compare the results obtained with those provided by other similar schemes in the literature, thus extracting valid conclusions regarding the real capabilities of our new proposals.

Once the detection of MANET attacks is addressed, we found it convenient to leverage them in order to adopt subsequent response measures against the intrusive events reported. However, despite that the three typical defense lines should theoretically interoperate to provide a more global and integral security approach, they are generally designed and adopted as independent solutions, without taking into consideration the necessity of effective intercommunication procedures. This

becomes specially critical in ad hoc environments [14]. Therefore, the last phase considered in this work has been to **integrate security solutions**. For that, we have first designed and developed a mechanism for the *notification and alert of security events*, whose main goal is to be used as an effective interoperation procedure between the detection and response modules. Also in the line of putting together security developments, although with more general purposes, we have implemented an *integral security framework*. This constitutes a valuable tool to deploy attacks and implement, integrate and evaluate new defense schemes in a controlled simulation oriented testbed environment for the research community.

According to all of the above, the specific tasks developed during the present thesis are:

I. *Study of MANET attacks*

- To elaborate a detailed list of the different attacks currently reported in MANET networks, proposing a novel taxonomy for a better classification.
- To select the most relevant attacks in order to focus on them in this work: *dropping* and *poisoning* attacks.
- To perform a thorough study on the state of the art of research in the field of security defenses against these attacks, classifying the different works in an organized and intuitive way.

II. *Development of schemes for the detection of attacks in MANETs*

- To design new detection schemes against the aforementioned attacks, which are a major security concern in MANETs.
- To evaluate the proper capabilities of the proposed detection schemes by means of simulation.

III. *Proposal for the interoperability between defense lines*

- To study the state of the art of the research regarding the interoperation between the different defense lines.
- To design a new communication solution aimed at solving the current limitations existing in this field for MANETs.
- To implement and evaluate the introduced solution.

IV. *Development of an integral security framework*

- To design a flexible architecture, appropriate for the implementation and evaluation of several network attacks and specific defensive solutions.

- To effectively develop the actual security framework, implementing different attacks as a proof of concept.
- To test the overall capabilities of the framework.

A.3. Main Contributions

The main contributions of this thesis regarding the tasks previously described are summarized in the next:

1. We present a review of the different security attacks currently reported in the field of MANET networks.
2. We propose a novel taxonomy to classify them, with the objective of developing more efficient and effective defense mechanisms.
3. We present a thorough bibliographical review of the most relevant works in the last years in the field of MANET attacks, focusing on two of the most disruptive threats: dropping and poisoning attacks. As a particular and relevant case of the latter type, sinkhole attacks are studied here.
4. We contribute with two specific proposals for the detection of the aforementioned attacks. One of them is based on a simple heuristic to distinguish between malicious dropping behaviors and legitimate discarding causes.
5. The second approach allows to detect sinkhole attacks by means of a collaborative scheme that collects information from the node's vicinity.
6. We also design and implement a new communication protocol mainly intended to interoperate among the traditional defensive lines. It might also be used to develop distributed and collaborative detection or reaction systems.
7. We develop a new integral security simulation framework for the evaluation of network attacks and security solutions, which relies on a flexible and versatile architecture.

A.3.1. Publications

In the following, we indicate the publications obtained more directly related to the main topics of the present thesis:

International Journals

1. *Submitted* → **L. Sánchez-Casado**, G. Maciá-Fernández, P. García-Teodoro and N. Aschenbruck. "Identification of Contamination Zones for Sinkhole Detection in MANETs". *Journal of Network and Computer Applications (Elsevier)*, 20 pages, 2014.

2. *Submitted* → **L. Sánchez-Casado**, G. Maciá-Fernández, and P. García-Teodoro. "A Model of Data Forwarding in MANETs for Lightweight Detection of Malicious Packet Dropping". *The Scientific World Journal (Hindawi)*, 25 pages, 2014.

Book Chapters

3. P. García-Teodoro, **L. Sánchez-Casado** and G. Maciá-Fernández. "Taxonomy and Holistic Detection of Security Attacks in MANETs". *Security for Multihop Wireless Networks*, S. Khan and J. Lloret (Eds.), CRC Press, pp. 1-12, 2014.

4. **L. Sánchez-Casado**, R. Magán-Carrión, P. García-Teodoro and J. E. Díaz-Verdejo. "Defenses Against Packet Dropping Attacks in Wireless Multihop Ad Hoc Networks". *Security for Multihop Wireless Networks*, S. Khan and J. Lloret (Eds.), CRC Press, pp. 377-400, 2014.

International Conferences

5. **L. Sánchez-Casado**, G. Maciá-Fernández, P. García-Teodoro and N. Aschenbruck. "A Novel Collaborative Approach for Sinkhole Detection in MANETs". *Workshop on Security in Ad Hoc Networks* (SecAN), pp. 42-55, 2014.

6. **L. Sánchez-Casado**, R. A. Rodríguez-Gómez, R. Magán-Carrión and G. Maciá-Fernández. "NETA: Evaluating the effects of NETwork Attacks. MANETs as a case study". *Advances in Security of Information and Communication Networks*, (SecNet), pp. 1-10, 2013.

7. **L. Sánchez-Casado**, G. Maciá-Fernández, and P. García-Teodoro. "An Efficient Cross-Layer Approach for Malicious Packet Dropping Detection in MANETs". *11th IEEE International Conference on Trust, Security and Privacy in Computing and Communications* (Trustcom), pp. 231-238, 2012.

National Conferences

8. **L. Sánchez-Casado**, R. Magán-Carrión, P. Garrido-Sánchez and P. García-Teodoro. "Protocolo para la Notificación y Alerta de Eventos de Seguridad en Redes Ad

hoc". Accepted in *Reunión Española sobre Criptología y Seguridad de la Información* (RECSI), 6 pages, 2014.

9. **L. Sánchez-Casado**, G. Maciá-Fernández, and P. García-Teodoro. "Indicadores de Ataques Sinkhole en MANETs". *XI Jornadas de Ingeniería Telemática* (JITEL), pp. 475-480, 2013.

10. **L. Sánchez-Casado**, R. A. Rodríguez-Gómez, R. Magán-Carrión and G. Maciá-Fernández. "NETA: un Framework para Simular y Evaluar Ataques en Redes Heterogéneas. MANETs como Caso de Estudio". *XI Jornadas de Ingeniería Telemática* (JITEL), pp. 487-492, 2013.

11. **L. Sánchez-Casado**, G. Maciá-Fernández, and P. García-Teodoro. "Multi-Layer Information for Detecting Malicious Packet Dropping Behaviors in MANETs". *Reunión Española sobre Criptología y Seguridad de la Información* (RECSI), pp. 57-62, 2012.

12. **L. Sánchez-Casado**, G. Maciá-Fernández, and P. García-Teodoro. "Caracterización de Servicios en Redes Ad Hoc Inalámbricas mediante Métricas Cross-Layer". *X Jornadas de Ingeniería Telemática* (JITEL), pp. 381-384, 2011.

Other publications which are less directly related to the main topic of this thesis, although also important for it, are:

International Journals

a) *Accepted* → R. A. Rodríguez-Gómez, G. Maciá-Fernández, **L. Sánchez-Casado** and P. García-Teodoro. "Analysis and Modeling of Resources Shared in the BitTorrent Network". *Transactions on Emerging Telecommunications Technologies (Wiley)*, 2014.

b) *Submitted* → S. Salah, G. Maciá-Fernández, J. E. Díaz-Verdejo and **L. Sánchez-Casado**. "A Model for Incident Tickets Correlation in Network Management". *Journal of Network and System Management*, *major revision*, 2014.

Books

c) G. Maciá Fernández, R. Magán Carrión, R. A. Rodríguez Gómez, **L. Sánchez Casado**: "Sistemas y Servicios Telemáticos". 2013. Ed. Avicam. ISBN: 978-84-941781-6-0.

A.4. Anomaly-based Multi-layer Intrusion Detection for MANETs

According to the aforementioned objectives and tasks, this work has been divided into three main parts.

The first part is related to the basics of security in MANET networks, providing a review of the different current attacks reported and trying to give some order to this field by proposing a novel taxonomy (publication 3) with two main goals: organize and classify security attacks in MANETs from a practical perspective, and provide guidelines to potentially build more flexible, effective and comprehensive detection approaches.

The second and more important part (as it corresponds to the central contribution of the thesis) is focused on the detection of two of the most disruptive threats in MANETs, dropping and poisoning attacks. As a particular and relevant case of the latter type, sinkhole attacks are considered here. A state of the art of research in the detection of these attacks can be found in publication 4. In this line, we discuss in Section A.4.2 a simple heuristic-based detection scheme aimed at distinguishing malicious dropping behaviors from different circumstances which can lead to legitimate packet discards (see publications 2, 7, 11 and 12). After that, in Section A.4.3 we describe a collaborative approach to detect sinkhole attacks in MANETs based on the existence of "contamination borders" (publications 1, 5 and 9).

Finally, the third part is intended to integrate the detection schemes previously developed with other defensive lines. Thus, we first introduce the development of a novel lightweight notification protocol used as a communication interface between detection and response defense modules (see publication 8). It has been explicitly designed to notify and alert about different security events in ad hoc networks. It might be also used for the distribution of information between entities in collaborative detection/response processes. On the other hand, and with a more ambitious purpose, we have subsequently developed an integral security framework for the simulation of network attacks and their defensive solutions. Its flexible architecture makes it highly extensible and versatile for the benchmarking of defensive schemes under controlled testing conditions (see publications 6 and 10).

Each of the parts is described more in depth in what follows.

First Part: Security Fundamentals in MANETs

A.4.1. Security Attacks in MANETs (Publication 3)

The study of network security attacks in MANETs shows that, in many attacks, several types, sub-types, and variants coexist, with subtle differences among them. Such diversity, artificial in some sense, can be justified by several causes, like a matter of degree or impact, or the motivation of the attacker. Besides, it can be found that the purpose of an attack is sometimes "mistaken" for the procedure that leads its execution. In addition, it can be checked that many research works analyze attacks against different specific protocols, creating a new variant when the target of the attack varies. In summary, we have today a big plethora of different names for attacks and there is no consensus about a definite list of attacks in MANETs.

Considering the above points, we can try to resolve the necessity for properly categorizing MANET attacks into more coherent classes, in order to improve and clarify this research field. A new classification would facilitate a better understanding of the vulnerabilities present in such environments and would enable the deployment of more flexible and effective detection approaches. This way, even though some classifications have already been proposed in the literature, like [18], [24], [25], [26] and [27], their practical utility from the aforementioned perspective is limited.

In the above context, we introduce a novel taxonomy for MANET attacks. Adhering to the definition of "taxonomy", several successive criteria are considered to classify the existing attack types. Thus, from a common *root*, *i.e.*, security attacks in MANETs, successive groups are obtained for the known attacks until each specific variant or *species* is derived. Conversely, defining the *species* allows us to derive a set of detection features that, beyond the particularities of each attack, would simplify the design of more effective detection environments. The different subsequent criteria used are: (*i*) *action* of the attacker, (*ii*) *effect* of the attack, (*iii*) *procedure* of the attack, and (*iv*) *function/service* attacked.

First, we classify attacks as active or passive, depending on whether the *action carried out by the attacker* to execute the attack affects the system in some way (active) or not (passive).

A second level considers the *effect of the attack*. Passive attacks imply sniffing information, what can directly affect confidentiality. Conversely, active attacks can produce three principal effects, which are related to either the entities taking part in the communication, or the transmitted information, or the quality of the service provided. These effects are as follows:

- *Impersonation*, affecting the authenticity of the entities in some way.

- *Spoofing*, which is a risk to the integrity of the information.
- *Interruption*, total or partial disruption of the service, affecting its availability.

A third level to differentiate MANET attacks is based on the *procedure followed to execute the attack*. Meanwhile passive attacks are possible by simply monitoring the channel, active attacks can be executed in several ways instead:

- *Impersonation*. This can be performed by falsifying the identity of the entities. Two procedures can be distinguished:
  - *Replication*, an identity already existing in the system is used.
  - *Invention*, a new specific identity is created for the attacker.
- *Spoofing*. This kind of attacks are executed by introducing "wrong" information at nodes and, consequently, causing "wrong" decision making. Two situations can appear:
  - *Poisoning*, by directly introducing fake information into communications.
  - *Delay*, by capturing packets and forwarding them with an artificial time delay, either positive or negative.
- *Interruption*. Three principal procedures can lead to the (total or partial) interruption of the service:
  - *Delay*, by introducing delays in communications, thus increasing the response time involved.
  - *Dropping*, by means of packet discarding, which implies a waste of resources to transmit information that will not reach its final destination.
  - *Flooding*, which will block the normal access to the resources of the system by artificially generating some kind of traffic.

In summary, we argue that every reported MANET attack can be classified in one of the seven categories obtained from the multi-dimensional classification *action* → *effect* → *procedure*, as shown in Fig. A.2. The existence of these classes allows to transcend particular differences due to the consideration of specific services and/or protocols.

At this point, the *species* of attacks can be obtained by applying a fourth classification criterion: the *target function attacked*, *i.e.*, the service and/or protocol objective of the attack. There are many attack sub-types/variants, so different functions (and associated protocols) are implemented in the network. Therefore, it is necessary to know the specific syntax of the corresponding function/service to be able to detect an attack.

Figure A.2: Taxonomy of security attacks in MANETs.

Dropping and Sinkhole Attacks in MANETs

After the study of the different reported attacks and the proposal of a novel taxonomy, we can conclude that two of the most disruptive threats in MANETs are dropping and poisoning attacks [9], [12]. In the first class, nodes maliciously drop received data or routing messages instead of forwarding them, thus disrupting the normal operation of the network. In the latter category, we focus on sinkhole attacks as a particular case of poisoning attacks, where nodes attempt to forge the source-destination routes in order to attract the surrounding traffic.

In this work, we consider two of the most widely used technologies for the deployment of MANETs, the IEEE 802.11 standard [35] and the AODV (*Ad hoc On-demand Distance Vector*) routing protocol [15]. We assume the reader's knowledge of the IEEE 802.11 standard, where the RTS (*Request To Send*)/CTS (*Clear To Send*) mechanism is used to avoid the hidden station problem, while the basics of AODV will be explained very briefly in the following.

In AODV, routes are established on demand via a *route discovery* process, *i.e.*, RREQ (*Route REQuest*) messages are broadcast by the nodes, flooding the network until any of them reaches the intended destination. Upon the reception of a RREQ message, the destination will then send a RREP (*Route REPly*) message backwards via the inverse route. Besides, AODV permits intermediate nodes to generate RREP messages if they have a valid route. Therefore, source and intermediate nodes are responsible for managing the routing information. To avoid routing loops, AODV employs *destination sequence numbers*, *i.e.*, monotonically increasing numbers specified for each destination node and updated whenever a node receives new information related to that destination, *i.e.*, if a received sequence number is greater than the last stored sequence number. Given the choice between two routes, a node will select that with the greatest sequence number.

The implementation of dropping attacks in MANETs is extremely easy. A malicious node may simply modify the forwarding function in the IP (*Internet Protocol*) protocol to perform the packet discard, either data or control related, in the desired way (totally, partially, selectively, etc.). However, to achieve greater impact in the network performance, it is common that these malicious nodes try to previously introduce themselves in the multi-hop routes (sinkhole attack).

To execute sinkhole attacks, malicious nodes can easily exploit the usage of sequence numbers in AODV. This way, a malicious node could modify or create a RREP message announcing an optimal metric and, if the sequence number is large enough, all other legitimate announced routes will be invalidated. As a consequence, the malicious node guarantees to be considered as the best route, thus being selected as the next hop on the path. If the sinkhole node replies to every received RREQ, it will eventually become a sink of most of the surrounding network traffic.

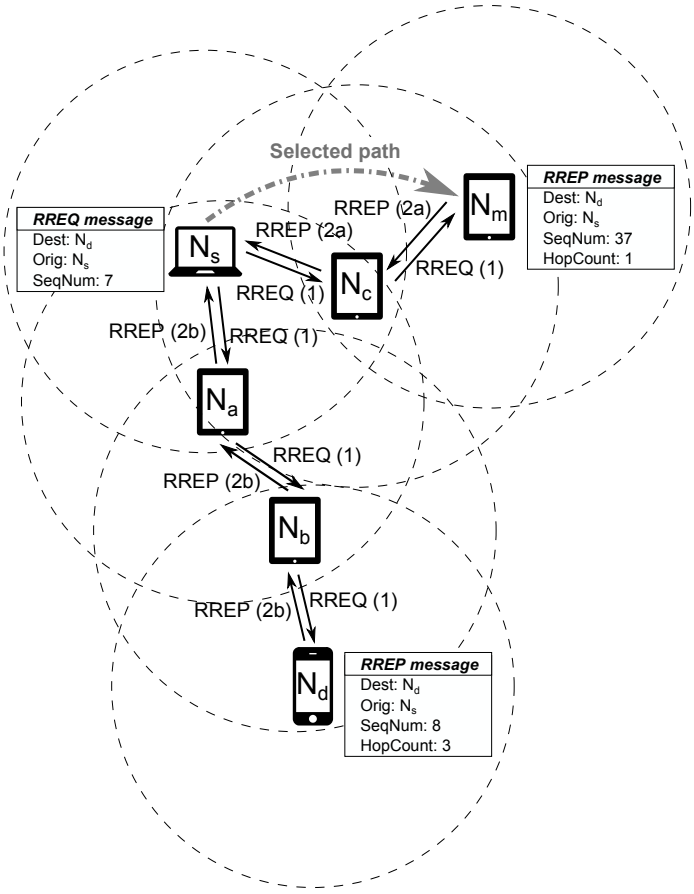


Figure A.3: Example of a sinkhole node, N_m, replying with a fake RREP.

Figure A.3 shows an example of a sinkhole attack. Here, the source node N_s broadcasts a RREQ message (1) asking for a route towards the destination N_d. This message is then forwarded by intermediate nodes (N_a, N_b, N_c). When the RREQ packet reaches the malicious node N_m, it replies with a fake RREP message (2a) claiming to have a fresher ($SeqNum = 37$) route towards N_d. At the same time, destination node N_d is replying with a RREP message (2b) that includes the legitimate value ($SeqNum = 8$). Therefore, despite receiving other legitimate replies, source node N_s will choose the route through N_c, which is considered the most recent. Thus, the traffic will eventually go through the malicious node N_m.

Although we focus our study on AODV, sinkhole attacks can be easily and similarly implemented in other protocols, like DYMO (*DYnamic MANET On-demand*) [43] or DSR (*Dynamic Source Routing*) [42]. Even though there are significant differences, these protocols still share some relevant concepts. Specifically, the use of some type of identifiers to determine the "freshness" of the route, whose application can also be exploited to launch a sinkhole attack. From this perspective, the detection approach presented in Section A.4.3 might be extended to other protocols.

Second Part: Attack Detection in MANETs

Before explaining the central contributions of this thesis regarding the attack detection topic, we have made a necessary review of the more important defensive solutions proposed in the literature to fight against dropping and sinkhole attacks. Although this review is not present in this summary, readers can find more details about it in publication 4.

A.4.2. Dropping Detection in MANETs (Publications 2, 7, 11 & 12)

Among others, *packet dropping* attack is one of the most disruptive threats in MANETs. Nodes exhibiting this behavior maliciously drop received data or routing messages instead of forwarding them, thus disrupting the normal operation of the network [12]. Different categories can be considered to classify this kind of attacks depending on the particular strategy adopted by the attacker (*blackhole*, *greyhole*, *selfish*, etc.).

This section introduces a novel IDS approach for recognizing generic malicious packet dropping behaviors in MANETs, based on a cross-layer methodology. In particular, statistics from both MAC (*Medium Access Control*) and network layers are collected and analyzed. This scheme relies on an analytical model which represents the forwarding process for a node in a MANET. This model properly includes circumstances which can lead to legitimate drops, such as collisions, channel errors or mobility related situations. This allows us to distinguish between these "normal" circumstances and actual malicious dropping behaviors. Two different approximations are deployed for the practical implementation: a local stand-alone approach, where every node informs about its own parameters, and a distributed-collection approach, where nodes collect information about neighbors, informing about it and sharing it to perform the detection.

A big number of solutions have been proposed in the literature to handle packet dropping in mobile ad hoc networks [52], like those in [98], [99], [105], [100] and [101]. However, most of these works only deal with mobility related situations by employing data mining techniques, which introduce a high computational overhead. Besides, the existence of legitimate reasons for dropping, like collisions or packet corruption, is not taken into account.

Forwarding Process in MANETs

This section presents the analytical model derived by us to the packet forwarding process followed by every node in a MANET. This model will constitute the basis for

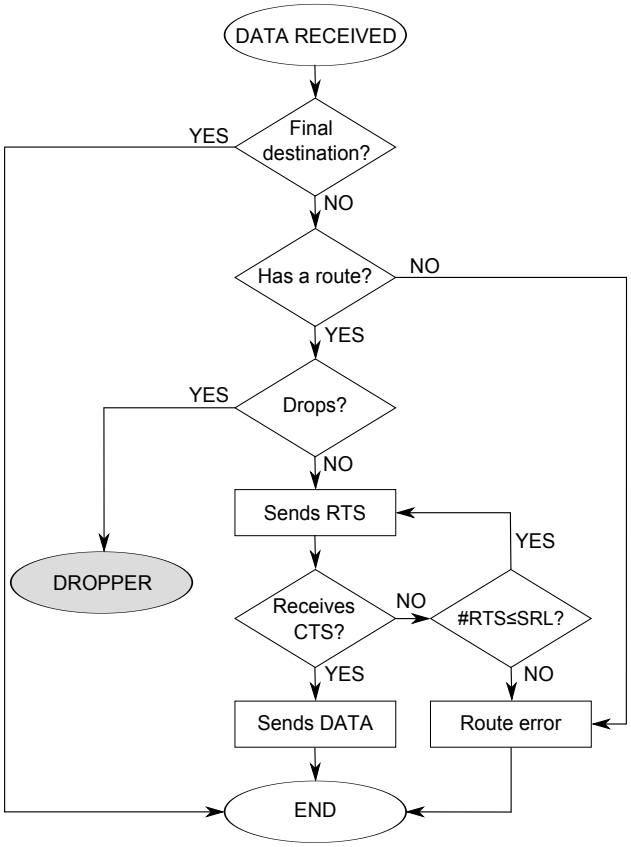


Figure A.4: Flowchart for the forwarding process in MANETs.

our proposed malicious packet dropping detection approach. This way, the operation by which a given node relays received data packets to a next hop implies several steps, which are properly explained in the following (see flowchart in Fig. A.4).

After a data packet is correctly received in a node, successive events must occur for the packet to be forwarded:

1. The node is not the final destination of the packet ($\overline{dest}$ event).
2. The node has a valid route to relay the packet to the destination ($rout$ event).
3. The node does not maliciously drop packets ($\overline{drop}$ event).

If all of the previous events occur, the node tries to forward the packet:

- First, the node will try to send a RTS message, *i.e.*, the box "Sends RTS" in Fig. A.4 is reached (RTS event, with associated probability P_{RTS}).

- Second, the node receives a CTS packet from the next hop when the corresponding RTS reached the latter, *i.e.*, the question "Receives CTS?" is raised in Fig. A.4 (CTS event, with probability P_{CTS}).

To obtain the probability of a node being a dropper, the above probabilities must be estimated. To obtain P_{RTS}, the mentioned events 1 to 3 are considered, *i.e.*, the node has a route for the destination and it is neither the final destination nor a dropper:

$$P_{RTS} = \Pr\left(RTS \mid \overline{dest}, rout\right) = (1 - P_{DROP}) \tag{A.1}$$

where P_{DROP} is the probability that the packet is maliciously discarded.

RTS and CTS packets can be lost due to legitimate reasons, which causes CTS not to be received, thus leading to a RTS retransmission, whose limit, SRL (*Short Retry Limit*), is fixed to 7 by default in the IEEE 802.11 protocol. Once this limit is exceeded, the corresponding packet is discarded, and the sender node assumes that the link towards the next hop is broken. We make the assumption that this limit is exceeded only in situations where the destination node has moved outside of the coverage area of the source. Therefore, the probability that the CTS is correctly received is divided into two terms: collisions or channel errors (P_{COL}), where RTS retransmissions do not reach the limit, and mobility situations P_{MOB}, which cause broken links. Thus, the probability that CTS event happens given that RTS event has occurred is:

$$P_{CTS} = \Pr(CTS \mid RTS) = 1 - (P_{COL} + P_{MOB}) \tag{A.2}$$

Finally, the data packet is forwarded (FWD event). Since both RTS and CTS events need to occur successfully, the forwarding probability, P_{FWD}, is computed as:

$$\begin{aligned} P_{FWD} &= \Pr\left(CTS, RTS \mid \overline{dest}, rout\right) = \Pr(CTS \mid RTS) \cdot \Pr\left(RTS \mid \overline{dest}, rout\right) \\ &= (1 - P_{DROP}) \cdot [1 - (P_{COL} + P_{MOB})] \end{aligned} \tag{A.3}$$

This way, the probability of packet dropping can be calculated from (A.3) as:

$$P_{DROP} = 1 - \frac{P_{FWD}}{[1 - (P_{COL} + P_{MOB})]} \tag{A.4}$$

This dropping probability is subsequently compared to a detection threshold, θ, so that the analyzed node is concluded to be malicious if P_{DROP} is greater than θ, and legitimate otherwise:

$$node = \begin{cases} \text{malicious}, & \text{if } P_{DROP} \geq \theta \\ \text{legitimate}, & \text{otherwise} \end{cases} \tag{A.5}$$

- **Parameters Estimation**

Here we discuss how to calculate the probabilities involved in Eq. (A.4): P_{FWD}, P_{COL} and P_{MOB}.

First, the estimator of P_{FWD} is obtained as the percentage of data packets forwarded with regard to those received:

$$\hat{P}_{FWD} = \frac{\#DATA_{FWD}}{\#DATA_{RECV}} \tag{A.6}$$

Regarding P_{COL}, the number of RTS packets sent by the node without a proper CTS and the total number of attempts to reserve the channel are computed. However, only those packets not related to broken links are taken into account. Thus, the estimator $\hat{P}_{COL}$ is:

$$\hat{P}_{COL} = \frac{\#RTS_{SENT} - \#CTS_{RECV}}{\#RTS_{SENT}} \tag{A.7}$$

Finally, the estimator for the probability of a broken link situation, P_{MOB}, is:

$$\hat{P}_{MOB} = \begin{cases} 1, & \text{if } \#RTS_{SENT} > SRL \\ 0, & \text{otherwise} \end{cases} \tag{A.8}$$

When AODV considers the link as broken, a route maintenance process is initiated, where two possibilities or scenarios may occur: if the link is closer to the source, a RERR (*Route ERRor*) message is sent backwards immediately; while if the link is closer to the destination, a local repair (by sending a RREQ) is tried. During a certain time, the node with a broken link continues receiving messages, although it is unable to forward them, thus behaving like a malicious node does. This period of time is even longer in the second scenario. Therefore, the decision about how long the probability $\hat{P}_{MOB}$ is considered equal to 1 (and therefore P_{DROP} considered 0 and the node legitimate) depends on the particular scenario that takes place. To know that, our IDS should monitor if any RREQ message has been sent after a broken link is detected.

Taking into account all of the above, the dropping probability is obtained from (A.4) as:

$$P_{DROP} = \begin{cases} 0, & \text{if } \hat{P}_{MOB} = 1 \\ 1 - \frac{\hat{P}_{FWD}}{1-\hat{P}_{COL}}, & \text{otherwise} \end{cases} \tag{A.9}$$

- **Enhanced Windowing**

The traditional way of acquiring some required features representing a system usually considers temporal observations over successive non-overlapped analysis windows of fixed duration. However, this approach presents two main drawbacks in our case:

- The first one occurs when the temporal window ends just after the transmission of a RTS packet, thus leading to undesirable effects due to discontinuities, since the whole circumstance characterizing a mobility related situation will not be caught in any of the temporal windows. Therefore, legitimate drops due to mobility will not be considered as such.
- The second is related to the fact that, even if during a certain interval there are no features to be collected or there are only a few, they will be analyzed anyway, thus obtaining biased information. In such a case, the IDS will consider a very high percentage of dropped packets, leading to the misclassification of legitimate nodes as malicious.

To remedy these issues, we propose an event-based windowing procedure instead of a time-based one. That is, the features are obtained for non-overlapping windows of P received data packets.

With event-based windows, the discontinuity problem is avoided, since the end of each window will always coincide with a data packet reception event, thus ensuring that mobility situations can be fully collected. On the other hand, the collection of statistics will always consider the same number of events, P, which attenuates the effect of biased information. Moreover, if a given node is not receiving traffic at all, it makes no sense to perform a detection process every certain time which only involves a waste of resources. Thus, the proposed event-based windowing method implies additional resources saving in nodes with scarce activity.

Implementing the Packet Dropping Detection Scheme

Beyond the theoretical development of our detection method, two different implementations can be deployed, which just differ in the way to collect the network features which characterize the nodes to be analyzed.

The first is a stand-alone approach, where the features collection process is locally provided by each node, which can access all the information needed to perform a more accurate detection. Note that this stand-alone approach is not really a feasible implementation, as it assumes that every node of the network is trustworthy. Yet, we think that this is an interesting case study, as it will give us the theoretical bounds of performance for our system and, for this reason, we will use it in the experimental evaluation.

Given the untrustworthy nature of the stand-alone features collection, a distributed gathering architecture is alternatively implemented. This relies on the use of monitor nodes promiscuously collecting and analyzing the features within their communication range and cooperating to provide a collaborative data collection process. As the monitor nodes work in promiscuous mode, the $\#CTS_{RECV}$ and $\#DATA_{RECV}$ features are replaced by their respective approximations, $\#CTS_{SENT}$ and $\#DATA_{SENT}$. That is, the original are indirectly "measured" through the latter ones. It must be noted that the use of these two features is just an approximation. However, in the next subsection we demonstrate that this estimation does not degrade significantly the performance of the system. When monitor nodes have collected all the needed information for a given node, its malicious behavior is estimated as usual.

It should be noted at this point that the use of monitor nodes implicitly assumes that they are trustworthy. This, however, is not strictly necessary, as some kind of voting process may be alternatively implemented to decide about potential differences in the values of the features received due to the existence of one or more malicious nodes that want to evade the detection. This way, the set of trustworthy monitor nodes can be substituted by the own neighbor nodes of a given one in the network.

Experimental Results

Several tests have been carried out to verify the proper performance of both dropping detection approaches, and the experimental results are discussed in what follows.

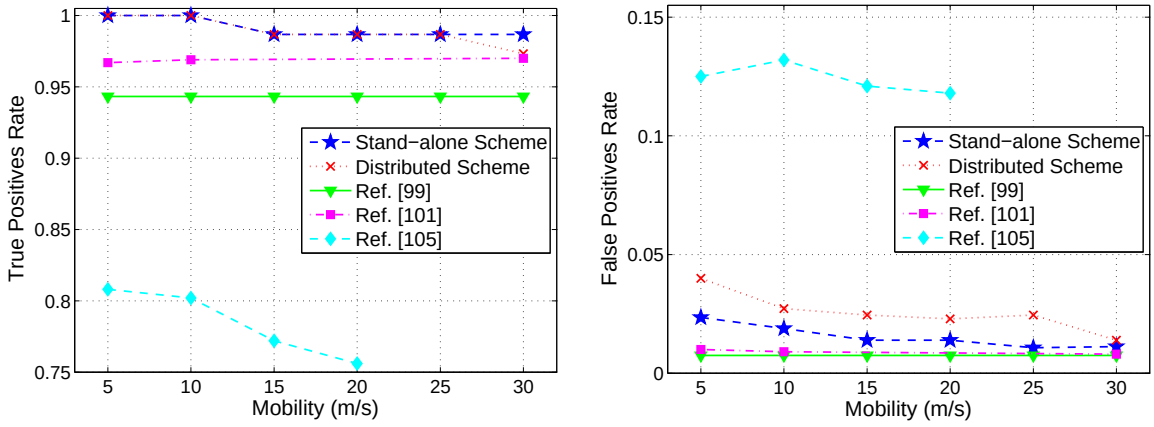
- **Experimental Environment**

NS-2 (*Network Simulator 2*) [139] is used to simulate several deployments of a MANET. The simulation area is restricted to a 1000 m x 1000 m square, with each node having a communication range of 250 m, and the *Two Ray Ground* [140] propagation model being used. AODV and IEEE 802.11b are chosen as the routing and MAC layer protocols, respectively, and the RTS/CTS mechanism is used to avoid the hidden station problem. The total number of nodes is 25, while the number of

malicious nodes performing a dropping attack varies from 1 to 20. There exist 20 application traffic flows, each one consisting of a CBR (*Constant Bit Rate*) connection, with 4 packets/second data and payload size equal to 512 bytes. The RWP (*Random Way Point*) [141] is used to model the mobility of the nodes, with a fixed minimum speed of 1 m/s and a maximum speed varying from 5 to 30 m/s. The pause time is set to 15 s. The malicious nodes are configured to drop 20% of the data packets going through them and supposed to be relayed. However, they participate normally in the routing process. The upper bound for the time that can take the local repairing process is selected to be 60 s.

- **Detection Performance**

We now evaluate the global effectiveness of the proposed detector, measured by computing two metrics, namely TPR (*True Positives Rate*) and FPR (*False Positives Rate*). From them, we obtain the ROC (*Receiver Operating Characteristic*) space by varying the decision threshold θ in (A.5). To capture the statistical performance, results have been derived by repeating 75 times (with different seeds) every simulation. We have compared the results obtained by our approach and those exhibited by other schemes in the literature, as [99], [105] or [101]. Fig. A.5 shows how our scheme overcomes them. For example, results in [105] are worse than ours, meanwhile similar capabilities are achieved in [99], obtaining a lower TPR but also a lower FPR. However, this scheme integrates three subsystems (a Bayes-based classifier, Markov chains and an association rule algorithm), thus resulting in a more complex approach. In a similar way, results in [101] are comparable to ours, but the system incurs in a huge overhead due to the use of non-linear SVM (*Support Vector Machine*) and FDA (*Fisher*



(a) TPR (b) FPR

Figure A.5: TPR (a) and FPR (b) for both implementations of our approach and other similar schemes.

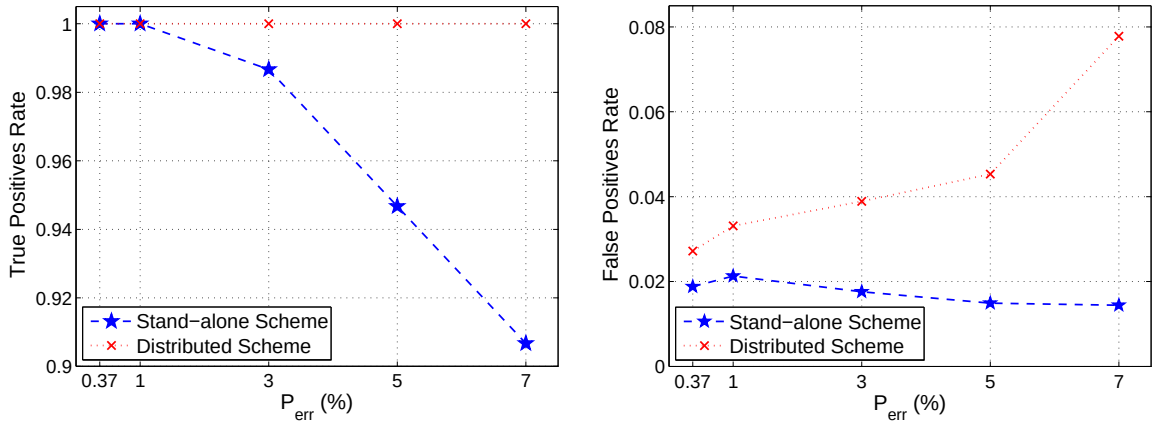
Discriminant Analysis). Besides, the results obtained for the distributed-collection IDS approach are a little bit worse than those get for the stand-alone case, since an approximation (through an indirect measurement) for two of the features is used, and thus the performance of this scheme becomes slightly deteriorated.

We have also carried out further experiments to evaluate the influence of different parameters over the detection results. These are: window size, nodes mobility and channel error.

Firstly, we have evaluated the influence of the event-based window size, by varying the number of received data packets, and using the values 50, 75, 100 and 125. The bigger the window size the better the detection capabilities, but at the expense of introducing greater delays in the detection process. Thus, we have selected 100 received data packets as a tradeoff value.

We also study the detection efficiency for different mobility speeds, from 5 to 30 m/s. Both implementations achieve excellent results, with TPR exceeding 97 % and FPR remaining below 4 % in all scenarios.

Our IDS approach has been also evaluated under different channel error probabilities, testing its performance when various channel characteristics, like shadowing or multipath fading, cause large packet losses. It is shown in Fig. A.6 that, even for large values of channel error probability (7 %), which can hide the dropping behavior of the malicious nodes, the scheme achieves excellent results: more than 90 % TPR and less than 10 % FPR.



(a) TPR (b) FPR

Figure A.6: TPR (a) and FPR (b) for both implementations of our approach and for different channel error probabilities.

In summary, we have verified the general good performance of our dropping detection approach. The results obtained clearly highlight the goodness of the IDS approach, which experienced 93% overall TPR with less than 4% FPR. This far overcomes the results exhibited by other similar schemes in the literature. It should be noted that the use of a simple model reduces the computational overhead present in more sophisticated approaches based on data mining algorithms, as well as it improves the detection performance under several circumstances not usually taken into account in previous works.

A.4.3. Sinkhole Detection in MANETs (Publications 1, 5 & 9)

Route poisoning attacks [9] are also highly disruptive threats in MANETs. Here, we focus on the *sinkhole* attack, possibly the most representative route poisoning attack, which is aimed at forging the multi-hop source-destination routes to attract the surrounding network traffic by providing fake routing information. That makes sinkhole nodes appear as the best path to some destinations, thus being selected by other legitimate nodes as a next hop on the forged route.

This section presents a novel behavior-based detection system that leverages the existence of what we call "contamination borders", *i.e.*, legitimate nodes under the influence of the attack and, at the same time, neighbors of others which are not contaminated. In these borders, routing information is more inconsistent and, therefore, behaves anomalously. By collecting and analyzing their own routing information and that belonging to their neighbors, contamination border nodes can determine the existence of sinkhole behaviors more precisely. Thus, we suggest a two-phase collaborative detection scheme. The first phase consists in a local pre-detection process, mainly devoted to minimize the traffic overhead. Only when this first process triggers an alarm, the detector will initiate the second phase, a collaborative procedure that collects some features from the neighbors to estimate the potential malicious behavior of a given node.

There exist many proposals in the literature to fight against sinkhole attacks in MANETs which simply monitor a set of collected features, usually the sequence numbers used by AODV or some kind of related metric, computed as a function of them [105], [106]. Furthermore, a number of slight variations also follow the approach of comparing the sequence number received in the RREP message with the one sent in the RREQ packet. Some of them can be found in [111], [112], [113], [114] and [116]. However, these schemes only consider the behavior of the sequence numbers locally, *i.e.*, without taking into account information of the network vicinity.

"Contamination Borders" in the Sinkhole Attack

As previously stated, our approach relies on the existence of contamination zones, formed by legitimate nodes under the influence of the attack. Some of these legitimate nodes conform the "contamination border" [143]. The peculiarity of these nodes is that they are simultaneously neighbors of contaminated nodes and of nodes which are not under the influence of the attack (*i.e.*, those that have the knowledge about legitimate routes). This way, when a non-contaminated node requests to one of these contamination border nodes a route that has been compromised, the latter will unintentionally reply with fake information. In such a situation, these border nodes behave in a similar way to how a malicious node would.

Under these circumstances, the only difference between a sinkhole node and a contaminated node is that the former deliberately tries to attract most of the surrounding traffic, whereas contaminated ones only act like the sinkhole for those requests related to fake routes learned from it. Then, we assign a higher probability of being malicious to those nodes that are labeled as suspicious for many routes. This way, we can conclude:

> **Corollary.** *Nodes in the "contamination zone" behave as sinkhole nodes for contaminated routes, but the existence of non-contaminated neighbors allows them to detect incoherencies in the routes.*

In Fig. A.7 it is possible to see an example of the evolution of the contamination zones and border nodes over time in a complex and realistic situation. The example depicts an static scenario (non-mobile nodes) with 25 randomly placed nodes. Legitimate nodes, from N_0 to N_{23}, are represented by blue circles; the sinkhole node, N_m, by a red triangle, meanwhile the legitimate nodes which have been contaminated are represented by magenta squares, connected by a blue dash-dotted line to the next hop of the contaminated route. Between t_0 and t_2, each legitimate node starts an application traffic flow towards a random destination.

We can observe that, at the beginning of the simulation, t_0, only N_m is acting as sinkhole. At time t_1, some legitimate nodes are already contaminated, *i.e.*, they contain at least one fake route due to the action of the sinkhole. In this situation, the node N_{22} will become a contamination border, since it is under the influence of the sinkhole N_m and, at the same time, it is neighbor of a non-contaminated legitimate node, N_9. As it will be explained in the following, by using the information provided by node N_9, it is more likely that N_{22} classifies N_m as malicious. At time t_3, after only 3 seconds, more than a half of the legitimate nodes in the network are within the contaminated area.

Most of the current detection approaches only consider the information which can be directly accessible by the node carrying out the detection process, *i.e.*, employing

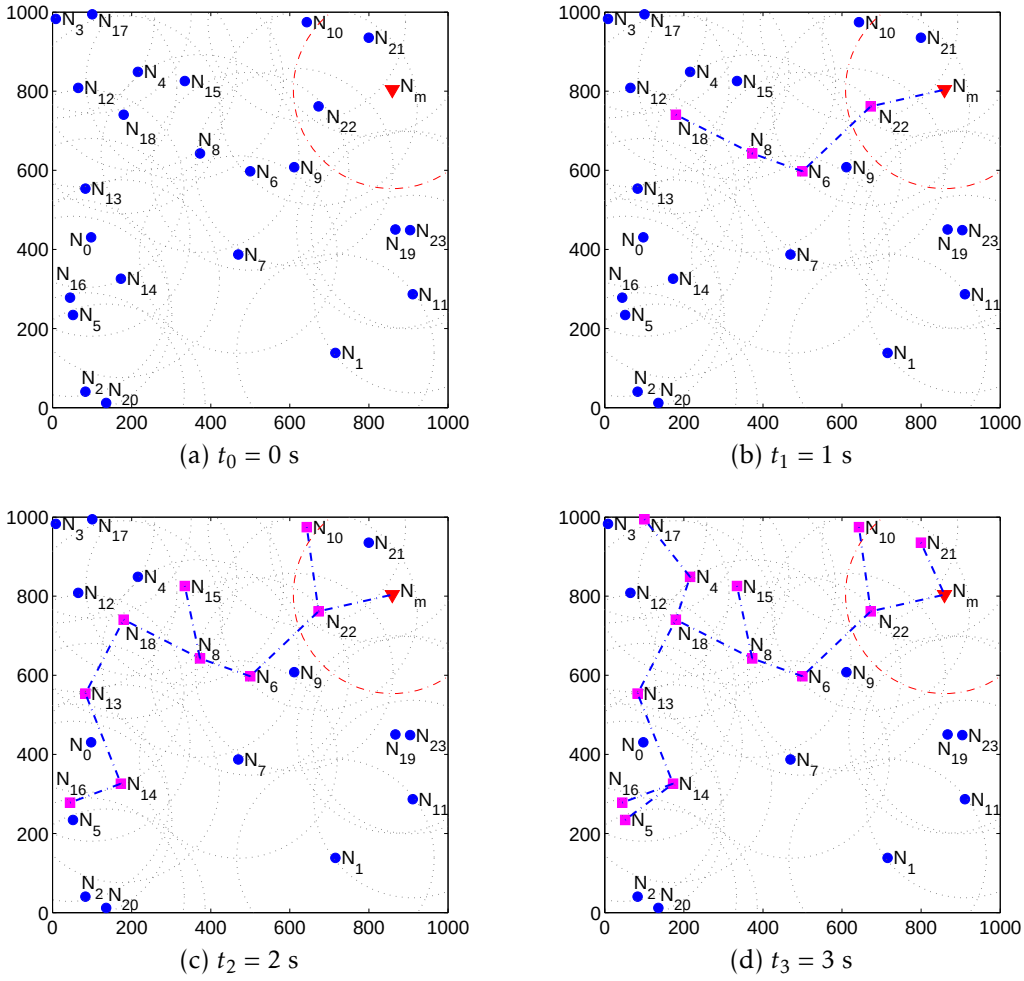


Figure A.7: Evolution of the contamination zones and border nodes under a sinkhole attack.

some metric related to the difference between sent and received sequence numbers. However, this approach suffers from some flaws which can lead to errors in the detection process and that our approach can overcome.

The first weakness is related to the fact that these approaches provide good results as long as the increased sequence numbers published by the sinkholes are high, that is, the difference between the sequence number received in the RREP and the one sent in the RREQ must be noticeable. However, the problem arises when sinkhole nodes are somehow smart and try to publish fake sequence numbers moderately high, thus hindering the detection process. On the other hand, contaminated legitimate nodes learning fake routes are able to publish them, being prone to be erroneously detected as sinkholes as well. Therefore, both facts can lead to an increase in the

misclassification of legitimate nodes as malicious, thus degrading the capabilities of these schemes.

Assuming that "contamination border" nodes behave like sinkhole nodes for contaminated routes and that they can detect incoherencies in the routes due to the access to the information of non-contaminated nodes (see *Corollary* above), we develop a method for the detection of sinkhole nodes based on the analysis of the evolution of the sequence numbers for the routes. In our approach, each node implementing the IDS first compares periodically the sequence number for a given route with its value in the last comparison (as in previous detection approaches). In addition, it may perform the same comparison in a more global way, with the route sequence numbers stored by its neighbors. This way, "contamination border" nodes will obtain higher differences in these comparisons, which results in the improvement of the detection capabilities.

Deploying the Sinkhole Detection Scheme

The specific implementation of the proposed sinkhole detector is presented here. Even though the detection process is locally performed by each node N_i running the detector, the features involved in such a process are, if necessary, collaboratively gathered from the node's vicinity. Our approach follows a sample-based procedure with period W and we will denote with superscript t the different parameters for the sampling interval of study. For every next hop node (NH) present in the routing table of N_i, the following features are collected:

- $\boldsymbol{D^t_{i,NH}}$: the set of destinations in the routing table of N_i using NH as next hop.
- $\boldsymbol{SN^t_{i,j}}$: sequence number at node N_i for the destination N_j.
- $\boldsymbol{NB^t_i}$: set of neighbors of node N_i.

We apply a two-phase heuristic to decide if a node is a sinkhole. The first phase (*pre-detection phase*) is mainly devoted to detect locally suspicions about nodes acting as sinkholes. Only if a node is considered suspicious the second phase (*collaborative phase*) will be triggered. In summary, the following detection procedure is executed:

Pre-detection phase

1) Firstly, the detector at every node N_i obtains for each next hop NH, a set of local suspicion values $LSV_{i,j}$, one for each possible destination N_j in $D^t_{i,NH}$:

$$LSV^t_{i,j} = SN^t_{i,j} - SN^{t-1}_{i,j} \tag{A.10}$$

2) If there exists at least one LSV value greater than a given threshold, θ_s, the node NH is considered suspicious:

$$NH = \begin{cases} suspicious, & \text{if } \exists\ N_j \in D^t_{i,NH} \ / \ LSV^t_{i,j} \geq \theta_s \\ legitimate, & \text{otherwise} \end{cases} \tag{A.11}$$

Only if the node NH is classified as suspicious (denoted as NH^*) in the first phase, the collaborative detection phase is triggered.

Collaborative detection phase

3) The detector at node N_i extracts a set of destinations N_j in D^t_{i,NH^*} employing NH^* as next hop. That is, all the destinations which are supposed to be compromised.

4) Then, N_i broadcasts a message requesting to its neighbors the sequence numbers for destinations N_j in D^t_{i,NH^*}.

5) After gathering the replies, N_i obtains the minimum sequence number of their neighbors for each destination N_j, and computes the difference between its own sequence numbers and this minimum value:

$$\Delta SN^t_{i,j} = SN^t_{i,j} - \min_{n \in NB^t_i} \{SN^t_{n,j}\} \tag{A.12}$$

6) A global suspicion value, GSV^t_{i,NH^*}, is obtained as the product of these differences, thus considering that nodes NH^* appearing in more routes are more likely to be a malicious sinkhole:

$$GSV^t_{i,NH^*} = \prod_{N_j \in D^t_{i,NH^*}} \left(1 + \Delta SN^t_{i,j}\right) \tag{A.13}$$

Note that we add one unit to the factors as, for a given compromised destination, the computed difference between sequence numbers might be zero.

7) If GSV^t_{i,NH^*} exceeds a second fixed threshold θ_d, the node NH^* is finally classified as a sinkhole:

$$NH^* = \begin{cases} malicious, & \text{if } GSV^t_{i,NH^*} \geq \theta_d \\ legitimate, & \text{otherwise} \end{cases} \tag{A.14}$$

The calculation of the malicious value is a simple process with low computational cost once the information from all the neighbors is gathered. Besides, the operation point of the system depends on the thresholds θ_s and θ_d.

- **Communication Protocol**

The collaborative detection phase implies communications between nodes. For that, two possibilities arise: *(i)* to use the own messages of AODV, or *(ii)* to design new messages to reduce the bandwidth consumption.

In the first option, when the detector classifies a given node as suspicious, it simply broadcasts, for every route using the suspicious node as next hop, a RREQ message to ask their neighbors for the sequence number associated to this route. Every neighbor having a valid sequence number will sent back a RREP message including the information. Despite its simplicity and compatibility with AODV, the problem is that for each possible destination, one communication flow is required, thus consuming more bandwidth in terms of sent packets.

A second option is to define some new messages specifically designed: a new request message, which includes the addresses of the several destinations for which the sequence number is required, and the associated reply packet. In this approach, only one communication flow is required, since it allows to ask for all the possible destinations in one single broadcast request, and to send all the information back through only one unicast reply, thus reducing the traffic overhead. These messages can be defined as new AODV message types or as messages for a different communication protocol. In this work we have chosen the latter, as it will be shown in Section A.4.4, where a new communication protocol specific for MANETs is defined, being the messages employed here a particularization of those designed for the new protocol. See, in particular, messages in Fig. A.12 and Fig. A.13.

Therefore, it is expected that, although the size of the new defined packets can be greater than that of the own AODV messages (depending on the expected number of destinations required), the reduction in the number of exchanged packets because of the use of our communication protocol results in a lower overhead.

Experimental Results

This section is first devoted to present the description of the experimental environment used to evaluate the detection scheme developed. Then, we show the different experiments performed and discuss the results obtained.

- **Experimental Environment**

We have simulated some MANET scenarios by using the network simulator OMNeT++ (*Objective Modular Network Test-bed in C++*) [144].

The simulation area for the network deployments is restricted to a 1000 m x 1000 m square. We have chosen IEEE 802.11g and AODV as MAC and network layer protocols respectively, and the RTS/CTS mechanism is activated. The total number of nodes is 25, 24 of them being legitimate and 1 acting as a sinkhole always replying with false RREP (adding a uniformly distributed random value between 20 and 30 units) to every received RREQ. The *Two Ray Ground* [140] propagation model is also used, and the communication range is set to 250 m.

In addition, more sophisticated and realistic models for the movement of the nodes and for the application traffic are also employed. RPGM (*Reference Point Group Mobility*) [149] is used to model the mobility of the nodes, the group size being 5 nodes and the maximum distance between a node and its corresponding group center being set to 250 m. We have considered the widely used tool BonnMotion [150] to generate the mobility patterns. The minimum speed for mobile nodes is fixed to 0.5 m/s and the maximum speed varies from 3 to 10 m/s, the pause time being set to 15 s. Each legitimate node has its own application traffic flow, simulating real time point to point voice traffic. Several calls per flow are obtained by modeling the IAT (*Inter-Arrival Time*), following an exponential distribution with λ equals to 7.5 seconds, and the CHT (*Call Holding Time*), modeled as a lognormal with mean, μ, set to 3.287, and standard deviation, σ, 0.891 [151]. For each call, one of the legitimate nodes is randomly chosen as destination, the call being treated as a CBR connection, with 4 packets/second data and payload size equal to 512 bytes.

- **Detection Performance**

We have evaluated the global effectiveness of the detector by computing the TPR and the FPR metrics, obtaining the associated ROC curve by varying the decision threshold θ_d in Eq. (A.14). Results have been derived by repeating 50 times (with different seeds) every simulation. First, the parameters W (sampling interval) and θ_s (local suspicious threshold) have been empirically obtained through an extensive experimentation as those providing a good tradeoff between detection capabilities and consumed bandwidth. This way, W and θ_s are fixed to 5 seconds and 20 units respectively.

We have compared the results obtained by our two-phase collaborative approach and those exhibited by other schemes that compute a local heuristic only considering sent and received sequence numbers, as [112] or [114]. Figure A.8 shows how, by including information from the neighbors, our collaborative scheme highly overcomes the results achieved by the local related approaches.

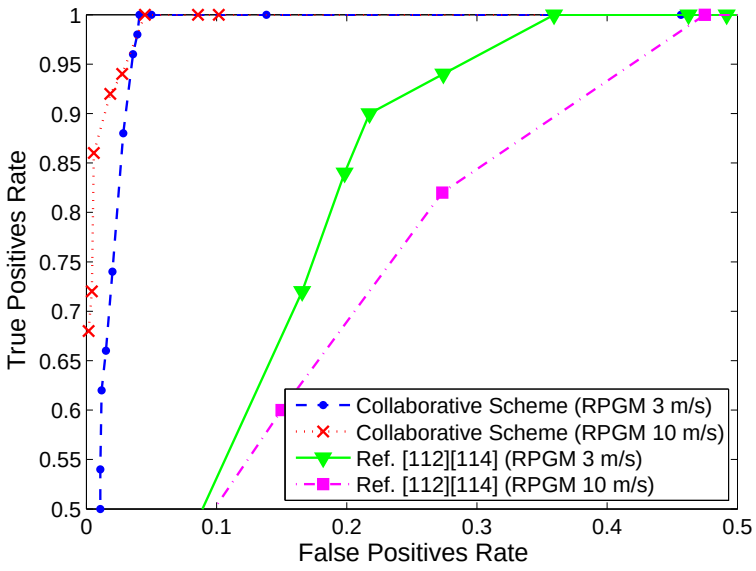


Figure A.8: Comparison between ROC curves for different sinkhole detection schemes.

Furthermore, beyond the good results provided by our scheme, we present more experiments intended to demonstrate the performance of our approach under a more realistic propagation model than Two Ray Ground. Specifically, we have used the *Nakagami multipath fading* propagation model [152], a generic probabilistic model where reception power follows a gamma distribution, reflecting different environmental circumstances well. The model employs a parameter m to specify the intensity of the fading effects, covering a wide range of fluctuation intensity. Therefore, the proposed detector is evaluated including a Nakagami multipath fading model, fixing the m-value to 4. Figure A.9 shows the detection results obtained for the scenarios described above, where the nodes move with a speed around 3 and 10 m/s.

It should be noted that, in environments with such a high level of losses due to propagation issues (reflections, scattering, fading, etc.), the usage of reactive protocols like AODV is not appropriate. In these situations, it is recommended to use proactive protocols which employ link quality metrics, like OLSR (*Optimized Link State Routing*). However, we can see that, even in these circumstances, our detector achieves remarkable results in terms of TPR and FPR.

Besides, we present another set of experiments to compare the performance of the proposed detector against the local related approaches under different sinkhole attack severities, considering a maximum speed of 10 m/s for nodes. Figure A.10 provides the ROC curves for two different sinkhole attack rates: 80 % and 50 %, and also for the extreme situation in which the sinkhole only sends forged RREP messages

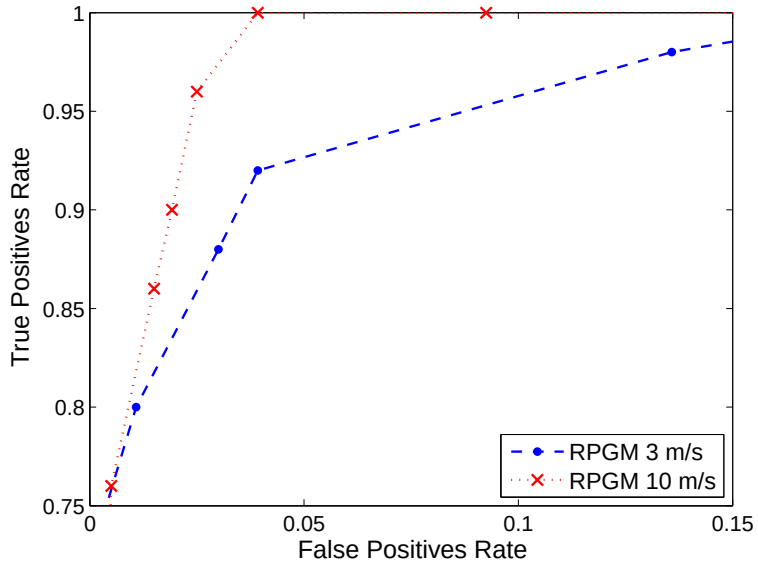


Figure A.9: ROC curve of our detector under the Nakagami propagation model.

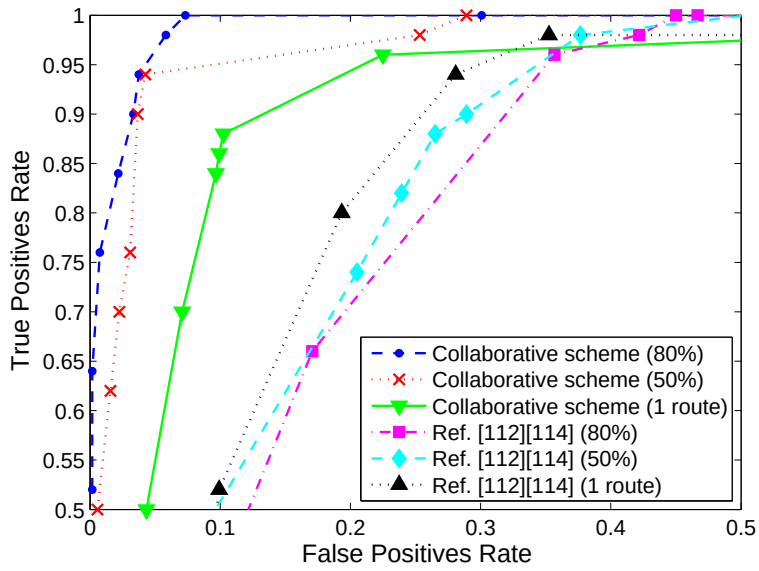


Figure A.10: ROC curves for different sinkhole detection schemes and different attack severities (RPGM 10 m/s).

when it is asked for a particular route, *i.e.*, the sinkhole only contaminates a given route.

As it can be seen again, the proposed detector is still able to achieve almost 90% TPR with 10% FPR in the extreme situation where only one route is contaminated.

We have studied the traffic overhead introduced during the collaborative detection phase as well, and we can conclude that our scheme is able to provide great detection capabilities while keeping the overhead within reasonable levels. Table A.1 shows the bandwidth consumed by the two approaches, AODV and new messages, for a maximum speed of 3 m/s.

In summary, we demonstrate that the proposed sinkhole detector achieves excellent results regarding the two considered metrics, TPR and FPR. By selecting

Table A.1: Bandwidth for the two approaches, AODV and new messages, for a maximum speed of 3 m/s.

Sampling interval (s)	θ_s	$BW^{aodv}_{pkt/s}$	$BW^{new}_{pkt/s}$	$BW^{aodv}_{B/s}$	$BW^{new}_{B/s}$
$W = 1$	0	41,56	34,83	843,03	904,78
	10	15,12	12,13	306,67	322,81
	20	13,09	10,56	265,49	280,10
	30	3,08	2,44	62,56	65,36
	40	2,33	1,83	47,44	49,35
$W = 5$	0	36,40	31,21	738,37	801,01
	10	13,72	11,10	278,24	293,95
	20	11,99	9,74	243,22	257,36
	30	3,06	2,42	62,08	64,83
	40	2,32	1,81	47,00	48,82
$W = 10$	0	24,39	21,04	494,66	538,35
	10	11,08	9,07	224,71	238,68
	20	9,76	7,97	197,93	209,99
	30	2,69	2,10	54,54	56,69
	40	1,99	1,55	40,33	41,92
$W = 30$	0	11,32	9,98	229,47	252,35
	10	6,27	5,26	127,08	136,57
	20	5,65	4,74	114,65	123,14
	30	2,23	1,80	45,23	47,76
	40	1,61	1,27	32,61	34,02
$W = 60$	0	5,84	5,24	118,43	131,41
	10	3,65	3,15	74,19	80,69
	20	3,34	2,86	67,78	73,52
	30	1,83	1,51	37,08	39,60
	40	1,22	0,98	24,77	26,05

the optimal operation point, TPR can achieve 100 % while FPR always keeps below 5 %, which confirms the capabilities of our model. Moreover, the traffic overhead introduced by the communications involved in the collaborative detection process is acceptable for this kind of ad hoc mobile environments.

Third Part: Integration of Security Solutions

The design of new detection schemes for dropping and sinkhole behaviors in MANETs constitutes, as indicated, the core of this thesis work. However, further interesting developments have been carried out in the line of integrating these schemes with other security solutions, thus providing stronger overall security. In what follows, two additional contributions are presented: a notification and alert security event procedure, which allows to launch subsequent response/reaction mechanisms to fight against detected attacks, and an integral security framework, intended to serve as a benchmarking tool to implement attacks, develop and integrate new defense solutions and evaluate their performance in order to conclude the real capabilities associated.

A.4.4. Notification and Alert of Security Events (Publication 8)

Although prevention, detection and response lines should interoperate to provide a more global and integral security solution, they are in general designed and adopted as separate and independent tools. Thus, in the literature there is a relative lack of proposals devoted to provide the necessary interaction between these modules.

It is possible to find some proposals for the notification of security related incidents, like those developed in [159], [162] and [168], which would be used to interact among different defensive modules. However, they are complex or resource-demanding approaches, and thus not suitable for MANET environments. Addressing the security-sensitive nature of exchanges between intrusion detection entities, the IDWG (*Intrusion Detection Working Group*) defined in the RFC 4766 ("Intrusion Detection Message Exchange Requirements") a number of requirements to meet in relation to the message format and the communication protocol to communicate entities. Based on these requirements, IDMEF (*Intrusion Detection Message Exchange Format*) [160] and IDXP (*Intrusion Detection eXchange Protocol*) [165] were designed. The IDMEF format is based on XML (*eXtensible Markup Language*), thus providing a huge versatility and flexibility. However, complexity and overhead are also its major drawbacks when used in resource-constrained environments, like MANETs, where a simpler (while also versatile and extensible) format is required. Likewise, IDXP provides a service for the reliable exchange of data between entities which complies with the requirements. However, it is built as a connection-oriented protocol, which is not suitable in this kind of networks, where connectionless transport protocols like UDP (*User Datagram Protocol*) are usually employed.

In this somewhat chaotic inter-defenses context, the proposal of a communication protocol specifically designed to notify security events in MANETs is justified. Thus, we study and develop here a novel protocol with this aim. It can be used as a

communication interface between the detection and response modules, as well as for the distribution of information between modules in collaborative detection and/or response processes. This is the case of the collaborative sinkhole detection proposed in Section A.4.3.

Protocol for the Notification and Alert of Security Events

The particular proposal presented here is versatile and efficient. It is implemented as an application layer protocol over UDP, in order to reduce delays and resource consumption. Besides, the notification procedure is flexible, allowing several functionalities and including several modalities for the message transmission, depending on the specific packet considered.

- **Functionalities**

Three different usages are initially considered:

1. *Alert notification*. Once an entity (typically a node in the network) detects the presence of malicious intrusive behaviors, the procedure for the notification of security alerts about them to other entities (the detector entity's vicinity, a central entity, the rest of the nodes in the network) is intended to communicate useful information about the detected event.
2. *Exchange of security information*. The potential exchange of security related information among the nodes mainly arises as a consequence of the works in [143] and [58], where collaborative IDSs based on the exchange of information among nodes are deployed. The applicability of the proposed exchange also includes centralized IDSs. In any case, there is a node asking other nodes in the network for information about a given analyzed node. In response, the specific information is provided.
3. *Asynchronous notification of security information*. The last functionality is intended to notify security information without requiring any previous request. In such a case, a given node, upon the occurrence of some anomalous or remarkable circumstance, could decide that it is particularly relevant and useful to send that information to other nodes, in order to perform a thorough investigation.

- **Protocol Operation**

As previously stated, the functionality is developed in terms of an application layer protocol over UDP, being employed by all those nodes in the network implementing some kind of defensive system (either prevention, detection or response).

To give support to the aforementioned functionalities, four types of messages are defined. The *alert notification* messages are sent whenever a given detection module determines the existence of an intrusive event. These messages should reach every node in the network in order to notify all of them about the detected attack. The *information request and reply* messages are exchanged in case a given node needs more global information than the locally collected one in order to perform cooperative detection/response procedures. Finally, an *asynchronous notification* message is defined to allow defensive modules to, proactively, provide useful information.

An important aspect in the design of the notification procedure is the method followed for the message transmission, which depends on the specific application and type of message considered. Given that the objective is to minimize the resource consumption, three modalities are defined in our protocol. *Broadcast* to the whole network (*flooding*) is used for the alert notification messages. Nodes receiving these messages will forward the reported event for its distribution, and therefore, each message must contain a unique identifier to avoid duplicate retransmissions. *Broadcast* to the k-hop neighbors could be used for the information request and for the asynchronous notification messages. These messages are sent by setting the TTL (*Time To Live*) field in the packets IP header to k. In the proposed functionalities we have employed $k = 1$. And finally, *unicast* is used for the information reply messages from each of the suppliers to the requesting node.

- **Message Format**

The format of the alert notification message is shown in Fig. A.11. The component fields are:

- *Message type*: to distinguish among the different usages considered.
- *Event type*: the particular type of attack observed (dropping, sinkhole, etc.).

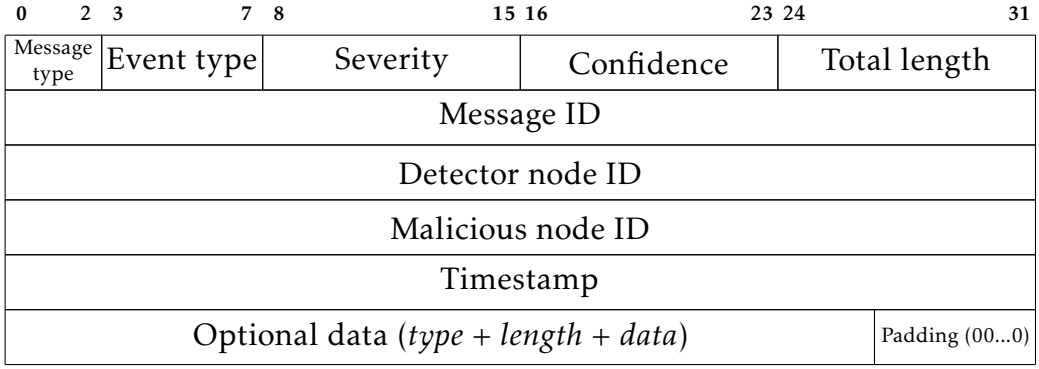


Figure A.11: Format of alert notification messages.

- *Severity*: the degree of affectation. For instance, the impact caused by a dropping attacker discarding only the 20% of the packets is less harmful than that produced by a node dropping every received packet.
- *Confidence*: the degree of certainty in the detection process. It is different a misuse-based detection, with around 100% confidence, than an anomaly-based detection, where the confidence will be (presumably) computed as a function of the deviation observed [84].
- *Message ID*: a monotonically increasing number associated to the message, used to avoid duplicate retransmissions. It is also useful to protect the protocol against reply attacks.
- *Detector node ID*: identifies the node which detected the reported incident. This ID typically refers to the IP address of the node.
- *Malicious node ID*: similar to the previous but related to the node being reported as malicious, necessary for the deployment of some response mechanisms.
- *Data (optional)*: it would be interesting the possibility of including additional useful information if needed (*e.g.*, the location of the malicious node). The format of this field must be: $< data\_type > \ < data\_length\_bytes > \ < data >$

Regarding the request messages (Fig. A.12), the information involved is:

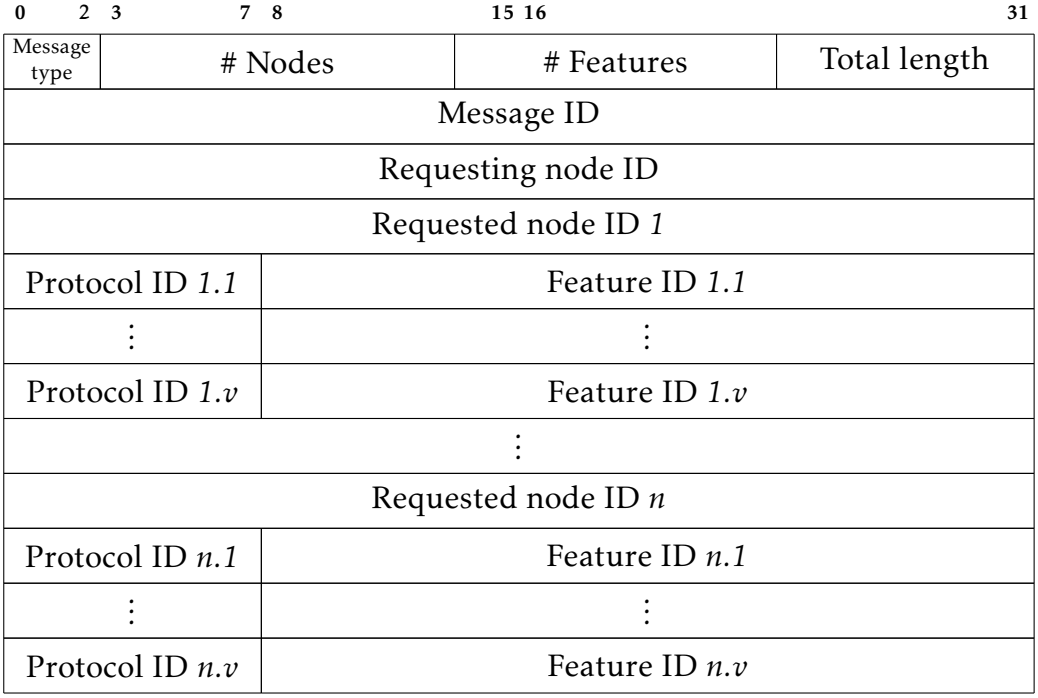


Figure A.12: Format of request messages for the exchange of information.

- *Message type*: security information request in this case.
- # *Nodes*: corresponding to the number of nodes whose features are required.
- # *Features*: amount of features required for every requested node.
- *Message ID*: similar to the field in the alert notification messages. However, in this case, the field is also employed to match requests and replies.
- *Requesting node ID*: used to univocally identify the node requesting the information, *i.e.*, the one presumably carrying out the collaborative detection process.
- *Requested node ID 1...n*: univocally identifies the different nodes whose information is required.

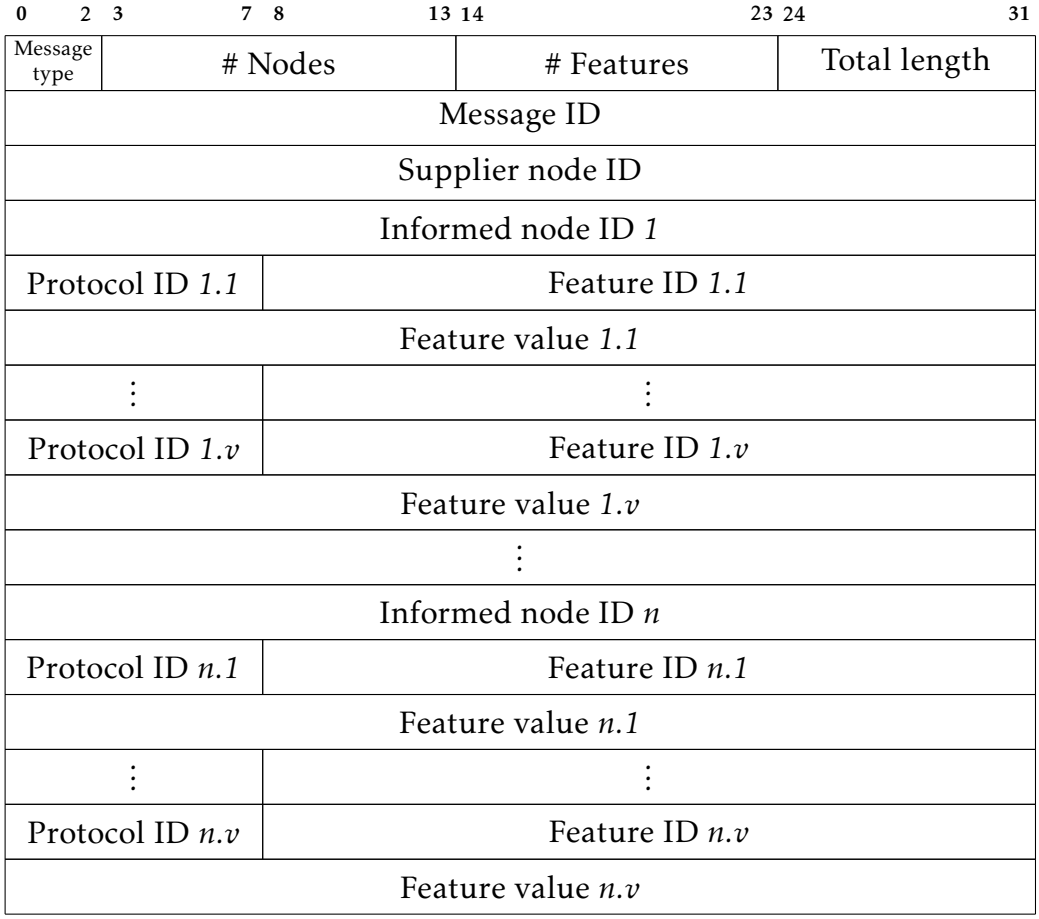


Figure A.13: Format of reply messages for the exchange of information.

- *Feature 1...v*: consecutive 32-bit fields representing each of the *v* features for which the information about the different requested nodes is claimed. Every feature is defined by two fields: *protocol ID* and *identifier* of the specific feature.

With regard to the reply messages (Fig. A.13), the main information returned is:

- *Supplier node ID*: represents, univocally, the node sending the information required.
- *Informed node ID 1...n*: to univocally identify the different nodes whose information is being provided.
- *Feature 1...v*: consecutive 64-bit fields, used to identify and inform about each of the *v* features solicited in the request message. After its identification (by using 32 bits, 8 of them to provide the protocol/procedure referred), its value is specified through a 32-bit field.

Table A.2 shows some of the features used in the IDS schemes so far deployed by the authors in A.4.2 and A.4.3. Thus, the protocol provides great flexibility, allowing

Table A.2: Example of features considered in the information exchange.

Protocol/Procedure	**Feature**	**Observations**
Miscellaneous	Sample period	In *s*
Topology	Speed	In *m/s*
	Acceleration	In m/s^2
	Location	GPS position
Physical	RSSI	From 0 to −80 dBm
MAC IEEE 802.11	$\#P_{RTS}$	RTS / CTS packets; sent and received
	$\#P_{CTS}$	
AODV	$\#P_{HELLO}$	HELLO / RREQ / RREP packets; sent, received, forwarded and discarded
	$\#P_{RREQ}$	
	$\#P_{RREP}$	
	SeqNum	Sequence number
	HopCount	Number of hops
Application	$\#P_{data}$	Data packets; sent, received and lost
	#Sessions	Number of sessions

its extension by simply defining new features. In particular, remind the exchange of *SeqNum* feature for the collaborative sinkhole detection in Section A.4.3.

Regarding the asynchronous notification messages, its format is the same used for the reply messages shown in Fig. A.13, with the following variations:

- *Message type*: asynchronous notification of security information.
- *Message ID*: identifier of the message itself, but not employed to match requests and replies.

All of these aspects, as well as the proper specification of the messages, have a direct impact over the communications performance of the monitored environment.

Performance Analysis

In this section we carry out a brief analysis of the bandwidth, BW (in bytes/s), consumed by the aforementioned protocol. Before computing the bandwidth, it is necessary to present some variables and their notations:

- $\boldsymbol{f_i^{a/rq/n}}$: frequency, in transmissions per second, of the messages sent by a requesting node N_i (for alerts, requests or asynchronous notifications), determined, among others, by the underlying detection process.
- $\boldsymbol{E[NB_i]}$: expected number of neighbors of the requesting node N_i. Considering L nodes with a coverage range of r meters, distributed in an $a \times b$ m^2 area, $L/a \cdot b$ will be the total node density, and $(L/ab)\pi r^2$ the number of nodes within the range of the requesting one. Subtracting the node itself:

$$E[NB_i] = \frac{(L-1)\cdot\pi\cdot r^2}{a\cdot b} \tag{A.15}$$

- $\boldsymbol{E[n]}$: expected number of nodes whose information is required/supplied.
- $\boldsymbol{E[v]}$: expected number of features required/supplied.
- $\boldsymbol{PS^{a/rq/rp/n}}$: size of the packets. The size of alert messages (without considering optional data) is 20 bytes, and that of the request, reply and asynchronous notification messages depends on the number of nodes and features, and is $12+4\cdot E[n]+4\cdot E[n]\cdot E[v]$ bytes for the former and $12+4\cdot E[n]+8\cdot E[n]\cdot E[v]$ bytes for the two latter, respectively.
- $\boldsymbol{p(I)}$: probability that a node has valid information about the requested features.

Once the notation has been defined, the specific performance depends on the particular application of the protocol.

- **Alert Notification**

The bandwidth consumed by the broadcast to all the network-based alert notification requires every node to forward the alert message, so the number of messages generated will be L, *i.e.*, the number of legitimate nodes. Consequently, the bandwidth (in bytes/s) in alert situations initiated by node N_i against node N_j is:

$$E[BW_{i,j}^{a}] = f_i^a \cdot PS^a \cdot L \tag{A.16}$$

- **Exchange of Security Information between Neighbors**

Regarding the second usage, one communication flow is required between the requesting node and its neighbors every time that security information is needed, starting with a broadcast request sent to the neighbors, which will be unicastly answered by those knowing the information. Therefore, the bandwidth involved in this case (in bytes/s) is:

$$E[BW_i^{ex}] = f_i^{rq} \cdot \left(PS^{rq} + PS^{rp} \cdot E[NB_i] \cdot p(I) \right) \tag{A.17}$$

Figure A.14 shows a comparison between the bandwidth introduced by the information exchange application from (A.17), by varying the number of neighbors, $E[NB_i]$, and for two different numbers of required nodes, $E[n] = 3$ and $E[n] = 10$. For simplicity, it is considered that only one feature is required ($E[v] = 1$). It is also assumed the worst case scenario, where all neighbors have the requested information ($p(I) = 1$). In addition the frequency of transmissions is also studied, first considering a worst (and unrealistic) scenario, where one communication flow is required every second ($f_i^{rq} = 1$), and another where communications are only needed every 10 seconds ($f_i^{rq} = 0.1$).

It can be observed that, even for a high number of features required and neighbors involved, our communication protocol introduces low overhead due to the exchange of information for collaborative detection/reaction schemes. This theoretical experimentation has only been performed for this exchange of security information usage, the worst one in terms of bandwidth consumption. Since the bandwidth consumption here is acceptable in MANET environments, the resulting bandwidth of the other two functionalities is expected to be acceptable too.

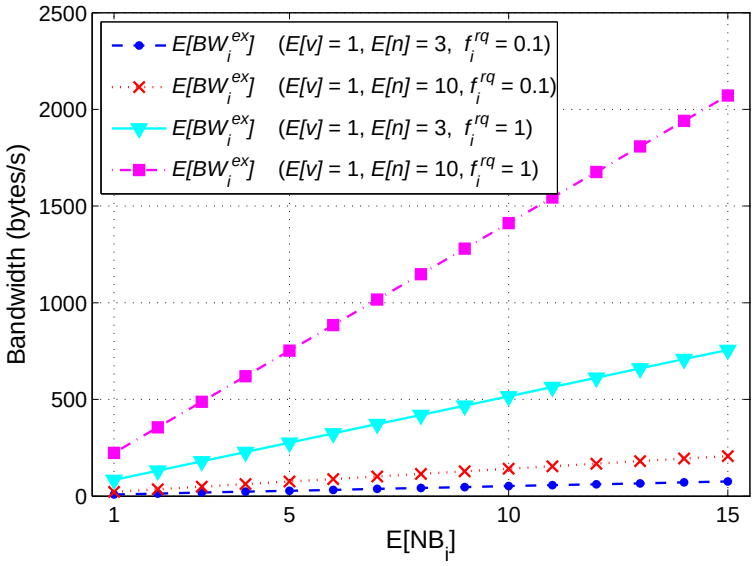


Figure A.14: Bandwidth consumed by the information exchange application for different parameters.

- **Asynchronous Notification of Security Information**

For the last usage, since it is an asynchronous notification, the communication consists of a single broadcast message sent from node N_i to their neighbors. Therefore, the expected value for the bandwidth (in bytes/s) is:

$$E[BW_i^n] = f_i^n \cdot PS^n \tag{A.18}$$

In summary, we propose a novel protocol for the notification of malicious activities against the security and for the exchange of security information between collaborative defense solutions. The procedure is suitable for decentralized environments, as ad hoc networks. The benefits of the presented proposal have been evidenced theoretically, and thus, the immediate objective of the authors is the effective implementation of the protocol and its performance evaluation under different experimental scenarios, in order to confirm the actual requirements involved.

A.4.5. NETA: Security Framework (Publications 6 & 10)

Security attack techniques evolve at a very high speed [173], [30], thus making the task of building defense mechanisms a hard mission. Simulation tools offer a good

compromise between cost and complexity in the development of such particular defenses, making it possible to design and configure different network deployments without having to spend a lot of money or worrying about valuable equipment.

Some of the most widely used simulators are, among others, NS-2 [139], NS-3 (*Network Simulator 3*) [179], GloMoSim (*Global Mobile system Simulator*) [180], OPNET (*OPtimized Network Engineering Tools*) [182] and OMNeT++ (*Objective Modular Network Test-bed in C++*) [144]. Nowadays, OMNeT++ is becoming one of the most popular due to the huge amount of frameworks (*e.g.*, INET, MiXiM, Castalia) available for it, as well as its flexibility and friendly-user interface, among other advantages. Based on such tools, it is desirable to have a common, extensible and versatile framework, able to combine the execution of the attacks and the particular defenses, in order to test them on multiple technologies, protocols and scenarios.

In this context, we introduce here NETA (*NETwork Attacks*) [145] [146], a novel framework built on top of the INET framework and the OMNeT++ simulator. Its flexible design is appropriate for the implementation and evaluation of many types of attacks and for the development of new defense techniques, thus making it suitable for the benchmarking of these solutions under the same testing conditions. NETA aims at saving efforts in the attack testing and defense development process, thus offering a useful tool for the research community in the network security field.

NETA: A Simulation Framework for NETwork Attacks

NETA has been developed by our research group, NESG (*Network Engineering & Security Group*) (http://nesg.ugr.es), as an OMNeT++ framework, built on top of the INET framework. The main idea is to implement nodes which can strike attacks, *attacker nodes*. To do this, the attacks are managed by the so-called *attack controllers*. These controllers manage one or more modules of a node by sending *control messages*. The messages are sent from attack controllers to specific modules that implement a modified behavior, such as that required by a specific attack. They are called *hacked modules*. For implementing this modified behavior, hacked modules are inherited or replicated from INET modules and are conveniently modified.

The creation of a new type of attacker node can be summarized as: (*i*) adding to the associated `.ned` file the controllers for the attacks that this node is expected to execute, (*ii*) creating the associated control messages and, (*iii*) substituting the modules needed by the attack controllers for their corresponding hacked modules. Figure A.15 shows the differences between a normal and an attacker node.

- **NETA Architecture**

As previously specified, the main components of an attack in NETA are: (*i*) attack controllers, (*ii*) control messages, and (*iii*) hacked modules.

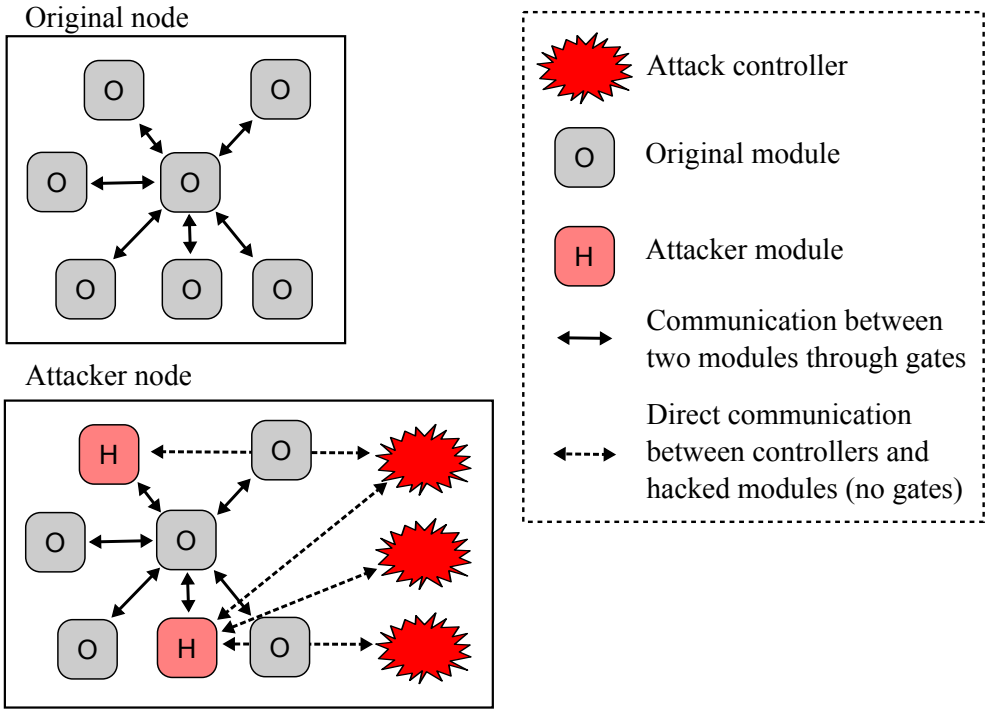


Figure A.15: Scheme comparison between an original node and its attacker version in NETA framework.

i. Attack Controllers. These are modules which control the execution of the attack, with the following properties:

- `attackType`: a name intended to differentiate an attack to the rest of them.
- `active`: a flag to indicate whether the attack is active in the simulation or not.
- `startTime`: the time at which the attack starts in the simulation.
- `endTime`: the time at which the attack ceases.
- `attack_specific_parameters`: different configuration parameters depending on the specific attack functionalities. For instance, we may define the generic probability of the attack, the sequence number added if the attack implemented is the sinkhole, or the delay introduced in delay attacks.

The tasks performed by an attack controller can be summarized as:

1. To obtain the different hacked modules involved in the execution of the attack.

2. To activate those hacked modules in the attack node by sending, at the starting time, activation messages which can contain configuration information.
3. To deactivate the hacked modules in the attack node by sending a deactivation message at the ending time.

ii. *Control Messages*. These messages are sent from attack controllers to the hacked modules involved in the attack execution, transmitting the information needed for the activation and deactivation of the attacks. Additionally, they can contain configuration information.

iii. *Hacked Modules*. These are the modules whose behavior is modified in order to strike an attack. There exists only one hacked module per modified module, and not a hacked module for every attack implementation. If two different attacks need to modify the same module, there will only exist one hacked module for them. This design improves the flexibility of the framework, allowing the execution of more than one attack simultaneously.

Implemented Attacks and Evaluation

As a proof of concept, in a first version of the NETA framework we have implemented the following three attacks:

- **IP Dropping Attack**. Nodes exhibiting this behavior intentionally drop, with a certain probability, received IP data packets instead of forwarding them. The main parameter defining the attack is:
  - `droppingAttackProbability`: the probability of dropping a packet, defined between 0 and 1.
- **IP Delay Attack**. Malicious nodes delay, with a given probability, IP data packets for a certain amount of time. The list of parameters is:
  - `delayAttackProbability`: the probability (between 0–1) of delaying a packet.
  - `delayAttackValue`: the specific delay time applied to the packet. Note that this parameter could be specified through a statistical distribution.
- **Sinkhole Attack**. Malicious nodes send fake routing information, claiming that they have an optimum route to the destination, thus causing other nodes to route data packets through them. The associated parameters are:

– `sinkholeAttackProbability`: the probability of answering a RREQ message with a fake RREP, defined between 0 and 1.

– `sinkOnlyWhenRouteInTable`: if set to *true*, the sinkhole node only sends fake RREPs to requests for destinations with a valid route in the attacker node. Otherwise (*false*), the node sends fake RREPs to any RREQ arriving, even if it does not know a valid route.

– `seqnoAdded`: the fake value added to the sequence number observed in the request. It can be different each time when specified as a statistical distribution.

– `numHops`: the fake number of hops returned by the attacker.

As a case study, a series of MANET deployments are simulated and several functional tests have been carried out to verify the proper performance of every implemented attack, measuring its impact on the network according to different metrics:

- PDR (*Packet Delivery Ratio*) and DR (*Dropping Ratio*) for dropping attack.
- E2ED (*End-to-End Delay*) for delay attack.
- AR (*Attraction Ratio*) for sinkhole attack.

Results obtained for the different metrics verify the proper operation of the implemented attacks. NETA is, however, in a preliminary phase of development, as our aim is to incorporate the detection schemes proposed in this thesis for dropping and sinkhole behaviors. Moreover, the implementation and integration in NETA of the developed notification protocol given in Section A.4.4 is also considered in the near future. Further developments in attack definition and defense lines are expected to be integrated in NETA, too. This way, some advances have been made in this line, although they must be evaluated more exhaustively.

Summarizing, NETA is intended to serve as a base reference framework to unify the development and simulation of attacks and defenses. Its flexible, extensible and versatile design is appropriate for the implementation and evaluation of many types of attacks and defensive techniques, doing it accurate for the benchmarking of current defense solutions under controlled testing conditions.

NETA is expected to become a useful tool for researchers focused on the network security field. For that, NETA framework is publicly available for download at http://nesg.ugr.es/index.php/en/neta and, despite its still short life, it has been widely downloaded worldwide.

Appendix B

Conclusions and Future Work

After describing the contributions of the work in Appendix A, now we summarize the main conclusions of this thesis work. Although they have been indicated in each of the previous sections, here we present them in a synthetic and unified manner. Additionally, we point out some open issues and future work to be addressed in the line of further research of this thesis.

B.1. Conclusions

MANETs have considerably evolved in recent years, leading to the appearance of different related technologies, architectures and applications. However, because of some inherent characteristics, the security in these networks is still an unsolved problem which needs to be properly addressed. A first step to provide security in these environments goes through having an appropriate **study and classification of current security threats**. Regarding this issue we can highlight the following achievements through this thesis:

- We have introduced a novel taxonomy for security attacks in MANETs, with that aim of organizing and classifying these threats from a practical perspective, allowing the potential building of more flexible and effective defensive approaches.

- We have also presented the main principles of MANET communications, providing some details about the operation of two of the most widely used protocols for MAC and network layers, IEEE 802.11 and AODV respectively.
- The way that dropping and sinkhole attacks can be implemented is described too.

After this necessary background, we have discussed that it is necessary to deploy **new detection schemes to protect MANET networks** against the previous specific security attacks. The main conclusions of the present work regarding this issue are summarized in the following:

- We have proposed an analytical model which represents the forwarding process in MANETs. This model properly includes the different circumstances which can lead to legitimate packet discards, like collisions, channel errors or mobility related situations, allowing to distinguish between such circumstances and actual malicious behaviors.
- Based on the previous analytical model, and following a cross-layer methodology, a novel IDS for the detection of packet dropping attacks is developed. Thus, a simple heuristic is employed for recognizing actual malicious packet dropping behaviors in MANETs.
- An enhanced event-based windowing is used instead of the classical time-based windowing commonly employed. This way, the features are obtained for non-overlapping windows of P received data packets.
- For the proposed IDS, two implementations can be deployed, depending on the way that the network features are collected: a local stand-alone approach and a distributed one.
- A thorough experimentation has been performed to evaluate the detection capabilities of the proposed system. Several tests have been carried out under several different circumstances, thus verifying the promising nature of our approach, which overcomes the results exhibited by similar schemes in the literature.
- Regarding the sinkhole attack, we have demonstrated the existence of "contamination borders", *i.e.*, legitimate nodes under the influence of the attack and, at the same time, neighbors of others which are not contaminated.
- We have shown how the routing information in these frontier nodes is more inconsistent and, therefore, behaves anomalously. By collecting and analyzing its own information and that belonging to their neighbors, border nodes can help in precisely determining the occurrence of sinkhole attacks.

- Based on that, a two-phase collaborative IDS scheme is presented: the first phase consists of a local pre-detection process which, if positive, launches a second phase where a cooperative mechanism that collects information from the vicinity is executed to accurately detect sinkhole attacks.
- We have suggested two possibilities for the previous inter-node information exchanging protocol. The first option is to use the own messages of the AODV protocol, which is simpler but introduces more overhead. The second is to use some specific messages, which implies a slight modification of the routing protocol but consumes less bandwidth and resources.
- Different tests have been carried under several conditions to evaluate the proper performance of the proposed detection approach. The results confirm the promising capabilities of our IDS.

With regard to the **integration of security solutions**, we can highlight the following main conclusions:

- We have contributed a novel protocol for the notification of security events in MANET environments, which can be used as an interface between the different defensive modules.
- Three possible usages have been discussed for this protocol. The first one allows to notify about the occurrence of security incidents to the whole network. To achieve this, a flooding like mechanism is used. The second application is about the exchange of security information, allowing to deploy distributed detection or response solutions. Broadcast requests to the neighbors and unicast replies from them are used in this case. The last usage of the proposed protocol allows the asynchronous notification of security information.
- A brief performance analysis evidences that, at least theoretically, the protocol is suitable for its use in MANETs.
- We have also contributed the development of NETA, a framework for the simulation of network attacks and defenses in MANET networks. The framework is built on top of the INET framework and the OMNeT++ simulator.
- The architecture of NETA is based on three different components: *attack controllers*, *control messages* and *hacked modules*. Its flexibility and versatility is appropriate for benchmarking attacks and solutions under the same testing conditions.
- As a proof of concept, three attacks have been implemented: IP dropping attack, IP delay attack and sinkhole attack. Their proper performance has been evaluated, the results proving the usability and capabilities of the framework.

- Currently in a preliminary state, we expect that NETA becomes an useful tool for researchers in the network security field. In this line, some defense schemes have been incorporated to NETA, although they must be evaluated in detail yet.

B.2. Future Work

The main future work pointed out from the present thesis, going from the more specific to the more general, are summarized in the following:

1. To implement effective methods for the adaptive determination of the detetion thresholds considered, since these values are strongly dependent on the specific network conditions.

2. To explore the possibility of including trust or reputation mechanisms in our detection proposals. These schemes can be applied as a response or prevention solution, thus allowing the feedback between the three traditional defensive lines.

3. Related to the previous point, to include, when applicable, mechanisms to deal with collusions, *i.e.*, those situations in which two or more attackers collude to evade the detection process or to increase the impact of the executed attack. Trust-based or reputation-based mechanisms seem a suitable solution for that.

4. To extend our detection approaches to other protocols different than AODV. Since the dropping approach relies on the analytical model for the forwarding process, the underlying routing protocol is not a strong limitation. Regarding the sinkhole scheme, we have shown that most other protocols also employ some identifiers in a similar way as the sequence numbers in AODV, which can be used as detection related features.

5. To build holistic detection approaches instead of attack-specific schemes, by extracting sets of common features independently of the target function attacked (*species*). This fact will presumably allow the definition of more comprehensive and complete detection systems.

6. To incorporate the actual notification protocol implementation in the NETA framework and to evaluate its requirements, in order to confirm its suitability for MANET environments.

7. To unify, in a single detection system, the different contributions here proposed (dropping and sinkhole attacks, dropping and sinkhole detectors, notification

protocol, etc.). The implementation of the NETA framework constitutes a huge progress in this goal.

8. To continue developing NETA by including new attacks and defense lines.

9. To apply and validate all the techniques developed to real scenarios, such as VANETs (*Vehicular Ad hoc NETwork*s) like environments or crisis management deployments.